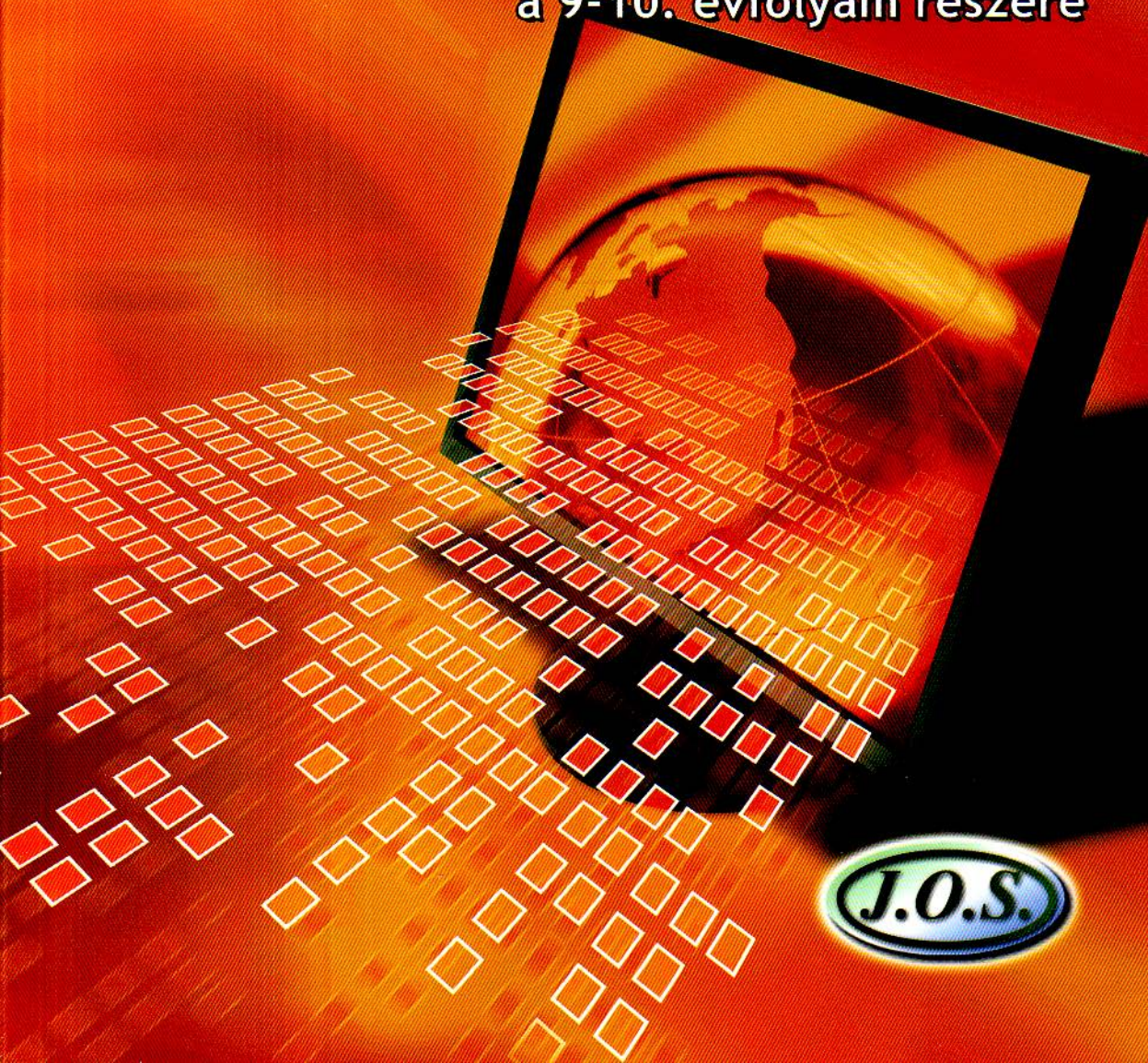


FARKAS CSABA

Informatikai ismeretek

a 9-10. évfolyam részére



*Gimnáziumok, szakközépiskolák alapóraszámában tanuló
csoportjai részére*

FARKAS CSABA

***Informatikai ismeretek
a 9-10. évfolyam részére***

Első kiadás

A könyvet az Oktatási Hivatal 15084-24/2006 határozati számon
2012. augusztus 31-ig tankönyvvé nyilvánította.

JEDLIK OKTATÁSI STÚDIÓ
Budapest, 2009



Minden jog fenntartva. Ezt a könyvet vagy annak részleteit a kiadó előzetes engedélye nélkül bármilyen formában vagy eszközzel reprodukálni, tárolni és közölni tilos.

Készült a 28/2000 (IX. 21.) OM rendelettel kiadott „Kerettanterv a gimnázium 9-12. évfolyamára” felhasználásával.

Az első kiadás változatlan utánnyomása, új borítóval.

Szakmai lektor: Danitz Béláné
Holczer József

Anyanyelvi lektor: Horváth Zsuzsanna
Venczel Katalin

Szerkesztette: Takács Attila

Borítóterv: Sarkadi Csaba, 2006
Tipográfia: Fodor Gábor Antal, 2006

©Kiadó: Jedlik Oktatási Stúdió Kft.
1212 Budapest, Tancsics Mihály u. 92.
Internet: <http://www.jos.hu>
E-mail: jos@jos.hu
Felelős kiadó: a Jedlik Oktatási Stúdió Kft. ügyvezetője
A kiadásért felel: Farkas Csaba

Nyomta: LAGrade Kft.
Felelős vezető: Szutter Lénárd

Terjedelme: 20,02 (A/5 ív)
Tömege: 380 gramm

ISBN: 963 87000 2 5
Kiadói kód: JO-0151

(Sz)ámítástechnika

Igaz történetek

Az informatika bevezetése a világon mindenhol nehézségekkel jár. A következő válogatás különböző számítástechnikai cégek ügyfélszolgálatától származik. Eredetileg a Wall Street Journalban jelent meg. Azóta e-mailben küldik egymásnak azok, akik úgy érzik, hogy végre találtak náluk kevésbé hozzáértőket.

Egy Dell-technikus azt ajánlotta egyik felhasználójuknak, hogy tegye vissza a flopit, amellyel bajlódtak, a meghajtóba, és zárja be az ajtót. A telefonáló megkérte a technikusot, hogy tartsa a vonalat. Hamarosan azt lehetett hallani, hogy a telefonáló leteszi a kagylót, keresztülvág a szobán, és becsukja a szoba ajtaját.

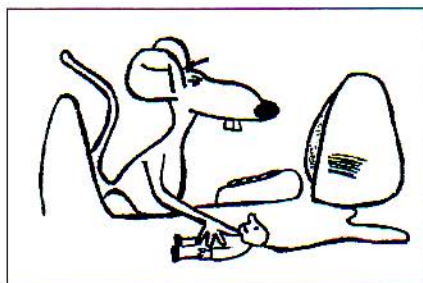
Ugyancsak egy Dell-felhasználó arra hivatkozott, hogy a billentyűzete nem működik. Pedig letisztította: fürdőkádját teleengedte szappanos vízzel és beleáztatta a billentyűzetet, majd minden billentyűt egyesével leszedett és lemosott.

Egy Dell-ügyfélszolgálatos olyan hívást fogadott, melyben a hívó vérben forgó szemekkel közölte vele, hogy a számítógépe rossznak és betegnek nevezte. Az ügyfélszolgálatos megmagyarázta, hogy a gép „Bad command” (rossz parancs) és „Invalid” (érvénytelen, más fordításban beteg, rokkant) válaszait nem szabad személyeskedésnek venni.

A Dell Computer Tech Supportnál egy felbőszült hívó azt állította, hogy nem képes számítógépét bekapcsolni. Miután megbizonyosodtak róla, hogy a gép be volt dugva a konnektorba, a technikus megkérdezte, hogy mi történik, ha benyomja a „Power” gombot. Erre a dühödött hölgy azt válaszolta: „Csak nyomom, nyomom ezt a lábpedált, és semmi sem történik.” (A lábpedálról kiderült, hogy az az egér.)

Egy Compaq technikai tanácsadó megkérte a felhasználót, hogy hibás lemezeiről küldjön be egy példányt. Néhány nap múlva egy levél érkezett a felhasználótól – a flopick fénymásolatával.

Egy IBM-ügyfélnek problémái voltak egy program telepítésével és hívta az ügyfélszolgálatot. „Betettem az első lemezt, és rendben volt. Azt írta ki, hogy tegyem be a másodikat, de ezzel már volt némi probléma. Aztán amikor azt írta ki, hogy tegyem be a harmadikat, az még csak bele sem ment.” Az ügyfél nem jött rá, hogy a „Tegyem be a 2. lemezt” utasítás egyben azt is jelentette, hogy előbb vegye ki az elsőt.



Szerepcsere

Igaz történet a Novell rendszergazdájától:

Hívó: Helló, ez az ügyfélszolgálat?

Technikus: Igen, segíthetek?

Hívó: A pohártartó a PC-men eltört és még garanciás. Hogy lehet megcsináltatni?

Technikus: Elnézést, pohártartót mondtott?

Hívó: Igen, a számítógép elülső részéhez van erősítve.

Technikus: Elnézést, ha nem egészen értem, de ezt valami akció során kaptá, esetleg egy kiállításon?

Hívó: Nem, a géppel együtt jött. Nem tudok semmiféle akcióról. Az van rajta, hogy „4×”. Ezen a ponton a technikus elnémította a telefont, mert nem bírta tovább nevetés nélkül. Az ügyfél a CD-meghajtó dobozkáját használta pohártartónak.

737.

Lássa el igaz (I), hamis (H) jelzőkkel a következő, számítógépes vírusokkal kapcsolatos mondatokat! [...] Egy vírusos lemez, közös dobozban tartva a többivel, megfertőzheti a másikat.

740.

Milyen módon lehet nagy valószínűséggel megakadályozni, hogy egy számítógép „vírusos” legyen! Karikázza be a helyes válaszok betűjelét! [...] b. megfelelő hőmérséklet és páratartalom biztosításával [...] d. minden használat után letakarjuk a számítógépet [...] f. hetente egyszer fertőtleníjtük a gép környezetét

229. Mi történik, ha egy számítógép „lefagy”?

a) A ventilátor túlhűti a processzort, és ez rendellenes működést eredményez.

689. Melyik állítás helyes? [...]

d) A vírus a határértéket meghaladó feszültségingadozás miatt keletkezik.

Részletek a számítógép-kezelő szakmai vizsgához készült Feladatbankból

Részletek az ECDL-példatárból

Valószínűleg mindenki azzal a jó érzéssel mosolyodott el az előző példák kapcsán, hogy lám, mások is hibáznak. Pedig az informatika elsajátítása során tévedni, hibákat elkövetni teljesen természetes dolog. Fontos: ha mi hibáztunk, akkor próbáljuk magunkat vidáman túltenni rajta, ha pedig más, akkor semmiképpen se gúnyoljuk ki, inkább segítsük őt a probléma megoldásában! Ma már az informatikára mindenkinek szüksége van, így ne feledkezzünk meg fogyatékkal élő embertársainkról sem, hiszen ez gyakran csupán minimális odafigyelést igényel. Például, ha egy weblap készítésekor a szöveges információt nem képként, hanem szöveggént helyezzük el, akkor ezzel azt a nem látó embertársaink számára is elérhetővé tesszük.

Hogyan tanuljuk az informatikát? Ha ki akarjuk használni a számítógép lehetőségeit, akkor természetesen néhány konkrét hardver és szoftver használatát alaposan meg kell ismernünk. Nem az a legfontosabb azonban, hogy pl. egy szövegszerkesztő programban melyik funkció hol van, hanem az, hogy milyen lehetőségeket ismer. A hangsúlyt tehát a szoftverek általános szolgáltatásaira kell helyezniünk, hogy tudásunk időtálló legyen.

Könyvünkben a hatályos kerettanterv szerint dolgoztuk fel az informatikai ismereteket. Azoknál a témaköröknél, amelyek szerepelnek a középszintű érettségi vizsgán is, a vizsgán elvárt mélységig tárgyaljuk a tananyagot. Az érettségi vizsgán számonkért fogalmakat csillaggal jelöltük meg. Mivel az informatikában nagyon sok az idegen (főleg angol) eredetű név és kifejezés, így ezek kiejtését szögletes zárójelben külön is feltüntettük.

Az informatika alapjai

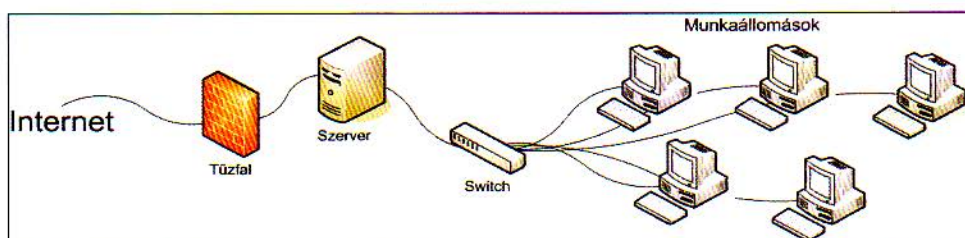
Az iskolai számítógépek használata

Az iskolai hálózat felépítése és használatának szabályai

A legtöbb munkahelyen, így az iskolában is, a számítógépek össze vannak kapcsolva, *helyi hálózatot* alkotnak. A helyi hálózat gépei nem egyenrangúak. A **felhasználók** a **munkaállomások*** mellett dolgoznak, míg a **szerverek*** szolgáltatásokat nyújtanak a **hálózati számítógépek részére**.

A számítógépek hálózatba kötése több előnnyel is jár:

- lehetővé válik az erőforrások megosztása (pl. nyomtató, közös tárterület),
- a közös adatokat egy naprakész közös adattárban tárolhatjuk,
- központilag megvalósítható a hálózat felügyelete (pl. a belépési és a hozzáférési jogosultságok kezelése).



Egyszerű helyi hálózat.

Ügyeljünk arra, hogy a gépeket összekötő vezetékek és hálózati elemek ne sérüljenek!

A számítógép-hálózat védelme érdekében **belépéskor felhasználói névvel és jelszóval kell azonosítanunk magunkat**. A felhasználók általában nem egyenrangúak, a **felhasználói név*** alapján dől el, hogy mit tehetünk és mit nem. A **hálózat biztonságos üzemeltetéséért a rendszergazda* felel**, ha elfelejtjük jelszavunkat, vagy további jogosultságokra is szükségünk van, hozzá kell fordulnunk.

Jóllehet a számítógépes hálózaton sokoldalú védelem állítható be, tökéletes védelem nincs. *Mások munkáját ne tegyük tönkre és ne hátráltassuk azzal, hogy fájljaikat töröljük, saját szórakoztatásunkra programokat telepítünk, vagy az alapértelmezett beállításokat módosítjuk!* A legtöbb hálózat rögzíti, ha egy adott felhasználó olyasmit művel, ami veszélyezteti a hálózat biztonságát. *Felhasználói nevünket és jelszavunkat ezért sohasem adjuk át másnak, mert kellemetlen helyzetbe hozhat minket, ha a nevünkben felelőtlenül dolgozik!*

Az informatika alapjai

A legtöbb iskolai számítógép-hálózat internet-hozzáféréssel is rendelkezik. Ilyenkor a hálózatot többnyire speciális eszközök (pl. tűzfalak) is védik, ezért a hálózat biztonsága érdekében előfordulhat, hogy nem minden internetszolgáltatás áll rendelkezésünkre. Ettől függetlenül magunk is törekedjünk arra, hogy felelőtlen viselkedésünkkel ne okozzunk gondot a többi felhasználónak. Így például *ne küldjünk e-mailt minden tanulónak*, ne használjuk iskolai címünket kereskedelmi vagy reklámtevékenységre, ne töltsünk le másokat sértő vagy illegális anyagokat. Lehetőleg *mindig ellenőrizzük, hogy az internetről származó programok és állományok vírusmentesek-e*. Hálózati ismereteinket pedig ne a hálózat feltörésével, hanem a rendszergazdai munkába való bekapcsolódással bővítsük.

Érintés- és balesetvédelem

A számítógépeket az iskolában *földelt csatlakozóval és érintésvédelmi kapcsolóval* üzemeltetik, ennek ellenére előfordulhat, hogy egy-egy vezeték vagy csatlakozó megsérül. Ilyen esetben feltétlenül értesítsük a felügyelő tanárokat.

Az iskolában *saját felelősségünkre se szedjük szét számítógépet vagy monitort*. Bizonyos áramköri elemek kikapcsolás után is tartalmazhatnak elektromos töltéseket, és megrázhathatnak. Sok olyan áramköri elem is van, amely viszont a testünkön felhalmozódó ún. statikus töltésektől sérülhet.

Jegyezzük meg, hogy *hol vannak a számítógépteremhez tartozó elektromos kapcsolók és biztosítékok*. Munka végén a számítógépet *mindig szabályosan állítsuk le és áramtalanítsuk*.

Nagy mennyiségű elektromos eszköz esetén óhatatlanul előfordul, hogy egy-egy eszköz meghibásodik, esetleg „leég”, azaz kellemetlen füstszag keletkezik. Ilyen esetben a számítógépet áramtalanítsuk.

Fontos tudni, hogy *az elektromos tűz nem oltható vízzel, és a poroltó is igen nagy kárt okozhat*, ám a speciális tűzoltó berendezés kötelezően megtalálható a teremben vagy annak környékén.

Ha a számítógépet, a monitort vagy más eszközt át kell helyezni, ügyeljünk arra, hogy *szállítás közben a vezetékek ne lógnanak*, s az eszközt szabályosan fogjuk meg.

A számítástechnikai eszközök általában kényesek a tisztaságra, emiatt a számítógépteremben *ne táplálkozzunk*.

Gyakori, hogy a golyós egér elkoszoldódik, ezért érdemesebb megtanulni a tisztítását, mint fölösleges csapkodással tönkretenni. A legtöbb hálózatban ezt a karbantartási műveletet a felhasználók is elvégezhetik, de a többi *hiba kijavítását vagy az eszközök cseréjét bizzuk a rendszergazdákra*.



A golyós egér tisztítása általában a felhasználók feladata.

Az egeret megfordítjuk, a golyó fedelét letekerjük (az irány általában szerepel rajta), majd a golyót kivesszük. Az ábrán a nyílakkal megjelölt két vékony görgőre tapadt koszt eltávolítjuk.

Az információ és mérése

Információ, adat, jel

A tágabb értelemben vett informatika az információ keletkezésével, továbbításával, tárolásával és feldolgozásával foglalkozik, tehát igen hatalmas területet fed le.

Az **információ*** olyan új ismeret, amely megszerzője számára szükséges, és korábbi tudása alapján értelmezhető. Szokták ezt úgy is fogalmazni, hogy az információ valamely meglévő bizonytalanságot szüntet meg. Egy középiskolás tanuló részére nem információ, hogy mikor volt a mohácsi vész, hiszen számára ez nem új ismeret. Ugyanez egy afrikai bennszülött számára szintén nem információ, mert nem tudja értelmezni.

Az **információ hordozója a jel***. A jel igen tág fogalom, a készülő ebéd illata éppúgy információt közöl, mint egy közlekedési jelzőtábla.

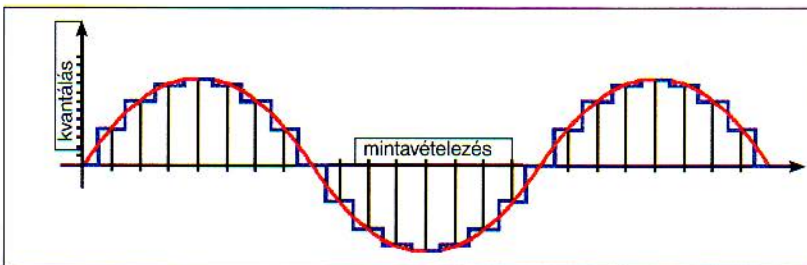
A szűkebb értelemben vett informatikán a számítógépes információfeldolgozást értik. Világos, hogy itt az információ speciális módon jelenik meg. A **számítástechnikai eszközökkel rögzített, azokkal feldolgozható és megjeleníthető információt adatnak*** nevezzük.

Az analóg és digitális jel

Egy jelet **analógnak*** nevezünk, ha két érték között tetszőleges értéket felvehet. Ilyen például az ember magassága vagy tömege. A **digitális jel*** csak előre meghatározott számú véges sok értéket vehet fel. Ilyen például egy felnőtt ember gyerekeinek a száma, vagy a boltban kapható pólók mérete. Ugyanez az információ megjelenhet akár mindkét módon is, gondoljunk pl. az analóg és a digitális órákra.

Az **analóg jeleket mintavételezéssel és kvantálással alakíthatjuk digitális jelekké**. A folyamatot jól nyomon követhetjük például a hang esetén.

A **mintavételezés*** azt jelenti, hogy az időben folytonosan változó hang értékeit csak bizonyos időközönként rögzítik. Az időköznek elég kicsinek kell ahhoz lennie, hogy a kapott mintát még folytonosnak halljuk, azaz „becsaphasson bennünket”. A mai hangkártyák másodpercenként 44 100 ... 96 000 között vesznek mintát.



Analóg jel és mintavételezéssel kapott változata. Az analóg jelet az x tengely mentén mintavételezik, majd az y tengely mentén kvantálják.

Az informatika alapjai

A **kvantálás*** azt jelenti, hogy a mért értékek tartományát véges sok egyenlő részre bontják, és a mintát az így kapott értékekhez kerekítik. Természetesen az eredeti és a kvantálaskor kapott értékek közötti eltérésnek olyan kicsinek kell lennie, hogy azt fülünk még ne vegye észre. A mai hangkártyák általában $2^{16} = 65\,536$ kvantálási szintet használnak.

Képek esetén a digitalizálás hasonlóan történik, csak ott nem bizonyos időközönként, hanem egymástól bizonyos távolságra vesznek mintát. A digitalizált képek jellemzőire a képfeldolgozással foglalkozó fejezetben térünk vissza.

A bináris számrendszer

A **bináris*** digitális jelek csupán két értéket vehetnek fel. Az így kapott rendszerek nagyon egyszerűek és rendkívül hibatűrőek. Tegyük fel például, hogy érkezik egy 0,5 V; 1,2 V; 4,8 V; 0,2 V elektromos jelsorozat. Ha tudjuk, hogy a két lehetséges érték 0 V és 5 V, akkor az eredeti értékek helyreállítása automatikus: 0 V; 0 V; 5 V; 0 V.

A fentiek miatt a számítógépek és a legtöbb informatikai eszköz (DVD-lejátszó, mobiltelefon stb.) binárisan kezelik az adatokat. A **bináris jelek matematikai leírásához a kettes számrendszert használják**, vagyis a két állapotnak a 0-t és az 1-et feleltetik meg. A **kettes számrendszerbeli számjegyeket biteknek*** (*binary digit* [bájnöri didzsit]) nevezik.



10 ujjunk segítségével a kettes számrendszerben 1024-ig tudunk elszámolni. Próbáljuk ki!

Ma a tízes számrendszert használjuk, ahol a számokat helyiértékesen írjuk fel:

$$3567 = 3 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0.$$

Hasonlóan írható fel egy négyjegyű (4 bites) szám a kettes számrendszerben:

$$1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11.$$

Az átalakításból leolvasható, hogy a helyiértékes számolás mindjárt megadja a lehetőséget az átváltásra is.

A kettes számrendszerben a műveletek elvégzése igen egyszerű. A **szorzótáblához** például mindössze 4 értéket kell megjegyeznünk: $0 \cdot 0 = 0$, $0 \cdot 1 = 0$, $1 \cdot 0 = 0$, és $1 \cdot 1 = 1$

Az **összeadás** kicsivel bonyolultabb, mert $1 + 1$ esetén átvitel is van: $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, és $1 + 1 = 10$.

A kettes számrendszerben végzett szorzás és összeadás szoros kapcsolatban van a matematikai logika ÉS, illetve VAGY műveletével. Megvalósításukra a kapuáramkörök kapcsán térünk vissza.

$$\begin{array}{r} 1101 \cdot 1001 \\ 1101 \\ 0000 \\ 0000 \\ \hline 1101 \\ 1100101 \end{array}$$

Szorzás kettes számrendszerben

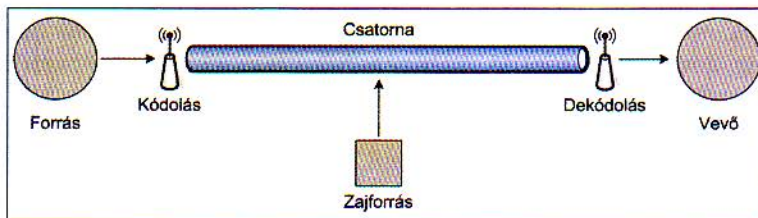
Az információátadás (kommunikáció) folyamata

Az információátadás az informatikai eszközök esetében (de lényegében az emberek esetében is) leegyszerűsítve a következőképpen épül fel.

A **forrás*** üzenetet vagy üzenetek egész sorát állítja elő, melyet továbbítani kíván a **vevőnek***. Ez az üzenet lehet pl. szöveg, kép, hang stb.

Az üzenet továbbítása a **csatorna*** segítségével történik. Ehhez az üzenetet a forrás oldalán úgy kell átalakítani, hogy azt a csatorna továbbítani tudja (**kódolás***), majd a vevő oldalán vissza kell alakítani (**dekódolás***). A csatorna által továbbított jelsorozatot **közleménynek*** nevezik. Fontos, hogy míg az üzenet maga a továbbítandó tartalom, addig a közlemény „csupán” az ennek megfelelő, a csatornán továbbított jelsorozat.

A csatorna tehát az a közeg, amely az információt továbbítja. Ez lehet kéteres kábel, lyukkártya, rádióhullám, nyomtatott papír stb. A csatornában a közlemény óhatatlanul sérül, az információhoz **zaj*** adódik hozzá, gondoljunk pl. a vibráló képernyőre, a szagató mobiltelefonra vagy a megsérült papírra.



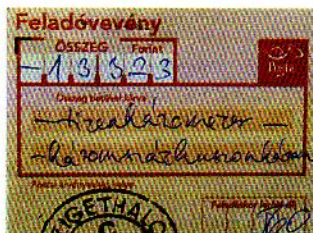
Az információátadás folyamata.

Hogyan valósul meg az ábra szerinti folyamat telefonálás közben?

A csatorna zajosságát nem a zaj mennyiségével, hanem a jelnek a zajhoz viszonyított értékével, a **jel/zaj aránnyal*** mérik. A hányados – változatlan zaj mellett – jelentősen javítható a jel erősítésével, például mobiltelefonok esetén a térerő növelésével.

A közleményben meglévő, újabb információt már nem adó elemeket **redundanciának*** nevezzük. Az emberi nyelv rendkívül redundáns: „Sok beszédnek sok az alja”. A redundancia elhagyásával a közlemény **tömöríthető**, így gyorsabban és olcsóbban továbbítható. A redundancia azonban nem feltétlenül káros jelenség. Megkönnyíti a közlemény értelmezését, illetve lehetővé teszi **ellenőrzését**, a hibák **javítását**. A postai csekken például a feladott összeget betűkkel is ki kell írni: a szöveg ugyanis a számokkal ellentétben nehezen hamisítható.

Amikor információt továbbítanak, szükségszerűen felmerül, hogy mekkora kapacitású csatornára van ehhez szükség: az információ mennyiségét valahogy mérni kell. Mivel a csatorna informatikai eszköz, ezért nem az információ tartalmát (az üzenetet), hanem a továbbított jelsorozatot (vagyis a közleményt) mérik.



Mi a redundancia szerepe a postai csekken?

Az információmennyiség

Először vizsgáljuk meg azt az esetet, amikor a közlemény egyetlen jeltől áll.

A legegyszerűbb esetben a jel kétféle lehet (pl. férfi vagy nő, elkészült-e az ebéd vagy sem, stb.), aminek kódja tehát 0 vagy 1. Így a közlemény egyetlen számjeggyel leírható. Ha a jel nyolcféle lehet, akkor a kód matematikai leírásához már háromjegyű szám szükséges: 000, 001, 010, 011, 100, 101, 110, 111. A közlemény az első esetben 1 bites, a második esetben pedig 3 bites.

Mivel a kettes számrendszerben két számjegy van, ezért 2 db egyjegyű; $2 \cdot 2$, azaz 4 db kétjegyű; $2 \cdot 2 \cdot 2 = 8$ db háromjegyű, tehát 2^n db n -jegyű szám lehetséges. Más szavakkal: egy biten 2, két biten 4, 8 biten $2^8=256$, n biten 2^n lehetőség ábrázolható.

Összefoglalva: az információ mértékegysége a bit*. Ha a jelszótunk 2^n elemű, akkor az egyetlen jeltől álló közlemény információmennyisége n bit.

Vizsgáljuk meg azt az esetet is, amikor a jelszót elemszáma nem a kettő hatványa. Ha a továbbítandó üzenet egy adott hónap neve, akkor ez 3 bites közleménnyel nem oldható meg, mivel az ehhez tartozó jelszót csak 8 elemű. Tehát 4 bites közleményre van szükség, ami némileg pazarló, mert nem használ ki minden lehetséges esetet. Az informatikában ezért a jelszót szinte mindig úgy választják meg, hogy 2^n eleme legyen. Például tipikus dolog, hogy a **karakterkészlet 256 elemű, azaz 8 biten ábrázolható**, a színeket viszont 24 biten tárolják, ami $2^{24} \approx 16,7$ millió színt jelent.

Ha a közlemény több jeltől áll, akkor a közlemény információmennyisége jelenként egyszerűen összeadódik. Például, ha a továbbítandó közlemény egy 100 karakterből álló szöveg, és egy-egy karaktert 8 biten ábrázolunk, akkor a továbbítandó információmennyiség $100 \cdot 8 = 800$ bit.



Az 1890-es népszámlálás adatait lyukkártyával rögzítették az Egyesült Államokban. A bal oldali ábrán az üzenet kódolásához használt lyukasztót látjuk egy lyukkártyával. A jobb oldali ábrán a lyukkártyaolvasó látható: ahol a tű átcsúszik a lyukon, ott a megfelelő óra egyet lép. A népszámlálás adatait néhány hét alatt feldolgozták, a rendszer kidolgozója, Herman Hollerith [holerit] erre a szakmai és üzleti sikerre alapozva hozta létre a későbbi IBM-et [ájbiem].

Az információmennyiség mértékegységei

A bit rendkívül kicsi egység, hiszen már egy betű kódolásához is 8 bit szükséges. Kézenfekvő választás volt ezért többszöröseként 8 bites egységet választani: **1 bájt* = 8 bit**. (Angolul a bájtot *byte*-nak írják, és a *by eight* [báj éjt], azaz nyolcasával kifejezésből származik.) Ennek megfelelően egy formázatlan szöveg annyi bájtos, ahány karaktert tartalmaz.

A bájt többszöröseire kétféle rendszer alakult ki:

- Az SI mértékrendszerben használt szokásos többszörösök (kilo-, mega-, giga-, ...) rendre ezerszeres szorzót jelentenek. **1000 bájt = 1 kilobájt***, **1000 kilobájt = 1 megabájt***, **1000 megabájt = 1 gigabájt***, a megfelelő rövidítések pedig rendre B, KB, MB, GB stb. Ezt elsősorban a hardvergyártók alkalmazzák, pl. a merevlemezek kapacitásának megadásánál (1 GB = 1 000 000 000 B).
- Mivel a számítógép kettes számrendszerben számol, számára nem a tíz, hanem a kettő hatványai természetesek. Mivel $2^{10} = 1024$, így ezt a váltószámot használva mind az ember, mind a gép elboldogul. Ebben az esetben már nem használhatók az SI rendszer elnevezései, ezért újakat vezettek be: **1024 bájt = 1 kibibájt***, **1024 kibibájt = 1 mebibájt***, **1024 mebibájt = 1 gibibájt***, a megfelelő rövidítések pedig rendre B, KiB, MiB, GiB stb.

A „kibi”, „mebi”,... elnevezéseket a Nemzetközi Elektrotechnikai Bizottság vezette be 1999-ben. (A „bi” szótöredék értelemszerűen a kettes számrendszerre utal.) Ez az elnevezés lassan terjed, sokan ma is előszeretettel használják az SI többszörösöket 1024-es szorzóval.

Adattovábbításkor azt adják meg, hogy a csatorna másodpercenként hány bitet továbbít, tehát **az adatátvitel mértékegysége a bit/s***. Természetesen ennek is többszöröseit használják. A hagyományos telefonvonalon elvileg 56 Kbit/s sebesség érhető el. Az egyre jobban terjedő ADSL 384 Kbit/s sebességgel indul, míg az üvegszál optikai kábelek átviteli sebessége több száz Mbit/s.



A kétkötetes Jókai regény összesen 640 oldal tartalmaz, oldalanként 33 sorral, soronként 50 karakterrel. Ez kb. 1 MB információ, ami bőven ráfér egy flopi-ra is.

Összehasonlításképpen: egy számítógép kb. 128-512 MB adatot tárol a memóriájában, míg egy CD-re kb. 640 MB, egy DVD-re minimum 4,7 GB adat fér. Ez utóbbi már képes egy több órás film tárolására is.

Becsüljük meg, hogy mennyibe kerül egy betű tárolása a könyv, a flopi, illetve a CD lemez esetében!

Az adatok tárolása

Az információ legkisebb egysége a *karakter* (betűhely), amely a betűk, számjegyek, írásjelek összefoglaló neve. (Ne keverjük az információ egységét, a karaktert, az információmennyiség mértékegységével, a bittel!) Az információ legkisebb, önállóan még értelmes részét *elemi adatnak** nevezzük. Hagyományosan az elemi adatokat három csoportba soroljuk: ezek a **szöveg, a szám és a logikai típusok**.

Hogyan lehet a számítógépen tárolni a nem szám jellegű adatokat, például a karaktereket? A megoldás módja a kódolás. *Kódolás** során egy jelkészlet elemeit rendre megfeleltetjük egy másik jelkészlet elemeinek. A karakterek esetében például minden karakternek megfeleltetünk egy számot, és a számítógép ezt a számot tárolja, illetve továbbítja.

A kódolás fogalma nem a számítógépekkel jelent meg: a Morse-ábécét, vagy a különböző titkosírásokat már jóval korábban is használták.

A karakterek kódolása

A karakterek kódolására sokféle kódkészlet alakítható ki, de ma alapvetően két kódrendszer terjedt el: az ASCII [aszki] és a UNICODE [junikód].

Az *ASCII kódkészlet** eredetileg 7 bites volt, azaz $2^7 = 128$ karaktert tartalmazott. Mivel ez kevésnek bizonyult (nem tartalmazza pl. a nemzeti karaktereket, így a magyar ékezetes jeleket sem), ezért később 8 bitesre bővítették, így ma **256 jelet tartalmaz**.

000 NUL	016 DLE	032 SP	048 0	064 @	080 P	096 `	112 p
001 SOH	017 DC1	033 !	049 1	065 A	081 Q	097 a	113 q
002 STX	018 DC2	034 "	050 2	066 B	082 R	098 b	114 r
003 ETX	019 DC3	035 #	051 3	067 C	083 S	099 c	115 s
004 EOT	020 DC4	036 \$	052 4	068 D	084 T	100 d	116 t
005 ENQ	021 NAK	037 %	053 5	069 E	085 U	101 e	117 u
006 ACK	022 SYN	038 &	054 6	070 F	086 V	102 f	118 v
007 BEL	023 ETB	039 '	055 7	071 G	087 W	103 g	119 w
008 BS	024 CAN	040 (056 8	072 H	088 X	104 h	120 x
009 HT	025 EM	041)	057 9	073 I	089 Y	105 i	121 y
010 LF	026 SUB	042 *	058 :	074 J	090 Z	106 j	122 z
011 VT	027 ESC	043 +	059 ;	075 K	091 [107 k	123 {
012 FF	028 FS	044 ,	060 <	076 L	092 \	108 l	124
013 CR	029 GS	045 -	061 =	077 M	093]	109 m	125 }
014 SO	030 RS	046 .	062 >	078 N	094 ^	110 n	126 ~
015 SI	031 US	047 /	063 ?	079 O	095 _	111 o	127 DEL

A 7 bites ASCII (American Standard Code for Information Interchange [amerikai sztenderd kód for informésn intercésndzs]) kódtábla. Pl. a 13-as kódú karakter a bekezdés vége, a 32-es a szóköz. Milyen számsor kódolja az ALMA, illetve az Alma szót?

Az eltérő nemzeti karakterek nagy száma miatt többféle (egyenként szabványosított) kibővített kódtáblát hoztak létre, melyek közös jellemzője, hogy az „alsó” 128 karakter egységes, a „felső” 128 karakter pedig eltérő. Az így kapott *kódlapokat** számozással különböztették meg, pl. az amerikai nemzeti kódlap a 437-es, a közép-európai (magyar) a 852-es számot kapta.

Sajnos a kódlapok bevezetése csak helyi megoldást hozott, hiszen ugyanaz a kód nyelvenként eltérő karaktert azonosít (pl. a 138-as kódú karakter a 852-es kódtáblában az Ő, a 437-esben az è, míg a cirill karakterkészletet tartalmazó 855-ösben az i). Az egyik országban helyesen kódolt üzenet így egy másik országban dekódolva más üzenetet eredményez. A problémát a **UNICODE*** bevezetésével oldották meg, amely **lényegében valamennyi nép valamennyi jelét tartalmazza**. Természetesen ennek ára van, a UNICODE **16 bites**, így a tárolás és továbbítás költségeit duplájára növeli.

A számok ábrázolása

Az egész számok ábrázolása kettes számrendszerben történik. A valós számok tárolására viszont kétféle lehetőség alakult ki: a fixpontos és a lebegőpontos.

Fixpontos számábrázolásnál* a számjegyek száma és a tizedesvessző helye rögzített. Ezeket a számokat lényegében egész számként kezelhetjük, csak meg kell jegyezni a tizedesvessző helyét. A pénztárgépeken és a benzinkutak kijelzőin is ilyen számokkal találkozunk.

Ha a tárolandó számok nagyságrendje nagyon eltérő, akkor a lebegőpontos számábrázolást használják. A **lebegőpontos számábrázolás*** a matematikaóránról ismert **normálalak kettes számrendszerbeli megfelelője**. (Például a Föld tömege normálalakban $5,973 \cdot 10^{24}$ kg, az elektroné pedig $9,109 \cdot 10^{-31}$ kg.)



A benzinárakat fixpontosan jelzi a kút oszlopa. Adjuk meg ezeket a számokat normálalakban is!

Logikai műveletek és kapuk

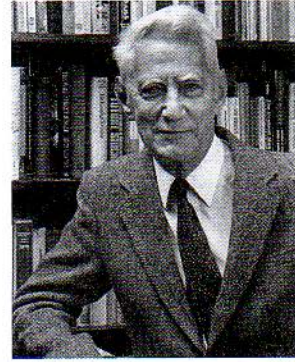
Logikai értékek

Az emberi gondolkodás és a számítógépen használt kettes számrendszerbeli műveletek között a matematikai logika teremti meg a kapcsolatot.

A **matematikai logika az állítások tényleges tartalmával nem foglalkozik, csak azzal, hogy az adott állítás igaz vagy hamis. Ezt az adott állítás logikai értékének* nevezzük.** Például a $2 < 3$ állítás logikai értéke igaz, míg a $2 = 3$ állításé hamis.

Fontos, hogy **minden állítás esetében eldönthető annak logikai értéke**. Ha egy kijelentés esetében ez nem tehető meg, akkor az nem tartozik a matematikai logika tárgyához. Ilyen állítások pl. a „Kati szép lány”, ami meglehetősen szubjektív, vagy az „Ez a mondat hamis” paradoxon.

Kézenfekvő a kapcsolat a kettes számrendszer és a matematikai logika között: az igaz logikai értéknek az 1, a hamis logikai értéknek a 0 feleltethető meg. Ez mindjárt rendezettséget is teremt közöttük: **hamis < igaz**.



A logika tudományos megalapozása Arisztotelész (Kr. e. 384-322) nevéhez fűződik (balra). A matematikai logikát George Boole [dzsordzs búll] (1815-1864) dolgozta ki, aki autodidakta úton vált tudóssá (középen). Boole eredményeit Claude E. Shannon [klód senon] (1916-2001), az információelmélet megalapozója használta először áramkörök tervezéséhez (jobbra).

Logikai műveletek

A logikai állításokkal különböző műveleteket végezhetünk. Ilyen művelet az állítás tagadása, vagy két állítás összekapcsolása az ÉS, illetve a VAGY kötőszóval. Az összetett állítás logikai értékét a matematikai logika az eredeti állítás tényleges tartalmától függetlenül definiálja.

Negáció* (tagadás). Az állítás tagadását a nyelvben a NEM szócskával, a matematikai logikában a NOT szócskával vagy a \neg jellel jelöljük.

Például, ha az állítás: A = „Az autó színe piros”, akkor a tagadása: NOT A vagy $\neg A$. Az A állítás egy kék autó esetében hamis, egy piros esetében igaz.

Értelemszerűen, **ha az eredeti állítás igaz, akkor tagadása hamis, és fordítva.**

Konjunkció (ÉS művelet). Ha két állítást ÉS kötőszóval kötünk össze, akkor a két állítás **konjunkciójáról* beszélünk.** A konjunkció jele az AND [end] szócska vagy a \wedge jel.

Például, ha A = „Az autó színe piros”, illetve B = „Az autó típusa Suzuki”, akkor az összetett állítás A AND B vagy $A \wedge B$. Az összetett állítás ezúttal hamis a piros Opel és a kék Suzuki esetében is, nemcsak a zöld Fordnál.

Definíció szerint **a konjunkció logikai értéke pontosan akkor igaz, ha mindkét állítás igaz, egyébként hamis.**

Diszjunkció (VAGY művelet). Ha két állítást VAGY kötőszóval kötünk össze, akkor a két állítás **diszjunkciójáról* beszélünk.** A diszjunkció jele az OR szócska vagy a \vee jel.

Például, ha A = „Az autó színe piros”, illetve B = „Az autó típusa Suzuki”, akkor az összetett állítás A OR B vagy $A \vee B$. Az összetett állítás igaz a piros Opel és a kék Suzuki esetében is, viszont a zöld Ford esetében már nem.

Definíció szerint **a diszjunkció logikai értéke pontosan akkor hamis, ha mindkét állítás hamis, egyébként igaz.**

Sajnos nyelvünk nem teljesen következetes az ÉS, illetve a VAGY használatát illetően. Például az „Én az almát és a szőlőt szeretem” kijelentés esetében az ÉS szócska a logikai VAGY-nak felel meg (egy gyümölcs nem lehet egyszerre alma és szőlő is).

A diszjunkció megengedi, hogy a két állítás egyszerre legyen igaz. Ez azonban nyilvánvalóan nem lehetséges, ha az „Este 8-kor leckét írtam” és az „Este 8-kor színházban voltam” állításokat kapcsoljuk össze. A VAGY kötőszóval ezért egy másik műveletet is definiálhatunk, ez a kizáró VAGY (*antivalencia*). **A kizáró VAGY akkor igaz, ha a két állítás közül pontosan az egyik igaz.** A számítástechnikában általában a XOR szócskával jelölik: $A \text{ XOR } B$.

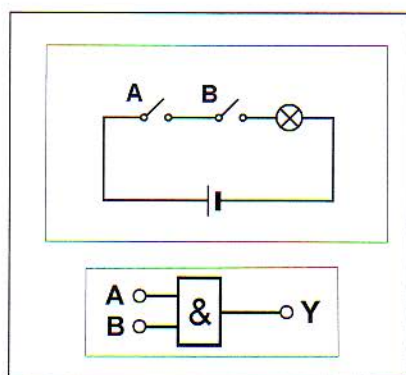
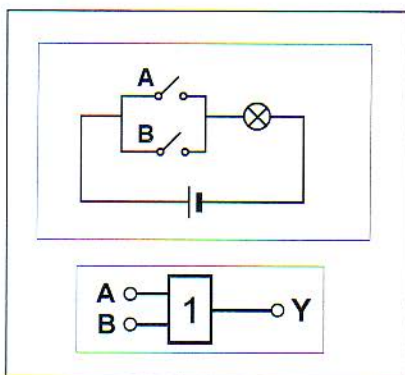
A	B	A AND B	A OR B	A XOR B
1	1	1	1	0
1	0	0	1	1
0	1	0	1	1
0	0	0	0	0

A logikai műveletek definícióját az ún. igazságtáblával foglalhatjuk össze.

A kapuáramkörök

A gyakorlati életben előforduló digitális áramkörök működése mindig leírható a logikai műveletek segítségével. Például a liftnek akkor kell elindulnia fölfelé, ha megnyomták egy felsőbb emeleten a hívógombot, ÉS be van csukva az ajtó.

Az ÉS, illetve a VAGY műveletek áramköreit két kapcsolóval és egy izzóval akár házilag is elkészíthetjük. **Az alapvető logikai műveleteket megvalósító áramköröket kapuáramköröknek nevezik.** Ilyen kapuáramkörökből épülnek fel a számítógép áramkörei is, „csupán” az elemek száma rendkívül nagy. Azt, hogy igen kis helyre igen nagy mennyiségű áramkört tudjanak elhelyezni (esetenként négyzetcentiméterenként akár egy milliót is), a félvezetők (pl. szilícium) tulajdonságai teszik lehetővé. A félvezetőkkel és alkalmazásukkal (pl. tranzisztorok) később a fizikai tanulmányaink során fogunk megismerkedni.



A VAGY, illetve az ÉS kapu egyszerű megvalósítása (felül) és szabványos jelölése (alul).

Összefoglaló kérdések, feladatok

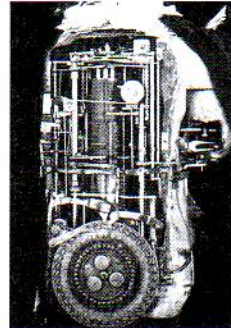
1. Milyen részekből áll egy iskolai hálózat? Milyen előnyei vannak a helyi hálózat kialakításának?
2. Mire kell ügyelnünk az iskolai hálózat használata során, hogy ne veszélyeztessük mások munkáját?
3. Milyen érintés- és balesetvédelmi szabályokat kell betartanunk a számítástechnika-teremben?
4. Mit jelentenek a következő fogalmak: információ, jel, adat? Mondjunk mindegyikre egy-egy példát a hétköznapi életből!
5. Mi a digitális és az analóg jel között a különbség? Mennyiben speciális a bináris digitális jel? Mi a bináris digitális jel előnye az analóg jellel szemben?
6. Mutassuk be a kommunikáció folyamatát egy példán! Hol jelenik meg a zaj?
7. Hány bit információ szükséges az év egy adott napjának meghatározásához?
8. Körülbelül hány bájt lehet a földrajzkönyvünk információtartalma képek nélkül? Adjuk meg az eredményt KB-ban és KiB-ban is!
9. Egy flopi, egy CD és egy DVD kapacitása rendre 1,44 MB, 650 MB és 4,7 GB. Hány flopi tartalma fér rá egy CD-re, illetve egy DVD-re?
10. Milyen előnyei és hátrányai vannak a UNICODE-nak az ASCII kóddal szemben?
11. Milyen módon lenne érdemes ábrázolni a következő adatokat: matematikaosztályzat, benzinár, az elemi részek tömege, irányítószám, testsúly? Melyik a kakukktojás, és miért?
12. Mi a következő kifejezések logikai értéke?
 $(2 < 3) \text{ AND } ((3 < 4) \text{ OR } (3 > 4))$,
 $(2 < 3) \text{ OR } ((3 < 4) \text{ AND } (4 < 5))$?
13. Mondjunk példákat a hétköznapi életből a következő összetett állításokra:
A OR B, A AND B, NOT(A OR B), NOT A OR NOT B!
14. Milyen kapcsolatot találhatunk a kettes számrendszerbeli összeadás és szorzás, valamint a logikai műveletek között?

Az első számítógépek

Jóllehet mechanikus (fogaskerekekkel működő) számológépeket, óraművel vezérelt harangjátékokat és egyéb szórakoztató automatákat már a XVII-XVIII. században is készítettek, a programozható számológép iránti valós igény csak a XIX-XX. században jelent meg.



*Les automates
Jaquet Droz
à Neuchâtel*



A Jaquet-Droz [zsák-dróz] óráscsalád a XVIII. században több automatát is készített. Ezek közül a három leghíresebb: az írnok, az orgona-játékosnő és a rajzoló. A képeken az írnok látható, melyet 40 betűből álló tetszőleges szövegre lehet programozni. A három automata ma is működik, és a svájci Neuchâtel [nösátel] város múzeumában tekinthető meg.

A korabeli matematikai, hajózási, csillagászati táblázatokat kézzel számolták. Az elvégzendő műveleteket elemi lépésekre bontották (pl. egy-egy összeadás vagy kivonás), s egy-egy ember mindig ugyanazt a lépést végezte. Ezt a munkát gépesítette CHARLES BABBAGE [csárlz bebedzs] (1792-1871) angol matematikus, aki 1822-ben építette meg első mechanikus gépét (*Difference Engine* [diferenz endzsin]).

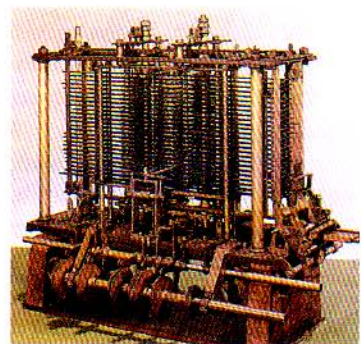
A gép sikerén felbuzdulva javasolta a lyukkártya vezérlésű (tehát programozható) változat elkészítését (*Analytical Engine* [enelitiköl endzsin]). Sajnos a gép a beleölt hatalmas összegek ellenére sem készült el, mert Babbage túlzott igényeket (50 helyi értékű számolás)



A *Difference Engine*
(1822)



Charles Babbage
(1792-1871)



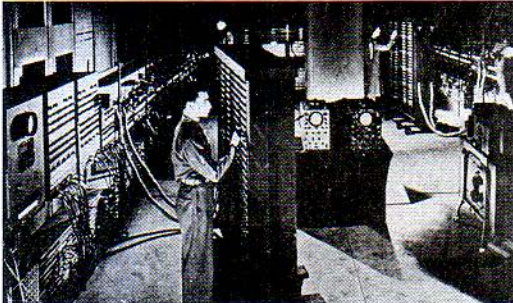
Az *Analytical Engine* makettje

támasztott a kor technikai színvonalával szemben. Ennek ellenére a modern számítógépek előfutárának tekinthető, mivel felépítésében már tartalmazta volna azok elemeit. (Bemeneti egység, vezérlőegység, feldolgozóegység, tárolás, kimeneti egység.)

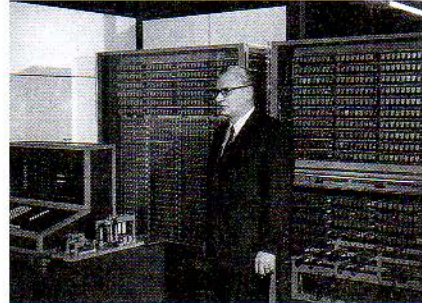
A Babbage terveihez közel álló *első mechanikus számítógépet*, a Mark I-et, HOWARD AIKEN vezetésével építette meg az IBM és a Harvard Egyetem 1939 és 1944 között. A gép gyakorlatilag hibátlanul működött 1959-ig.

Az *első elektronikus számítógépeket* a II. világháborúban készítették, többnyire a hadi számítások elvégzésére (lőtáblázatok készítése, titkosírás megfejtése). Természetesen a szemben álló felek között csak minimális információcsere volt lehetséges.

Az első működő elektronikus számítógépet Z3 [cet 3] néven KONRAD ZUSE [cuze] építette Németországban 1941-ben. Ő kapcsolóként reléket használt, így a gép egy műveletet kb. 3 s alatt hajtott végre. Zuse már a kettes számrendszert használta, a gép 64 db (22 bites) számot tudott tárolni.



Az ENIAC (1945). Mai fogalmaink szerint mekkora volt a memóriája?



Konrad Zuse az 1964-ben újraépített Z3-mal

Az Angliában fejlesztett *Colossus* 1943-ban készült el, és a reléknél lényegesen gyorsabb, ám kevésbé megbízható elektroncsövekkel dolgozott. Az 1500 elektroncsövet tartalmazó gép másodpercenként 25 000 karakter feldolgozására volt képes. ALAN TURING [elen tyuring] vezetésével (aki 1936-ban elsőként adta meg a programozható számítógép matematikai modelljét) sikeresen használták a rejtjelezett német rádióüzenetek megfejtésére.

A legismertebb korabeli számítógép az Egyesült Államokban kifejlesztett ENIAC [eniák], melyet lőtáblázatok számítására terveztek. A gép J. P. ECKERT [ekert], H. G. GOLDSTINE [goldstájn] és J. W. MAUCHLY [móslí] vezetésével 1945-ben készült el. Mintegy 18 000 elektroncsövet tartalmazott, így a relés megoldásnál lényegesen gyorsabb volt: másodpercenként kb. 5000 műveletet tudott elvégezni. A számjegyeket decimálisan tárolta, így egy-egy számjegy tárolásához 10 elektroncső volt szükséges. Érdeemes a félmillió dollárba került gép fizikai méreteire is kitérni: hossza 30 méter, magassága 3 méter volt, és 30 tonnát nyomott.

Az ENIAC programozása a régi telefonközpontokéhoz hasonlított, a programot egy huzalos dugaszolótableta segítségével alakították ki.

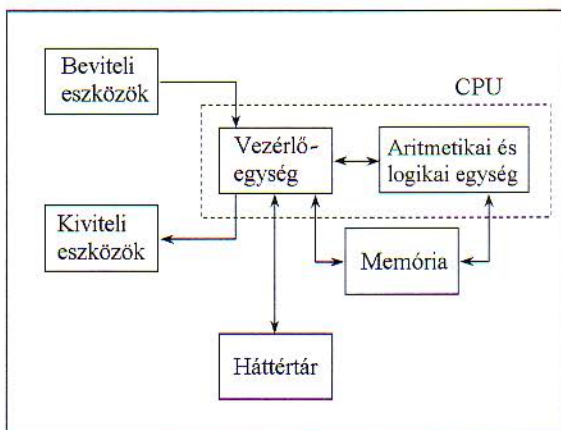
A számítógépek hamarosan a világ többi részén is megjelentek, pl. a Szovjetunióban az első számítógépet 1951-ben SZ. A. LEBEGYEV építette MESZM néven.

Hazánkban az első számítógépet, az M-3-at 1958-59-ben szovjet dokumentáció alapján építette a Magyar Tudományos Akadémia Kibernetikai Kutató Csoportja.

Az ENIAC kapcsán ismerkedett meg a számítástechnikával NEUMANN JÁNOS (1903-1957), aki kidolgozta a modern elektronikus számítógépek működésének alapelveit. Neumann János Budapesten született, és a farsori gimnáziumban érettségizett. Csaknem egy időben szerezte meg matematikusi diplomáját Budapesten és a vegyészmérnököt Zürichben. 1933-ig a németországi Göttingen híres egyetemén tanított, majd 1933-ban az Egyesült Államokba távozott, ahol a princetoni egyetemre került. Hamarosan bekapcsolódott a kibontakozó II. világháború miatt fontossá vált katonai kutatásokba, ennek kapcsán ismerkedett meg az ENIAC-kal. Később részt vett a hidrogénbomba kifejlesztésében is.

Neumann János a számítógépek további fejlesztésére vonatkozó javaslatait 1945-ben foglalta össze a „First Draft of a Report on the EDVAC” [főrszt dráft ov e riport on dö edvác] c. jelentésében. Ezek az ún. *Neumann-elvek** a következők:

- A számítógép legyen teljesen elektronikus,
- használjon kettes számrendszert,
- a programokat és az adatokat ugyanabban a belső tárban tárolja,
- alkalmazzon vezérlőegységet, amely a beírt utasításokat emberi beavatkozás nélkül egymás után végrehajtja. (Ez az elv megfelel a Turing által készített matematikai modellnek, ezért úgy is szoktak fogalmazni, hogy „legyen univerzális Turing-gép”.)



Neumann János és a Neumann-elvű számítógépek blokkvázlata.

Milyen területen ért el Neumann János jelentős tudományos eredményeket az informatikán kívül? Keressünk adatokat a könyvtárban vagy az interneten!

A mai számítógépek általában megfelelnek a Neumann-elv követelményeinek. Az ún. nem Neumann-elvű számítógépek (pl. „sejtautomaták”) párhuzamosan képesek műveleteket végezni. Az ilyen számítógépek még kísérleti stádiumban vannak.

A számítógép felépítése

A számítógép elektromos és mechanikus alkatrészeit *hardvernek** nevezik, míg a programokat és adatokat *szoftvernek**. Tekintsük át a számítógépek hardver jellemzőit!

A számítógépek csoportosítása

A legnagyobb teljesítményű számítógépek a **szuperszámítógépek**. Ezek valamilyen nagy számolási igényű célra (pl. tudományos kutatások, meteorológiai előrejelzések, filmeffektek) készült igen drága célszámítógépek.

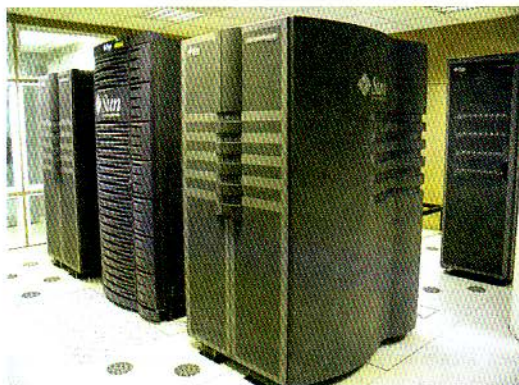
A **mainframe** [méjnfrem] számítógépeket általában nagyvállalati környezetben, a cég központi adatbázisának kezelésére, levelezésének lebonyolítására stb. használják. A mainframe gépekhez egyszerre sok felhasználó kapcsolódhat, a gép képes egy időben nagyon sok program futtatására.

A **miniszámítógépek** szintén több felhasználó egyidejű kiszolgálására alkalmasak, de a kis- és közepes vállalatoknál használják őket. Az asztali számítógépekből álló hálózatok sok helyről kiszorították őket.

Az **asztali számítógépek** (*desktop* [deszktop]) többnyire egy felhasználó kiszolgálására alkalmasak. Nagyon sokféle eszköz (periféria) csatlakoztatható hozzájuk, és áruk is rendkívül kedvező, ezért mind az otthoni, mind a munkahelyi környezetben széles körben elterjedtek.

A **hordozható számítógépek** (*laptop, notebook* [leptop, noutbuk]) felépítése, működése lényegében megegyezik a desktop gépekével, a hordozhatóság miatt azonban különleges kialakításúak, ezért drágábbak az ugyanolyan kapacitású asztali gépeknél.

A **PDA** (*Personal Digital Assistant* [perszönöl didzsitöl eszisztent]) vagy **pocketPC** [pákitpíszí] sok esetben nem nagyobb egy zsebszámológépnél. Kisebb mérete miatt a programok szerényebb képességűek, és többnyire ROM-ba vannak égetve. Sok esetben egybeépítik a mobiltelefonnal (smartphone [szmártfón], „okos telefon”).



Magyarország legnagyobb teljesítményű számítógépe az NIIF Iroda Szuperszámítógép Központjában található, és az akadémiai kutatásokat szolgálja. A számítógép 4+1 egységből áll, összességében 276 processzort és 360 GB memóriát tartalmaz (2005-ös adatok).

A hardver tagozódása

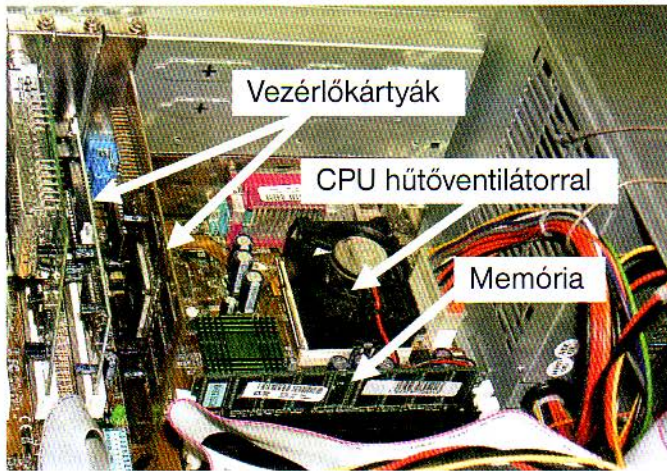
A hardver eszközöket két csoportba soroljuk, ezek a központi egység és a perifériák.

A **központi egység*** feladata a számítások végzése, a számítógép többi részének vezérlése, és az adatok tárolása a gép bekapcsolt állapotában. Az első két feladatot a **központi feldolgozóegység** (CPU) végzi, míg az utóbbit a **központi tár** látja el.

Fizikailag a központi egység a gép dobozában, az ún. **alaplapon*** helyezkedik el. Az alaplapon vezetékek sorozata köti össze a központi egység részeit, e vezetékek összessége a **busz**. A gép többi eleme az alaplapon lévő csatlakozókhoz (**slotok** [szlot]) egy-egy **vezérlőkártyának** nevezett áramkörrel csatlakozik. A kártyák kivezetéseit a gép hátoldalán a gép szétszedése nélkül is megtekinthetjük.

A **perifériák*** szerepe az adatok bevitelle (**beviteli eszközök**), megjelenítése (**kiviteli eszközök**), az adatok és programok tárolása (**háttértárak**), valamint a számítógépek közötti kapcsolat biztosítása (**kommunikációs eszközök**).

A **tápegység*** általában a számítógép házában található. A 230 V-os hálózati váltóáramból a számítógép számára szükséges 5 illetve 12 voltos egyenáramot állít elő. Áramkimaradás esetére a szerverek gyakran szünetmentes tápegységet (**UPS**) használnak.



Ha kinyitjuk a PC dobozát, az alaplapon megtekinthetjük a központi egységet és a vezérlőkártyákat.

A személyi számítógépek

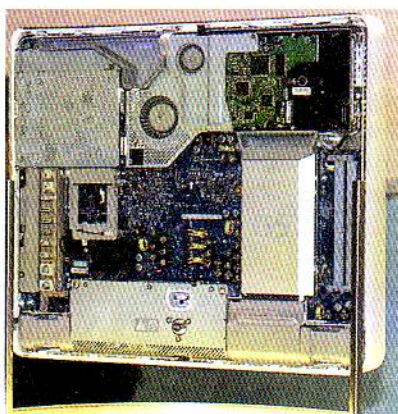
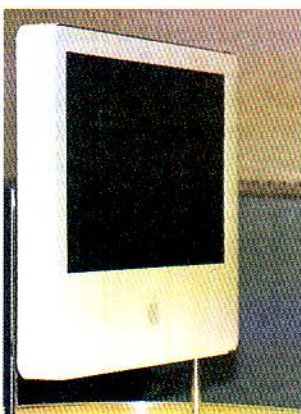
A személyi számítógépek kategóriájában alapvetően kétféle számítógép terjedt el. Üzleti célra elsősorban az **IBM kompatibilis** vagy egyszerűen **PC** (**Personal Computer** [pörszönöl kompjutör]) számítógépeket használják. Grafikai alkalmazásokra, és főleg a tengerentúlon az oktatásban is inkább az **Apple Macintosh** [epl mekintos] gépek terjedtek el.

Az informatika alapjai

A kétféle számítógép között jelentős a filozófiai eltérés. Macintosh gépeket kizárólag az Apple cég gyárt, a forgalomba hozott gépek pedig egyetlen adattal (pl. iMac G5) jellemezhetők, akár az autók vagy a mosógépek.

A PC-ket az IBM vezette be a piacra. Fő jellemzője a *nyitott architektúra*, vagyis a gép továbbfejleszhető, bővíthető alkatrészeinek egyszerű cseréjével vagy újabbak beépítésével. Alkatrészeket bárki gyárthat, ha az megfelel a kialakult szabványoknak, így a gépet sok esetben a vásárlás helyén igényeinknek megfelelően rakják össze. Egy PC esetében **konfigurációnak nevezük azoknak a hardverelemeknek az összességét, amelyekből a számítógép felépül.**

A fentiek egyik következménye, hogy a PC-k általában olcsóbbak, de a kevésbé márkás alkatrészekből gyakran házilag összerakott gépek kevésbé megbízhatóak. A továbbiakban főleg a hazánkban legelterjedtebb PC számítógépek jellemzőit tekintjük át.



Az ábrán látható Macintosh számítógép a kb. 17-18 cm vastag monitorba van beépítve.

A központi egység

A központi egység részeit az *órajel hangolja össze*. Az órajel szabályos időközönként érkező áramimpulzus, melyet az alaplapon lévő órajelgenerátor állít elő, és a központi egység alkatrészei ehhez igazodnak. Egyébként azonos paraméterek esetén a nagyobb órajel nagyobb sebességű gépet jelent. Vigyáznunk kell azonban arra, hogy különböző gépeket (pl. Macintosh és PC) csak az órajel alapján nem lehet összehasonlítani.

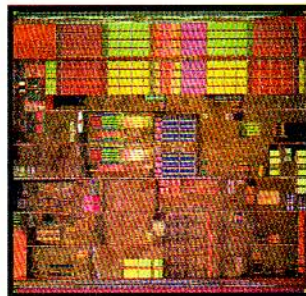
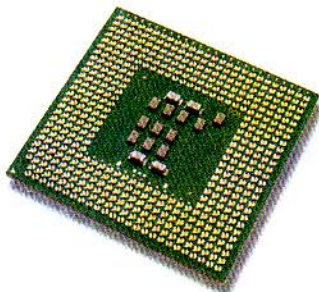
A központi feldolgozóegység

A *központi feldolgozóegység** (Central Processing Unit [szentrál processzing junit], CPU) a személyi számítógépek esetében egy integrált áramkör, amely három fő részből áll.

Az *aritmetikai és logikai egység* (ALU, Arithmetic and Logic Unit [aritmetikai és logikai junit]) a számításokat végzi, a *vezérlőegység* (CU, Control Unit [kontrol junit]) pedig a számítógépet vezérli. A *regiszterek* azokat az adatokat tárolják, amelyekkel az ALU, illetve a CU éppen dolgozik.

A processzorok jellemzésére általában két adatot szoktak megadni: az egyik az órajel nagysága, a másik, hogy hány bites számokkal tud számolni.

A mai processzorok többnyire 32 bitesek, de már megjelentek a 64 bites változatok is. Ez azt jelenti, hogy egy lépésben 32 (illetve 64) bites számokkal képesek műveletet végezni. Tipikus órajelük 2-4 GHz körül van. Természetesen a különböző felhasználási területekre eltérő processzorokat fejlesztettek ki, így másfajta processzor való a szerverekbe, illetve az otthoni és az irodai számítógépekbe. A laptopokba speciális, kisebb energiafelvételű processzorokat gyártanak. PC-kben használt processzorok például az *Intel Pentium* vagy az *AMD Athlon* család tagjai.



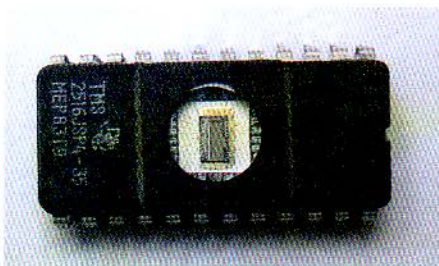
Egy mikroprocesszor alulról és belülről

A központi tár (memória)

A központi tár működése szempontjából két részre tagolódik, a ROM-ra és a RAM-ra.

A **ROM*** (*Read Only Memory* [ríd onli memori]) **csak olvasható memória, tartalmát a gép kikapcsolás után is megőrzi**, ezért a gép működéséhez nélkülözhetetlen adatok és programok vannak benne. Az IBM kompatibilis gépekben a gép működéséhez feltétlenül szükséges alapvető rutinokat a **ROM BIOS** (*Basic Input/Output System* [bézik input autput szisztem]) tárolja.

A ROM típusú memóriák egyik fajtája az ún. **EPROM**, amely alkalmas eszközzel írható (EPROM égető), ill. törölhető (ultraibolya fényvel, pl. kvarclámpa). A ROM egy másik



Az EPROM elektromosan írható, és a rajta lévő ablakon át ultraibolya fényvel törölhető.



A bélyeg méretű memóriakártyák kapacitása ma már elérheti az 1 GB-ot is.

változata a flash [fles] ROM, flash card, memóriakártya vagy *EEPROM*, melynek tartalma elektromosan törölhető is. A hordozható számítógépekben, mobiltelefonokban, digitális fényképezőgépekben, MP3 lejátszóknak stb. gyakran a winchesterek helyett használják, mivel kisméretű, és nem tartalmaz mozgó alkatrészeket. Vásárlásakor figyelniük kell arra, hogy többféle szabványa is létezik, s ezek eszközfüggők.

A **RAM*** (*Random Access Memory* [rendom ekszesz memori]) **írható és olvasható memória, tartalmát kikapcsoláskor elveszti.** Az itt elhelyezkedő adatok lényegesen gyorsabban érhetők el, mint a háttértárakon lévők, ezért az indítandó programot az operációs rendszer indulás előtt betölti a memóriába. A RAM egyik fajtája a lassabb, de olcsóbb *DRAM*, általában a számítógépek memóriáját ez alkotja. A gyorsabb, de drágább *SRAM*-ot inkább gyorsítótárként (*cache* [kes]) használják. A grafikus felületek terjedésével a gépek memóriaigénye megnőtt, 2005-ben tipikusnak mondható az 512 MB, sőt 1 GB méretű RAM is.

A *DRAM*-nak több változata van. A 90-es években használt *EDO-RAM*-ot mára teljesen kiszorította a lényegesen gyorsabb és nagyobb kapacitású *SD-RAM*, illetve annak továbbfejlesztett változatai, a *DDR-RAM*, valamint az *RD-RAM*. A különböző RAM verziók között nem elsősorban a működési elvben van eltérés, hanem a belső felépítésben és az adatok továbbításának módjában.

Beviteli eszközök

A **billentyűzet*** (*keyboard* [kíbord], *tasztatúra*) az egyik leggyakrabban használt, és talán leggyorsabb beviteli eszköz. Általában 101-105 gombot tartalmaz, a nemzeti írásjelektől függően többféle (amerikai, német, magyar stb.) változata van.

A **billentyűzet részei: a funkcióbillentyűk** (F1–F12), **az alfanumerikus blokk** (ez a rész tartalmazza a betűket), **a vezérlő blokk** (Insert, Home [houm] stb. gombok), **a kurzormozgató blokk** („nyílas” gombok) **és a numerikus blokk** (amely a Num Lock [nám lok] billentyű állásától függően kurzormozgató is lehet). Gyakori, hogy a billentyűzet felső részén a be- és kikapcsoláshoz vagy a böngészéshez további gombokat is elhelyeznek.



Braille-írást [brájl] használó billentyűzet és kijelző látássérültek részére. Az eszköz a képernyőn lévő adatokat Braille-írással jeleníti meg a felső részen lévő csíkon .

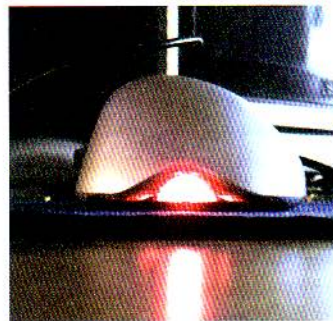
Figyeljünk oda arra, hogy a billentyűzet használatakor kezünk lehetőleg vízszintes helyzetben legyen, vagy inkább alacsonyabban!

Az **egér*** (*mouse* [mausz]) a grafikus felületek meghatározó beviteli eszköze („kattintott egérkor”). Az egérnek két fajtája terjedt el, az **optomechanikus egér alján egy golyó érzékeli a mozgást, míg az optikai egerek a kibocsátott és visszavert fény változása alapján tájékozódnak**. Terjednek a vezeték nélküli egerek is, amelyek távirányítókhoz hasonló módon infravörös fényvel vagy rádióhullámokkal tartják a kapcsolatot a számítógéppel.

Az egér helyes használatakor kezünk az egéren pihen, és a gombok számától függően ujjaink a gombokon fekszenek. Az egeret alkarunk megtámasztásával, csuklóból mozgatjuk. Az egér vásárlásakor figyeljünk oda arra, hogy mérete megfelelő legyen.



Az egér 1968. december 8-án született a Stanfordi [sztenford] Kutatóintézetben, ahol Douglas Engelbart [dáglesz engelbert] és csapata fejlesztette ki.



Optikai egér. Hasonlítsuk össze a golyós és az optikai egér használatának előnyeit és hátrányait!

További pozicionáló eszköz a **hanyattgér** (*track ball* [trek ból]), a golyós egér hanyatt fektetett változata, ilyenkor ujjainkkal közvetlenül a golyót görgetjük. Főleg laptopokon használják. Eredetileg pozicionáló eszköznek készült a **botkormány** (*joystick* [dzsojsztik]) is, amelyet ma már inkább játékeszközként ismerünk. A **fényceruza** (*light pen* [lájnt pen]) segítségével a képernyőn megjelenő adatok között választhatunk az adott rész színe alapján. Végül könnyű kezelhetősége miatt egyre jobban terjed az **érintőképernyő** (*touch [tács] monitor*), amely a fizikai érintés helyét jelzi a számítógép felé.

Képek bevitelére használjuk a **digitális fényképezőgépet***, illetve a **lapolvasót*** (*scanner* [szkenner]). Ma elsősorban az ún. síkágvas szkennerek használatosak, amelyek külső megjelenése a fénymásolóra hasonlít, „csupán” a behelyezett lapról valamilyen képformátumban fájl készül. Jellemző adataikra a grafikai fejezetben térünk vissza.

A számítógép előtt ülve a helytelen szék és asztal gerincferdülést, derékfájást, keringési zavarokat okozhat. Az asztal magassága akkor megfelelő, ha elé ülve talpunkat a földre tudjuk helyezni, és az asztalra téve kezünket, az alkarunk és felkarunk legalább 90 fokos szöget zár be. A szék pedig lehetőleg legyen forgatható, háttámlájának szöge és magassága pedig állítható. (Részletesebben ld. 50/1999 EüM rendelet.)

A monitorok

A monitorokat színkezelés szerint két csoportra bontjuk: a monochrom monitorok szí-
nét a felhasználó nem módosíthatja, míg a színes (color [kálör]) monitorok esetében vá-
laszthat a színekészletből, az ún. színpalettaból.

A monitorok* általában kétféle üzemmódot ismernek. *Alfanumerikus* módban
csak karakterek jeleníthetők meg, míg *grafikus módban* a képernyő képpontokra,
ún. **pixelekre van felosztva, és a képpontok egyenként is megjeleníthetők.**

A mai operációs rendszerek általában grafikus módban dolgoznak, és a 800×600-as
minimális felbontást várják el. A felhasználók zöme ténylegesen 1024×768-as vagy ennél
nagyobb felbontást használ.

A megjeleníthető színek száma: 16, 256, 65536 vagy 16,7 millió (*true color* [trú kálör]).
Ezek között az operációs rendszer beállításainak segítségével választhatunk.

A monitorok méretét a *képátmérővel* jellemzik. Ennek tipikus értéke ma 17 inch, de a
grafika és a képfeldolgozás terén ennél jóval nagyobb méretű monitorokat alkalmaznak.

Működés szempontjából kétféle monitort
használunk.

A *katódsugárcsőves** (CRT) monitorok a té-
vékészülékhez hasonlóan működnek: egy elekt-
ronnyaláb rajzolja fel belülről a képet. A moni-
tor *képfriessítési frekvenciája* mutatja meg, hogy
másodpercenként hány képet vált, ez 70 Hz alatt
már káros lehet a szemünkre.

A *TFT-monitoroknál** (*Thin Film Transistor*
[tin film tranzisztor] vagy aktív mátrix) minden
képponthez külön áramkört elem (tranzisztor)
tartozik, így éles, nem vibráló képet kapunk.
A korai TFT-monitoroknál a képpont színének
módosítása még lassú volt, ezért film nézésére
nem voltak alkalmasak. Ma már ezt a problémát
is sikerült kiküszöbölni.

A nyomtatók

A *nyomtatók** (*printerek*) az adatokat papíron jelenítik meg. A nyomtatók minőségét
dpi-ben mérik. A *dpi (dot per inch)* azt mutatja meg, hogy a nyomtató hány **különbö-
ző pontot tud elhelyezni egy 1 hüvelyk (2,54 cm) hosszúságú szakaszon.** A szokásos
irodai nyomtatók esetében ez az érték 600-1200 dpi körül van. A személyi számítógépek
esetében alapvetően négyféle nyomtatási eljárást használnak.

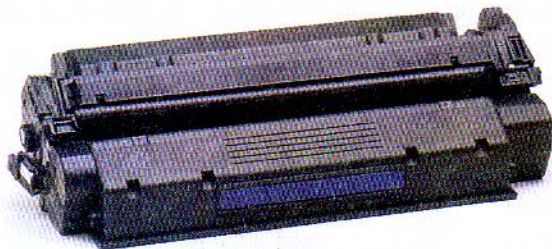
A *mátrixnyomtató** fejében tűk helyezkednek el. A papír előtt festékszalag van, a tűk
a festékszalagon át a papírra ütnek, ennek segítségével hagynak azon nyomot.

A mátrixnyomtatók helyét ma már átvették a lézer- és a tintasugaras nyomtatók. Mind
a mai napig használják azonban számlák és nyugták készítésére, mivel a mátrixnyomtató
többpéldányos nyomtatásra is alkalmas.

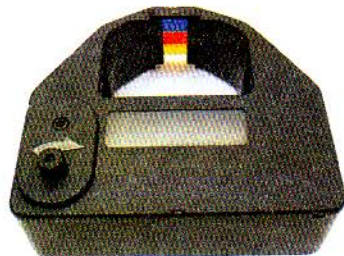


A PDA érintőképernyője egyszerre
beviteli eszköz is.

A **lézernyomtató*** (*laserjet* [lézerdzset]) működése hasonló a fénymásolóéhoz: az elektromosan feltöltött hengerre lézervény rajzolja fel a kívánt ábrát. Ahol lézervény éri a hengert, ott a töltése megváltozik, így ide feltapad a speciális festékpó, melyet a henger átad az alatta áthaladó lapnak. Végül a nyomtató fixáló része a festékpórt a lapra égeti 200-300 °C-on.



Lézernyomtató toner. A toner nemcsak a festékpórt, hanem a hengert is tartalmazza. Mivel környezetszennyező, nem szabad a kukába dobni.



A mátrixnyomtató festékszalagja is lehet színes, ekkor a fej háromszor megy végig a soron.

Bár színes változata is létezik, fő felhasználási területén, az irodai munkában ma még többnyire fekete-fehér lézernyomtatókkal találkozunk.

A **tintasugaras nyomtató*** (*ink jet*, ill. *bubble jet* [bábl dzset]) esetében a papír előtt egy fej mozog, amely apró tintacseppeket lő a papírra. A tintacsepp kilövése kétféleképpen történhet. A *termikus eljárás* esetében elpárologtatják a tinta egy részét, és az így keletkező gőz szorítja ki a cseppeket. A másik megoldás, hogy *piezoelektromos kristályt* használnak, amely az áram hatására hirtelen kitágul.

A színes tintasugaras nyomtatók speciális papírra akár fotó minőségű képet is tudnak nyomtatni. Irodai célra az otthoni változatoknál nagyobb kapacitású tintasugaras nyomtatókat fejlesztettek ki, melyek alapszínenként külön-külön tintapatront használnak.

A **hőnyomtatók*** esetében apró tűk melegítik fel a papírt, amely tartalmazza a hő hatására megváltozó színű festéket. Hátránya, hogy a papíron lévő információ egy idő után elhalványul. Főleg faxkészülékekben, pénzkidő automatákban és parkolóórákban alkalmazzák.

További kiviteli eszközök

A **rajzgépeket*** (*plottereket*) a mérnöki munkához használják. A toll egy csavarment segítségével vízszintesen vagy függőlegesen mozoghat egy papíron (síplotter), így a ferde vonal vízszintes és függőleges vonaldarabkákból áll. Másik megoldás, hogy a papírt függőlegesen mozgatják, és a toll csak vízszintesen mozoghat (dobplotter).

A **hangkártya*** segítségével digitális formában rögzített hangot, zenét játszhatunk le. A mai hangkártyák általában a zenét hífi minőségben képesek visszaadni, és gyakran a multiplex mozikban szokásos effektusokra is képesek. Többnyire össze vannak kötve a számítógépben lévő CD-ROM-mal, így alkalmasak az audio CD-k közvetlen lejátszására is.

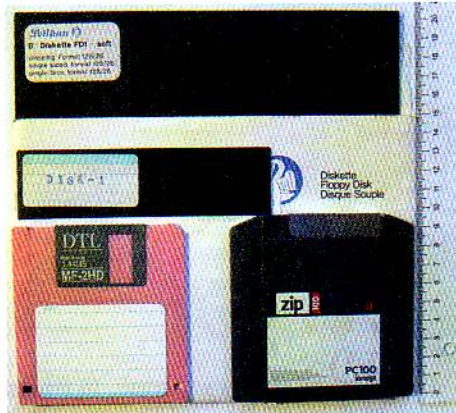
Mágneses elven működő háttértárak

A **flopi**, a **merevlemez** és a **szalagos táruk** mágneses alapon működnek. Ez azt jelenti, hogy a lemez vagy szalag mágnesezhető bevonatból készült felületét (pl. valamilyen fémoxid) az író-olvasó fej íráskor mágnesezi, olvasáskor pedig a felület mágnességének változását érzékeli.

A **hajlékonylemez*** vagy **flopi** (*Floppy Disc* [flopi diszk], *FD*) jelentősége a tárolókapacitása miatt csökken, ma már nem minden asztali számítógépben található meg a lemez írásához/olvasásához szükséges lemezegység (*Floppy Disc Drive* [flopi diszk drávj], *FDD*). A flopit elsősorban kisebb terjedelmű adatállományok tárolására és szállítására használjuk.

A lemezen **sávok*** (azaz koncentrikus körök) mentén helyezkednek el az adatok, a sávokat **szektorokra*** osztjuk. A lemeznek ez az ún. **logikai szerkezete** a **formázáskor** jön létre, melyet akár háziilag is elvégezhetünk. Ma már lényegében csak az 1,44 MB kapacitású, 3,5" átmérőjű HD jelzésű lemezeket használjuk, az ettől eltérő szabványú lemezekben lévő adatokat érdemes más adattárra másolni.

Az **a: drive** [a drávj] és a **zip drive** [cip drávj] a hajlékony lemez utódjának tekinthető. Működésük, felépítésük hasonló a flopiéhoz, ám a fej jobb pozícionálásával elérték, hogy a sávok száma sokszorosára nőtt, így kapacitásuk meghaladja a 100 MB-ot.



A hajlékonylemez mérete csökkent, miközben kapacitásuk nőtt. A képen látható lemezek mérete: 8, 5 1/4 illetve 3,5 inch. A bal oldali 3,5"-os flopi kapacitása 1,44 MB, a mellette lévő zip lemezé pedig 100 MB. Ma már ezt a technológiát kiszorította az optikai lemezek és a pendrive-ok elterjedése.



Merevlemez (*winchester*) szétszedett állapotban. A lemez fordulatszáma percenként 4500 és 12000 között van, az átlagos elérési idő (amíg a fej egy adott sáv fölé ér) 6 és 12 ms között.

A legfontosabb mágneses háttértár a **merevlemez*** (*winchester* [vincseszter], *Hard Disc Drive* [hárd diszk drávj], *HDD*), amely **programjaink és adataink folyamatos tárolására szolgál.**

A winchesterben több lemez lehet egymás felett, ezek közé nyúlnak be az író/olvasó fejek. A fejek együtt mozognak, így az egymás alatt lévő sávok egyszerre érhetők el. **A winchesteren az egymás alatt elhelyezkedő sávokat *cilindernek* nevezzük.** A fejeket a lemezek nagy sebességű forgatásakor kialakuló léggárna a lemezek fölött tartja, s így azok nem érnek hozzájuk. A fej és a lemezek távolsága nagyon kicsi, ezért a lemezeket szűrővel védett tokban tartják, hogy megóvják őket a különben bekerülő porszemek okozta sérülésektől. Mivel a winchester lemezei stabilan rögzítettek, így a sávok és szektorok sűrűbben helyezhetők el. A mai merevlemezek kapacitása általában meghaladja a 100 GB-ot, percnkénti fordulatszáma pedig 7200 körül van.

A merevlemez munkatár. A rajta lévő adatok a hardver- és szoftverhibák miatt, a vírusok károkozása vagy véletlen felhasználói törlés miatt könnyen megsérülhetnek. Célszerű ezért **az adatok folyamatos mentése (*archiválása*) más tárolóra is.**

A **mágnesszalagos egységeket ma már csak archiválásra használják.** A *streamer* [sztrímer] speciális, csak számítógépes egység, míg a *dat* a hangrögzítés eszköze is. Működési elvük a kazettás magnóéhoz hasonló, az adatok itt is kazettára kerülnek, bár a kétfajta kazetta paraméterei eltérőek.

Optikai elven működő háttértárak

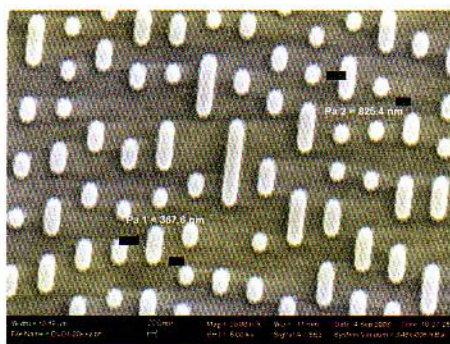
Az optikai táraknál a lemezek írása vagy olvasása vagy mindkettő optikai elven történik, erre lézertényt használnak. Működésük elve egyszerű: a felületre eső fénynyaláb vagy visszaverődik (logikai 1), vagy a felület tulajdonságai miatt szétszóródik (logikai 0), így az érzékelő vagy érzékeli a visszavert fénysugarat, vagy nem.

A csak olvasható optikai lemezek esetén a fényt kiemelkedések szórják szét, míg az írható lemezek esetében a felület elnyeli azt. A mágneses lemezektől eltérően az adatok spirálalakban vannak elhelyezve.

A **CD* (*Compact Disc* [kompakt diszk]) kapacitása 650-800 MB.** A meghajtók legfontosabb paramétere a sebessége, melyet a zenei CD-lejátszók sebességének többszöröseként adnak meg. Ez az érték ma tipikusan 40–50 körül van.

A **DVD* (*Digital Versatile Disc* [didzsitől vörstötáj] diszk]) a CD továbbfejlesztett változata, kapacitása minimum 4,7 GB.** A DVD-olvasók a CD lemezeket is kezelik.

A CD-ROM, illetve a DVD-ROM préseléssel, nagy tömegben, olcsón állítható elő. A CD-re 70-80 perc jó minőségű zenei felvétel, míg a DVD-re egy mozifilm is elfér, ezért mindkettő a megfelelő analóg eszközök (kazettás magnó, videomagnó) vetélytársa.



DVD-ROM felületéről pásztázó elektronmikroszkóppal készült felvétel, 20 000-szeres nagyításban. (Készítette: Prof. Tong Leung, [lojng] University of Waterloo [Juniövrsziti ov vóterlu], Kanada.)

Az informatika alapjai

A CD R illetve a DVD R egyszer írható, míg a CD RW és a DVD RW törölhető és újraírható. Az árak csökkenésével az íráshoz szükséges meghajtók is terjednek. A DVD esetében néha problémát okoz, hogy kétfajta R, illetve RW szabvány van, a + illetve a – jelű („pluszos” illetve „mínuszos”).

A pendrive

Az utóbbi évek egyik leggyorsabban fejlődő háttértára a *pendrive** [pendrájv], amely egy EEPROM típusú (tehát elektromosan törölhető) memóriát tartalmaz. Maga az eszköz az ún. USB portra csatlakoztatható, és a mai operációs rendszerek (Windows XP) azonnal felismerik. Kapacitása dinamikusan nő, 2005-ben tipikus mérete 512 MB volt.



Pendrive

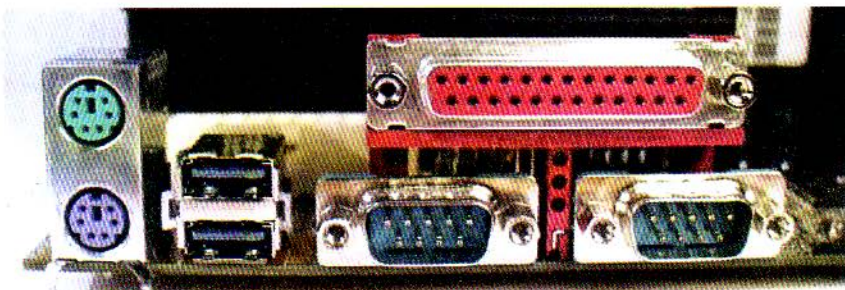
A számítógépen lévő csatlakozók

A számítógépet külső eszközökkel, illetve egy másik számítógéppel a hátoldalán található csatlakozók (*portok*) segítségével kapcsolhatjuk össze. Ezek többnyire közvetlenül az alaplaphoz csatlakoznak, de saját vezérlőkártyán is elhelyezkedhetnek. A számítógép egyes részei közötti kapcsolódási felületet *interfésznek** (*interface*) nevezzük.

A *soros port** egy adatvonalon teszi lehetővé az adatáramlást, az adatok bájtonként, a bájtok bitenként egymás után haladnak. A 9 vagy 25 pólusú csatlakozó mindkét irányú adattovábbításra képes, az operációs rendszer COM1, COM2, ... néven azonosítja. Régebbi egereket, analóg telefonmodemet, néha a nyomtatót szokták rá csatlakoztatni.

A *párhuzamos port** esetében nyolc adatvonal van, ezért az adatok bájtonként haladnak át. A 25 vagy 36 pólusú csatlakozó esetében a dugón vannak a tüskék. A párhuzamos portot az operációs rendszer LPT1, LPT2, ... néven azonosítja, a párhuzamos portra csatlakoztathatjuk általában a nyomtatót.

Az *USB** (*Universal Serial Bus* [juniverzöl szíriöl bász]) port is soros elven működik, sebessége azonban lényegesen nagyobb a soros és a párhuzamos portnál. Az USB eszközök a gép bekapcsolt állapotában is csatlakoztathatók (*hot plugging* [pláging]).



Balról jobbra: két kerek PS/2-es port, 2 db USB port, majd fölül a 25 pólusú párhuzamos, alatta 2 db 9 pólusú soros port. Keressük meg ezeket a csatlakozókat az iskolai számítógépen is!

Az USB-nek két változata van, a lassabb USB 1 (max. 12 Mb/s) és a lényegesen gyorsabb USB 2 (max. 480 Mb/s). Természetesen USB 2 eszközt csatlakoztathatunk az USB 1 portra is, de akkor nem tudjuk kihasználni a sebességét. Az USB portok használatát a régebbi operációs rendszerek nem támogatják. **USB portra gyakorlatilag bármely újabb periféria** (egér, nyomtató, ADSL modem stb.) **csatlakozhat**, akár olyan külső eszközök is, mint pl. a mobiltelefon adatkábele vagy a digitális fényképezőgép.

A PS/2 portok az alaplapon található, kör alakú csatlakozók. **A PS/2 portra* csatlakoztathatjuk a billentyűzetet** (többnyire lila), **illetve az egeret** (többnyire zöld).

Általában a hangkártyán találjuk meg a **game portot** [gém], amely a játékokhoz használatos botkormány csatlakozója.

Hálózati berendezések

A modem* át számítógépünket – többnyire a telefonhálózat segítségével – **összeköthetjük egy távoli számítógéppel**, pl. csatlakoztathatjuk az internetre. A modem egyik oldalon tehát a számítógéphez csatlakozik, a másik oldalon pedig a telefonhálózathoz (vagy a kábeltévé hálózathoz). A modem lehet *belső*, ekkor valamelyik slotba kell dugni, vagy *külső*, ekkor a soros portra vagy az USB portra csatlakozik.

A modem a számítógép által használt jeleket úgy alakítja át (*modulálja*), hogy azt továbbítani lehessen az analóg hálózaton. Természetesen a célállomáson egy másik modem alakítja vissza (*demodulálja*) a jeleket.

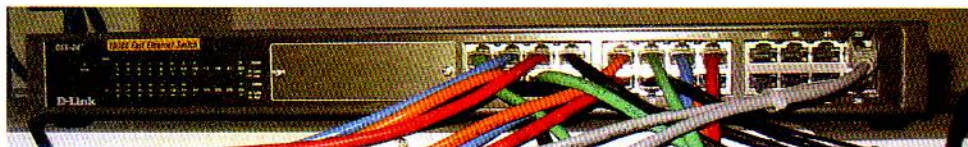
A modem sebességét általában bit/s-ban adják meg, de néha találkozhatunk a baudal (ejtsd „bód”) is. A **baud** [bód] *azt adja meg, hogy a modem másodpercenként hány modulációt végez*. Ha a modem egyszerre 1 bitet alakít át, akkor ez megegyezik a bit/s-mal, ha azonban egy lépésben többet, akkor értelemszerűen kisebb annál.

Mint láttuk, az analóg modem sebessége legfeljebb 56 kbit/s, egy ADSL modem sebessége viszont akár több Mbit/s is lehet.

Ha helyi hálózatot építünk, akkor minden gépben el kell helyeznünk egy **hálózati kártyát***, amely a számítógép jeleit továbbítja a többi gép felé, illetve az érkező adatokat a számítógép felé.

A hálózati kártyák összekapcsolása sokféleképpen lehetséges. Gyakori a telefonközpont-hoz hasonló ún. **switch*** [szvics] használata. Ekkor valamennyi hálózati kártya a switch-hez kapcsolódik, és azon keresztül áramlanak az adatok a megfelelő gépek között.

A hálózati kártya és a switch összeköthető vezetékekkel (pl. a telefonvezetékhez hasonló **UTP-kábellel***, vagy a kábeltévé-hálózat vezetékéhez hasonló **koaxiális kábellel***), de megvalósítható vezeték nélkül is (pl. rádióhullámokkal).



Switch. A beérkező UTP-kábelek „másik végén” számítógépek vagy további switchek vannak.

Számítógép-generációk

Az előző fejezetekben megismerkedtünk a mai személyi számítógép felépítésével, jellemző eszközeivel. Befejezésül érdemes áttekinteni a számítógépek fejlődésének főbb állomásait az ENIAC-tól napjainkig.

A számítógép fejlődését generációkra szokás bontani. **Az ENIAC előtti számoló- és számítógépeket nulladik generációnak* nevezik.**

Az első generációs* számítógépek (1946-1958) elektroncsövekkel készültek. Főbb jellemzőik a *hatalmas méret* (400 m³-es központi egység), a nagy fogyasztás és megbízhatatlan működés. A gépek tárolókapacitása kb. 1000 bájt, *sebessége kicsi*, mindössze néhány ezer művelet másodpercenként. Háttértárként *lyukkártyát*, illetve *lyukszalagot* használtak. Ekkor készült el az első programozási nyelv, a *Fortran*.

A második generációs* gépek már félvezetőket (diódák, tranzisztorok) **használtak** (1958-1965). Ezzel együtt jelentősen nőtt a megbízhatóságuk, *csökkent a méretük, és megnőtt a sebességük is* (kb. egymillió művelet másodpercenként). Háttértárként megjelenik a *mágnesszalag* és a *mágneslemez*. A gépek kezelését már *operációs rendszer* segíti. Ekkor kezdik el a számítógépeket ügyviteli célokra is használni.



A harmadik generációs PDP 11 1970-ben készült.

A harmadik generáció* (1965-1972) magával hozza az integrált áramkörök használatát. Számptalan új, ma is használt *periféria* jelenik meg, pl. a monitorok, nyomtatók, rajzgépek, háttértárként pedig általánossá válik a merevlemez. A számítógépek sebessége már 10–15 millió művelet másodpercenként.

A géppel együtt szállított szoftver értéke eléri a vételár 50 %-át, számptalan ma is elterjedt programozási nyelv (Pascal, C stb.) születik meg. Megjelennek a *többfelhasználós operációs rendszerek*. A megbízható hardver és a bőséges szoftver a számítógépek *széleskörű alkalmazását* teszik lehetővé.

A negyedik generáció* (1972-1990) fő jellemzője a mikroprocesszor. Ezzel elkezdődik a miniatürizálás, az egy cm²-re eső áramköri elemek száma hamarosan eléri az egymilliót, így a gép központi egysége 0,1 m³ alá csökken. Megjelennek a *mikroszámítógépek*, ezzel a számítástechnika bevonul az otthonokba. Velük együtt elterjednek a játék- és

alkalmazói programok, illetve a Basic [bézik] programozási nyelv. Egyre több *technikai eszközbe* kezdik beépíteni a mikroszámítógépes eszközöket. Ebben a korszakban készülnek el az első *szuperszámítógépek*, illetve terjedésnek indulnak a *számítógép-hálózatok*. Új programozási technikák alakulnak ki (strukturált programozás, objektum orientált programozás), amely hatékonyabbá teszi a szoftverfejlesztést.



Az első személyi számítógép, az Altair [alter], 1975-ben készült. Az adatokat a dobozon lévő kapcsolókkal lehetett bevinni a 4 kilobájt memóriába, az eredményt pedig a lámpák felgyulladására jelezte.

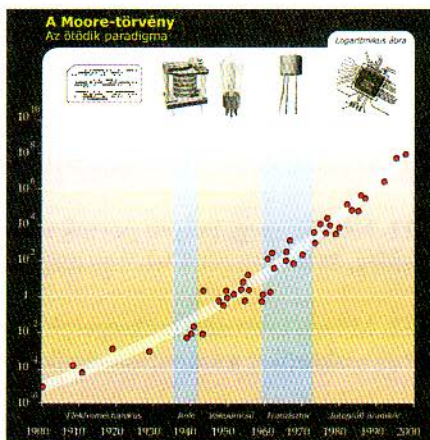


A legendás Commodore 64 [komodor] 1982-ben jelent meg. Addig ismeretlen grafikai és zenei teljesítménye a világ legnagyobb példányszámban eladott gépévé tette.

Ma az ötödik generációban* élünk. A korszak fő jellemzője a *többprocesszoros rendszerek* megjelenése és a helyi, illetve *nagy területű hálózatok* (pl. az internet) elterjedése. Az 1 cm²-re eső *áramkörti elemek száma tovább nő*, ma már egy integrált áramkör akár 100 milliónál több áramkörti elemet is tartalmazhat. A többprocesszoros rendszerekkel megjelentek az első *párhuzamos működésű* (tehát nem Neumann-elvű) számítógépek.

Fölmerül a kérdés: hogyan tovább? Gordon Moore [múr] nevéhez fűzik (valójában többen is felismerték, illetve módosították) a következő tapasztalati törvényt. **Az egy lapkára szerelt áramkörti elemek száma körülbelül 18 hónaponként megduplázódik.** Ez azt jelenti, hogy a mikroprocesszorok teljesítménye változatlan ár mellett kb. másfél évenként kétszeresére nő.

Jelenleg (2005-ben) az áramkörti elemek távolsága egy szilícium lapkán kb. 90 nm. Mivel az atomok mérete kb. 0,2 nm, így mai tudásunk alapján ez a távolság kb. 4 nm-ig csökkenthető. A további fejlődéshez tehát kb. 10 év múlva a gyártóknak új technológiát kell bevezetniük.



Az ábra az 1000 dollárért vásárolható másodpercenkénti számítások számát szemlélteti. Hányzorosára nőtt ez száz év alatt? Mit mondhatunk a növekedés üteméről?

Összefoglaló kérdések, feladatok

1. Hogyan csoportosítják teljesítményük alapján a számítógépeket? Melyik csoportot milyen célra érdemes használni? Miért?
2. Hogyan tagozódik a hardver? Mik a központi egység feladatai? Mutassuk meg számítógépünkön, hogy az egyes perifériák hol helyezkednek el, és milyen célt szolgálnak!
3. Hogyan épül fel a központi feldolgozó egység? Melyek a ma használt processzorok főbb paraméterei?
4. Mi a ROM és mi a RAM? Milyen fajták vannak? Mekkora a tárolókapacitásuk?
5. Soroljuk fel, és mutassuk is meg a billentyűzet részeit! Mi a következő gombok szerepe: F1, Insert, Home, Del, NumLock, Shift, CapsLock?
6. Milyen pozícionáló eszközöket alkalmaznak? Adjuk meg azt is, hogy melyiket milyen célra érdemes használni!
7. Milyen elven működnek a mai nyomtatók? Milyen nyomtatót érdemes választani otthoni képnymtatásra, irodai célokra, illetve bolti számlázásra? Miért?
8. Helyettesítheti-e az érintőképernyő a hagyományos monitort és billentyűzetet? Mondjunk érveket mellette és ellene is!
9. Rajzoljuk fel vázlatosan a winchester felépítését! Milyen adatokkal jellemezhetjük? Mire kell figyelni a mindennapos használata során?
10. Milyen fajtái vannak a CD, illetve a DVD-lemeznek? Milyen adatokkal jellemezhetjük ezeket, illetve a hozzájuk tartozó meghajtókat?
11. Nézzük meg, milyen portok vannak gépünkön! Melyiket milyen célra használjuk jelenleg, és milyen célra lehetne még használni?
12. Milyen eszközök szükségesek egy helyi hálózat kiépítéséhez? Mutassuk meg ezeket az eszközöket a számítástechnika teremben!
13. Kérjünk árjegyzéket egy számítástechnika szaküzletben! Állítsunk össze ez alapján egy olyan konfigurációt, amely alkalmas irodai munkára! Állítsunk össze egy olyat is, amely alkalmas a legújabb játékok futtatására!
14. Hardvergyártók kapcsán általában amerikai és távol-keleti cégeket szoktak felsorolni az emberek, pedig neves európai cégek is vannak. Nevezzünk meg néhányat!

Az operációs rendszer

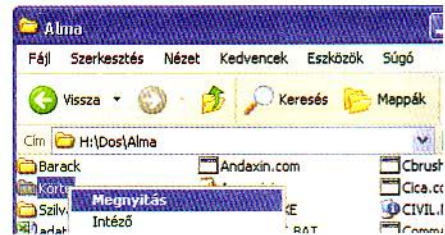
Az operációs rendszer fogalma és néhány jellemzője

Az operációs rendszer feladata

A számítógép bekapcsolása után automatikusan elindul a ROM memóriában lévő BIOS [biosz] nevű program, amely a hardver eszközök ellenőrzése után megkeresi és betölti az operációs rendszert. Az operációs rendszer azonos hardver esetén is eltérő lehet gépenként, és az is *gyakori, hogy a gép indításakor a felhasználó több operációs rendszer közül választhat*. Az operációs rendszer többnyire a winchesteren található, de betölthető a flopiról, a CD-ről, ritkábban pedig akár a pendrive-ról is. **Az operációs rendszer* egy program** (sőt több programból álló programcsomag), amely nélkül a mai számítógépek nem működhetnek. Főbb feladatai a következők:

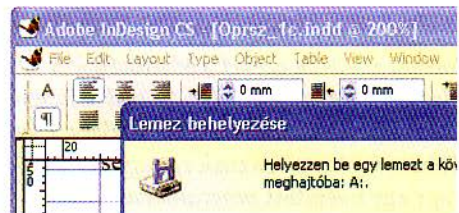
Kapcsolattartás a felhasználóval. A felhasználó az operációs rendszer segítségével adja ki utasításait a gép számára. Ebből a szempontból megkülönböztetünk karakteres, illetve grafikus felületű operációs rendszereket. **A grafikus felület* (Graphical User Interface, GUI [grafiköl jüzer interfész]) előnye, hogy könnyebben kezelhető, ugyanakkor hátránya, hogy erősebb hardver szükséges hozzá.**

```
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-9 on an i686
login: glen
Password:
You have new mail.
[glen@phoenix glen]$ ls -l public_html
total 20
-rw-rw-r-- 1 glen glen 1076 Mar 16 11:54 add.php
-rw-rw-r-- 1 glen glen 352 Mar 16 08:32 myguestbook
-rw-rw-r-- 1 glen glen 182 Mar 16 15:31 nov.php
drwxr-xr-x 7 glen glen 4096 Mar 2 09:45 phpMyAdmin
-rw-rw-r-- 1 glen glen 211 Mar 23 11:55 time.html
[glen@phoenix glen]$
[glen@phoenix glen]$ cd public_html
```



Fájlok listázása és mappába lépés karakteres (Linux) és grafikus (Windows) felületen.
A felhasználók szívesebben választják a grafikus felületet. Miért?

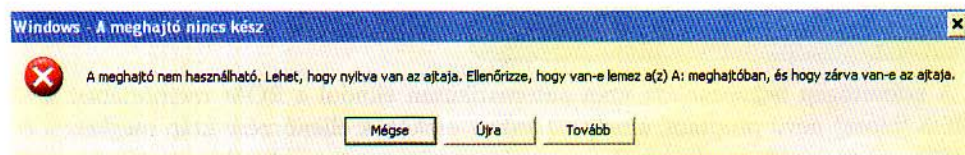
Programok indítása, futtatása, befejezése. A felhasználó által kiválasztott programot az operációs rendszer keresi meg a háttértárolón, tölti be és indítja el. Gondoskodik arról, hogy miközben a program fut rendelkezésére álljanak a szükséges eszközök. Megakadályozza, hogy a programok hibás működése kárt okozzon, és a befejeződő program után elvégzi a szükséges „takarítást”.



Az erőforrásokat az operációs rendszer kezeli. Honnan látjuk az ábrán, hogy a kértést nem a szoftver, hanem az operációs rendszer küldte?

Az erőforrások kezelése. Az alkalmazói programok a különböző erőforrásokhoz (pl. háttértárak, nyomtató) *csak az operációs rendszeren keresztül* férhetnek hozzá. Így lehetőség van az erőforrások megfelelő megosztására a futó alkalmazások között.

Hiba- és kivételkezelés. Az operációs rendszer feladata, hogy program- vagy hardverhiba esetén gondoskodjon a hiba kezeléséről. Pl. értesíti a felhasználót, hogy tegye vissza a mágneslemez, hibás a CD, vagy nincs papír a nyomtatóban.



Az operációs rendszer gondoskodik a hibák kezeléséről is. Példánkban a felhasználó fájl beolvasása közben kivette a flopit. A számítógép nem fagy le, az operációs rendszer tájékoztatja a felhasználót a problémáról és a lehetséges megoldásokról. Próbáljuk meg előidézni ezt a hibaüzenetet!

Az adatvédelem biztosítása. Az operációs rendszer feladata, hogy illetéktelenek ne férjenek hozzá mások adataihoz. Belépéskor a felhasználó nevével és jelszavával azonosítja magát. Ettől függenek jogosultságai: mely mappákba léphet be, hozhat-e létre, illetve törölhet-e ott fájlokat, lehetősége van-e programok telepítésére stb.

A futó programok közötti kapcsolat biztosítása. Gyakori, hogy különböző programok között adatcsere szükséges. Egy megoldás az operációs rendszer által kezelt *vágólap*, mely lehetővé teszi az adatok átadását.

A modern operációs rendszerek **egyszerű felületet biztosítanak a programok számára**. A felhasználó így lényegesen könnyebben sajátíthatja el a programok használatát, hiszen azok kezelését csak egyszer kell megtanulnia. A programozók pedig kész elemekből építkezve jelentős munkát takaríthatnak meg.

A Windows [vindóz] esetében pl. *Windows-alkalmazásoknak* nevezik azokat a programokat, amelyek ezeket a lehetőségeket kihasználják. Bár a Windows képes a korábbi DOS [dosz] alá írt programok futtatására is, ezek nem tekinthetők Windows-alkalmazásoknak.

Az operációs rendszerek általában **segédprogramokkal is segítik a felhasználót**. Többnyire tartalmaznak *egy egyszerű szövegszerkesztőt* (pl. Windowsban a Jegyzettömb) vagy egy beépített *programozási nyelvet* (pl. QBASIC [kubézik] a DOS-ban) is stb.

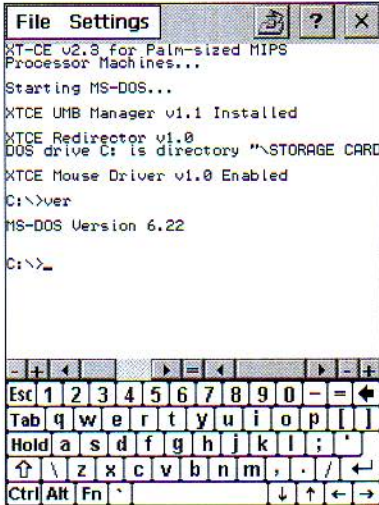
Azt, hogy egy adott gépen milyen programokat használhatunk, az éppen futó operációs rendszer dönti el. Egy DOS alá írt játék futhat egy PDA-n is, ha arra telepítettek DOS-t. A DOS operációs rendszer alatt ugyanakkor nem futtatható a Windowshoz készített Excel, még akkor sem, ha a gép egyébként rendelkezik a szükséges hardverfeltételekkel.



A felhasználó olyan mappába kívánt belépni, amely számára nincs engedélyezve. Az operációs rendszer megtagadta a belépést.

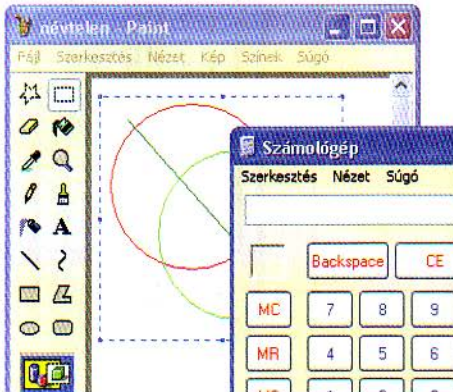
Az operációs rendszerek csoportosítása

Az éppen futó programot *taszknak** (task) nevezik. Az egy időben futtatható programok számától függően az operációs rendszer lehet *egytaszkos** vagy *multitaszkos**. Tipikus egytaszkos operációs rendszer a klasszikus DOS, míg pl. a Windows XP vagy a Linux multitaszkos.



A DOS operációs rendszer egy PDA-n. Használhatók vele a régi játékok is.

• *Köteget** (batch [becs]) üzemmód esetén a felhasználó egy fájlban „adja át” az elvégzendő feladatok listáját, melyet az operációs rendszer a felhasználó további beavatkozása nélkül hajt végre.



Multitaszkos operációs rendszerben párhuzamosan futhatnak a programok.

Hogyan futhat egy processzor esetében párhuzamosan több taszok? Erre egy megoldás az *időosztásos* (time-sharing [tájmsering]) rendszer, amikor a processzor a futó taszok mindegyikére egy-egy rövid időseletet szán. Ha az első taszokra az ember számára belátható időn belül újra sor kerül, úgy érezzük, hogy a taszok párhuzamosan futnak.

A felhasználók szempontjából az operációs rendszer lehet *egyfelhasználós** vagy *többfelhasználós**. Ez utóbbi esetben a gép bekapcsolása és a felhasználó beléptetése szétválik. Az operációs rendszer a felhasználót bejelentkezési neve és jelszava alapján azonosítja, ettől függően különböző jogosultságokat biztosít számára.

Az operációs rendszer többféle üzemmódban dolgozhat, de nem minden operációs rendszer támogatja valamennyi üzemmódot.

```

Echo off
PROMPT $P$G
VERIFY ON
BREAK ON

SET DIRCMD=/O:N
SET TEMP=C:\
SET TMP=C:\

PATH=A:\;C:\

IF NOT "%config%"=="YES_CD" GOTO NO_CD
LH MSCDEX.EXE /D:mscd001
GOTO QUIT

:NO_CD
IF NOT "%config%"=="NO_CD" GOTO QUIT
GOTO QUIT

:QUIT
    
```

Az Autoexec.bat fájl a DOS minden indításakor elvégzi a szükséges műveleteket.

Operációs rendszer

- *Interaktív** üzemmód esetén a felhasználó által kiadott parancsot az operációs rendszer azonnal végrehajtja, majd várja a következő utasítást. (Természetesen több taszk vagy több felhasználó esetében elképzelhető, hogy egy-egy erőforrásra, pl. a nyomtatóra várni kell.)
- *Valós idejű** (*real-time* [riltájm]) üzemmód esetében a számítógép konkrét fizikai folyamatban vesz részt, pl. közlekedési jelzőlámpákat irányít. Ilyenkor az erőforrások számával biztosítani kell, hogy a kiadott utasítást tényleg az előírt időpillanatban végre lehessen hajtani.

A PC-s operációs rendszerek

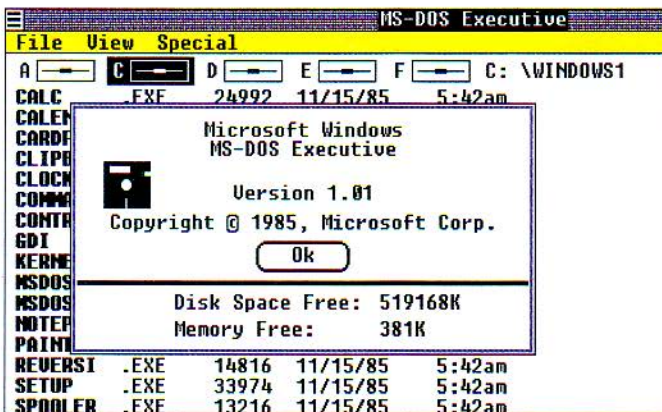
A PC-s számítógépek egyik első operációs rendszere a *DOS** (*Disc Operating System* [disk operéjting szisztem]) volt. Az 1981-ben megjelent *DOS egyfelhasználós, egytaszkos, 16 bites, karakteres felületű operációs rendszer, így viszonylag szerény a hardverigénye*. Közel 15 éves pályafutása során nagyon sok program készült hozzá, ezek egy részét, főleg a grafikat kevésbé igénylő számlázó és könyvelő programok terén mind a mai napig használják.

A *Windows**-rendszer hosszú ideig csupán grafikus felületet biztosított a DOS-hoz, vagyis a Windows 3.1-es verziója még nem volt igazi operációs rendszer. *Mint 32 bites, grafikus felületű, multitaszkos, önálló operációs rendszert hosszú ideig kétféle változatban fejlesztette a Microsoft.*

Az 1995-ben megjelent *Windows 95* és utódai a *Windows 98* majd *Windows Me* [mí] elsősorban az otthoni felhasználók részére készültek.

A *Windows NT* [enti] első változata 1993-ban készült el, valódi többfelhasználós operációs rendszerként elsősorban a vállalati felhasználás céljára. Ennek utóda a *Windows 2000*, illetve a ma széles körben elterjedt *Windows XP* (*eXPerience* [ekszpírienz]).

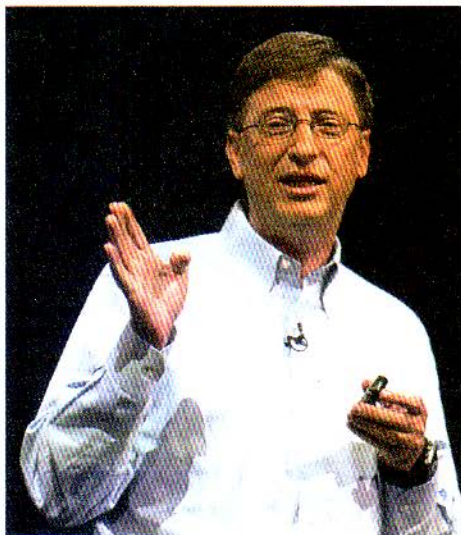
A *Linuxot** 1991-ben a helsinki egyetem másodéves hallgatója, Linus Torvalds [linusz torváldsz] kezdte fejleszteni a nagygépes UNIX [juniks] operációs rendszer mintájára. Az eredeti rendszer 32 bites, multitaszkos, többfelhasználós és karakteres felületű volt, *de ma már 64 bites, illetve grafikus felületű változata is van.*



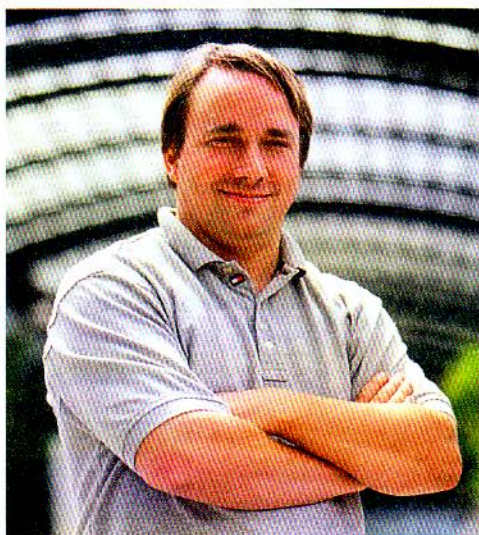
A Windows első verziója 1985-ben készült.
Hosszú ideig csak grafikus felület volt a DOS-hoz.



A Windows CE a PDA-k operációs rendszere.



Bill (hivatalosan: William Henry) Gates [viliem henri géjtsz] 1955. október 28-án született Seattle-ben [sietl]. 13 éves korában az iskola által bérelt gépen ismerkedett meg a Basic [bézik] programozási nyelvvel, 15 évesen pedig már megrendelésre írt programokat. Paul Allen [pól álen] barátjával 1975 júliusában alapította meg a Microsoftot [májkrosoft]. Ma a világ egyik leggazdagabb embere.



Linus Benedict Torvalds [linusz benedikt torválds] 1969. december 28-án született Helsinkiben. Másodéves egyetemista korában a 386-os processzor lehetőségeinek jobb kihasználására készítette el a Linux első változatát. 1997 óta az Egyesült Államokban a Szilícium Völgyben dolgozik, de mindmáig a Linux rendszermagjának vezető fejlesztője.

Az eredeti rendszer nyílt forráskódú és ingyenes termék, így nem jár hozzá hivatalos terméktámogatás sem. Adott felhasználói csoportok részére többen készítenek Linux alapú programcsomagokat ún. **disztribúciókat** (*Debian, Suse [szuze], Red Hat [red het], Uhu stb.*), de ezek már tartalmazhatnak nem ingyenes komponenseket is, és többnyire a terméktámogatás sem ingyenes.

A grafikus felület

A grafikus felületű operációs rendszert még az 1970-es évek elején dolgozták ki a Xerox cég Palo Alto-i kutatóközpontjában, itt találták fel a felület kezeléséhez szükséges egeret is. Manapság szinte minden személyi számítógép átvette az ott bevezetett módszereket, nem véletlen tehát, hogy ezek meglehetősen egyformán kezelhetők.

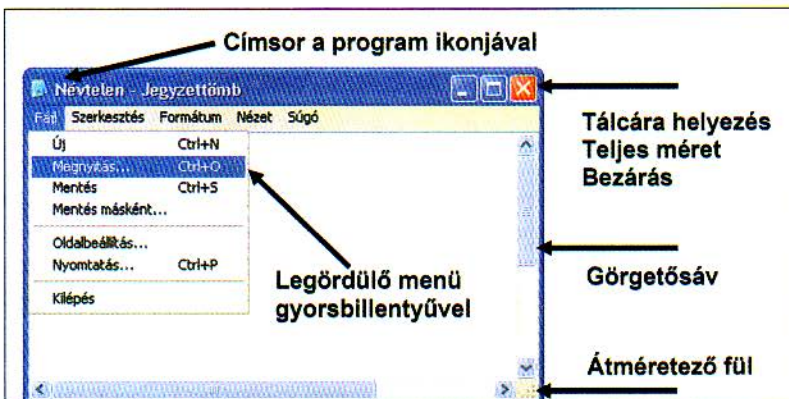
Foglaljuk össze a grafikus felület főbb elemeit!

- A grafikus felületen többnyire az egér segítségével választjuk ki, hogy milyen műveletet szeretnénk végezni. Hordozható számítógépeken egér helyett más megoldást használnak, pl. érintőképernyő, touchpad [tácsped], hanyattgér. (A műveletek elvégzésére természetesen a billentyűzetet is használhatjuk, azonban annak kezelése kényelmetlenebb.)



A Suse Linux (balra) és a Windows XP (jobbra) jellegzetes képernyője. Milyen közös elemeket találhatunk a két operációs rendszer felületén?

- Az operációs rendszer munkaterülete az **asztal***, melynek megjelenése módosítható.
- Az asztalon elhelyezett **ikonok*** egy-egy programot vagy adatállományt szimbolizálnak, melyet többnyire az ikonra történő két egérekattintással nyithatunk meg.
- Az asztal egy kitüntetett része, a **tálca*** tartalmazza az éppen megnyitott alkalmazások felsorolását, melyek között egérekattintással válthatunk.
- A parancsok közötti választás egyik legegyszerűbb eszköze a **menürendszer***. Az operációs rendszer menüje egy kitüntetett gombra (Windowsban a Start gomb) való kattintással nyitható meg. Így elérhetjük azokat a programokat is, melyeknek nincs ikonjuk az asztalon.



Programablak részei a Windows-rendszerben. Ha az alkalmazás címsorában lévő ikonjára kattintunk, a megjelenő menüpontokkal billentyűzet segítségével is elvégezhetjük az ablak kezelésére vonatkozó műveleteket. Próbáljuk meg a billentyűzet segítségével áthelyezni, illetve átméretezni az ablakot!

- A programok egy **ablaknak*** nevezett téglalap alakú területen jelennek meg. Az ablak az egér húzásával áthelyezhető, illetve átméretezhető. Felső része tartalmazza az ablak azonosítására szolgáló adatokat (*címsor*) és a program menürendszerét.
- Az alkalmazásokban a **legördülő menüt*** használják, ahol a főmenü pontjai egy-egy menülistát azonosítanak. Gyakori, hogy a menüpontokhoz *billentyűzetkombinációt* is rendelnek, melynek lenyomásával azok egér nélkül is elérhetők.

A grafikus felület elemei

Karakteres felületen az adatok bevitele egyszerűen a szöveg beírásával történik. A grafikus felületen azonban többféle eszköz is rendelkezésünkre áll, például két eset közül választhatunk a jelölőnégyzettel vagy számok esetében a léptetővel. Az alábbi táblázat a grafikus felület leggyakoribb beviteli eszközeit tartalmazza.



A legegyszerűbb adatbeviteli eszköz a *szövegmező* vagy *beviteli mező*. Van többsoros változata is: ilyen például a Jegyzetömb munkaterülete.



Míg az összetartozó *rádiógombokból* (balra) mindig csak egy lehet kiválasztott, addig a *jelölőnégyzetekből* (jobbra) többet is megjelölhetünk.



A *gombokra* kattintva az adatbevítelt véglegesíthetjük (OK) vagy elvethetjük (Mégse).



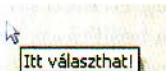
A *folyamatjelző* egy hosszabb tevékenység állapotát mutatja, pl. hány százalékban készült el a formázás.



A *numerikus léptető* segítségével számszerű adatot adhatunk meg, pl. egy betű nagyságát.



A *lista* (balra) egy vagy több elem kiválasztására szolgál. Ha a lista túl hosszú, megjelenik rajta a *görgetősáv* is. A *kombinált listába* (jobbra) be is írhatunk új elemeket. A listaelemek ennél a változatnál a nyílhegyre kattintva gördülnek le.



Nem beviteli, hanem tájékoztató eszköz a *buborék*, amely akkor jelenik meg, ha valamely elemre rámutatunk az egérrel.

Az egér kezelése

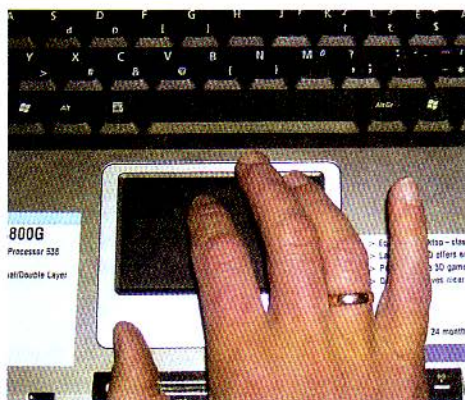
A grafikus felületek legfontosabb pozícionáló eszköze az egér vagy a neki megfelelő más pozícionáló eszköz (trackball [trekball], touchpad [tácsped], easyball [iziboll] stb.). Bár mindenki ismeri és használja, érdemes összefoglalni, hogy milyen műveleteket végezhetünk vele PC-s környezetben.

- *Rámutatás:* a mutatót egy adott ikonra toljuk.
- *Kijelölés:* rámutatás után egyet kattintunk az egér bal gombjával.
- *Húzás:* egy ikonra rámutatunk, majd az egeret lenyomott bal gomb mellett mozgatjuk. Ilyenkor a kijelölt ikon követi az egérkurzort. Ezt nevezik *fogd és vidd* (*drag and drop* [drágendrop]) technikának.
- *Indítás:* rámutatunk egy ikonra, és duplán kattintunk. Ez többnyire az alkalmazás indítását jelenti.
- *Kattintás az egér jobb gombjával:* rámutatunk egy ikonra, és lenyomjuk az egér jobb gombját. Ilyenkor megjelenik a helyi menü, ahol választhatunk az adott objektum leggyakoribb lehetőségei közül.

(Mivel a Macintosh számítógépeken egygombos egeret használnak, ott a műveletek részben eltérnek a fentiektől.)



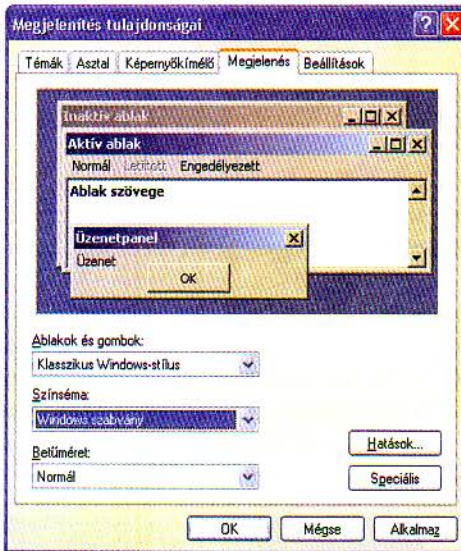
Az easyball egy nagyméretű hanyattgér. A kisgyerekek részére fejlesztették, akik az egeret még nem tudják „megfogni”.



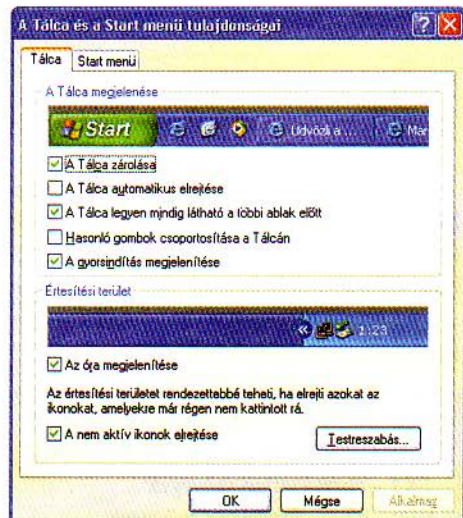
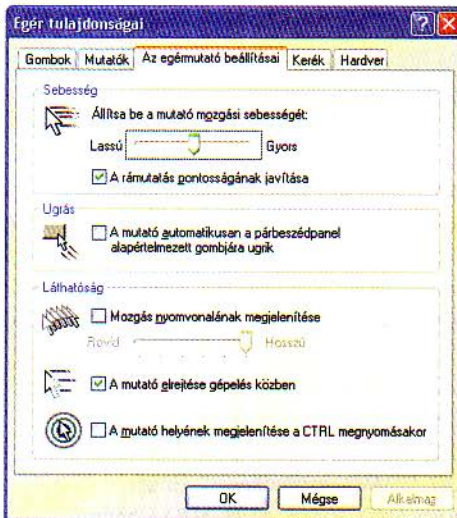
Laptopok esetén az egeret gyakran tapipad (touchpad) helyettesíti. Miért? Milyen további megoldásokkal találkozhatunk?

A grafikus felület testreszabása

A grafikus felület számtalan kényelmi és vizuális elemet tartalmaz. A grafikus felület megjelenését a felhasználók feladataiknak és ízlésüknek megfelelően módosíthatják, ezt a folyamatot hívják *testreszabásnak**. Világos, hogy a beállítások az alkalmazott operációs rendszertől, de még annak egyes verzióitól függően is eltérhetnek. Az ábrákon a Windows XP néhány beállítását tekinthetjük át.



A vizuális jellemzőket könnyen elérhetjük az asztal helyi menüjének **Tulajdonság** pontjával. A **Megjelenés** fülön megadhatjuk az ablakok és gombok főbb jellemzőit, a **Beállítások** fülön pedig a képernyőfelbontást és a színmélységet. Ellenőrizzük ezeket a beállításokat a gépünkön!



Az egér beállításait tartalmazó ablakot a **Start** menü **Vezérlőpult** pontjával érhetjük el. Olvassuk le az ábráról, hogy milyen beállítási lehetőségeink vannak! Ha lehet, próbáljuk is ki ezeket!

A tálca testreszabása a tálca helyi menüjéből a **Tulajdonságok** ponttal érhető el. Hogyan lehet a tálcát áthelyezni felülre? Milyen előnyei és hátrányai vannak, ha a tálca mindig látható a többi ablak előtt?

A fájlrendszer

A fájl létrehozása, jellemző adatai

Számítógépes környezetben adatainkat elektronikus formában tároljuk, így munkánk később újra megnyitható, módosítható, folytatható, megjeleníthető papíron stb. **Az összetartozó adatok halmazát, melyet a számítógépen egy egységben tárolunk, adatfájlnak* vagy egyszerűen fájlnak* nevezünk.**

Az adatfájl (pl. egy formázatlan szöveg) *mindig valamilyen alkalmazói programhoz tartozik* (pl. a Jegyzetkönyvhöz), amely a benne lévő adatokat kezelni tudja. Egy adott formátumú fájl kezelésére *több alkalmazás is megfelelő lehet*, pl. a formázatlan szöveges dokumentumot a Microsoft Word [májkroszoft vörd] is tudja kezelni. Az adatfájlt így általában az adott alkalmazás segítségével hozzuk létre. Néha az operációs rendszer segítségével is létrehozhatunk üres, tehát adatokat nem tartalmazó, de adott formátumú fájlt.



A Windowsban fájlt létrehozhatunk akár az asztalon is a helyi menü Új menüpontjával, de többnyire az alkalmazás Fájlnak a menüjének Mentés másként menüpontjával szoktuk munkánkat elmenteni.

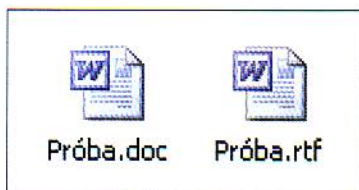
A fájlt a legtöbb PC-s operációs rendszer esetében két adat azonosítja: a neve és a kiterjesztése, a kettőt többnyire egy pont választja el egymástól. A fájl neve gyakorlatilag tetszőleges lehet, míg a kiterjesztést* a fájlban tárolt adatok típusa határozza meg, bár használata nem kötelező. Például Windows környezetben a Minta.doc esetében a Minta a fájl neve, a felhasználó által választott neve, míg a doc kiterjesztés arra utal, hogy a fájl formázott szöveget tartalmaz.

EXE, COM, BAT – futtatható programok	TMP – ideiglenes munkafájl
BMP, GIF, JPG, JPEG, TIF, TGA – képek	PPT, PPS – bemutatók (diasorozatok)
TXT – formázatlan szövegek	AVI, WMV – mozgóképek, filmek
DOC, RTF – formázott szövegek	WAV, SND, WMA, MP3, MID – hang
XLS, XLC, WKS, WQ1 – táblázatok	ZIP, ARJ, LHA, RAR – tömörített állomány
DBF, MDB – adatbázisok	BAK – biztonsági másolat
PRN, PS, EPS – nyomtatófájlok	CLP – vágólap tartalom

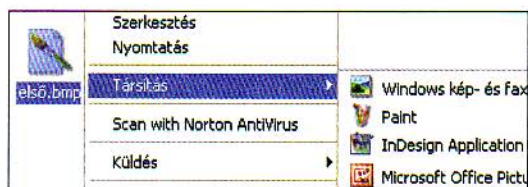
A leggyakoribb fájlkiterjesztések. Melyekkel találkoztunk eddigi tanulmányaink során?

A fájlok nevére és kiterjesztésére a különböző operációs rendszerek gyakran egyéb feltételeket is előírnak. Így a hagyományos DOS rendszerben a fájl neve legfeljebb 8, a kiterjesztése legfeljebb 3 karakter lehetett. A Windowsban a fájlnev lehet hosszabb is, és tartalmazhat szóközt vagy pontot is, ekkor a kiterjesztés az utolsó pont utáni rész. (Néhány karaktert, pl. \, ;, / továbbra sem használhatunk.) A Linux a kisbetűket és nagybetűket különbözőnek tekinti, és itt a kiterjesztéseknek is kisebb a szerepe.

A fájlban tárolt adatokat általában csak valamilyen program segítségével tudjuk értelmezni, feldolgozni. A munka megkönnyítése érdekében **a legtöbb operációs rendszerben az egyes fájl típusokhoz társíthatunk* egy-egy alapértelmezett programot.** A Windows esetében a társított programot a fájl neve előtt álló ikon jelzi. Ezzel az ikonnal a társított alkalmazás indítható el, amely mindjárt be is tölti az adott fájlt.



A kiterjesztés különböző fájl típusra utal, de a társított program ugyanaz.

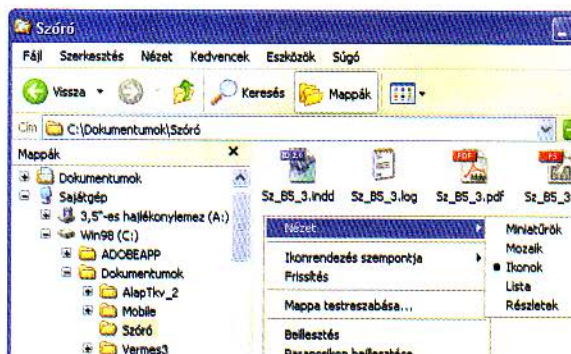


A Windowsban a helyi menü Társítás pontjával megadható, hogy melyik alkalmazásban nyíljon meg a fájl.

Meghajtók és mappák

Egyes háttértárakon a fájlok száma meghaladhatja a tízezret is, ezért azokat érdemes csoportosítani. A **mappa*** vagy **alkönyvtár** a fájlok egy csoportját tartalmazza. A mappát a neve azonosítja, és további mappákat is tartalmazhat. Így alakul ki a **mappaszerkezet*** vagy könyvtárszerkezet. A mappaszerkezetben általában **bármely két mappa között pontosan egy út vezet**, ezt a szerkezetet a matematikában fának nevezik. A mappáknak kiterjesztésük is lehet, de ezt viszonylag ritkán használják ki.

A **fájlok és mappák fizikailag a meghajtókon*** helyezkednek el. A meghajtó lehet a winchester, a flopi, a CD, a pendrive stb. Gyakori, hogy a winchestert több részre (ún. **partícióra**) bontják, ekkor minden partíciót önálló meghajtóként kezelhetünk.



A Windowsban a **Start** menüből indítható a fájl szerkezet kezelésére szolgáló **Sajátgép**. A **Mappák** gombra kattintva az ablak két panelre tagolódik. A bal oldali a mappaszerkezetet mutatja, míg a jobb oldali a kiválasztott mappa tartalmát. A mappa tartalmának megjelenítését a helyi menü **Nézet** menüpontjával szabályozhatjuk. Próbáljuk ki, hogy mit jelentenek az egyes nézetek!

Operációs rendszer

A meghajtókat a legtöbb PC-s operációs rendszer egy betűvel azonosítja, a DOS nyomán. *A*; illetve *B*: jelöli a flopmeghajtót, *C*: a winchestert stb. A flopi- illetve a CD-meghajtóba többféle flopi, CD-ROM stb. helyezhető, ezért ezek különböző neveket kaphatnak, melyeket az operációs rendszer ki is jelez.

Némileg más úton jár a Linux, ahol a meghajtók is betagolódnak a mappaszerkezetbe, így mint mappák jelennek meg.

A könyvtárszerkezet kezelése

Egy adott mappa vagy fájl helyét az elérési út írja le. **Az elérési út* megadja, hogyan jutunk el a meghajtó azonosítójától egymásba ágyazott mappák sorozatán át a keresett fájlig vagy mappáig.** A Windows-rendszerben elválasztójelként \ (visszaper) jelet használnak, így egy tipikus elérési út:

`C:\Windows\Help\notepad.hlp.`

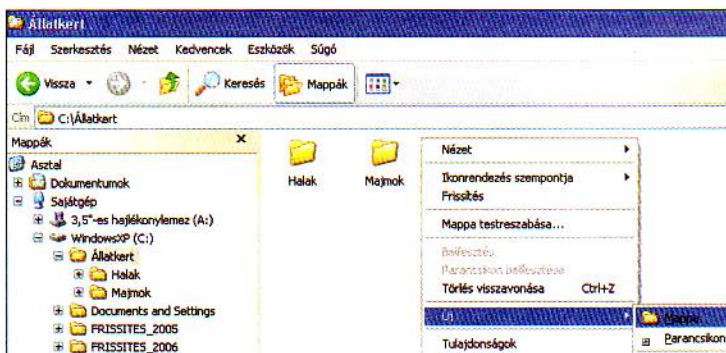
(Több operációs rendszerben, így pl. a Unixban is, az elválasztó a „sima” per: /.)

Az elérési út megadásánál kitüntetett szerepet tölt be a **gyökér*** (*root* [rút]), ahonnan az adott meghajtón valamennyi elérési út indul. Példánkban a `C:\` mappa a gyökér.

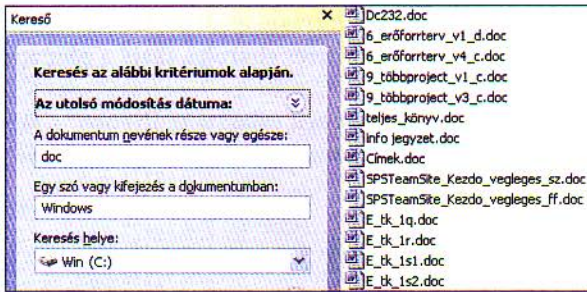
Munka közben az összetartozó fájlokat általában egy adott mappába helyezjük. Ezt a legtöbb alkalmazói program az első mentés vagy megnyitás után felajánlja, a fájlkezelő programok pedig külön is jelzik. (A Windows Sajtógép pl. a címsorba írja ki.) **Az éppen használt mappát aktuális mappának* nevezük.**

A tényleges munka közben általában két tipikus feladat szokott felmerülni, amelyre az operációs rendszerek eltérő megvalósítást kínálhatnak. Az egyik az **új mappa létrehozása**, melybe munkánk során az összetartozó fájlokat tesszük. A másik a **korábban elmentett fájl keresése**, amelynek vagy a helyére, vagy a nevére, esetleg egyikre sem emlékszünk már.

A fájlokkal és mappákkal az **alapvető műveletek (megnyitás, törlés, átnevezés, nyomtatás)** a grafikus felület menürendszerével értelemszerűen elvégezhetők. Sok esetben érdemes azonban megjegyezni a billentyűzetkombinációkat is. Így megnyitáshoz általában elegendő lenyomni az *Enter* gombot, törléshez a *Delete* [dilit] gombot, míg Windowsban az átnevezéshez használhatjuk az *F2* gombot is.



Új mappa létrehozása Windowsban. Mi lesz az új mappa elérési útja?



Fájl keresése Windows XP-ben. Állapítsuk meg az ábra alapján, hogy milyen tulajdonságú fájlokat listázunk!

Fájlok és mappák másolása és mozgatása

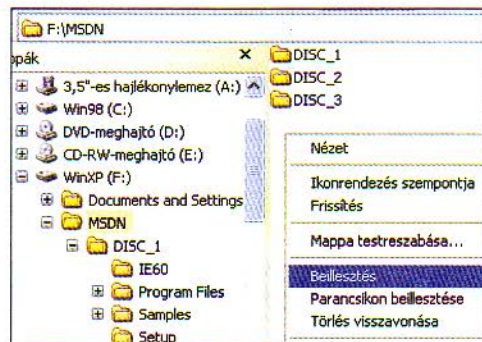
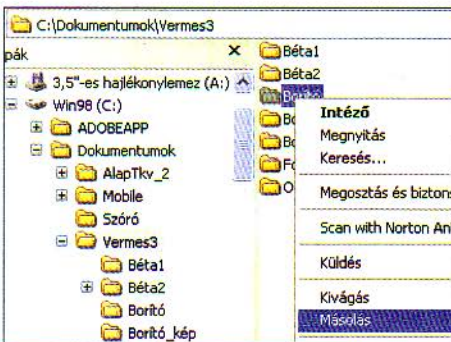
Gyakori művelet, hogy bizonyos fájlokat és mappákat máshogy szeretnénk elrendezni a mappaszerkezetben, például szeretnénk egy másik háttértárra áthelyezni. Ilyenkor az adott fájlokat és mappákat át kell másolnunk vagy mozgatnunk. A mozgatás a másolástól csak annyiban tér el, hogy ott az eredeti helyről a fájlok és mappák törlődnek.

Karakteres felületen a másolás parancsa meglehetősen bonyolult, hiszen meg kell adni mind a forrás-, mind a célmappa helyét. Ráadásul ha elírjuk oly módon, hogy az a rendszer számára értelmes, akkor nem hibát jelez, hanem más műveletet hajt végre.

```
F:\>COPY H:\DOS\ALMA\SKI.LGC C:\TMP\FOOT.ERR
1 fájl másolása történt meg.
```

A másolás általános parancsa a DOS-ban. A COPY [kopi] parancsot előbb a forrásfájl követi, majd a célfájl, ami most új nevet kap.

Grafikus felületen a **másolást*** általában többféle módon is meg lehet oldani. Például *egérrel a szükséges fájlokat a forrásmappából a célmappába húzhatjuk*, vagy egy menüpont kiválasztása után az operációs rendszer végigvezet bennünket az egyes lépéseken.



Mappa másolása a Windowsban. A forrásmappába lépve (bal oldali ábra) kijelöljük a másolandó mappát, és a helyi menü **Másolás** lehetőségére kattintunk. Belépünk a célmappába (jobb oldali ábra), és a helyi menü **Beillesztés** pontját választjuk. Mit, honnan és hová másoltunk?

Gyakori megoldás a vágólap használata. Ilyenkor a forrásmappában kijelölt fájlok elérési útját a *Másolás* parancs hatására az operációs rendszer eltárolja, majd a célmappába lépve a *Beillesztés* parancs hatására odamásolja.

Mozgatáskor* a *Másolás* helyett a *Kivágás* lehetőséget kell választani. Ilyenkor a fájlok csak a beillesztés végrehajtása után törölődnek a forrásmappából. (A vágólap kezelésére a későbbiekben még visszatérünk.)



Fájlok és mappák csoportos kiválasztása

Fájlműveleteknél igen gyakori, hogy nem egy fájlt, hanem *több, hasonló paraméterekkel rendelkező* (pl. azonos betűvel kezdődő) *fájlt kell egyszerre* másolni vagy törölni. Fontos, hogy megismerkedjünk ezekkel a lehetőségekkel, mert gyakran az alkalmazói programokban is használják őket.

Fájlok csoportos megadására a karakteres felületeken a **dzsóker karaktereket** vezeték be, amelyek **bizonyos karaktereket vagy karaktercsoportokat helyettesíthetnek**. Gyakori a DOS-ban alkalmazott megoldás: a **? egy tetszőleges karaktert**, a *** pedig tetszőleges karaktersorozat** helyettesíthet. Például a MOSO, MASA, MOSODA, MOS6 szavak esetében a MOS? jelentheti a MOSO és a MOS6 szavakat, míg a MOS* a MOSO, MOSODA, MOS6 szavakat.

Grafikus felületen a **csoportos kijelöléshez** az egér mellett általában a billentyűzet gombjait is használni kell. A Windowsban például a *CTRL* gomb folyamatos nyomva tartása mellett egyesével jelölhetünk ki több fájlt, míg ha a fájlok egymást követik, a *SHIFT* gomb nyomva tartása mellett elég rákattintani a kijelölendő fájlok közül az elsőre és az utolsóra. A két módszert felváltva is használhatjuk.

A fájlok áttekintését és így a csoportos kijelölést is jelentősen megkönnyíti, ha **az adatokat rendezve jelenítjük meg** a képernyőn. Grafikus felületeknél általában ehhez elegendő egyszerűen rákattintani az adott tulajdonságra a megjelenő listán.

Név	Méret	Típus	Módosítva
 CONFIG.SYS	1 KB	Rendszerfájl	1999.01.19. 21:12
 Csocso.sys	6 KB	Rendszerfájl	1998.05.15. 21:01
 Agyak.txt	1 KB	Szöveges dokument...	1996.01.02. 14:42
 Almagyak.txt	1 KB	Szöveges dokument...	1996.10.08. 14:44

Ha a Sajátgépen a **Részletek** nézetet választjuk, akkor a fájlok adatait az egyes tulajdonságokra kattintva rendezhetjük. Példánkban a fájlok **Típus** szerint vannak rendezve, így a *SHIFT* nyomva tartása mellett az elsőre és utolsóra kattintva könnyen kijelölhetjük a szöveges dokumentumokat.

A fájlok és mappák tulajdonságai

Az operációs rendszerek a fájlokról és mappákról nagyon sok olyan beállítást tárolnak, amelyeket egyszerű listázáskor nem jelenítenek meg. Ilyenek például az **attribútumok***, amelyek a **fájlok viselkedését leíró logikai értékek**.

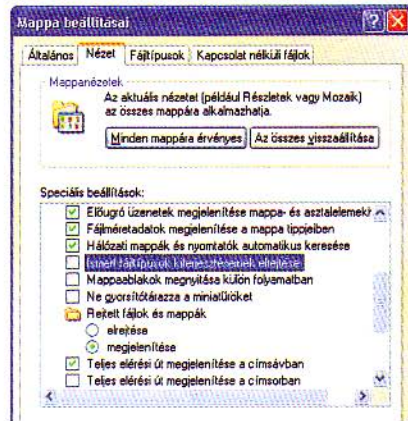
Az attribútumok operációs rendszertől függően különfélék lehetnek. A PC-s operációs rendszerek két leggyakrabban használt attribútuma az *írásvédett* (vagy *Read Only* [rid onli]), illetve a rejtett (*Hidden* [hidn]).

Az *írásvédett** fájlok például **nem módosíthatók**, a *rejtett** fájlok pedig (ha nem módosítottuk az alapbeállításokat) **listázáskor nem jelennek meg**, így nem jelölhetők ki pl. törlésre. Mivel ezeket a tulajdonságokat a felhasználó maga állíthatja be, csak a véletlen módosítás, illetve törlés ellen jelentenek védelmet, a szándékos kísérlet ellen nem.

A Windows-rendszer egyik érdekessége, hogy azoknál a fájloknál, amelyekhez a Windows társított alkalmazást, **megadható, hogy a kiterjesztések megjelenjenek-e**. Alapértelmezés szerint a kiterjesztések nem látszanak, de sokszor **célszerű bekapcsolni őket**. (Például a *fontos.txt.exe* nevű fájl a kiterjesztés kikapcsolása után ártatlan szöveges dokumentumnak tűnik. Ha azonban kettőt kattintunk rá, nem a Jegyzetömb indul el, hanem maga a program. Ezzel a módszerrel álcáztak már vírusokat is.)



A helyi menü **Tulajdonságok** menüpontjával tekinthetjük meg és állíthatjuk be az attribútumokat. Olvassuk le az *annyi.bmp* fájl tulajdonságait!



A Sajátgép **Eszközök** menüjének **Mappa beállításai** pontjával kapcsolhatjuk ki az ismert fájltypusok kiterjesztésének megjelenítését. Mikor hátrányos ez a beállítás?

A meghajtók kezelése

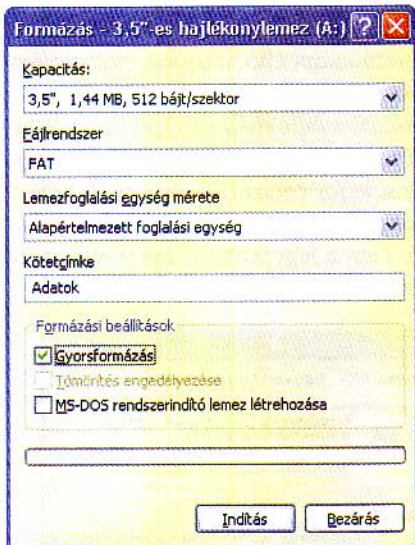
Az adatokat fizikailag a háttértárak tartalmazzák, ezek felhasználói szinten viszonylag kevés beavatkozást igényelnek. Legfontosabb adataikat, azaz a kapacitásukat, illetve a nevüket az operációs rendszer egyszerű parancsaival lekérdezhethetjük.

A **winchester munkatár**, itt tároljuk programjainkat és adatainkat. Mivel igen nagy rajta a forgalom, könnyen bekövetkezhet adatvesztés kártékony programok, véletlen törlés, hardver- vagy szoftverhiba miatt. Fontos, hogy főleg **az adatokról folyamatosan készítsünk biztonsági mentést (archiválás)** valamely más háttértárra, pl. CD-re.

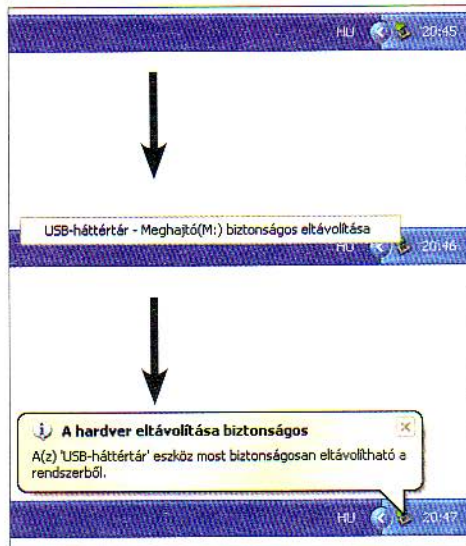
Flopik esetében a **formázás azt jelenti, hogy létrehozunk rajta az adatok tárolásához szükséges szerkezetet**. A legtöbb lemezt már gyárilag formázzák, ebben az esetben az utólagos formázás szerepe a lemez törlése, illetve hibamentességének ellenőrzése.

Operációs rendszer

Pendrive használata esetén figyelniük kell arra, hogy ha írás/olvasás közben kihúzzuk a gépből, akkor fizikailag is sérülhet. **Célszerű ezért a pendrive eltávolítása előtt az operációs rendszer megfelelő parancsával lezárni a pendrive-val való adatcserét.**



Hajlékonylemez formázása az A: meghajtó helyi menüjéből indítható. Fontos tudni, hogy a gyorsformázás nem ellenőrzi a lemez felületét!



A pendrive eltávolításának lépései. Kattintsunk az USB háttértár biztonságos eltávolítása ikonra, és csak akkor húzzuk le az eszközt, amikor a Windows jelzi, hogy a fájlműveletek befejeződtek.

Segédprogramok

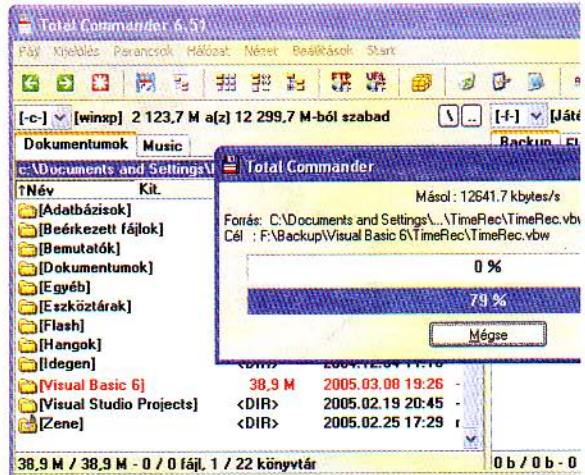
A segédprogramok jellemzése

Az operációs rendszerek a szorosabban vett alapfunkciók mellett *mindig tartalmaznak különböző segédprogramokat* is, amelyek a felhasználó munkáját megkönnyítik. Ilyenek pl. egy egyszerű szövegszerkesztő (pl. a Windowsban a Jegyzettömb), a munka automatizálását segítő programozási eszköz (pl. DOS esetén a batch nyelv, Windowsban a Visual Basic Script [vizuál bézik szkript]), különböző karbantartó programok stb.

Az egyes operációs rendszerekhez külső gyártók is készítenek segédprogramokat, ezeket két csoportba sorolják. **A shell [sell] programok az operációs rendszer használatát könnyítik meg, a utility [jutiliti] programok pedig új funkciókkal bővítik.** Igen elterjedt shell programok például a különböző „Commander”-ek [kommander], melyekkel a fájlkezelést könnyíthetjük meg (Norton Commander, Windows Commander stb.). Tipikus utility programok a vírusirtók, a tömörítők, a médiakezelők vagy például a CD-író programok.



Peter [pítör] Norton (1943-) a Norton Commander és Norton Utility készítője. Cégét 1990-ben eladta, de nevét a vevő is megtartotta (pl. Norton Antivírus).



A kétablakos Total Commander jelentősen megkönnyíti a fájl- és mappakezelést. Ennek ellenére legyünk tisztában az operációs rendszer eszközeivel is, hiszen előfordulhat, hogy olyan gépen kell dolgoznunk, ahol nincs telepítve, de Windows alatt Sajátgép mindig van.

A vágólap (clipboard)

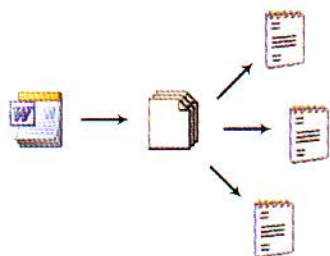
A vágólap* egy átmeneti tároló a memóriában, ahová bármely alkalmazásból adatot helyezhetünk el, majd ezt onnan ugyanabba vagy egy másik alkalmazásba beszúrhatjuk. A vágólap mindaddig megőrzi a tartalmát, amíg egy másik adattal felül nem írjuk, vagy az operációs rendszerből ki nem lépünk. Tartalma ezért többször is beszúrható. A vágólapot az operációs rendszer háttérben futó segédprogramja kezeli, amely gyakran arra is lehetőséget ad, hogy a vágólap tartalmát fájlba mentjük.

A vágólap felhasználói szinten három műveletet támogat. **Másolás*** (Copy, CTRL C) a kijelölt adat a vágólapra másolódik, **kivágás*** (Cut [kát], CTRL X) pedig az eredeti helyéről törlődik is. **Beillesztés*** (Paste [pész], CTRL V) a vágólap tartalma a kijelölt helyre másolódik a célalkalmazásban, pl. a kurzorhoz.

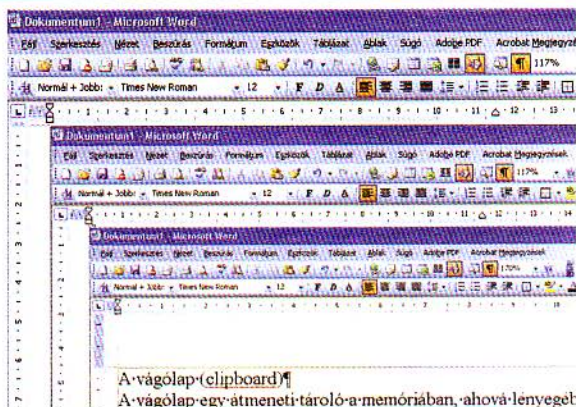
Az adat gyakorlatilag bármilyen típusú lehet, legrosszabb esetben a célalkalmazás nem tudja értelmezni, és ezért nem fogadja beillesztés. Így például képet a Jegyzetombbe nem tudunk beilleszteni.

A vágólap kezelését az egyes alkalmazások tovább bővíthetik vagy le is tilthatják. Például a Microsoft Office 2003 már 24 elemű vágólapot is képes kezelni, míg az Adobe Acrobat [adobi akrobat] a vágólap használata letiltható PDF fájlok megnyitása esetén.

Képlomás* alatt azt értjük, hogy a képernyőt vagy annak egy részletét egy új képbe másoljuk. Ezt gyakran az operációs rendszer is támogatja. A Windowsban például a teljes képernyőt a Print Screen [szkrín], az aktív ablakot az ALT Print Screen gombok lenyomásával másolhatjuk a vágólapra. Természetesen igen sok, többnyire szabadon terjeszthető utility programot is találhatunk erre a célra.



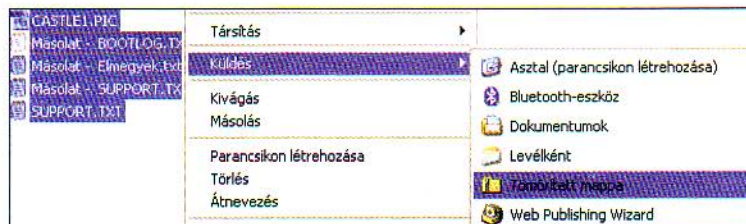
Forrásdokumentum, vágólap és céldokumentumok



A képlomás és beillesztés többszöri végrehajtása „kép a képben” hatást eredményez. Próbáljuk ki ezt a funkciót!

Az adatok tömörítése

Tömörítés segítségével adatainkat olyan alakra hozhatjuk, amelyben kisebb helyet foglalnak el a háttértáron. A tömörített állomány akár az eredeti harmada vagy még kisebb is lehet, így könnyebben szállítható (kevesebb flopi szükséges hozzá), illetve olcsóbban továbbítható (rövidebb idő alatt ér át a hálózaton). A tömörítési eljárásokat alapvetően két csoportba soroljuk. **Veszteségmentes tömörítés*** során az adatok információtartalma nem változik, míg **veszteséges tömörítés*** esetén egy része visszaállíthatatlanul elvész.



A Windows XP-ben a kijelölt fájlkból a helyi menü segítségével készíthetünk egy tömörített állományt, amit a továbbiakban a mappákhoz hasonlóan kezelhetünk. Veszteséges vagy veszteségmentes ez a tömörítés?

Veszteségmentesen kell tömöríteni a programokat és a dokumentumokat. Ilyenkor, az állományok „kitömörítésekor” visszaáll az eredeti állapotuk. Az eljárás lényege, hogy a gyakran ismétlődő szótöredékeket egy rövidebb azonosítóval helyettesítjük. (Ebben a könyvben pl. a Windows szót helyettesíthetnénk egy #W azonosítóval.)

Veszteséges tömörítést alkalmaznak a képek, hangok, filmek tárolásánál. Természetesen ez a minőség romlását eredményezi. Ez azonban nem, vagy csak csekély mértékben vehető észre, miközben a fájl mérete drasztikusan csökken. Néhány közismert veszteséges tömörítés: JPEG (képeknél), MPEG (filmeknél), MP3 (hangoknál).

A fájlkezelésben általában veszteségmentes tömörítést használunk. Ehhez igen sok program szerezhető be, ezeknél nem elsősorban a program neve, hanem a használt fájlformátum lényeges, pl. ZIP, ARJ, RAR stb. *Sok esetben maga az operációs rendszer is támogatja a tömörítést.* Például Windows XP esetében a ZIP fájlok mint „tömörített mappák” jelennek meg, és kényelmesen mappaként kezelhetők.

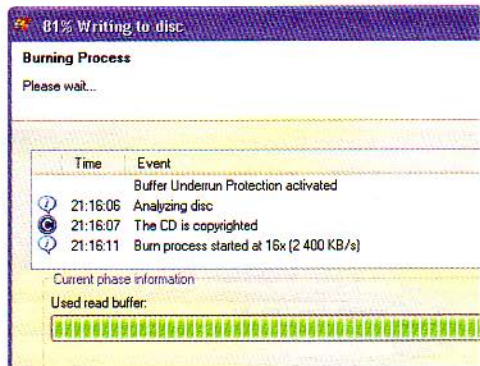
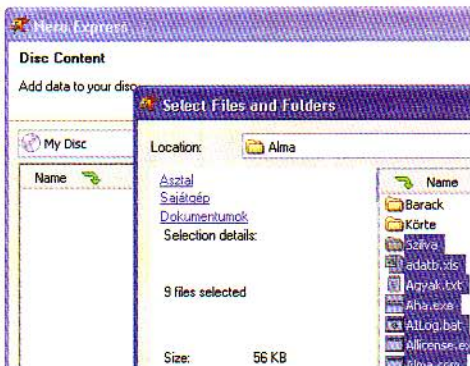


A WinRAR tömörítőprogram. A tömörítőprogramok sokféle formátumot ismernek. Megadható a tömörítés aránya, a tömörített állomány pedig feldarabolható, titkosítható, lehet önkicsomagoló stb. Olvassuk le az ábráról a tömörített fájl nevét, típusát, és eredeti méretét!

Adatok archiválása CD-re

A CD lemez a viszonylag nagy kapacitásával és alacsony árával az egyik leghatékonyabb adathordozó. Hátránya, hogy mivel az adatokat más elven tárolja, mint a winchester, ezért a CD-re írás kicsit nehézkes. Az adatokat törölni csak újraírható CD-ről lehet, de ezeknél is a lemez formázásával. Ha a CD-re viszonylag kevés adatot írtunk, lehetőségünk van arra, hogy később további adatokat tegyünk rá.

A CD-írás* két lépésből áll. *Első lépésként kijelöljük a CD-re másolandó állományokat, a második lépés azok tényleges kiírása a lemezre.* A CD-írást sok esetben már az operációs rendszer (pl. Windows XP) is támogatja, de a felhasználók előszeretettel használnak külső CD-író programokat azok extra szolgáltatásai miatt. Ugyanakkor érdemes figyelni arra, hogy *egy adott programmal írt CD-t később egy másik programmal nem feltétlenül tudunk továbbírni,* de újraírható CD-k esetén a CD ekkor is törölhető.



CD-írás Neroval. A varázsló végigvezet az egyes lépéseken.

A CD-író programokkal nemcsak adat, hanem audio vagy video CD-t is készíthetünk.

Vírusok, vírusvédelem

A **vírusok*** károkozás céljára létrehozott, **önreprodukáló** (tehát szaporodni képes) **programok**. A vírusok létrehozásának célja lehet az illegális fájlmásolás megakadályozása, konkurens cég vagy politikai irányzat adatainak, programjainak megsemmisítése, vagy egyszerűen bizonyítási kényszer. Az *internet elterjedésével egy-egy „jól megírt” vírus pillanatok alatt megfertőzhet több millió számítógépet is*, óriási kárt okozva.

A vírusok hatása lehet bizonyos fájlok letörlése, a winchester átkonvertálása, grafikus vagy zenei hatások, bizonyos menüpontok vagy ikonok eltűnése, dokumentumok váratlan módosulása, gépünkről induló e-mail áradat stb.

A vírusokat a következő módon szokták csoportosítani:

- **BOOT [bút] vírusok***. Valamennyi flopi, illetve winchester tartalmaz egy boot szektor nevű részt, amely az operációs rendszer indításához szükséges, és akkor is jelen van, ha az adott lemez nem tartalmazza az operációs rendszert. Ha a vírus ezt a részt támadja meg, egy bent felejtett flopi esetén is képes aktivizálni magát.

- **Appendelő vírusok***. A futtatható állományokhoz (EXE, COM) fűzik hozzá magukat, és az ilyen fájlok indításával aktivizálódnak.

- **Trójai falovak***. Hasznos programnak álcázott pusztító célú programok.

- **Férgek***. Általában nem szaporodnak, hanem az adott rendszer adatainak (pl. rendszergazda jelszava) megszerzése a céljuk. Működésük során rejtve maradnak, feladatuk elvégzése után gyakran megsemmisítik önmagukat.

- **Makróvírusok***. A legtöbb modern alkalmazói program lehetővé teszi, hogy az alapvető programozási ismeretekkel rendelkező felhasználók új funkciókkal bővítsék. Ezek a beépülő programok a **makrók**, amelyek a dokumentumokkal együtt vihetők át az egyik gépről a másikra. Sajnos a makró is készülhet pusztító szándékkal, ilyenkor megfertőzheti a többi ugyanolyan típusú fájlt. Ekkor beszélünk makróvírusokról.

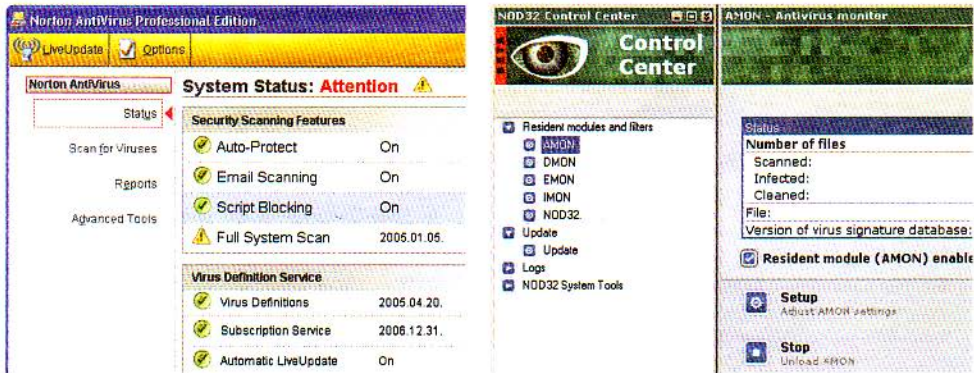
A **vírusok elleni védekezésben fontos elem a megelőzés. Az idegen forrásokból származó fájlokat** (pl. internetről letöltött dokumentumok) vagy az ismeretlen eredetű (többnyire shareware [server] vagy freeware [fríver]) programokat **mindig célszerű ellenőrizni**. Ugyancsak fontos, hogy programjainkról, **dokumentumainkról készítsünk biztonsági másolatot**.



„Légy szíves, nézd meg a tőlem kapott szerelmes levelet!” Ha a címzett megnyitotta a levelet, akkor a vírus a levelezőprogram címlistáján lévő minden címre elküldte magát. A hatalmas levélforgalom túlterhelte az internetet, a kiesett munkaidő kb. 10 milliárd dollár kárt okozott. Milyen típusú vírus volt az „Iloveyou” [ájlvájú]?

Az aktív védekezés fontos eleme a **vírusirtó program* használata**. Ezek általában kétféle módon működhetnek. Egyfelől a már ismert vírusok jellegzetes kódrészleteit keresik. Másfelől bizonyos kódrészletek, mint például az önmagát másoló programrész, a vírusokra jellemzőek, ezek a vírus ismerete nélkül is megtalálhatók (*heurisztikus analízis*). Míg az első módszerhez folyamatosan frissíteni kell a vírusdefiníciós fájlt, addig a második tévesen is riaszthat, ezért a mai vírusirtók a két módszert általában kombinálják.

Az ún. rezidens vírusirtó programok folyamatosan a memóriában tartózkodnak, és valamennyi fájlműveletet ellenőrzik. Használatuk folyamatos internetkapcsolat esetén feltétlenül ajánlott. Előnyük, hogy működésük automatikus, és a vírusdefiníciós fájlok az interneten át folyamatosan frissíthetők.



A Norton Antivírus (balra) és a NOD32 (jobbra) jellegzetes képernyője. Mindkettő rezidens, tartalmaz heurisztikus analízist, és vírusdefiníciós fájljaikat az interneten keresztül folyamatosan frissítik. Mit is jelentenek ezek a fogalmak?

Törölt állományok helyreállítása

A főlöslegessé vált fájlok és mappák törlése helyet szabadít fel a merevlemezen, azonban gyakran előfordul, hogy a nemrég letörölt állományra mégis szükségünk van. Ezért a legtöbb operációs rendszer törléskor nem távolítja el véglegesen a letörölt állományokat, hanem egy átmeneti tárolóba (Windowsban pl. *lomtár**) teszi.

A lomtár mérete többnyire beállítható. Ha megtelik, akkor a benne lévő állományok egy részét az operációs rendszer véglegesen törli, így szabadít fel helyet. Ha a megadott maximális méretnél nagyobb fájlt törölünk, akkor azt az operációs rendszer kénytelen véglegesen törölni, hiszen nem fér el benne.



A lomtár ikonja az egyes operációs rendszerekben.

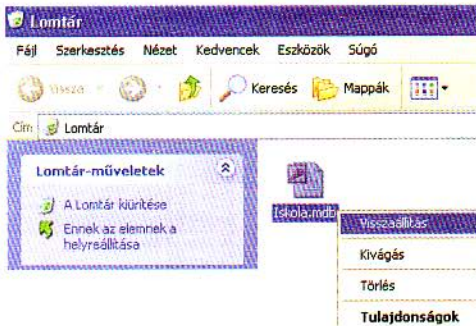
Balról jobbra: Mac OS X, Suse Linux, Uhu Linux, Windows 98, Windows XP.

Operációs rendszer

A lomtár *nem minden meghajtó fájljait képes tárolni*. Operációs rendszertől függően ezt magunk is beállíthatjuk, de a legtöbb esetben a floppy, a pendrive, a CD vagy a hálózati meghajtóról törölt állományok nem kerülnek a lomtárba, így később nem is állíthatók vissza.

A lomtárban lévő fájlokat **a lomtár megfelelő parancsaival a lomtárból az eredeti helyére visszaállíthatjuk**, vagy véglegesen törölhetjük.

A lomtár használata nem kötelező, *törléskor előírhatjuk a fájlok azonnali és végleges eltávolítását is* (pl. Windows XP esetén a SHIFT Del billentyűkombinációval).



- Lemez meghajtók egymástól független beállítása
- Azonos beállítás használata az összes meghajtóra:

A törölt fájlok ne kerüljenek át a Lomtárba, hanem azonnal semmisüljenek meg.



10%

A Lomtár maximális mérete (az egyes meghajtók százalékában)

- Törési jóváhagyás megjelenítése

Belépve a lomtárba a kijelölt fájlokat a helyi menü vagy a munkaablak parancsaival visszaállíthatjuk vagy véglegesen törölhetjük. Milyen fájlok nem kerülnek törléskor a lomtárba?

A Windowsban a lomtár tulajdonságait a helyi menü **Tulajdonságok** menüpontjával érhetjük el. A lomtár maximális méretét a csúszka húzásával szabályozhatjuk.

Együttműködés többfelhasználós környezetben

Az informatikai biztonság

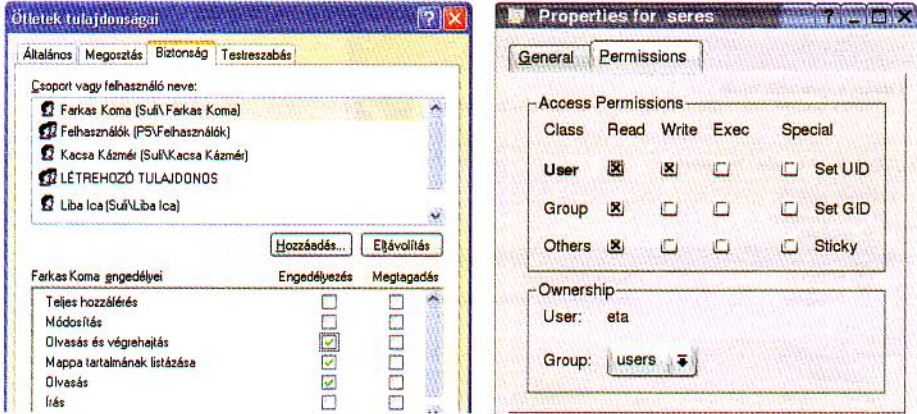
Többfelhasználós környezetben – akár egy gépen dolgoznak többen, akár hálózatra vannak kapcsolva a számítógépek –, fontos szerepet kap a biztonság kérdése. **Az informatikai biztonság*** alapvetően két dolgot jelent: az egyik az **információs rendszer megbízható működése**, a másik az **információvédelem**.

Az **információs rendszer megbízható működése** azt jelenti, hogy a szükséges adatok és a kezelésükhöz szükséges hardver- és szoftvereszközök folyamatosan rendelkezésre állnak. A hardver- és szoftverhibák ellen gyakori mentéssel, biztonsági másolatok készítésével védekezhetünk. Ez elsősorban a felhasználók feladata annak ellenére, hogy hálózatos környezetben a nagy közös táruk adatait a rendszergazdák időnként lementik más tároló eszközre, pl. DVD-re. (Ezt backupnak [bekáp] nevezik).

Az **információvédelem** azt jelenti, hogy az adatokhoz csak azok férnek hozzá, illetve az adatokat csak azok módosíthatják, akiknek ehhez joguk van. Ehhez természetesen szükséges a felhasználók megfelelő azonosítása, amit az operációs rendszerek a felhasználói név és jelszó segítségével oldanak meg. A meghajtókhoz, a mappákhoz illetve a fájlokhoz

való **hozzáférési jogokat*** az operációs rendszerek belépési névhez rendelik hozzá. Így a belépési név és jelszó páros dönti el, hogy mit tehetünk a rendszerben és mit nem. A jelszó akkor biztonságos, ha számokat, kis- és nagybetűket egyaránt tartalmaz.

Általában mindig megtalálható két különleges felhasználó. Az egyik a *rendszergazda* (*administrator*), aki kiemelt jogokkal rendelkezik. Mások jelszavát ő sem ismerheti meg, de szükség esetén módosíthatja. A másik a többnyire jelszóval sem védett *vendég* (*guest* [geszt]), akinek csak minimális jogai vannak: lényegében csak olvashat bizonyos fájlokat.

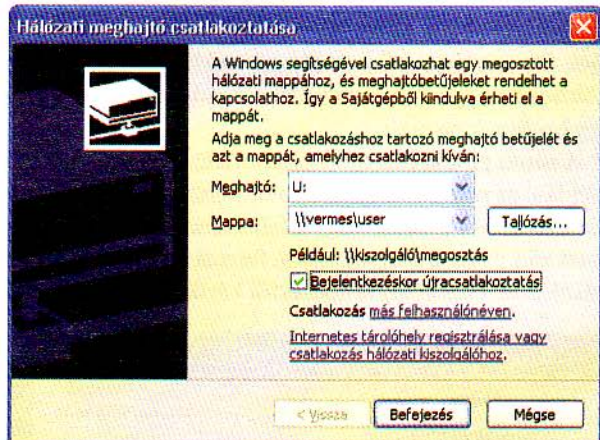


Hozzáférési jogok beállításához kattintsunk a fájl ikonjára jobb gombbal, és a **Tulajdonságok** (**Properties** [propertiz]) pontban válasszuk a **Biztonság** (**Permissions** [pörmisönsz]) fület. Balra: Windows, jobbra: Linux.

Hálózati meghajtó

A hálózatok fontos szolgáltatása, hogy a közös adatokat csupán egy helyen, például egy szerveren, egy alkönyvtárban **tároljuk**. Ehhez a felhasználók hozzáférési lehetőséget kapnak.

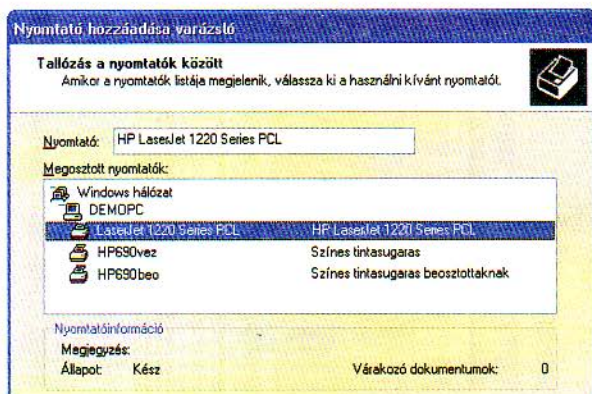
A Windows-rendszerben az ilyen alkönyvtárat *megosztott mappának* nevezik. A felhasználók a megosztott mappát hálózati meghajtóként látják, és a többi meghajtóhoz hasonlóan kezelhetik. Előírható, hogy mely felhasználók milyen jogokkal férjenek hozzá a megosztott mappához.



Hálózati meghajtó csatlakoztatása Windows XP-ben. Az U: meghajtó ténylegesen a vermes nevű gép user néven megosztott mappája. Miért előnyös a hálózati meghajtó használata?

Hálózati nyomtató csatlakoztatása

A hálózatok hasznos szolgáltatása, hogy a mappákhoz hasonlóan a **nyomtatók is megoszthatók**. Ez lehetővé teszi, hogy egy adott nyomtatót a felhasználók gyakorlatilag bármely számítógépről elérjenek. Mivel a legtöbb hálózatban a felhasználók viszonylag ritkán nyomtatnak, **így viszonylag kevés, központilag felügyelt nyomtató segítségével valamennyi felhasználó igényeit ki lehet elégíteni**.

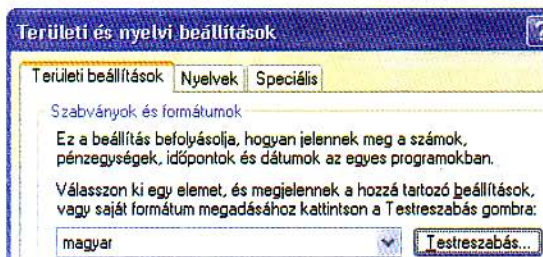


Hálózati nyomtató telepítése. A telepítendő nyomtatót csak ki kell választani a listáról. Telepítés után úgy használhatjuk, mint a helyi nyomtatókat. Mik a hálózati nyomtató használatának előnyei és hátrányai?

Egyéni munkakörnyezet kialakítása

Többfelhasználós környezetben az operációs rendszer lehetővé teszi, hogy a felhasználók **egyéni munkakörnyezetet** alakítsanak ki. Ez azt jelenti, hogy a felhasználók beállíthatják a nekik megfelelő háttérteret, felbontást, megosztott mappákat, hálózati nyomtatókat stb. Helyi hálózat esetén ezek a beállítások kilépéskor a szerverre mentődnek. Ha ezután a felhasználó egy másik gépen lép be, akkor beállításai arra a gépre is lementődnek, így ott is a megszokott munkakörnyezetet kapja. Az egyéni munkakörnyezet profilnak nevezik.

A profil az alkalmazói programokkal kapcsolatos beállításokat is tartalmazhat, ilyenek például az utoljára mentett dokumentumok listája vagy a **területi és nyelvi beállítások***. Ez utóbbi azt jelenti, hogy a felhasználó az operációs rendszerben állíthatja be az általa használt dátum, pénznem stb. formátumokat. Az alkalmazói programok ezeket a beállításokat az operációs rendszertől kéri le, így biztosítják az egységes megjelenést.



A Windowsban a Vezérlőpultból érhető el a Területi és nyelvi beállításokat. Ellenőrizzük ezeket a beállításokat a gépünkön! Megoldható-e, hogy egy angol nyelvű operációs rendszer magyar területi beállításokkal működjön?

7. A Windows rendszerben egy fájl neve *meglepi.txt.exe*. Mi a fájl kiterjesztése? Elképzelhető-e, hogy az *exe* végződés nem jelenik meg a monitoron? Milyen esetben lehet ez kellemetlen?

8. Soroljuk fel a számítógépünkön elérhető háttértárat! Mi az azonosítójuk, fizikailag milyen elven működnek, hol találhatóak, mekkora a kapacitásuk és mire kell vigyázni a kezelésüknél?

9. Soroljunk fel minél több érvet, hogy miért és miről érdemes biztonsági másolatot készíteni! Ennek ellenére miért nem készítenek sokan biztonsági másolatot?

10. Milyen lehetőségeket kínál az általunk használt operációs rendszer a meghajtók, mappák és fájlok adatainak megjelenítésére?

11. Keressük meg gépünkön azokat a fájlokat, amelyek *txt* kiterjesztésűek és tartalmazzák a *Windows* szót!

12. Hogyan másolhatunk át egy fájlt floppyra vagy pendrive-ra? Próbáljuk meg a folyamatot úgy összefoglalni, hogy megértse az az idősebb rokonunk is, aki még kezdő számítógép-felhasználó!

13. Mi a vágólap? Milyen műveleteket lehet vele végezni? Hányféle lehetőséget tudunk felsorolni ezek megvalósítására?

14. Miért nem célszerű a dokumentumokat veszteségesen tömöríteni? Tömörítsünk be veszteségmentesen egy bittérképes ábrát, egy formátatlan szöveget tartalmazó dokumentumot és egy futtatható fájlt! Melyik esetben mennyi a megtakarított hely?

15. Soroljuk fel a vírusok fajtáit és terjedésük módját! Mondjunk példát az utóbbi idők „leghíresebb” internetes vírusfertőzésére! Mennyi idő alatt terjedt el a vírus? Mennyire becsülik az okozott kárt?

16. Biztos, hogy helyet szabadítunk fel, ha letöröltünk egy fájlt? Ha az általunk használt operációs rendszer kezeli a lomtárat, nézzük meg a tartalmát! Mekkora lehet a maximális mérete?

17. Mit jelent az informatikai biztonság? Hogyan valósítják meg ezt iskolánk esetében?

18. Hogyan állíthatjuk be az általunk használt operációs rendszer esetében a hozzáférési jogokat? Ki és milyen hozzáférési jogokat állíthat be?

19. Mint ismeretes, az Európai Unió közös pénze az euró, a tervek szerint hazánk is csatlakozni fog az eurózónához. Hogyan állítható be az operációs rendszerben, hogy az alapértelmezett pénznem az euró legyen? Elképzelhető-e, hogy az operációs rendszer helyes beállításainak ellenére egy alkalmazói program más formátumot lát? Miért?

Kommunikáció hálózaton

A hálózatok áttekintése

A számítógép-hálózatok csoportosítása

Korunk jellemzője, hogy a számítógépek nem önállóan működnek, hanem egymással összekapcsolva hálózatokat alkotnak.

A számítógéphálózatokat leggyakrabban kiterjedésük alapján szokták csoportosítani. Tekintsük át, milyen kategóriák alakultak ki!

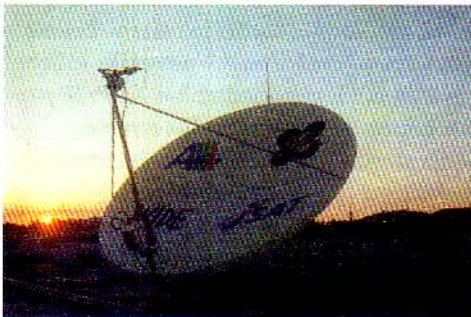
Ha összekapcsoljuk saját elektronikai eszközeinket (mobiltelefonunkat, laptopunkat, asztali számítógépünket) **személyi hálózatot*** (PAN, Personal Area Network [perszonális hálózat]) **hozunk létre**. A PAN eszközeit összeköthetjük vezetékekkel is, de gyakoribbak a vezeték nélküli megoldások. Ilyen például a rádióhullámokkal működő *bluetooth* vagy az infravörös sugarakat használó *IrDA*.

Helyi hálózatot* (LAN, Local Area Network [helyi hálózat]) **alakítanak ki, ha a számítógépeket egy intézményen belül kötik össze**. Ilyen például az iskolai hálózat is.

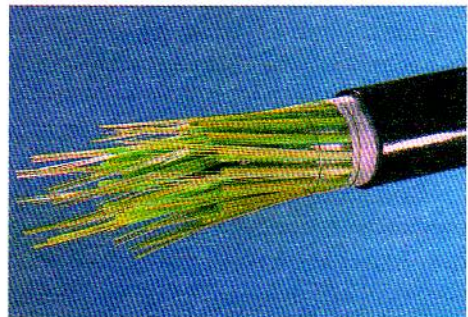
A városi hálózat* (MAN, Metropolitan Area Network [metropoliten hálózat]) **egy viszonylag kis területen** (pl. egy városban) **összekapcsolt helyi hálózatokból jön létre**. Városi hálózathoz jutunk például, ha egy településen összekapcsoljuk az iskolák helyi hálózatait.

Ha a helyi és a városi hálózatokat kontinentális méretekben kapcsoljuk össze, akkor nagy területű hálózathoz* (WAN, Wide Area Network [világ hálózat]) **jutunk**. Ennek tipikus képviselője az internet.

A hálózatok összekapcsolása történhet *telefonkábel* (pl. a telefonhálózaton keresztül), *üvegszállal* vagy *műholdak* segítségével is.



A parabolaantennáról küldött jel műhold segítségével jut el a vevőhöz. Az antenna viszonylag olcsón és gyorsan telepíthető, azonban az adatátviteli sebesség csak kb. 1-2 Gb/s.



Az optikai kábel vékony üvegszálakat tartalmaz. Egy üvegszálban egyszerre több fénysugár is haladhat, így akár 400 Gb/s sávszélesség is elérhető vele. Kiépítése költséges.

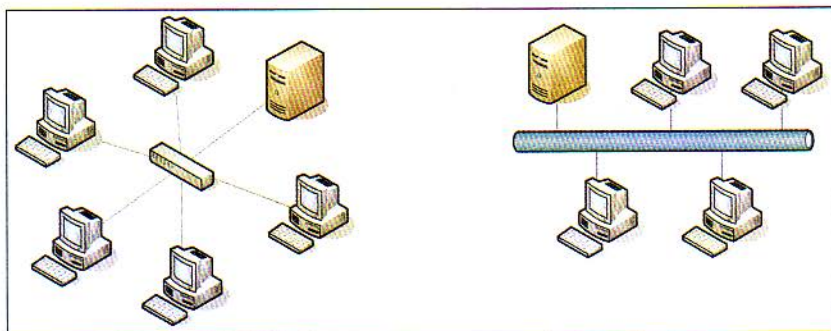
Helyi hálózatok

Mint azt a korábbiakban láttuk, a számítógépek helyi hálózatba kötése több előnnyel is jár. Lehetővé teszi az erőforrások megosztását, a közös adatokat egy adattárban tárolhatjuk, és központilag megvalósítható a hálózat felügyelete is.

A helyi hálózat kialakításához szükségünk van olyan *hardvereszközökre*, amelyek lehetővé teszik a gépek összekapcsolását, és *hálózati szoftverekre*, amelyek megvalósítják a gépek közötti kommunikációt. A hálózat folyamatos üzemeltetéséhez pedig megfelelő felkészültségű szakembert is (*rendszergazdát*) kell alkalmazni.

A helyi hálózat fizikai kialakításához a számítógépekben lennie kell egy-egy hálózati kártyának, ehhez csatlakoznak a gépeket összekötő vezetékek. A **hálózati kábelek elrendezését topológiának* nevezük.**

A helyi hálózatok esetében alapvetően kétféle topológiát használnak. **Csillagtopológia** esetén valamennyi számítógép egy központi berendezéshez (pl. switch [szvics]) csatlakozik. **Síntopológia** esetén a számítógépek egyetlen vezetékre vannak felfűzve.



A csillagtopológia esetén a munkállomások és a szerverek egy központi berendezéshez csatlakoznak, síntopológia esetén pedig egyetlen vezetékre vannak felfűzve. Milyen előnyei és hátrányai vannak a kétféle kiépítésnek?

Ahhoz, hogy a hálózat gépei együtt tudjanak működni, szabványosítani kellett a kommunikáció módját. Az **adatátviteli szabványt protokollnak* nevezik.** Jóllehet többféle protokoll létezik, de helyi hálózatok esetén is a legelterjedtebb a **TCP/IP protokoll***, amely az **internet által összekapcsolt gépek kommunikációs nyelve.**

A helyi hálózatokat az erőforrások elérési módja szerint három fő csoportba sorolják:

- **Egyenrangú gépek hálózata** (*peer-to-peer hálózat* [pírtupír]). Bármelyik gép megoszthatja nyomtatóját vagy valamelyik meghajtóját úgy, hogy a többiek is hozzáférhessenek. A peer-to-peer hálózat kialakításához szükséges funkciókat a mai operációs rendszerek, pl. a Windows XP vagy a Linux már eleve tartalmazzák.

- A **kiszolgáló-ügyfél** (*server-client* [szerver klájent]) **hálózatban** a gépek nem egyenrangúak: a **szerverek* szolgáltatásokat nyújtanak a munkállomások (kliensek) részére.** A szerverre ún. szerver operációs rendszert kell telepíteni, ilyen pl. a Windows Server 2003, a Novell Netware vagy a Linux.

• Az elosztott (*host-terminal*) hálózatokban a programok egy központi számítógépen (*host*) futnak, a többi számítógép (*terminálok*) csupán beviszi és megjeleníti az adatokat. Ehhez természetesen nagy teljesítményű központi számítógép (*mainframe*) szükséges.

Az internet: a hálózatok hálózata

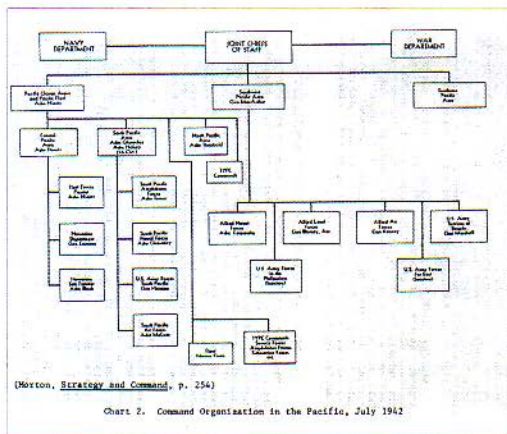
Az internet kialakulása

Az internet elődjét az *ARPANET*-et az Amerikai Védelmi Minisztérium kutatóintézetében (*Advanced Research Project Agency* [edvenszd riszörccs prodzsekt édzsönszi]) fejlesztették ki az 1960-as években. A hálózatot 1969-ben helyezték üzembe az USA négy egyetemén, és a nyilvános telefonhálózaton át kötötték össze. Mivel katonai célok-ból hozták létre, megtervezésénél fontos szerepet kapott, hogy ha egy esetleges külső támadás esetén egyes részei elpusztulnának is, a megmaradó rész továbbra is üzemképes legyen (pl. atomcsapás, terrorista merénylet). Ezt a következő eszközökkel érték el:

• Az adatátvitel módját *hardverfüggetlen* módon szabványosították. Így különböző típusú eszközök (az internetes hűtőszekrénytől a mobiltelefonig) köthetők a hálózatba, ráadásul nagyon sokféle módon (telefonkábel, ıvegszál, rádióhullámok stb.).



Az ARPANET fejlesztőcsapata. A hálózat kezdetben négy egyetemet kötött össze az Egyesült Államokban a nyilvános telefonhálózaton keresztül.



A hagyományos szolgálati út nem teszi lehetővé az egyes szintek megkerülését (www-cgsc.army.mil). Az úthálózat viszont lehetővé teszi, hogy egy-egy útszakaszt vagy várost elkerüljünk. Mondjunk példát arra, amikor az első, illetve arra is, amikor a második megoldás alkalmazása célszerűbb!

• **Nincs abszolút központja**, felépítése leginkább a közlekedési úthálózathoz hasonló. Ahogy egyik városból a másik városba útfelbontás esetén többnyire más úton is eljuthatunk, úgy egy esetleges külső támadásból eredő *szakadás esetén az adatok más útvonalon haladhatnak*.

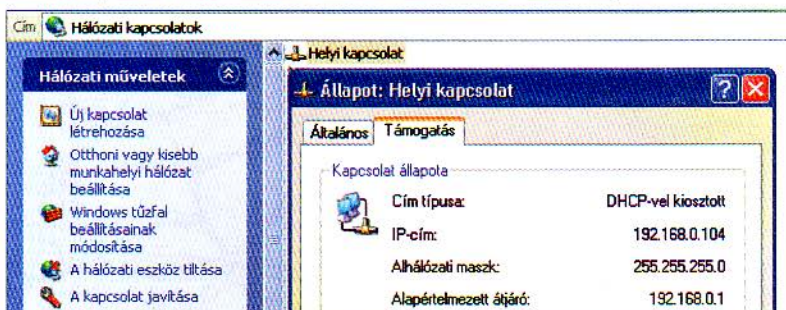
• A kommunikációban gyakori, hogy bizonyos időben („csúcsidő”) rengeteg adat áramlik, míg máskor alig van terhelve a hálózat („éjszakai tarifa”). Ezért nem szerencsés, ha két számítógép folyamatosan lefoglal egy vonalat, és emiatt egy másik (esetleg fontosabb) „hívó” várakozni kényszerül. Megoldásként azt találták ki, hogy **az adatokat csomagokra bontják, ezeket** ellátják a megfelelő azonosító adatokkal (feladó, címzett, a csomag sorszáma stb.), és **egyenként küldik át a hálózaton**. Így egy adott vonalon különböző helyekre áramló csomagok követhetik egymást.

• **Az összetartozó csomagok a hálózat terheltségétől függően akár különböző vonalakon keresztül is utazhatnak**. A csomagok irányítását speciális számítógépek (*routerek* [rúterek]) végzik. A beérkező csomagokat a célállomáson futó programok összerakják, a sérült vagy hiányzó csomagokból pedig ismétlést kérnek a küldő számítógépétől.

Mivel az ARPANET katonai hálózatként jött létre, csak az USA Védelmi Minisztériumával kapcsolatban álló intézmények használhatták, miközben a sikerek miatt alaposan megnőtt rá az igény. Ezért a katonai hálózatot **1983-ban** különválasztották (MILNET), és a maradék részből **létrejött a mai fogalmaink szerinti internet**.

A számítógépek azonosítása az interneten

Az internetre kapcsolt minden eszköz egy 32 bites számot kap, melyet **IP-címnek*** neveznek. Az egyszerűbb kezelhetőség érdekében az IP-címeket 8 bites csoportosításban, tízes számrendszerbe váltják át, és így adják meg: 121.134.167.13. (Az internet terjedésével az így rendelkezésünkre álló IP címek rohamosan fogynak, ezért kidolgozták és részben már használják is a 128 bites IP címeket alkalmazó szabványt.)



Számítógépünk IP címét a Windows Vezérlőpultján a Hálózati kapcsolatoknál érhetjük el. Az alhálózati maszk azt mutatja meg, hogy a cím mely része azonosítja a hálózatot. Az Alapértelmezett átjáró az a számítógép, amely hálózatunkat összeköti az internettel. Ellenőrizzük ezeket a beállításokat gépünkön!

Mivel ezeket a számokat nem egyszerű megjegyezni, ezért inkább *szótöredékekből álló ún. internetcímekkel** helyettesítik. Az internetcím alapján az IP-számot külön szerverek (DNS, Domain Name Service [domén név szörvisz]) keresik vissza automatikusan. Az internetcím felépítése hierarchikus, és hátulról előre haladva kell értelmezni őket.

Például a *www.mkogy.hu* esetében a *hu* végződés jelzi, hogy az internetcímet Magyarországon jegyezték be (*hu* a *fődomain* [domén] vagy a *TLD*, *Top Level Domain* [top levöl domén]). Ezen belül a számítógép az *mkogy* azonosítójú hálózaton van (*mkogy* a *domain* vagy az *SLD*, *Second Level Domain* [szekond levöl domén]). Végül a gép az *mkogy* domainen belül *www* azonosítóval rendelkezik, a kialakult gyakorlat szerint így nevezik a webszolgáltatást végző számítógépet.

A kétbetűs fődomain általában az országra utal, kivétel az Egyesült Államok, ahol a hárombetűs fődomainek a szolgáltató jellegét adják meg: *gov* = állami intézmény, *com* = üzleti szféra, *mil* = katonai szervezet, *edu* = oktatási intézmény stb.

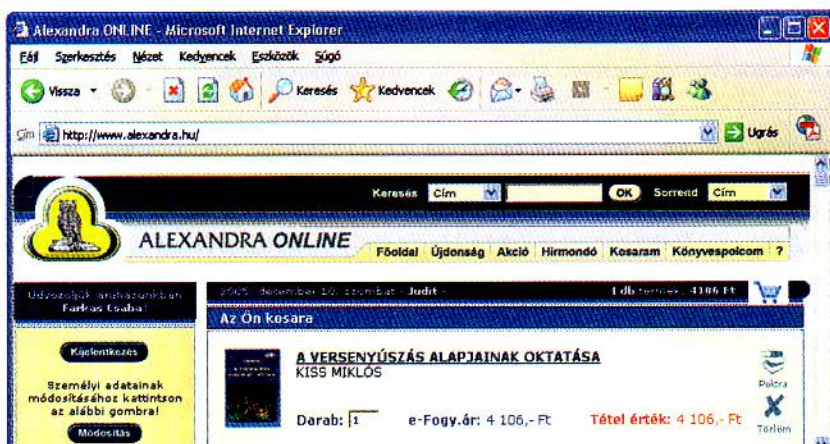
Gyakori megoldás, hogy az internetre csatlakozó otthoni gépünk csak az „internetezés idejére” kap IP-címet a szolgáltató ún. *DHCP-szerverétől*. Ezt *dinamikus IP-címnek* nevezik; ezzel a megoldással a rendelkezésre álló IP-cím tartomány jobban kihasználható.

A világháló használata

A World Wide Web

Az internet egyik leggyakrabban használt szolgáltatása a World Wide Web [vörld vájd veb] vagy egyszerűen web, terjedő magyar elnevezéssel *világháló*.

Cégek, intézmények részére a webhely fenntartása rendkívül előnyös, mivel viszonylag olcsón, ugyanakkor naprakészen tudják tájékoztatni ügyfeleiket. A webhely multi-médiás, tehát egyaránt tartalmazhat szöveget, képet, hangot, filmet stb.*, így gyakorlatilag minden információ elhelyezhető rajta. Nem véletlen, hogy az internetcím többnyire a telefonszámmal egyenrangú helyet kap a cégek elérhetőségében. A felhasználók a világhálón böngészve, kényelmesen és egyszerűen juthatnak hozzá a szükséges adatokhoz, akár vásárolhatnak is. Eközben a világháló lehetőséget ad a kikapcsolódásra és szórakozásra is.



Vásárlás az interneten. A vevő otthonról rendelhet, az árut a futár házhoz szállítja. Milyen árucikkeket célszerű így megrendelni? Mire kell ügyelni a megrendelés során?

A világháló hátránya, hogy sok webhelyen (főleg az ingyenes szolgáltatók tárhelyein) az információ ellenőrizetlen módon jelenik meg. Így nem ritka a téves adatokat tartalmazó vagy illegálisan elhelyezett információ sem, sőt néha akaratlanul károkozó programokat is letölthetünk. A böngészés tehát egyfelől nagy szabadságot ad a felhasználónak, ugyanakkor felelősségteljes magatartást is vár el tőle.

A világháló működése

A webhelyet tartalmazó számítógépet a webhely internetcíme azonosítja, például így: *www.microsoft.com*. Ezen belül az adatok egy mappaszerkezetben helyezkednek el, tehát a fájlokat a fájlkezelésben tanult módon adhatjuk meg. Ezt a mappaszerkezetet a **webszerver* program** kezeli és **teszi elérhetővé az internet irányába**.

Tekintsük a következő példát: *http://www.microsoft.com/hun/oktatas/default.msp*x. Itt

http: az adatátvitel szabványára utal (*Hypertext Transfer Protocol*),

//: a webhely címét vezeti be („duplaper”),

www.microsoft.com a webhely internetcíme,

*/hun/oktatas/default.msp*x a fájl elérési útja a webhelyen belül.

A webhely megtekintéséhez egy **böngészőprogramra* van szükségünk** a számítógépünkön. A böngésző felveszi a kapcsolatot a megadott webszerverrel, és megjeleníti onnan az alapértelmezett oldalt. Természetesen többféle böngészőprogram létezik, pl. az Internet Explorer, Netscape [netszkép] Navigator, Opera, FireFox [fájörfoksz] stb.



A Netscape Browser [netszkép brauzer] 8 böngésző grafikus felülete

Böngészés közben a webhely tartalma weblapokban* jelenik meg. Mivel mind a webszerver, mind a böngészőprogram sokféle lehet, fontos, hogy a weblapok tartalmát egységes módon kezeljék. A **weblapok leírását a HTML-szabvány* rögzíti** (*HyperText Markup Language* [hiperteksztr márkáplengvidzs]). A webszerveren lényegében minden weblaphoz tartozik egy HTML-kódú (de nem feltétlenül *html* kiterjesztésű) fájl. Ez tartalmazza, hogy az adott lapon hogyan jelenjenek meg a szövegek, és hogyan kerüljenek rá a képek, hangok, filmek stb. Ez utóbbiak *ténylegesen más fájlokban helyezkednek el*, ezért önmagában a HTML-fájl mentése nem jelenti a weblap tartalmának teljes lementését.

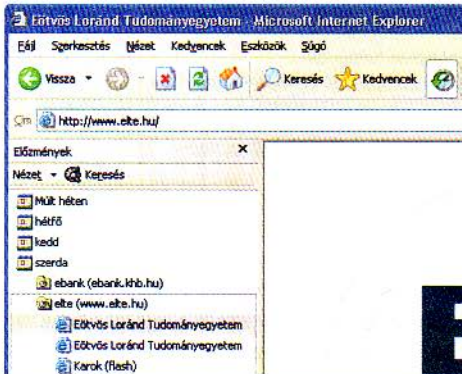
A **weblapok közötti átjárhatóságot a hivatkozások* (linkek) biztosítják**, ezekre kattintva egy másik weblapra jutunk. A hivatkozás *ténylegesen a célweblap címét tartalmazza*, de a HTML-szabvány lehetővé teszi, hogy a felhasználó helyett egy számára könnyebben érthető információt lásson.

A böngészők használata

Mivel böngészéskor a világhálón keresünk, illetve a webhely fájljai között lépkedünk, **a böngésző kezelése hasonló a grafikus fájlkezelő programokéhoz.**

A keresett webhely címét egy beviteli mezőben adhatjuk meg, az alapértelmezett *http:* és gyakran a címben szereplő *www* rész is elhagyható. Ha már az adott webhelyen jártunk, akkor azt (bizonyos ideig) a *címlista legördítésével is kiválaszthatjuk.*

Böngészés közben lehetőségünk van a már felkeresett lapok között *előre-hátra* lépkedni, a lassan letöltődő lapok letöltését *leállítani* vagy *újrakezdeni*, illetve a *kezdőlaphoz* visszatérni. A kezdőlapot minden esetben magunk is megadhatjuk.



A korábban meglátogatott webhelyek (előzmények) megtekintése az Internet Explorerben (balra) illetve a Mozilla FireFox böngészőben (jobbra). Hasonlítsuk össze a két böngésző felületét!

A böngészők közötti fő eltérés az extra szolgáltatásokban, illetve az internetről jövő kellemetlen vagy káros jelenségek kezelésében van. Gyakorlatilag minden böngésző lehetővé teszi, hogy a már meglátogatott webhelyeket listaszerűen megtekintsük (*előzmények*), elmentsük *kedvenceink* közé, és hogy *keressünk* a világhálón. A *védelem* sokféle lehet, ilyen pl. az előugró ablakok blokkolása, az aktív elemek működésének megakadályozása, vagy bizonyos tartalmak tiltása.

A látássérültek többsége a világhálót beszélő rendszerrel (ritkábban Braille-kijelzővel [brájl]) használja. Ezek a szoftverek értelemszerűen csak a karakteres tartalmakat képesek felolvasni, illetve kijelezni. Mivel a mai weblapokat egyre inkább a képi elemek túlsúlya jellemzi, ezért egyre több webhely esetén készítenek olyan alternatív változatot is (*akadálymentes weblap*) amely karakteresen is értelmezhető és használható (például elektronikus könyvtárak).



Az akadálymentes webhely karakteresen is értelmezhető és kezelhető.

Keresés a világhálón

Az interneten való információkeresést erre kialakított webhelyek segítségével célszerű végezni. Ezek alapvetően kétféle módon működhetnek.

A **tematikus keresők*** esetében **adott témakörök közül választhatunk, amelyet aztán több lépésben fokozatosan szűkíthetünk.** Például az iskolák listájához a *hudir* keresőben a következő úton juthatunk el:

Oktatás, kultúra > Iskolák, egyetemek > Iskolák > Középiskolák

The screenshot shows a web browser window with the URL http://www.hudir.hu/hudir/Oktatas_kultura/Iskolak_egyetemek/Iskolak. The page content includes:

- Iskolák** (Schools)
- Elérési útvonal: HuDir -> Oktatás, kultúra -> Iskolák, egyetemek -> Iskolák
- További ajánlott témák
- A tárgykör részletezése: A tárgykörhöz tartozó honlapok
 1. [KÉK Oktatási Központ](#) Képesség és kreativitás.
 2. [Forrai Magániskola - Kéttannyelvű Szakközépiskola](#) (Budapest)
 3. [Védikus Bölcsélettudományi Szabadegyetem](#)
- Szakmunkásképzők
- Általános iskolák
- Középiskolák

A középiskolák példában szereplő keresése a *hudir* tematikus keresővel

A **kulcsszavas keresők*** segítségével **adott szavak vagy kifejezések előfordulásait kereshetjük a webhelyeken.** Mivel a web méretei miatt igen nagy számú találat várható, a legegyszerűbb esetben is érdemes bekapcsolni, hogy csak a magyar nyelvű lapok között keressünk.

A nagy mennyiségű találatot a kulcsszavas keresők esetében a **speciális keresés** segítségével szűkíthetjük. Ilyenkor megadhatjuk, hogy

mely szavak forduljanak elő biztosan (*logikai ÉS kapcsolat*),

melyek biztosan nem (*logikai tagadás*),

melyek közül szerepeljen valamelyik (*logikai VAGY kapcsolat*),

illetve megadhatunk szóközüket tartalmazó kifejezéseket is.

A keresés gyakran ország, méret, dátum, fájlformátum stb. feltételek megadásával tovább finomítható.

The screenshot shows the Google search interface with the following settings:

- Találatok keresése:**
 - Tartalmazzák ezen szavak **mindegyikét**: matematika
 - Tartalmazzák pontosan ezt a **kifejezést**: emelt szintű érettségi
 - Tartalmazzák ezen szavak **legalább egyikét**: feladatok példák
 - Nem tartalmazzák** ezeket a szavakat: sikertelen bukás
- Nyelv:** Ilyen nyelvű lapok közti keresés: magyar
- Fájlformátum:** Kizárólag olyan találatok megjelenítése, melyek fájlformátuma a következő: bármilyen formátum
- Dátum:** Ekkor frissített weblapok: bármikor
- Előfordulások:** Olyan találatok megjelenítése, amikben a keresett kifejezések itt fordulnak elő: a lapon bárhol
- Tartomány:** Kizárólag olyan találatok megjelenítése, melyek ezen a webhelyen vagy tartományon találhatóak: [Pt.: google.com, .org](#) [További inf.](#)

Speciális keresés a Google [gúgl] segítségével. Gyakorlásként fogalmazzuk meg, hogy milyen tartalmú lapok jelennek meg a fenti beállítások esetén!

Elektronikus levelezés

Az elektronikus levél (e-mail)

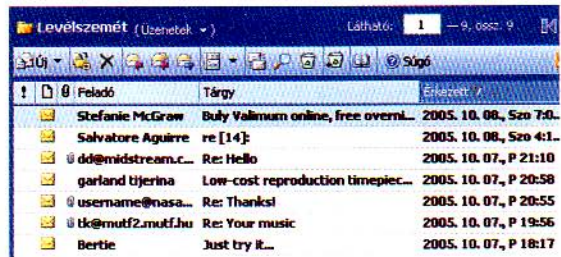
Az e-mail [ímél] az sms „nagytestvére”, nevének megfelelően a hagyományos (postai) levél elektronikus változata. **Az elektronikus levél* (e-mail*) elkészítéséhez toll és papír helyett szövegkezelő programot, továbbításához pedig az inernetet használjuk.**

Előnye, hogy számítógép előtt ülve megírása és elküldése kényelmes, gyors és olcsó, és bármilyen elektronikus anyagot (képet, dokumentumot stb.) *csatolhatunk* hozzá. Napjainkban az elektronikus levelezés *titkosítása és hitelesítése is megoldott*, tehát csak idő kérdése, hogy a hivatalos ügyintézésben is elterjedjen.

Hátránya, hogy minden próbálkozás ellenére sem tud olyan személyessé válni, mint egy saját kézzel, esetleg illatosított papírra írt levél. A fő problémát az olcsósága okozza, emiatt gyakran több **kéretlen reklámlevél (spam [szpem])** érkezik elektronikus postaládánkba, mint valódi.



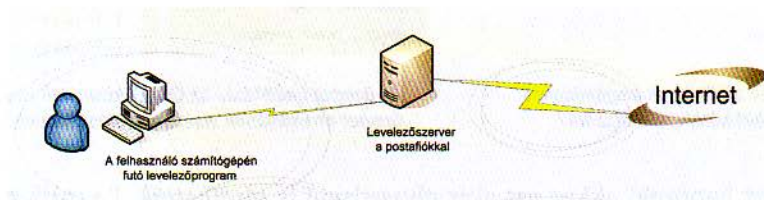
A spam eredetileg egy húskonzervet jelentett. Adjuk meg, miről ismerhető fel egy mai spam!



A kéretlen reklámleveleket a legtöbb levelezőprogram megpróbálja kiszűrni. Miért nem célszerű a levélszemlétkébe került leveleket azonnal törölni?

Az elektronikus levelezés működése

Az elektronikus levelek kezeléséhez szükségünk van egy „postaládára” vagy postafiókra, ahová leveleink megérkeznek. A **postafiókot***, amely lényegében egy mappa, **egy folyamatosan bekapcsolt állapotban lévő szerveren kell elhelyezni.**



A levelezőszerver folyamatos internetkapcsolattal rendelkezik, ehhez akkor és onnan kapcsolódunk, amikor és ahonnan akarunk és tudunk, de gyakran ez is az interneten át valósul meg.

Ennek megfelelően e-mail címünk két részből épül fel. Egyfelől tartalmaznia kell a levelezőszerver címét (pl. *akarhol.hu*), másfelől az ezen lévő postafiókunk azonosítóját (pl. *valaki*). A két részt az angol at (jelentése -nál, -nél) szócska rövidítése, a @ jel (kukac jel) kapcsolja össze: *valaki@akarhol.hu*.

A levelezőszerverre ugyan folyamatosan érkehetnek leveleink, de csak bizonyos időnként nézzük meg postafiókunk tartalmát. **A postafiók tartalmának megtekintéséhez egy levelezőprogramra* van szükségünk, amely képes felvenni a kapcsolatot a levelezőszerverrel.** *Legegyszerűbb esetben erre a célra egy böngésző is elegendő*, de ha további kényelmi szolgáltatásokat is szeretnénk használni, akkor valamilyen célszoftverre (pl. Outlook [áutluk]) van szükségünk.

Új elektronikus postafiókot igen könnyen létrehozhatunk, mivel nagyon sok *ingyenes szolgáltató* van. Ilyenkor természetesen meg kell adni a kért azonosítót, továbbá néhány adatot, amely az életkorunkra, foglalkozásunkra, érdeklődési körünkre vonatkozik. Tudnunk kell azonban, hogy sok szolgáltató automatikusan megszünteti postafiókunkat, ha azon bizonyos ideig nincs forgalom.

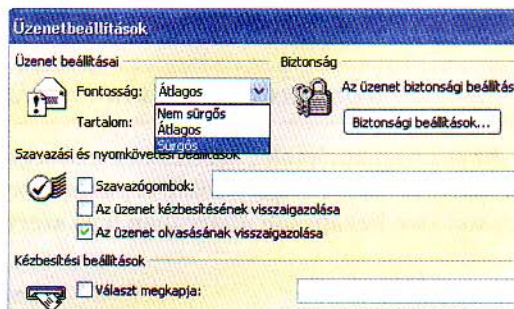
Az elektronikus levelek kezelése

Új levél írásakor kötelezően meg kell adnunk a címzett e-mail címét. (Egy levélnek több címzettje is lehet.) Illik továbbá megadni a levél tárgyát is, ellenkező esetben a levelet spamnek nézhetik.

Lehetőségünk van a levélről tetszőleges számú címzettnek *másolatot** (CC [szíszí], Carbon Copy [karbon kopi]), **illetve titkos másolatot*** (BCC [bíszíszí], Blind Carbon Copy [bláind karbon kopi] vagy *vakmásolat*) **küldeni**. Arról, hogy ki kapott másolatot, a címzett is értesül, míg arról, hogy ki kapott titkos másolatot, a címzett nem fog tudni, így pl. nem is tud neki válaszolni.



Új levél írása az Outlook segítségével.
Mely adatokat kötelező megadni?

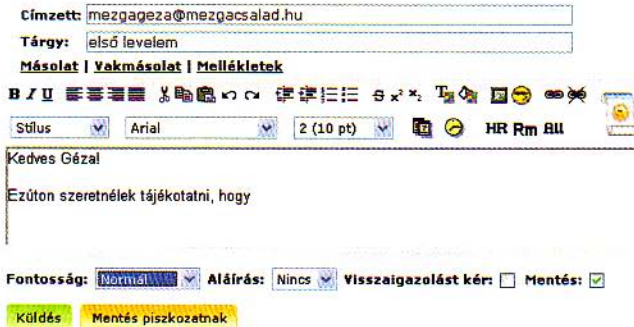


Az üzenet beállításai az Outlookban. Mit jelent „Az üzenet olvasásának visszaigazolása” lehetőség?

Ha levelet kaptunk, akkor azt akár olvasatlanul is törölhetjük. Ez spam esetében célszerű is, hiszen a levél tartalmazhat ártalmas, pl. vírusos mellékletet. Olvasás után **válaszolhatunk a címzettnek vagy** mindenkinek, aki a levelet tudomásunk szerint megkapta. Ilyenkor a levél tárgya *Re: (vagy Vá:)* szócskával egészül ki, a címzett mezőt

pedig a levelezőszoftver automatikusan kitölti. Ha a levél nem ránk tartozik, akkor azt **továbbküldhetjük** a tényleges címzettnek. Ilyenkor természetesen a címet nekünk kell megadnunk, azonban a levél tárgyát a szoftver a *Fw: (vagy Tv:)* szócskával egészíti ki.

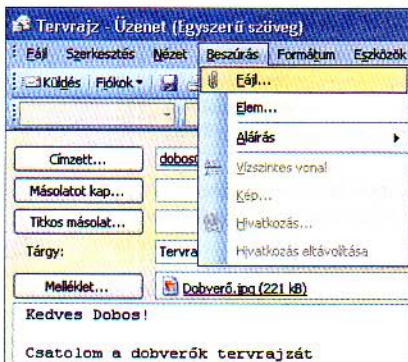
A legtöbb levelezőrendszer lehetővé teszi a levelek nyomkövetését. *Megadhatjuk a levél fontosságát*, vagyis, hogy levelünk sürgős, átlagos vagy nem sürgős. Fontos adat lehet, hogy a levél eljutott-e a címzett postafiókjába, illetve elolvasta-e a címzett. Ennek megfelelően *visszaigazolást kérhetünk a levél kézbesítéséről*, illetve elolvasásáról. Ez utóbbi visszaigazolása nem feltétlenül automatikus, a legtöbb rendszer rákérdez, hogy elküldje-e a visszaigazolást a feladónak.



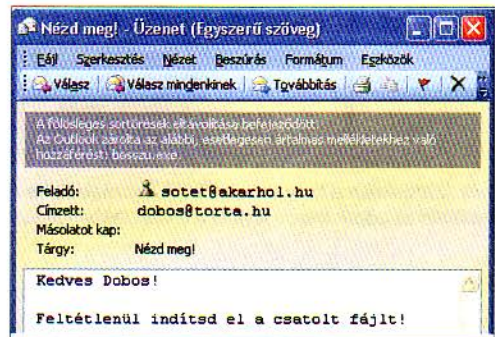
Új levél írása a vipmail [vipmél] ingyenes szolgáltató webes felületén. Milyen előnyei és hátrányai vannak a webes felületnek egy levelező célprogrammal szemben?

Fájl csatolása

Az elektronikus levélhez fájlokat is **csatolhatunk***, így partnerünknek képeket, dokumentumokat is küldhetünk. Maga a csatolás a legtöbb esetben egyszerű: a megfelelő



*Az Outlookban a **Beszúrás** menü **Fájl** menüpontjával csatolhatunk egy fájlt. Milyen korlátokra kell ügyelnünk?*



A károsnak ítélt fájlokat az Outlook automatikusan blokkolja. Milyen előnyei és milyen hátrányai lehetnek ennek?

menüpontra vagy a megfelelő gombra való kattintással ki kell választanunk gépünk fájl-szerkezetéből a csatolandó fájlokat.

Csatolásakor figyelembe kell vennünk, hogy egyrészt a nagyméretű fájlok küldése és fogadása időigényes, másrészt *a postafiókok és a küldhető-fogadható levelek méretét a legtöbb szolgáltató korlátozza*. Ugyancsak gondot okozhat az is, hogy bizonyos fájlokat a levelezőszoftverek a felhasználó érdekében automatikusan blokkolnak, így akkor sem kapják meg, ha a fájl egyébként nem káros.

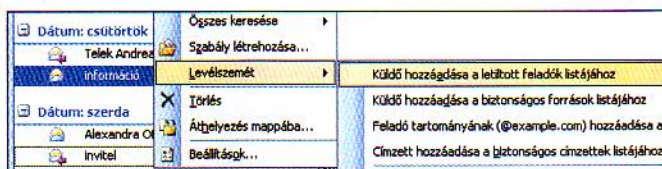
A levelek csoportosítása és szűrése

Leveleinket a levelezőszoftver *mappákban** tárolja, alapértelmezés szerint a levél-forgalom irányától függően. Így külön mappába kerülnek a *beérkezett üzenetek*, az *elküldött levelek*, illetve a *törölt elemek*. Természetesen **mappákat magunk is létrehozhatunk**, hogy a leveleket rendszerezve tároljuk, és a leveleket ezekbe áthelyezhetjük.



Nagyszámú levél esetén célszerű azokat saját mappákban rendszerezni, ilyen például a fontos nevű mappa. (Balra a freemail [frímél], jobbra az Outlook látható.)

Sok levelezőszoftver lehetővé teszi a *levelek csoportosítását* akár egy adott mappában is különböző szempontok szerint. Például: feladó, mikor érkezett stb., így akár nagyszámú levelet is kényelmesen kezelhetünk.



Az Outlookban a helyi menü *Küldő hozzáadása a letiltott feladók listájához* menüpontra kiválasztása esetén az adott feladó levelei a továbbiakban automatikusan a *Levélszemét* mappába kerülnek.

Nagyon hasznos szolgáltatás a levelek szűrése. Egyszerűbb esetben ez a *levélszemét** *automatikus szűrését* jelenti. (A kéretlen reklámlevelek bizonyos formai és tartalmi jellemzők alapján jól beazonosíthatók.)

Általános esetben a *felhasználó megadhatja, hogy adott feltételeket kielégítő leveleket a szoftver hogyan kezeljen*. Például előírhatja, hogy a „Kedvezményes” tárgyú levelek, melyek tartalmazzák a „Csak Önnek és csak most” szöveget, automatikusan a *törölt elemek* mappába kerüljenek.



Szűrő létrehozása

Szűrő opciók elrejtése

Válassza ki a keresési feltételt - Adja meg a szűrési feltételt, ami alapján eldönthető, hogy mit kell csinálni a bejövő üzenettel. Használja a "Próba Keresés" funkciót, hogy láthassa, hogy mely üzenetek lettek volna szűrve ezzel a szűrési feltétellel.

Feladó: Tartalmazza az alábbi szavakat:

Címzett: Amelyben nincs:

Tárgy: A következő melléklettel rendelkezik

Szűrő létrehozása Gmailben (dzsímél vagy géfél).
Milyen leveleket szűrünk ki az ábrán látható beállításokkal?

Az internet további szolgáltatásainak áttekintése

Az internet szolgáltatásait három csoportba osztjuk:

- elektronikus levelezés (e-mail, levelezési lista, hírcsoportok),
- közvetlen kapcsolat (telnet, chat [cset]),
- keresés adatbázisokban (ftp, gopher [gófer], www).

Ezek megvalósítása ma már többnyire böngészőn keresztül történik, de gyakran külön célprogramokat is használhatunk hozzá.

A levelezési lista és a fórum

Gyakori, hogy egy-egy témakörben több ember is szeretne az elektronikus levélhez hasonló módon információt váltani. Ennek elterjedt módja az e-mail alapú levelezési lista és a webes felületen megvalósítható fórum.

A levelezési lista* lényege, hogy a lista tagjai minden, a lista címére küldött levelet automatikusan megkapnak. Ha valaki fel akar tenni egy kérdést (pl. „Hol lehet bam-

Szerző	Téma: ASP oktatás
<p>szlzs Top 25 Poster Tag 05 nov. 2002 óta Össz üzenet: 4</p>	<p>ASP oktatás Küldte: 05 nov. 2002 12:53 du. Mi kell minimálisan ASP oktatásához minden gépre? Mit lehet tenni, ha nincs szerver erre a célra?</p> <p>post reply</p>
<p>MiZo Top 25 Poster moderator Tag 05 nov. 2002 óta Össz üzenet: 46</p>	<p>Re: ASP oktatás Küldte: 05 nov. 2002 12:53 du. ASP-hez minimálisan egy Internet Information Server (IIS) webszerver kell, ez a Windows 2000 Server alaplól felrakja, de Windows 2000 Professional alá is felrakható (Vezérlőpult\Programok hozzáadása\Windows komponensek). Ha adatbázis elérést is szeretnénk nézni, azt lehet egy Access-szel is próbálgatni. Vagy használható az MSDE is, annak teljesen olyan a felülete, mint egy SQL szerveré.</p> <p>Üdv, MiZo</p> <p>post reply</p>

Részlet egy fórumról. Az szlzs nevű felhasználó kérdésére MiZo válaszolt.
A legtöbb fórumra csak regisztrált felhasználók küldhetnek üzeneteket. Miért?
Regisztráljuk be magunkat egy számunkra érdekes fórumra! Milyen lépéseken kell végigmennünk?

buszrügvet kapni?”), akkor azt elküldi a „kínai konyha” lista címére. Innen továbbítják a lista minden tagjának. Aki a kérdésre reagál, az is a lista címére küldi a választ, így a lista minden tagja azt is megkapja. Közös érdeklődési kör esetén gyakori, hogy sokan nem is szólnak hozzá, csak olvasgatják a kérdéseket és a válaszokat.

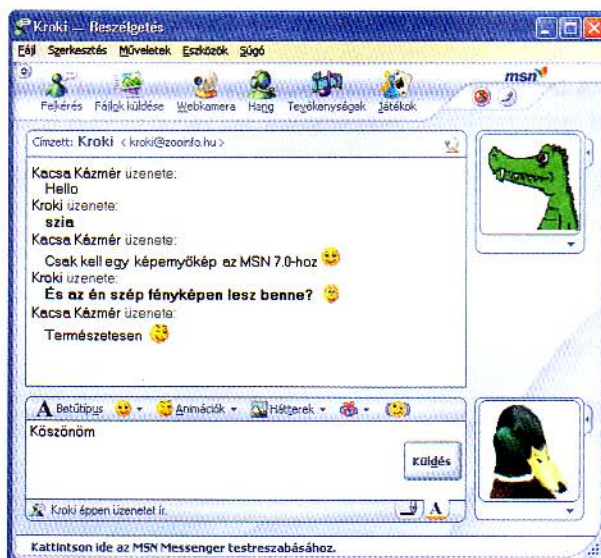
Egy listára feliratkozni („előfizetni”) úgy kell, hogy e-mailt küldünk a lista feliratkozási címére, vagy webes felületen regisztrálunk. Mivel a listák napi forgalma több tucat levél is lehet, érdemes – még feliratkozás előtt – megtudakolni a leiratkozás módját. **A lista zárt, ha csak a lista tagjai írhatnak a listára levelet, és moderált, ha a listára kerülő leveleket elküldés előtt ellenőrzik.** A legtöbb listán reklámcélú levelek küldése tilos.

A fórum* esetén a kérdéseket és hozzászólásokat témánként csoportosítva rendszerezik és egy webszerveren tárolják. A fórumok tartalmát bárki megtekintheti, azonban csak az szólhat hozzá, aki az adott fórumon regisztráltatta magát.

Csevegés (Chat) és azonnali üzenetküldés (Messenger)

A csevegésben részt vevő számítógépek között folyamatos adatforgalom van, melyet egy szervergép vezérel. **A csevegésben* részt vevők által írt üzenetek** így a szerveren keresztül **valamennyi résztvevőhöz** lényegében **azonnal eljutnak.**

Egy régebbi megoldás az *IRC Chat* [cset], ahol az adatforgalmat több, összekapcsolt szerver végzi. A csevegő megjelöli, hogy milyen témában kíván beszélgetni („csatorna”), s a továbbiakban az adott csatornára érkező minden üzenetet megkapja, és válaszolhat is. Ha valamelyik csevegő személye felkelti az érdeklődését, privát csatornát nyithatnak, amelyen keresztül már csak ketten cseveghetnek.



Az MSN Messenger használata. A listán megtekinthetjük partnereink állapotát (a kapcsolatfelvétellel kölcsönös), és a kiválasztott partnerrel beszélgetést kezdhetünk. Természetesen többen is tárgyalhatnak egyszerre. Milyen feltételek szükségesek a Messenger használatához?

A csevegést ma általában webes felületen valósítják meg. Több portál egyik szolgáltatója a felhasználók közötti csevegés biztosítása. Ehhez előbb regisztrálni kell magunkat, s csak az ekkor megadott felhasználói névvel és jelszóval léphetünk be.

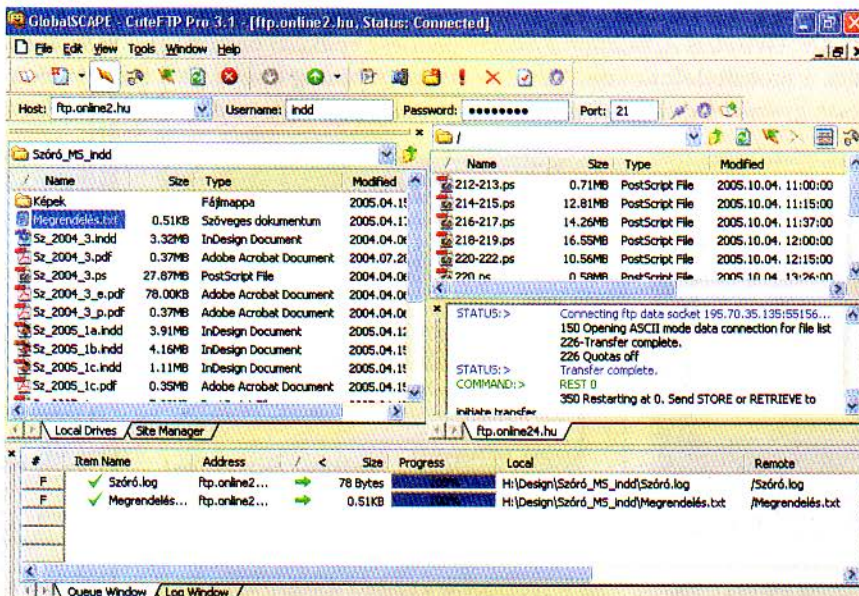
Az **azonnali üzenetküldés*** lehetőségét több operációs rendszer (pl. Windows XP) már operációs rendszer szinten támogatja (Windows Messenger [meszendzser]). Ebben az esetben egy előre megadott, működő e-mail címmel azonosítjuk magunkat. A továbbiakban kölcsönösen „meghívhatjuk” azokat, akik szintén beléptek a szolgáltatásba, ettől kezdve ha valahol belépünk a rendszerbe, értesülnek róla.

A szolgáltatás jelzi az elérhető partnerek listáját, s ha valamelyikükkel kapcsolatba kívánunk lépni, csupán ki kell választanunk az illetőt. A tényleges kommunikáció már külön ablakban zajlik, ekkor fájlokat is küldhetünk egymásnak, webkamerával láthatjuk is egymást, stb. Ugyancsak hasznos, hogy az azonnali üzenetküldés lehetővé teszi a több partnerrel egyidejűleg folytatott telekonferenciát is.

FTP és gopher

Az interneten elhelyezett információkat, fájlokat ma már elsősorban böngészővel, webes felületen érjük el. Az adatfájlok elérésére azonban más módszerek is kialakultak, és ezeket részben még ma is használják.

Az **FTP* (File Transfer Protocol [fáj]l transzfer protokoll)** fájlokat tartalmazó archívumokhoz való hozzáférést, a fájlok le- és feltöltését teszi lehetővé. Gyakran így érhetünk el ingyenes programokat, zene vagy filmfájlokat. Jóllehet a szolgáltatás elérhető parancssorból is, ma már többnyire böngészőn vagy kényelmesebben kezelhető célprogramokon keresztül (pl. CuteFTP [kjut]) használják.



Az FTP-szolgáltatás elérése a fájlkezelőkhöz hasonlóan használható CuteFTP programmal.

Kommunikáció hálózaton

Az FTP-szerverek minden esetben jelszavas hitelesítést igényelnek, de a legtöbb esetben beléphetünk *anonymous* néven is. Ilyenkor jelszóként az e-mail címünket illik megadni.

A *gopher** [gófer] menürendszer segítségével szöveges állományokat tartalmazó adatbázisokhoz való hozzáférést tesz lehetővé. Ilyen pl. a *www.mek.iif.hu* címen elérhető magyar elektronikus könyvtár, de gyakran a levelezési listákat is így archiválják.

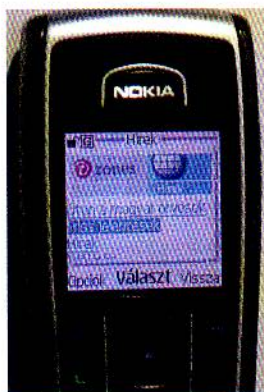
```
ftp> open ftp.microsoft.com
A kapcsolat létrejött - ftp.microsoft.com.
220 Microsoft FTP Service
Felhasználó (ftp.microsoft.com:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Jelszó:
230>Welcome to ftp.microsoft.com.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
dr-xr-xr-x  1 owner  group           0 Nov 25  2002 bussys
dr-xr-xr-x  1 owner  group           0 May 21  2001 deskapps
dr-xr-xr-x  1 owner  group           0 Apr 20  2001 developr
dr-xr-xr-x  1 owner  group           0 Nov 18  2002 KBHelp
```

Az FTP szolgáltatás eléréséhez a parancssorba az ftp parancsot kell beírni. A felhasználó által beírt parancsokat pirossal kiemeltük. Próbáljuk ki a fenti szolgáltatást!

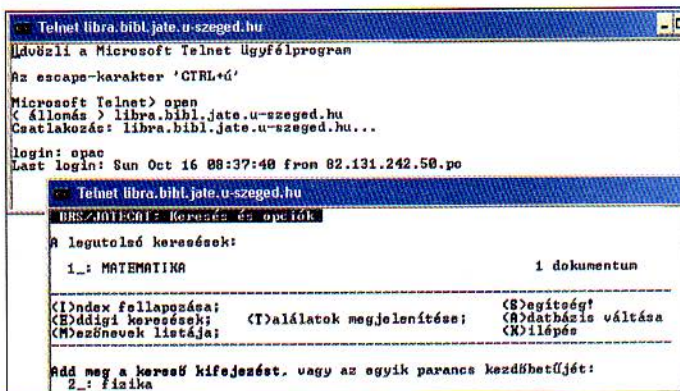
További internetes szolgáltatások

A *telnet** szolgáltatás segítségével beléphetünk egy távoli számítógépre, és azon dolgozhatunk. Természetesen ehhez megfelelő hitelesítés szükséges, és a jogosultságaink is meglehetősen korlátozottak. Korábban igen széles körben használták nagy könyvtárak katalógusában való keresésre, de a webes felület elterjedésével jelentősége csökkent.

A *WAP** (Wireless Application Protocol [vájrlész aplikésn protokoll]) a **WWW megvalósítása a mobiltelefonokon**. Mivel az átviteli sebesség kisebb, és a mobiltelefon kijelzője csak gyengébb minőségű grafikát képes megjeleníteni, ezért a WAP-lapok kevesebb



Az internet elérése mobiltelefonon keresztül WAP segítségével történik.



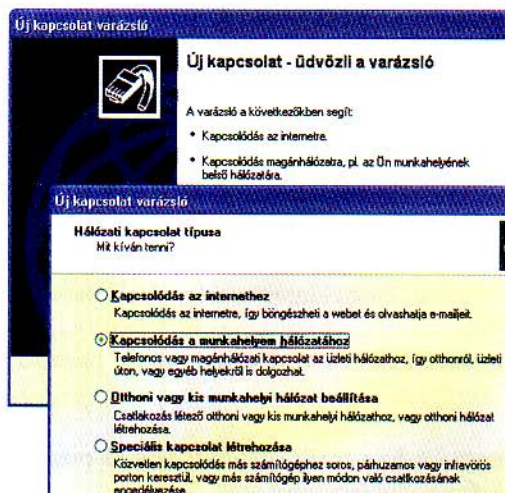
Telnet segítségével a szegedi egyetem könyvtárának katalógusában keresünk régi matematika- és fizikakönyveket. Indításhoz a parancssorba be kell írni a telnet utasítást (címsor). Próbáljuk ki a fenti szolgáltatást!

multimédiás anyagot tartalmaznak és lényegre törőek. A WAP, a mobiltelefonok újabb generációjának megjelenésével, a GPRS [dzsípiáresz] kapcsolat megvalósításával napjainkban kezd terjedni.

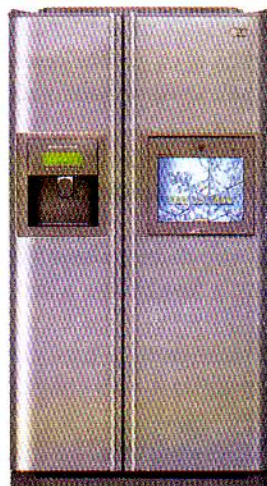
Az internetes technológia lehetővé teszi cégek, intézmények belső információáramlásának egyszerűen kezelhető, gyors és áttekinthető megvalósítását. Ez a felismerés vezetett a virtuális magánhálózatok (VPN), illetve az intranetes hálózatok kialakításához.

Az intranet* egy cég vagy intézmény belső hálózata, amely teljes egészében az internetes megoldásokat használja. Mivel zárt, ezért biztonságossá tehető, de mivel használata az internet szolgáltatásain nyugszik, könnyen kezelhető.

A **VPN* (Virtual Private Network [virtuál prájvit netvörk]) esetében a cég vagy intézmény ténylegesen is az interneten át valósítja meg, hogy egyes „külső” gépek kapcsolódjanak a belső hálózathoz.** A külső felhasználó egy már meglévő internetes hozzáféréssel keresztül (amiről neki kell gondoskodnia) hoz létre egy védett kapcsolatot cége szerveivel. Ezt követően már úgy tud dolgozni, mintha az intézmény belső hálózatán, egy számítógép előtt ülne. Így például eléri saját dokumentumait, a belső nyomtatókat stb.



A VPN kapcsolat létrehozását az operációs rendszerek is támogatják (Windows XP).



Milyen plusz szolgáltatásokat nyújt az internetes hűtőszekrény?

Ma még inkább érdekesség, hogy az internet kezd megjelenni a háztartási gépekben is. Az egyik első ilyen eszköz az *internetes hűtőszekrény* volt, amely a számítógép és a hűtőszekrény kombinációja: az ajtón egy érintőképernyőt tartalmaz. Nyilvántartja és az interneten át elérhetővé teszi a benne lévő élelmiszerek listáját, valamint megrendelhetjük vele az utánpótlást is. Elkészült már az *internetes mosógép* is, amely felkapcsolódik a netre, és letölti az adott ruhafajtához leginkább megfelelő mosási programot. Az *internetes mikrohullámú sütő* érintőképernyőjével lehetővé teszi a böngészést, pl. receptek letöltését.

Összefoglaló kérdések, feladatok

1. Hasonlítsuk össze a PAN-, LAN-, MAN- és WAN-hálózatokat kiterjedtségük, felhasználási területük és megvalósításuk módja alapján!
2. Milyen internet-hozzáférési lehetőségeink vannak lakóhelyünk környékén? Derítsük ki ezek technikai megvalósításának módját, a felkínált sáv szélességet és a költségeket! Hasonlítsuk össze az adatokat! Mit tapasztalunk?
3. Hasonlítsuk össze az internet működését a nyilvános telefonhálózat működésével! Milyen eltéréseket tudunk mondani a vonalak foglaltságára és a hálózat biztonságára vonatkozólag?
4. Mi iskolánk internetcíme? Mit jelentenek a cím egyes részei?
5. Keressük meg iskolánkat tematikus kereső és kulcsszavas kereső segítségével is! Végezzük el ez utóbbi keresést úgy is, hogy az iskola nevét és címét nem adjuk meg, hanem csak az iskola jellemzőire keresünk rá!
6. Melyek az Európai Unió főbb szervezetei? Keressük meg ezek honlapját az interneten! Törekedjünk magyar nyelvű információk gyűjtésére!
7. Soroljuk fel, milyen lehetőségeket kínál fel az általunk használt böngészőprogram a biztonság fokozására, a keresésre és a korábban megtekintett weblapok elérésére!
8. Értelmezzük e-mail címünket: mit jelentenek benne az egyes részek? Soroljunk fel néhány hibás e-mail címet is!
9. Milyen adatokat kell, illetve lehet megadni, ha e-mailt küldünk? Adjunk meg néhány jellemzőt, ami valószínűleg kéréslen reklámlevélre (spamre) utal!
10. Milyen szolgáltatásokat kínál fel az általunk használt levelezőprogram az alapvető lehetőségeken túl? Milyen esetekben lehet ezeket jól kihasználni?
11. Hogyan tudunk futtatható állományt elküldeni partnerünknek úgy, hogy a levelezőprogram biztosan ne blokkolja?
12. Hasonlítsuk össze az elektronikus kapcsolattartás következő lehetőségeit: SMS, e-mail, fórum, azonnali üzenetküldés, levelezési lista, MMS! Mikor célszerű ezeket használni, és ez milyen költségekkel jár?
13. Mit jelentenek a következő fogalmak: FTP, telnet, WAP, gopher, VPN, intranet? Melyik érhető el böngészőn keresztül? Mi elterjedésük vagy visszaszorulásuk oka?

Dokumentumkészítés számítógéppel

Grafika

Grafikus adatok kezelése

Pixel- és vektorgrafika

A grafikus adatok kezelésére alapvetően kétféle módszer alakult ki.

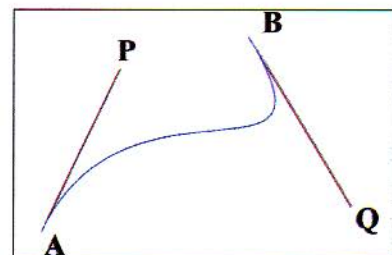
A **pixelgrafika*** (*rasztergrafika vagy bittérkép*) a képeket képpontokként kezeli és tárolja. A képpont jellemző adata a helye és a színe, a képpontok mérete azonos. A pixelgrafikus ábrák téglalap alakúak, és a háttértáron általában nagy helyet foglalnak el. **Átméretezéskor a kép minősége romlik**, viszont a képek megjelenítéséhez nem szükséges nagy matematikai teljesítmény.

A **vektorgrafika*** az ábrákat vonalakra bontja, és ezeket egymáshoz viszonyítva matematikai úton írja le. (Szakaszok esetében például az egymáshoz viszonyított hosszukkal és szögükkel, görbék esetében pedig a görbületek irányával és mértékével.) **Az ábrák minősége átméretezéskor nem változik**, ugyanakkor a megjelenítéshez komoly számolási teljesítmény szükséges.

A kétféle módszer alkalmazási területe eltérő. A mérnöki gyakorlatban szinte kizárólag a vektorgrafikát használják, hiszen ez teszi lehetővé a műszaki rajzok matematikai pontosságú elkészítését és átméretezését. A betűk alakját is érdemes vektorgrafikusan tárolni, hiszen azok méretét is nagy mértékben módosítjuk. Ugyanakkor a fényképek feldolgozására lényegében csak a pixelgrafika használható, mivel azok általában nem bonthatók matematikai görbékre.



Ha a pixelgrafikus képet nagyítjuk, akkor a kép minősége romlik, míg a vektorgrafikus kép minősége nem változik. A vektorgrafikus képek megjelenítése ugyanakkor számolásgényes.



A gyakran használt Beziér-görbét négy pont írja le: az A és B végpontok, valamint a görbület irányát és nagyságát megadó P és Q vezérpontok.

Színkezelés

A fizika törvényei szerint a színek három adattal írhatók le. Attól függően, hogy milyen célból kell ezt a három adatot megadnunk, különböző leírási módszereket használnak.

Az **RGB-színrendszer***: a vörös (*Red*), a zöld (*Green* [grín]) és a kék (*Blue* [blú]) fény segítségével a szivárvány minden színe előállítható, ha a három alapszín megfelelő arányban keverjük. Ezt a módszert használja a monitor, a tévékészülék, a projektor, tehát azok az eszközök, amelyek a színeket közvetlen fénykibocsátással állítják elő.

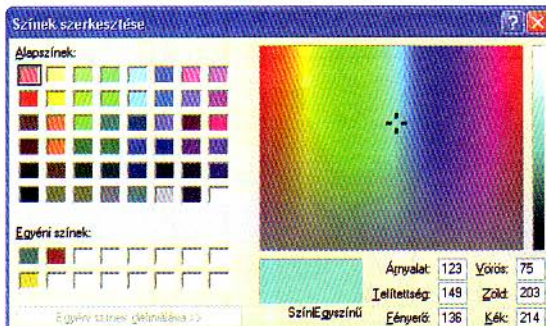
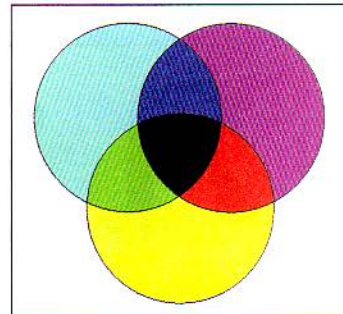
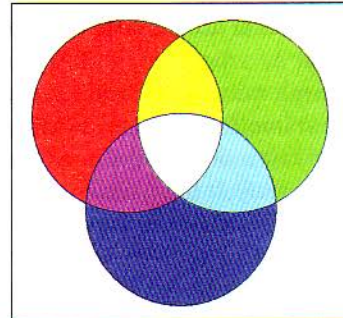
Az RGB-színek keverésekor például vörös + zöld = sárga; zöld + kék = kékeszöld (cián); kék + vörös = bíbor; vörös + zöld + kék = fehér.

A **CMYK-színrendszert*** a festékeknel használják. A festékek nem kibocsájtják a fényt, hanem a rájuk eső fény egy részét visszaverik, míg más részét elnyelik. Így más szabályok szerint keverednek, tehát mások az alapszínek is: a kékeszöld (*Cyan* [cián]), a bíbor (*Magenta* [madzsenta]) és a sárga (*Yellow* [jelou]). Ezekből nem lehet tökéletes feketét kikeverni, ezért **negyedik alapszínként a feketét is hozzá kell venni** (kulcsszín, *Key* [kí]).

CMYK-keveréskor például: sárga + bíbor = vörös; bíbor + kékeszöld = kék; kékeszöld + sárga = zöld; sárga + bíbor + kékeszöld = (elvileg) fekete.

A **HSB-színrendszer** a fentiekől eltérően nem valamely eszközhöz, hanem inkább az ember színérzékeléséhez kapcsolódik. A három adat ebben az esetben az árnyalat (*Hue* [hjú]), a telítettség (*Saturation* [szecsörésn]) és a fényerő (*Brightness* [brájtnezsz]).

Az **árnyalat** adja meg, hogy melyik színről van szó (pl. egy ház vakolata zöld). A **telítettség** azt mutatja meg, hogy mennyire tiszta az adott szín, azaz milyen messze van az adott fényerejű szürkétől. Végül a **fényerő** a szemünkbe érkező fény mennyiségét méri. Ugyanazt a szint, pl. egy ház vakolatának zöldjét, másnak látjuk napsütésben, mint borús időben.



A színeket a Paintben akár az RGB, akár a HSB-színrendszer szerint is megadhatjuk. Adjuk meg a zöldes-kék, a sárga, és a bíbor szín szinkordinátáit!

Felül az RGB-, alul a CMYK-színkeverés színekörét látjuk. Mi a kapcsolatot a két rendszer alapszínei között?

Képfarmátumok

A leggyakoribb képfarmátumok, amelyekkel találkozhatunk, a következők:

BMP: a Microsoft Paint [májkroszoft péjnt] által használt pixelgrafikus szabvány. Könnyen kezelhető, tömörítetlen, nagyméretű fájlok, melyeket minden Windows-alkalmazás ismer. A színek száma kiválasztható (16, 256, true color [trú kálör] vagy fekete-fehér).

GIF: a CompuServe [kompjuszörv] által kifejlesztett pixelgrafikus szabvány. Mivel csak 256 elemű színpalettát használ és az adatokat tömöríti, ezért a fájlok mérete kicsi. Tartalmazhat a kép átlátszóságára vonatkozó információt is, és képes egy fájlban akár több képet elhelyezni, így mozgóképek tárolására is alkalmas (animált GIF).

PNG: a GIF utódjaként emlegetett formátum, amely hatékonyabb tömörítést tartalmaz, elsősorban hálózati környezetre tervezték.

JPEG: az ISO [izo] által nemzetközi szabványként elfogadott eljárás, 1982–87 között egy nemzetközi bizottság dolgozta ki. A beépített tömörítési eljárás veszteséges, azaz nagy tömörítési arány esetén a kép minősége romlik. A JPG típusú kép létrehozásakor a tömörítési arány többnyire megadható. A filmeknél használt MPEG kódolásnál is a JPEG formátumból indultak ki.

TIFF: főleg a nyomdai előkészítésnél használt pixelgrafikus formátum, melyet az alkalmazások közötti hordozhatóságra terveztek.

WMF: a Windowsban pl. a ClipArt [klipárt] képeinek leírására használt vektorgrafikus szabvány.

PDF: gyakran mint vektorgrafikus szabványt emlegetik, de ténylegesen a formázott szöveget és grafikát egyaránt tartalmazó dokumentumok platformfüggetlen megosztására vezették be. A PDF formátumú dokumentum gyakorlatilag minden számítógépen megtekinthető az ingyenes Acrobat Reader [akrobat rider] segítségével. További előnye, hogy módosítása és nyomtatása szükség esetén letiltható.

Rajzok, ábrák készítése

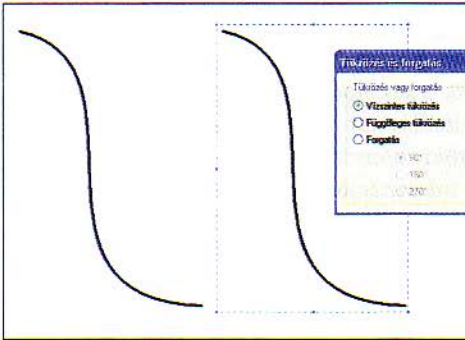
A legegyszerűbb rajzolóprogramok is igen sok kész elemet kínálnak, melyekből ábrákat elkészíthetjük. Ilyenek a szakaszok, az egymáshoz csatlakozó szakaszok, a körök, az ellipszisek, a téglalapok, a lekerekített sarkú téglalapok stb.

A vektorgrafikus programok több kész elemet kínálnak fel (nyilak, csillagok, speciális műszaki elemek), míg a pixelgrafikus programok a szabadkézi rajzolásra való eszközökben (különbféle ecsetek, tollak, radírok) adnak több lehetőséget. Ennek ellenére a legegyszerűbb pixelgrafikus programok (pl. a Paint) is felkínálják a Beziér-görbét [bézié], és a legegyszerűbb vektorgrafikus programok (pl. a Microsoft Office [májkroszoft ofisz] alkalmazások rajz eszköztára) is tartalmazzák a szabadkézi rajz lehetőségét.

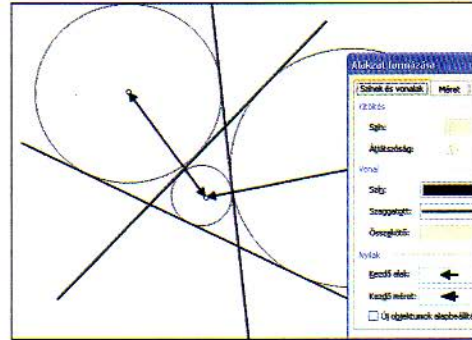
A kétféle program között az eltérés a grafikus adatok tárolásában és a későbbi módosítás lehetőségében van, vagyis kiválasztásukkor ezeknek kell elsődleges szempontoknak lenniük.

A pixelgrafikus programok* az elhelyezett objektumokat, így a görbéket is, képpontokká alakítják. Elhelyezésük után a görbe tulajdonságait (pl. görbületének iránya,

mértéke) már nem tudjuk elérni vagy módosítani. Ugyanakkor **széles körű lehetőséget adnak képpontok kijelölésére** (adott alakú terület, adott színű pontok, kontúrokkal határolt területek stb.) és azok **törlésére, átszínezésére stb.**



Virág készítése Painttel. Görbét rajzolunk, a görbét tartalmazó téglalapot megduplázzuk, majd a másolatot tükrözzük. A görbe tulajdonságai utólag már nem módosíthatók.



Geometriai ábra készítése MS Office rajz eszköztárával. A nyíl utólag is kijelölhető, és tulajdonságai módosíthatók. A hegyét azonban nem tudjuk radírral törölni.

A **vektorgrafikus programok*** az alakzatokat görbéként tárolják, még a **szabadkézi rajzot is görbévé alakítják**. A görbék a későbbiekben is kiválaszthatók, tulajdonságaik módosíthatók. **Lehetőségünk van különböző görbék csoportba foglalására** (és így egy alakzatként való kezelésére), **az egymást takaró elemek előre vagy hátra küldésére**. Ugyanakkor nem tudjuk a görbék egyes részleteit radírral törölni vagy ecsettel kijavítani.

A professzionális grafikus programok (pl. Adobe PhotoShop [adobi fotosop]) egyik hasznos eszköze a **réteg**. A rétegeket **legegyszerűbben egymásra helyezett fóliákként képzelhetjük el**, melyek az ábra elemeit vagy azok korrekcióit külön-külön tartalmazzák. A grafikus ezeket **a rétegeket egyenként módosíthatja**, kikapcsolhatja, bekapcsolhatja, törölheti, illetve új rétegeket vehet fel, ezáltal igen nagy szabadsági fokot kap. (A rétegeket nem minden grafikus formátum tárolja, így pl. a BMP vagy a GIF formátum sem.)



Rétegek a PhotoShopban. Az „egér” külön rétegen szerepel, így egy mozdulattal eltüntethető, illetve a kép többi részétől függetlenül szerkeszthető.

Digitális képek kezelése

A digitális képek készítésének és megjelenítésének főbb eszközei

A digitális fényképezőgépek elterjedésével a fényképek elkészítése sok esetben az otthoni számítógépek és a színes nyomtatók világába került.

A megjelenítő eszközök jellemzője a **felbontás***, ami **megmutatja, hogy egységnyi hosszúságú** (általában 1 inch, azaz 2,54 cm) **szakaszon hány képpontot jelenít meg az eszköz. Mértékegysége a dpi (dot per inch).**

A **képernyő** (beleértve a projektort is) a képpontokat az RGB-színkeverés szabályai szerint, a vörös, zöld és kék színekből keveri ki. A képpontok legkisebb távolságát (különösen a TFT-monitoroknál) a hardver feltételei általában meghatározzák. **Monitorok esetében** a képek 72-96 dpi felbontásban jelennek meg. Mivel ez az érték a monitorokon bizonyos határok között állítható, ezért inkább a **képpontok számát adják meg: pl. 800×600, 1024×768 vagy 1280×1024.**

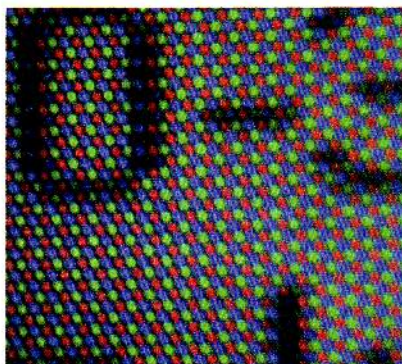
Ugyanaz a kép különböző monitorokon teljesen eltérő színeket mutathat, mivel a monitorok beállítása, a helyiség megvilágítása, illetve a szín megjelenítését szolgáló anyagok eltérőek lehetnek. Egy adott monitor színvisszaadása módosul az anyag öregedése során is. Ezért a professzionális célokra használt eszközök *színhelyes beállításához különböző kalibrációs módszereket dolgoztak ki.*

A képeket többnyire *papírra* nyomtatjuk. A nyomtatáskor használt festékek a CMYK-színkeverést használják, tehát az RGB-színkeverés szabályai szerint tárolt képet konvertálni kell. Sajnos a *CMYK-színek száma kisebb az RGB-színek számánál*, ezért ilyenkor színvesztés is előfordulhat.

Figyelembe véve az emberi szem felbontóképességét, **egy képet akkor látunk jó minőségűnek, ha a felbontás nyomtatáskor kb. 300 dpi.** Mivel egy képeslap kb. 10×15 cm, azaz kb. 4×6 inch méretű, ez azt jelenti, hogy egy képeslap méretű kép előállításához kb. $1200 \times 1800 = 2\,160\,000$, vagyis kb. 2,16 millió képpont szükséges.

A **digitális fényképezőgépek*** egyik legjellemzőbb paramétere, **hogy hány millió képpontot rögzítenek**, azaz hány „megapixel” (1 MP = egy millió képpont). Mint láttuk, egy levelezőlap méretű képhez kb. 2,16 MP szükséges.

Az **optikai zoom** [zúm] azt mutatja meg, **hogy a lencserendszer hányszoros nagyítást tesz lehetővé.** Ezáltal a nehezen megközelíthető tárgyakat „közelebbre hozhatjuk”, vagy kisméretű tárgyról is készíthetünk felvételt. Fontos tudnunk, hogy *a digitális zoom esetében* a hiányzó képpontokat a meglévőkből a fényképezőgép matematikai úton számolja ki, *a kép ténylegesen nem tartalmaz több részletet.*



Az RGB-színrendszernek megfelelő képpontok a monitoron

A kép vágása és forgatása

Bár a digitális fényképezőgéppel készült képeket a legtöbb esetben akár magáról a gépről közvetlenül megjeleníthetjük tévéképernyőn vagy nyomtatón, gyakran szükséges a képek bizonyos mértékű módosítása. A **leggyakoribb korrekciós* műveletek a kép vágása, forgatása, illetve a kép világosságának és kontrasztjának beállítása**. Ehhez igen sok egyszerű program érhető el, pl. az MS Office részét képező Picture Manager [pikcsör menedzser]. Igen sok freeware [fríver] vagy shareware [server] program is rendelkezésünkre áll, mint pl. az IrfanView [irfanvjú] vagy az ACDSee [eiszídíszi]. A professzionális programok, pl. az Adobe PhotoShop vagy a Corel PhotoPaint [korel fotopéint] ugyan sokkal többet tudnak, elsajátításuk azonban nehezebb.

A **kép vágása** során figyelembe kell venni, hogy a kapott részkép felbontása és méretarányai megfelelnek-e a megjelenítő eszköz paramétereinek. Ez vágáskor a legtöbb képfeldolgozó programban külön is beállítható.

Gyakori művelet a **kép forgatása** is: a ferdén fényképezett képet vágás előtt vissza kell forgatni, hogy a függőleges vonalak tényleg függőlegesen álljanak. Erre főleg képek szkennelése esetén lehet szükségünk.



A kép forgatása és vágása a Microsoft Office programcsomag részét képező Picture Manager munkaablakának megfelelő menüpontjaival. Vágáskor beállítható a részkép oldalainak aránya is.

A fényerő és a kontraszt

A **fényerő*** azt adja meg, hogy a kép adott részéről mennyi fény jut a szemünkbe. Ha egy sötétre sikerült kép fényerejét növeljük, akkor a képfeldolgozó program általában **fehér színt ad a képhez**, így az világosabb lesz.

A **kontraszt*** a **legsötétebb és legvilágosabb rész különbsége** (viszonyítva a legvilágosabb részhez). Ha a kép kontrasztját növeljük, akkor a program **az árnyalatokat eltávolítja egymástól**, így a kép részletei jobban kiemelkednek.

Természetesen egyik esetben sem kerülnek új részletek a képhez, csupán a meglévő képpontok színe módosul.



Fényerő automatikus kijáratása

A kijelölt képek fényerejének automatikus beállításához kattintson a Fényerő kijáratása gombra.

Fényerő kijáratása

Fényerő- és kontrasztbeállítások

Fényerő:

40

Kontraszt:

40

Speciális beállítások

Egyebek

Középtónus: csak a középtónusok kijáratása

0

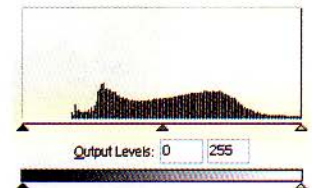
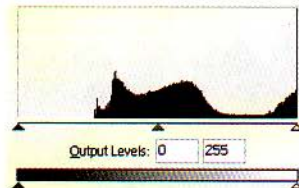
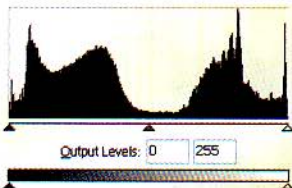
Egy sötét kép fényerejének és kontrasztjának növelése a Picture Manager programban

Érdeemes egy fekete-fehér kép módosításán végignézni, hogy pontosan mi is történik a fényerő, illetve a kontraszt módosítása során.

A képpontok szürkefekozatonkénti megoszlását a *hisztogram* mutatja. A hisztogram lényegében egy oszlopdiaagram. Az oszlopok magassága bal oldalon a legsötétebb, jobb oldalon pedig a legvilágosabb pontok számával arányos.

Az ábráról leolvasható, hogy amikor a képet *világosítjuk*, a sötét árnyalatok elmozdulnak a fehér pont felé, eközben a legvilágosabb árnyalatok eltűnnek. A *kontraszt növelésénél* pedig a program a megmaradt árnyalatokat távolítja el egymástól, így a hisztogramon csíkok jelennek meg. (Ezekhez a fokozatokhoz nem tartozik képpont.)

A kapott képet ugyan jobbnak látjuk, ténylegesen a képen lévő árnyalatok száma, így a kép részletgazdagsága csökkent. A képet tehát akkor tudjuk utólag jól korrigálni, ha több színt rögzítünk, mint amennyit a szemünk meg tud különböztetni.

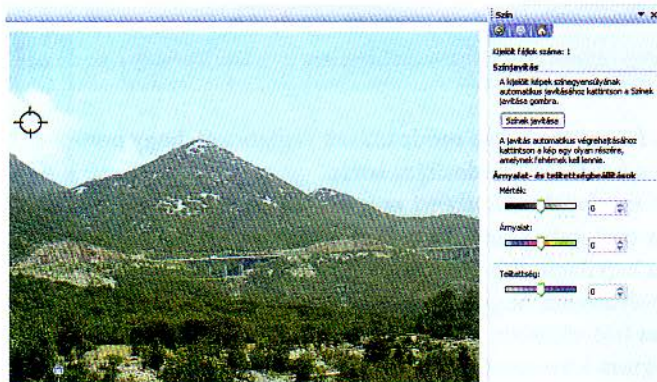


Egy fekete-fehér kép fényerejének és kontrasztjának növelése. Bal oldalon az eredeti képet látjuk alatta a képpontok eloszlását szemléltető hisztogrammal. Középen a fényerő növelése, majd jobb oldalon a kontraszt növelése utáni állapot. Jól látható, hogy az árnyalatok száma csökkent. (A képjavítás és a hisztogramok a PhotoShop programmal készültek.)

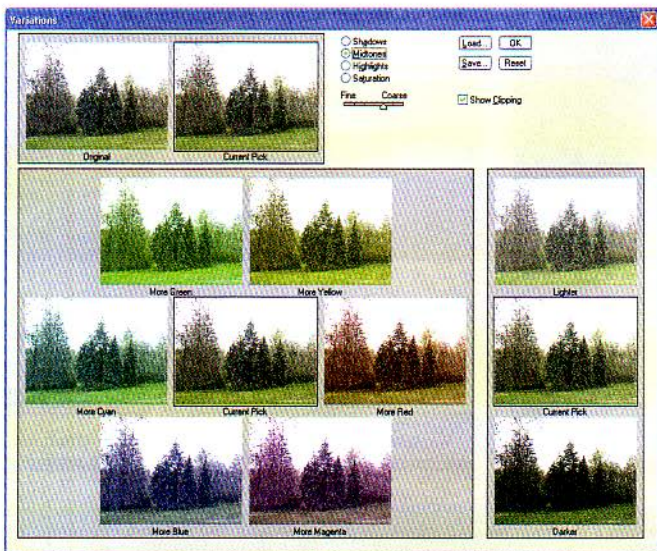
Színkorrekció

A tárgyak színét a megvilágítás is befolyásolja, pl. a neonfényvel megvilágított üzletben vásárolt ruhát más színűnek látjuk az utcán. A *lámpafény*nél készített képek inkább sárgás árnyalatúak, míg a hegyekben készített képek kékesebbek. A digitális fényképezőgépeken ez gyakran már a kép készítésekor megadható, így a szint automatikusan korrigálják.

A hibás színek utólagos korrekciója nem egyszerű feladat, ezért a képfeldolgozó programok beépített „varázslókkal” segítik az amatőr fotósok munkáját.



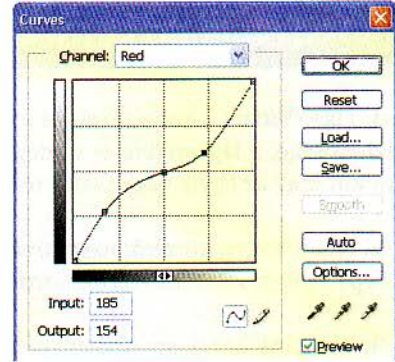
Automatikus színkorrekció Picture Managerrel.
Elegendő kiválasztani egy fehér területet.



A kezdő felhasználó munkáját a professzionális képfeldolgozó programok is varázslókkal segítik (PhotoShop).

A professzionális képfeldolgozó programok lehetővé teszik, hogy a képek színkomponenseit a grafikus külön-külön is módosítsa. Az egyes komponenseket *színcsatornáknak* nevezik, például egy RGB-kép esetében a három színcsatorna a vörös, a zöld, illetve a kék. (Az ún. *alfa csatornáknak* a kép átlátszóságát adják meg.)

A korrekció egyik módja, hogy görbék segítségével megadjuk: milyen bemeneti értékekhez milyen kimeneti értékek tartozzanak az egyes csatornák esetén. Ehhez természetesen nagy gyakorlat szükséges.



A vörös színcsatorna módosítása: a 185 fényerejű legyen csak 154 (PhotoShop).

A „vörösszem-effektus”

Az ember szemébe villanó vaku fénye sok esetben nem a szivárványhártyáról, hanem a szem belsejében lévő, vörös színű retináról verődik vissza. A kapott vörös szemet gyakran tovább rontja a szemlencse fénytörése.

A vörösszem-effektus eltüntetésére a legtöbb képfeldolgozó program varázslókat kínál. Fontos azonban tudnunk, hogy az eredmény többnyire szürke szem lesz, mivel az eredeti szín hiányzik a képről.



A vörösszem-effektus eltávolítása Picture Managerrel



Kép tömörítése webes formátumúra a Picture Managerben

Képek konvertálása

Az egyes képfarmátumok felhasználási területei eltérőek, ezért gyakran szükségünk lehet a képek más formátumúra történő konvertálására. Ezt általában a *Mentés másként* (vagy *Save As* [széjv ez]) menüponttal tehetjük meg.

A képszerkesztő programok külön támogatást biztosítanak arra az esetre, ha egy képet az interneten szeretnénk elhelyezni. Ebben az esetben ugyanis mind a képpontok, mind a színek számát érdemes lecsökkenteni, hogy kisebb fájlméretet kapjunk. A kapott minőségromlás a monitor alacsony felbontása miatt böngészéskor nem okoz problémát, de papírra már nem lehet megfelelő minőségben kinyomtatni a képet.

Összefoglaló kérdések, feladatok

1. Hasonlítsuk össze a pixel- és a vektorgrafikát az adatok tárolásának módja, átméretezhetősége, a fájl mérete és a megjelenítés eszközigénye alapján! Mondjunk példákat, amikor csak az egyik vagy csak a másik használható!

2. Nézzük meg közről nagyítóval a tévéképernyőt vagy a monitort! Hogyan alakul ki a kép? Milyen színrendszert használnak ebben az esetben?

3. Menjünk közel egy óriásplakáthoz, és nézzük meg, hogyan alakul ki a kép! Milyen színrendszert használtak a készítésénél? Hogyan lehet ezt megállapítani?

4. Mit jelentenek a HSB-színrendszer esetén a következő fogalmak: árnyalat, fényerő, telítettség?

5. Soroljuk fel a leggyakoribb képformátumokat, és jellemezzük őket röviden! Melyiket hol használják, és miért?

6. Készítsük el Painttel a következő közlekedési jelzőtáblákat: stop tábla, főútvonal, zsákutca, behajtani tilos!

7. Rajzoljuk meg az Európai Unió zászlóját Painttel, illetve a Word rajzeszköztárával is!

8. Mely paramétereket szokták megadni a digitális fényképezőgépeknél? Mit jelentenek ezek? Melyiket érdemes vásárlásnál figyelembe venni?

9. Gyakori, hogy ugyanaz a kép máshogy néz ki a képernyőn és a papíron. Miért? Hogyan lehetne ezt a különbséget csökkenteni?

10. Hogyan lehet az általunk használt képfeldolgozó programmal a képet vágni, forogtatni, nyomtatni, illetve a képet más formátumban menteni?

11. Mit jelentenek a következő fogalmak: fényerő és kontraszt? Mi történik, amikor a kép fényerejét, illetve kontrasztját növeljük?

12. Lámpafénynél és nappali megvilágításban készült képeket teszünk egymás mellé. Mit tapasztalunk? Hogyan oldható meg a probléma?

13. Mi az a „vörösszem-effektus”, és hogyan javítható?

14. Napjainkban a képfeldolgozó programokkal megoldható, hogy egy kétgyermekes családot ábrázoló nyári képből háromgyermekes családot ábrázoló őszi kép készüljön. Mi a véleményünk ennek a ténynek a jogi és emberi vonatkozásairól?

Dokumentumkészítés számítógéppel

Szövegszerkesztés

A szövegszerkesztő programok jellegzetességei

A szövegszerkesztők előtt

Az ismeretek írásba foglalása, így a szövegszerkesztés igénye is, egyidős az emberiséggel. Kezdetben a *képirást* használták, amikor a vázlatos rajzok eseményeket örökítettek meg. Ilyenek például az észak-amerikai indiánok feljegyzései. Ezt fokozatosan váltották fel az egyes fogalmakat visszaadó *ideogrammak* (pl. kínai írásjelek), illetve az európai kultúrában ma is használt *hangírás*, amely betűkkel rögzíti az egyes hangokat.

Föníciai	Modern héber	Korai görög	Klasszikus görög	Etruszk	Klasszikus latin	Modern latin
𐤀	א	Α	Α	Α	A	A
𐤁	ב	Β	Β	Β	B	B
𐤂	ג	Γ	Γ	Γ	C	C
𐤃	ד	Δ	Δ	Δ	D	D
𐤄	ה	Ε	Ε	Ε	E	E



Néhány ma használt latin nagybetű kialakulása. A római kőtáblákon használt nagybetűk már a mai nagybetűinknek felelnek meg. Hogyan változott az „A” betű az idők során?

Az írás rögzítésére használt írószerek igen változatosak voltak. A régi magyarok által is használt *rovásírásnál* a jeleket fába (pl. botokba) vésték. A babiloniak fapálcikáikkal puha agyagtáblákba nyomták *ék alakú* jegyeiket, az agyagtáblákat pedig kiegészítették. A rómaiak viasszal bevont falapokra írtak a *stilus* nevű írószerszámmal.

A hellenizmus korától kezdve egyre inkább elterjedt a vágott hegyű toll, mellyel kikészített állati bőrre, *pergamenre* írtak. Később a pergament felváltotta a *papír*. *Töltőtollal* a XVIII. sz. óta használnak, s bár a golyóstollal már a XIX. században kísérleteztek, ténylegesen csak a XX. sz. második felében terjedt el.

A kézzel írt középkori *kódexek* lassan készültek. A folyamatot először *fametszetek* készítésével próbálták felgyorsítani: egy falap felületéből kifaragták az oldal tükörképét, és ezzel „pecsételtek”. Az áttörést JOHANNES GUTENBERG érte el a XV. sz. elején azzal, hogy szétszedhető és újra összerakható betűkből készült nyomóformát alkalmazott.

Szövegszerkesztés

A hivatalos dokumentumok készítéséhez a *nyomdagép* túlságosan termelékeny, hiszen itt csupán egy-két példány előállítására van szükség. Ehhez az első *írógépet* a XVIII. sz. elején készítették, de végleges formáját a XIX. sz. második felében érte el.

etiam atq; etiam rogandus est: ut eum in pace tranquilla: iusticie
obseruatione: suorum dilectione: regni incremento: & diuturna
demū uite incolumitate tenere: seruare: & augere dignetur.

Finita Bude Anno dñi. M. CCCC. LXXIII
in uigilia penthecostes: per Andream Hess

Részlet a legrégebb magyar nyomtatványból. A Mátyás király udvarában élt Hess András nyomdász által Budán kiadott *Chronica Hungarorum* [kronika hungarorum] zárószorai. Melyik évben is készült?

A szövegszerkesztők fogalma

A *szövegszerkesztő programok** lehetővé teszik a szöveg beírását, megformázását, későbbi felhasználásra való elmentését, illetve kinyomtatását. A szövegszerkesztő programok kimenete tehát általában papíron jelenik meg, többnyire mindössze pár példányban.

A *szövegszerkesztő programokhoz hasonló szövegkezelő funkciókkal rendelkeznek a kiadványszerkesztő, a webszerkesztő és a prezentációkészítő programok is*. A négy program közötti eltérések a kimenet sajátosságaiból következnek. A kiadványszerkesztő programok esetében ugyanis ez nyomdai eszközökkel papírra történik, igen nagy példányszámban. A webszerkesztő és a prezentációkészítő esetében pedig elektronikus eszközön, pl. monitoron vagy kivetítőn jelenik meg a dokumentum.

A szövegszerkesztő programokat meg kell különböztetnünk az *editoroktól* is. **Az editor programok formázatlan szövegek kezelésére készültek**. Ilyen például az operációs rendszereknél tárgyalt Jegyzettömb is.

A **mai szövegszerkesztők WYSIWYG*** [vizivig] (*What you see is, what you get* [vát jú szi iz vát jú get], vagy magyarul AMILÁSZKA – *Amit látsz, azt kapod*) **rendszerűek, tehát a képernyőn látott elrendezés megfelel a nyomtatási képeknek**.



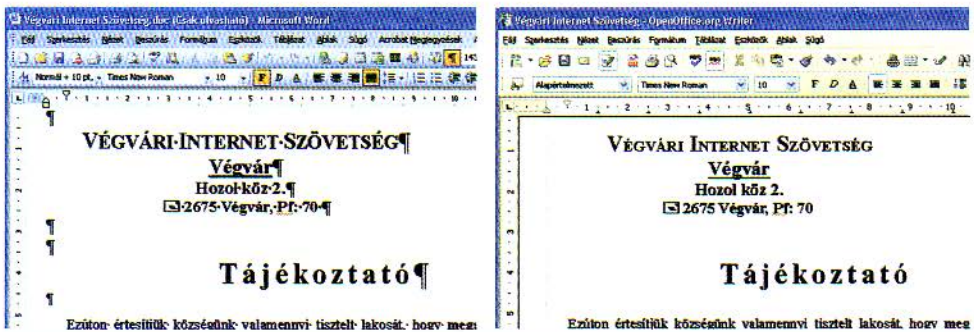
A DOS alatt futó Context szövegszerkesztő felülete.

A papíron a nyomtatón beállított betűtípus jelenik meg, a programból csak a betűstílus szabályozható. Érdemes megjegyezni, hogy a telepített program mérete összesen 340 KB volt. Mekkora helyet foglal el nagyságrendileg a háttértáron egy mai szövegszerkesztő program?

A szövegszerkesztés tehát nem a számítógépekkel együtt jelent meg, főbb elvei már régen kialakultak. (Gondoljunk pl. a középkori kódexekre!) Ezért a számítógépes szövegszerkesztő programok funkciói értelemszerűen hasonlítanak egymásra. *Érdeemes tehát a hangsúlyt a konkrét megvalósítás helyett az általános alapelvekre és funkciókra helyezni, mert így könnyen át tudunk állni egy újabb verzióra vagy egy másik szövegszerkesztőre.*

A szövegszerkesztők felülete

A grafikus felületek elterjedésével a szövegszerkesztők megjelenése is egységessé vált. A fejezet ábráihoz három teljesen különböző gyártótól származó szövegszerkesztő program képernyőképeit fogjuk használni, ezek a *Microsoft Word* [májkroszoft vörd], az *Ability Write* [abiliti rájt] és az *OpenOffice.org Writer* [openofisz org rájter].



A *Microsoft Word* (balra) és az *OpenOffice.org Writer* (jobbra) szövegszerkesztő jellemző képernyője. Milyen hasonlóságokat látunk a két felület között?

A szövegszerkesztők funkcióinak elérése

Tekintsük át röviden, milyen módszerek alakultak ki az alkalmazói programokban az egyes funkciók elérésére!



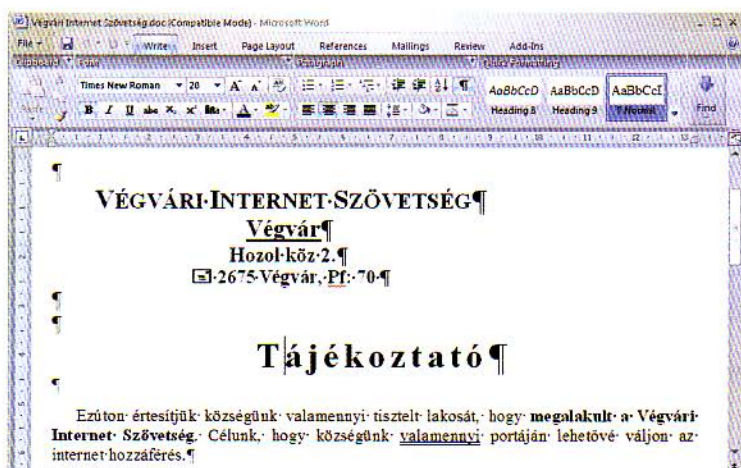
A *Nézet* menü *Eszközök* pontjával bővíthetjük a megjelenő eszköztárat (*OpenOffice.org Writer*). Próbáljuk ki ezt a funkciót az általunk használt szövegszerkesztő programban!



Az *Eszközök* menü *Testreszabás* pontjával két sorban helyezhetjük el a *Szokásos* és a *Formázás* eszköztárat (*Microsoft Word*). Végezzük el ezt a beállítást!

Menü*. A képernyő tetején jelenik meg, és az adott program által ismert összes funkció elérhető belőle. Ha a program túl sok funkciót ismer, akkor a szükséges parancs megkeresése nehézkes, ezért többnyire testreszabható. Ha valamely elemre az egér jobb gombjával kattintunk, akkor a rá jellemző műveletek között választhatunk a **helyi menüben***.

Eszköztárak*. A funkciók gyorsabb elérését lehetővé tevő gombsorok. Az eszköztárak száma is gyakran módosítható.



A Microsoft Office 2007-es verziójában kombinálják a menüt az eszköztárral, így a felület könnyebben átlátható.

A menü és az eszköztár kombinációjával egy kényelmesebben kezelhető felülethez juthatunk. Ezt a megoldást használja pl. a Microsoft Office 2007-es verziója.

Billentyűkombinációk*. Leggyorsabban a billentyűkombinációkkal dolgozhatunk. Ezek egy része egy adott rendszerben szabványosnak tekinthető, ezért néhányat (pl. mentés) érdemes megjegyeznünk.

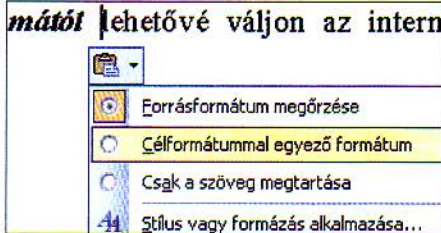
CTRL N – Új dokumentum (new [nyúl])	CTRL S – Fájl mentése (save [széjv])
CTRL O – Fájl megnyitása (open)	CTRL P – Fájl nyomtatása (print)

Néhány gyakori billentyűkombináció a dokumentumfájlok kezeléséhez. (A magyar verzióban is az angol billentyűkombinációkat használják.)

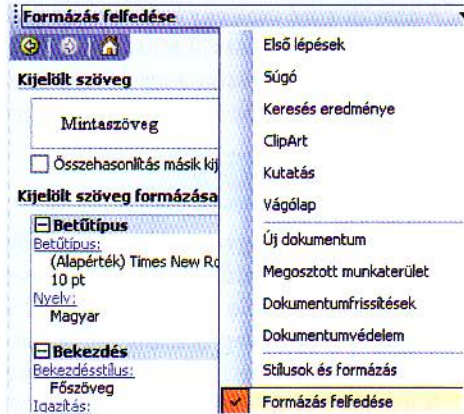
Munkaablak. Vannak olyan összetettebb műveletek, melyeket a menüből csak több különböző menüpont kiválasztásával érhetünk el. Ilyen például a kijelölt rész valamennyi formátumának áttekintése. A munkaablak a képernyő jobb oldalán jelenik meg, és a kiválasztott műveletcsoportot egyben tartalmazza. (Például az MS Wordben a *Formázás felfedése* munkaablakban kényelmesen áttekinthetjük a kijelölt rész összes formátumát.)

Intelligens címke. Gyakori, hogy egy adott művelet után rendszeresen további műveleteket is végezni kell. Például, ha egy félig kész szövegbe beillesztünk egy újabb szöveg-

részt, akkor annak formátumát a már meglévő szöveghez kell igazítani. A beillesztés helyén megjelenő intelligens címke ilyenkor automatikusan megjelenik, és lehetővé teszi a művelet kiválasztását.



Az intelligens címkével megadhatjuk a beillesztett szöveg formátumát (Microsoft Word).

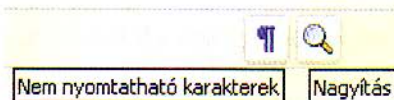


A munkaablak segítségével megjelenítjük a kijelölt rész formátumát (Microsoft Word).

A szöveg megjelenítése

Szöveg beírásakor a szöveget teljes szélességében szeretnénk látni, míg formázáskor szükségünk van arra is, hogy az adott szöveg hol helyezkedik el az oldalon. Ennek megfelelően a legtöbb szövegszerkesztőben a szöveg nagyítását is módosíthatjuk.

Munkánkat segíti, ha a képernyőn azt is figyelemmel tudjuk kísérni, hol nyomtuk le a szóközt vagy bekezdés vége (ENTER) gombokat. Az ilyen **nem nyomtatható karakterek*** a szöveg megjelenését vezérlik, helyüket speciális szimbólumok jelzik a képernyőn. Ezek megjelenítését általában az eszköztáron tudjuk be- vagy kikapcsolni.



A nem nyomtatható karakterek ki- és bekapcsolása, illetve a nagyítás beállítása általában az eszköztár segítségével érhető el (OpenOffice.org Writer).

A szöveg begépelése, részei és kijelölése

A szövegszerkesztésben célszerű előbb a „nyers” szöveget bevinni, majd a formázásokat (beleértve a nyelvi hibák javítását is) a szöveg bevitele után elvégezni.

A bevitel tiltott műveletei

A szöveg helyes beírásának két legfontosabb szabálya a következő:

Az ENTER gombot csak a bekezdés végén nyomjuk le, a bekezdés sorokra tördelését bízzuk a szövegszerkesztő programra. Mivel új bekezdést akkor indítunk, amikor új gondolatot kezdünk, a szövegszerkesztők lehetővé teszik, hogy azokat egységesen, de egymástól elkülönítve formázhassuk meg.

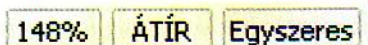
A szavak közé csak egy szóközt tegyünk, és **ne használjuk a szóköz gombot szavak, címek stb. igazítására!** Látni fogjuk, hogy a karakterek szélessége sok mindentől függ, így a szóközők szélessége még két egymást követő sorban sem feltétlenül azonos. Ez azt jelenti, hogy a szóköz nem használható pozícionáló eszközként.

A hibák javítása

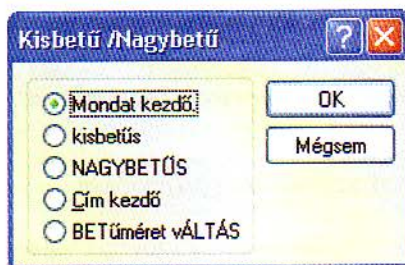
Tekintsük át a leggyakoribb gépelési hibákat és azok javításának módját!

Kimaradt egy betű. A számítógépeken általában kétféle beviteli mód van: *beszúró*, illetve *átíró*, a kétféle üzemmód között az *Insert* [inzört] gombbal válthatunk. Ha egy kimaradt betűt utólag beírunk, akkor az beszúró módban helyet csinál magának, míg átíró módban a következő betű helyére kerül.

Bekerült egy fölösleges betű. A fölösleges betűt kétféleképpen törölhetjük ki. Vagy a kurzort a betű mögé visszük és lenyomjuk a *Backspace* [bekszpész] gombot, vagy a kurzorral a betű elé állunk és lenyomjuk a *Delete* [dilit] billentyűt.



Gyakran az állapotsoron lévő ÁTÍR (vagy BESZÚR) feliratra kattintással is válthatunk a kétféle mód között (OpenOffice.org Writer).



A **Formátum** menü **Kisbetű–nagybetű** pontjával javíthatjuk a véletlenül lenyomott Caps Lock által okozott hibát (Ability Write).

Véletlenül lenyomtunk az Enter gombot. Az Enter gomb lenyomását általában a ¶ nem nyomtatható karakter jelzi. Ha a nem nyomtatható karakterek képernyőnkön nem jelennek meg, kapcsoljuk be azok megjelenítését az eszköztáron, majd a ¶ karaktert a betűkhöz hasonlóan töröljük ki.

Csupa nagybetűvel írtunk. A hiba oka a véletlenül lenyomott *Caps Lock* [kepszlok] billentyű. Ilyen esetekre a legtöbb szövegszerkesztő program lehetővé teszi, hogy a szövegrészt kijelöljük, és a betűket kisbetűsre módosítsuk.

A szöveg részei

A papír alapú megjelenítéskor a dokumentum legkisebb egysége a **leütés**, nagyobb egységei a **szó**, a **sor**, illetve az **oldal**. A leütés fogalma nemcsak a betűket és írásjeleket jelenti, hanem beletartozik a szavakat elválasztó szóköz is. A *szöveg terjedelmét* általában a *leütések számával adják meg*. (Egy írógéppel készített szabványos oldal 25-30 soros volt, és soronként 50-60 leütést tartalmazott, ami összesen 1200-1800 leütés.)

Az elektronikus szövegszerkesztésben a leütésnek a **karakter*** fogalma felel meg. A következő egység a **bekezdés***, mely két **bekezdésvége jel** (¶) közötti rész. A bekezdés első sorát gyakran eltérően formázzák, például beljebb kezdik (első sor behúzása). Általában nem szép, ha a **bekezdés utolsó sora**, a **kimenetsor** túl rövid.

A szövegszerkesztésben **egy fájl egy dokumentumot tartalmaz**, ez azonban szakaszokra tagolódhat. A **szakaszok*** a dokumentumok olyan részei, melyeknek oldalbeállításai (pl. lapméret, hasábok száma, margóviszonyok) **eltérőek lehetnek**. Ilyen például egy tankönyv végén a kislexikon vagy a szakácskönyv végén a tárgymutató.

A kijelölés módjai

Ha a dokumentumban egy adott szövegrészt meg szeretnénk formázni, akkor azt általában ki kell jelölni.

A kijelölést leggyakrabban egerrel végezzük. A legegyszerűbb megoldás az, hogy a **kijelölendő részen lenyomott bal gombbal végighúzzuk az egérkurzort**. Ha az egeret a sorok előtti oszlopban húzzuk lefelé, akkor így teljes sorokat jelölhetünk ki.

Gyakran kényelmesebb a billentyűzetet használni. Ilyenkor általánosnak tekinthető megoldás, hogy a **SHIFT gomb lenyomása mellett a kurzormozgató billentyűkkel végigvezetjük a szövegkurzort** a kijelölendő szövegrészen.

A **kijelölt részt egérekattintással vagy valamelyik funkciógombbal növelhetjük**, ennek konkrét megvalósítása függ az adott programtól is. Az OpenOffice.org Writerben például egy kattintás az adott szót, míg kettő az adott mondatot jelöli ki. A Microsoft Word esetében pedig egy kattintás az adott szót, kettő az adott bekezdést, három pedig a teljes dokumentumot.

A **funkciógomb** használatára jó példa a Microsoft Word. Itt az **F8** egymás utáni lenyomása rendre kijelöli a szót, a mondatot, a bekezdést, majd a teljes dokumentumot.

Téglalap alakú rész kijelölésére (**hasábblokk**) viszonylag ritkán van szükségünk a betűk eltérő szélessége miatt. Viszonylag ritka az is, hogy egyidejűleg több **különálló szövegrészt** jelöljünk ki. Ezt a két művelet emiatt nem minden szövegszerkesztő program támogatja.

A **teljes dokumentum kijelölésére gyakorlatilag minden mai szoftver a CTRL A billentyűzetkombinációt használja**. (Érdemes megemlíteni, hogy ez alól az Microsoft Word régebbi változatai kivételek, ott a CTRL SHIFT O kombinációt kellett használni.)

A kijelölésre nem minden esetben van szükség. **Azt a szót vagy bekezdést, amelyen a kurzor áll, általában nem kell külön jelölni**, ha a betű- vagy bekezdésformátumot szeretnénk módosítani.

...sége, nem csoda, hogy itt ringott a keresztény r
rális fellegvára. ¶
Itt épületegyüttesét szimbólumnak is tekinthetjü
középen zömök tornyával, kétoldalt pedig a kö
na nyugat felé kitárt karokkal állna ott. Valóban
továbbította a keresztény kultúrát az ország lak
n választékos jubileumi kiállítás, hanem a főap
teréves történetéről tanúskodnak. Az utóbbi 15

bencés használja, nyilván "Széphalom" mintájára.
(Pannónia-szent-hegyének) vagy Szent-Márton kol
mezővárost pedig Győrszentmártonnak, melyre cs
(nyilván a szent nevek számának csokkentésére). A
magyarokat is megigzerte a táj szépsége, nem csod
bölcsője, s lett annak később kulturális fellegvára.
A kolostor műt szíjában kialakult épületegyütte
vagy észak felől közelítünk feléje, középen zömök
főapáti szárnyával, úgy tűnik mintha nyugat felé k
ezer éven át ezt tette befogadta és továbbította a k
csaka most megnyit mértéktartóan választékos ju
egyéb emlékei és Magyarország ezeréves történet
restaurálási munkálatai alapján rendkívül értékes,

Nem összefüggő terület kijelöléséhez a CTRL gombot, hasábblokk kijelöléséhez az Alt gombot kell lenyomva tartani (Microsoft Word).
Mikor használhatók ki ezek a lehetőségek? Mondjunk egy-egy példát!

Betűformázás

A betűk formai jellemzői

A betű három jellemző adata: a típusa, a stílusa és a mérete.

A *típus** a betűk és egyéb jelek közös grafikus jegyek szerint megtervezett együttese. A betűtípust a neve azonosítja. A legismertebb típusok a Times [tájmsz], az Arial (vagy Helvetica [helvetika]), illetve a Courier [kurír].

Egy adott betűtípusnak nagyon sok változata van: a betű lehet félkövé, dőlt, aláhúzott, kiskapitális, keskeny, széles stb. A **betűtípus különböző változatait betűstílusoknak* nevezük**. A betűtípusok tervezésénél *a változatokat külön-külön tervezik*, a dőlt betű például nem egyszerűen az álló betű megdőntött változata. Ha egy betű valamely változata nem áll rendelkezésre, akkor a szövegszerkesztő programok azt az alapbetűtípusból matematikai úton előállítják (*hamis dőlt, hamis félkövé*).



Ha nem áll rendelkezésünkre a megfelelő betűváltozat, a szövegszerkesztő program az alapbetűből torzítással állítja azt elő. A bal oldali a valódi, a jobb oldali pedig a hamis dőlt betű. Milyen különbségeket látunk?

A *betű méretét** pontokban adják meg, a *pont** az inch [incs] (2,54 cm) 1/72-ed része. A szokásos betűméret 10–12 pont, de pl. a kiadványszerkesztésben gyakran használják a pont többszörösét is: $1 \text{ pica} = 12 \text{ pont}$ [pika].

A betűtípusok csoportosítása

Első megközelítésben a betűtípusokat két csoportba sorolhatjuk. A *talpas* (vagy *serif* [serif]) **betűk** (pl. Times) **változó vonalvastagságúak, a száruk talpakban végződnek**. Az ilyen betűk könnyen megkülönböztethetők, a talpak vezetik a szemet, ezért jól olvashatók. Főleg könyvek, újságok esetében használják, amit folyamatosan kell olvasnunk.

Elektronikus megjelenítés, pl. a monitor vagy projektor esetén, ahol kisebb a felbontás, a jobban felismerhető talp nélküli (sans serif [szan serif]) betűket használják. Ezek a **talp nélküli betűk** többnyire állandó vonalvastagságúak (pl. Arial).

Times, Arial, Brush Script, Jokermann

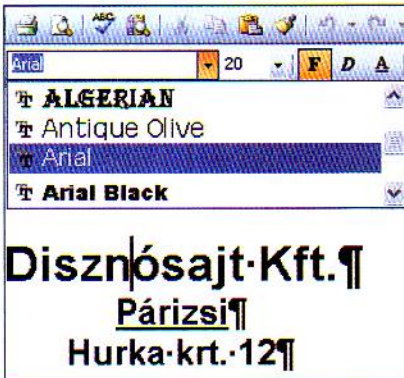
Néhány jellegzetes betűtípus: talpas, talp nélküli, kézirást utánozó és díszbetű. Melyiket mire használjuk?

Természetesen további csoportokat és alcsoportokat is megkülönböztetnek, ilyenek például a *kézírást utánzó* (script [szkript]) betűtípusok (pl. Brush Script [brás szkript]) vagy a *dísz- és reklámbetűk* (pl. Jokermann [dzsókermen]).

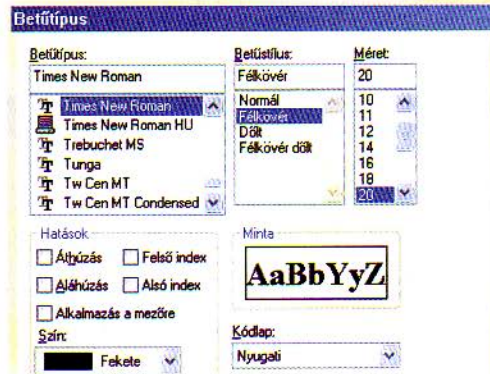
A betűformázás lehetőségei

A betűformázás gyakori feladat, ezért többféle módon is elérhető.

A *betűtípus*, *betűméret* és a három legfontosabb *betűstílus* (félkövér, dőlt, aláhúzott), illetve a *betűszín* legkényelmesebben az *eszköztár megfelelő ikonjaival választható ki*. Legördítve a betűtípus listát a kiválasztott betűtípus neve többnyire már saját formátumában jelenik meg, így hamar eldönthetjük, hogy tetszik-e egyáltalán.



Az eszköztár ikonjaival módosíthatjuk az elérhető betűformátumokat (Microsoft Word). Olvassuk le az ábrán a kurzornál lévő betű formátumát!



Valamennyi betűformátum eléréséhez a **Formátum** menü **Betűtípus** menüpontját kell választanunk (Ability Write). Miért célszerűbb többnyire mégis az eszköztár használata?

Természetesen a *menü segítségével valamennyi betűformátum elérhető*. Kényelmes, ha a szövegszerkesztő a kiválasztott formátumot mindjárt egy mintán be is mutatja.

Leggyorsabban a billentyűkombinációk segítségével formázhatunk. A három alapstílus billentyűkombinációit külön is érdemes megjegyeznünk, mivel ezek az alkalmazói programokban általában egységesek.

A szövegszerkesztők további lehetőségeket is kínálhatnak a betűformátumok megtekintésére, illetve beállítására. A Microsoft Wordben pl. ilyen lehetőség a munkaablak már említett *Formázás felfedése* pontja.

CTRL B – félkövér (bold)
 CTRL I – dőlt (italic [itelik])
 CTRL U – aláhúzott (underline [ándórlájn])

A szövegszerkesztőkben többnyire a megfelelő angol kezdőbetűkkel érhetőek el a betűstílusok. (A Microsoft Word korábbi változataiban a magyar kezdőbetűket használták.)

Szimbólumok beillesztése

A billentyűzetre értelemszerűen csak a leggyakoribb karakterek kerültek. Így például hiába keressük a műszaki életben használt speciális jeleket (fok, görög betűk) vagy a nemzeti valuták jelöléseit. Azokat a karaktereket, amelyeket a billentyűzet segítségével közvetlenül nem vihetünk be, *szimbólumoknak** nevezzük. Az igény miatt gyakoriak a csak szimbólumokat tartalmazó karakterkészletek (pl. Symbol [szimból]). A szimbólumok kezelése és formázása a betűkével azonos, az eltérés csupán a bevitel módjában van.

A szimbólumokat kétféleképpen vihetjük be. Az egyik lehetőség a hozzájuk tartozó karakterkód megadása, ezt pl. Windowsban az ALT Gr gomb lenyomása mellett, a numerikus billentyűzeten írhatjuk be. A másik lehetőség, hogy a megfelelő menüponttal megjelenítjük a lehetséges szimbólumok táblázatát, ahonnan a szükséges jelet az egérrel választhatjuk ki.



A szimbólumok táblázatát a **Beszúrás** menü **Szimbólum** vagy **Különleges karakterek** pontjával érhetjük el. Például a céges fejléceknél használatos szimbólumok a Wingdings betűtípusban találhatóak (balra Microsoft Word, jobbra OpenOffice.org Write).

Tervezzünk ezek felhasználásával egy fejléces papírt iskolánk részére!

Bekezdésformázás

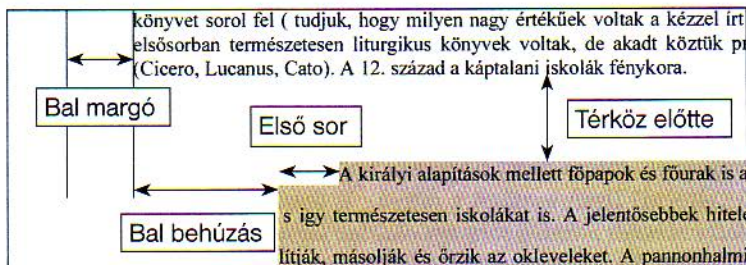
Bekezdésformátumok

Ha új gondolatot kezdünk, akkor azt írásban új bekezdés indításával jelezzük. Mint láttuk, a szövegszerkesztésben bekezdésnek két bekezdésvége jel közötti részt nevezünk. A bekezdések formázásánál három lehetőségünk van.

Az *igazítás** azt adja meg, hogy hogyan helyezkednek el egymáshoz képest a bekezdés sorai. A szövegszerkesztők általában négy lehetőséget ismernek: *balra zárt**, *jobbra zárt**, *középre zárt**, illetve *sorkizárt**; ritkábban megjelenik egy ötödik lehetőség is, az elosztott. A *sorkizárt* esetben a szövegszerkesztők a szóközök szélességét állítják be úgy, hogy a szöveg bal széle és jobb széle egymás alatt legyen (ettől az első sor és a kimenetsor eltérhet). Az *elosztott* elrendezésben az utolsó sor jobb széle is a többi alá kerül, rövid kimenetsor esetén hatalmas szóközöket eredményezve.

A **behúzás*** a bekezdés bal, illetve jobb szélének távolsága a margótól. Az *első* sort folyószövegben a jobb olvashatóság kedvéért gyakran beljebb kezdik. Ha viszont a bekezdés többi sorához képest az első sor kijebb van, akkor *függő behúzásról* beszélünk.

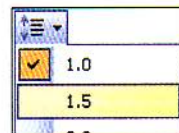
A **térköz*** a bekezdés függőleges elrendezését adja meg: az előző, illetve a következő bekezdéstől mért távolságot, valamint a sorközt.



A behúzás és térköz fogalma. Hogyan állíthatók ezek be az általunk használt programban?

A bekezdésformátumok beállítása

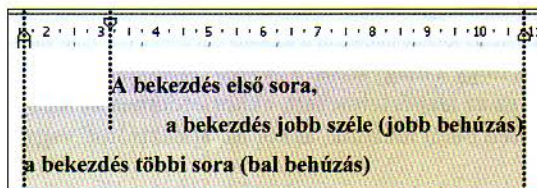
Az *igazítást* gyakorlatilag minden olyan alkalmazói programban, amelyek kezelik a szöveget is, az *eszköztár megfelelő ikonjaival* módosíthatjuk. Az eszköztárat érdemes áttanulmányoznunk, mert igen gyakran további lehetőségeket is kínál a bekezdés gyors megformázására.



Az eszköztár ikonjai a bekezdések formázásához. Balra az igazítás beállítása (Ability Write), középen a behúzás csökkentése és növelése (OpenOffice.org Writer), jobbra a sorköz beállítása (Microsoft Word). Hasonlítsuk össze az ábrán látható ikonokat az általunk használt szövegszerkesztővé!

A behúzást egyszerűen módosíthatjuk a **vonalzón** lévő jelek húzásával is. A módszer előnye, hogy a képernyőn közvetlenül látjuk a formázandó bekezdést. Aláírást például úgy helyezhetünk el, hogy a bal behúzás jelét jobbra toljuk.

A bekezdésformátumok módosítására a **legnagyobb szabadságot ezúttal is a menü adja**, ahol minden lehetőséget megtalálunk. A menü segítségével a megfelelő értékeket pontosan is beállíthatjuk, míg pl. a vonalzón történő húzással nem feltétlenül.



A behúzás beállítása a vonalzó segítségével (Microsoft Word). Készítsünk el egy aláírásmintát a vonalzó segítségével egy dokumentum aláírási sora alá!

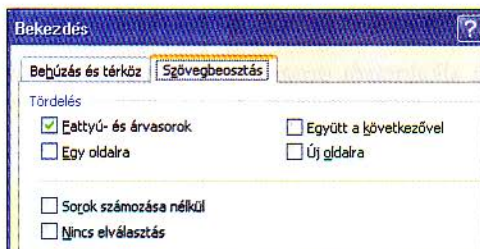
Bekezdések laphatáron

Ha a bekezdés laphatárra kerül, a szöveg áttekinthetőségét jelentősen ronthatja, ha a sorok rosszul helyezkednek el.

A bekezdés első sorát *árvasornak* nevezik, ha az egyedül áll a lap alján, utolsó sorát pedig *fattyúsornak*, ha az egyedül áll a lap tetején. A szövegszerkesztő programok általában automatikusan úgy küszöbölik ezt ki, hogy egy sort (az árvasort vagy fattyúsor esetén még egy sort) átvisznek a következő oldalra. Természetesen ez a funkció igény esetén kikapcsolható.

Szakszövegeknél előfordulhat, hogy egy hosszabb bekezdést egyáltalán nem szabad laphatáron „eltörni”. Ilyen esetben a fattyú- és árvasorok kezelése helyett *előírhatjuk, hogy az adott bekezdés minden sora egy oldalon legyen.*

Hasonló módon, ha a cím az egyik oldal aljára kerül, miközben a hozzá tartozó rész a következő oldal tetején folytatódik, a szöveget az olvasó nehezebben tudja követni. Ezért többnyire *megadható, hogy a címet tartalmazó bekezdés a következő bekezdéssel egy oldalra kerüljön, vagy az adott cím mindenképpen új oldalt kezdjen.*



A bekezdés tulajdonságai között rendelkezünk arról, hogy mi történjen, ha egy bekezdés laphatárra kerül (Microsoft Word). Mire használhatjuk az egyes lehetőségeket? Mondjunk egy-egy példát!



A bekezdés tulajdonságainak pontos beállítása a **Formátum** menü **Bekezdés** pontjával történik (OpenOffice.org Writer). A 0,21 cm-es bekezdés feletti térköz hány pontnak felel meg?

A szöveg helyes bevitele és javítása

Az írásjelek használatának szabályai

A mondatvégi írásjelek (pont, kérdőjel, felkiáltójel), illetve a vessző, pontosvessző, kettőspont elé nem teszünk szóközt, utána viszont igen.

Az idézendő szöveg és az idézőjelek közé nem kerül szóköz, a kezdő idézőjel elé és a záró idézőjel után viszont igen. Ügyeljünk arra, hogy a magyarban a „99-es” idézőjeleket használjuk, ha a szövegszerkesztőnkben nem ez az alapértelmezett, akkor érdemes az idézőjeleket beírás után lecserélni. Ha az idézetben belül idézünk, akkor az 'aposztrófof' vagy a »francia idézőjeleket« kell alkalmaznunk.

A gondolatjel előtt és után szóköz van (kivéve, ha vessző követi), maga a gondolatjel a magyarban az elválasztójelnél hosszabb. Gyakran a CTRL és a „szürke mínusz” beírásával érhető el, de ez szövegszerkesztőnként eltérhet.

A nyitó zárójel után, illetve a csukó zárójel előtt nincs szóköz. Ha egész mondatot zárójelbe teszünk, akkor a mondatvégi írásjel is belekerül, egyébként azonban nem. Ma már nem illik zárójel helyett a / jelet használni.

Szavak és kifejezések elválasztása a sorok végén

A bekezdéseket a szövegszerkesztő program a szóközőknél, illetve az elválasztójelek-nél automatikusan sorokra tördeli. Mivel ez nem minden esetben felel meg az előírások-nak, lehetőségünk van speciális jelek alkalmazásával ezt módosítani.

A **nem törhető szóköz** normál szóköz helyett használható, ha azt akarjuk, hogy a két szó mindig ugyanabban a sorban maradjon. Célszerű használni például nagy helyiértékű számok tagolásánál: 123°456°000. (A képernyőn látható jele papíron nem jelenik meg.)

A **nem törhető kötőjelnél** a két szó nem választható el. Például egy telefonszám nem választható el a kötőjelnél, miközben annak meg kell jelennie a papíron: 123–4567.

A **lág elválasztójel** vagy **feltételes kötőjel** az elválasztás helyét jelzi. Ha az adott szó a sor végére kerül, akkor a szövegszerkesztő program a feltételes kötőjelnél elválasztja a szót, és kiteszi a kötőjelet, azonban a kötőjel nyomtatásban nem jelenik meg.

CTRL SHIFT szóköz – nem törhető szóköz
 CTRL SHIFT kötőjel (nem mínusz!) – nem törhető kötőjel
 CTRL kötőjel (nem mínusz!) – feltételes kötőjel
 CTRL mínusz – gondolatjel
 SHIFT ENTER – kézi sortörés (adott bekezdésen belül)
 CTRL ENTER – kézi oldaltörés

Néhány különleges karakter bevitele a Microsoft Word szövegszerkesztőben

Matematikai, műszaki és pénzügyi jelek beírása

Számok beírása. A tizedesvessző, illetve a sorszámoknál a pont előtt nincs szóköz. Mivel az irányítószám nem sorszám, utána nem szabad pontot tenni.

Képletek beírása. A számokat, műveleti jeleket, zárójeleket és a függvények nevét (pl. lg, sin, sgn) álló, a változókat dőlt betűvel írjuk. A műveleti jelek előtt és mögött tegyünk egy-egy szóközt:

$$x^2 + 3x + 4 = -\sin(x + 2)$$

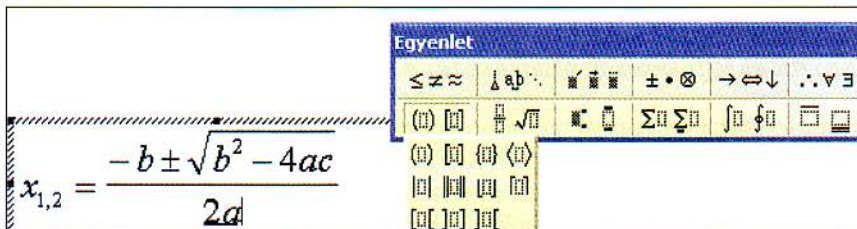
Igényesebb képletek beírására használjunk inkább egyenletszerkesztő programot. Gyakran használják *szorzásjelként* a keresztet, pl. 2×3 . Ezt ne x -el írjuk, hiszen a megfelelő jel a magyar billentyűzeten megtalálható (*Alt Gr ú*).

A számokat négy számjegyig egybeírjuk, a fölött ezres tagolást használunk. Az ezres tagolást a magyarban a szóköz jelöli: 2 345 678, pontot csak a banki bizonylatoknál szokás használni.

A **százalékjel** közvetlenül a szám után áll, tehát nem kell elé szóközt tenni: 12%.

A *pénznem jelét* a nemzeti szabványok rögzítik. A forint jele pl. követi a számot, de előtte egy nem törhető szóközt kell tenni: 12 Ft, és nem 12FT vagy 12ft.

A *fizikai mennyiségek* esetén a mérőszám és a mértékegység közé nem törhető szóközt teszünk: 34 kg. A fizikai mennyiségek általában kisbetűvel kezdődnek, kivéve azokat, amelyek személynévből származnak.

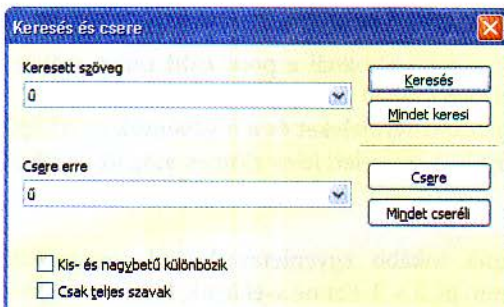


Képletek beírásához a **Beszúrás** menü **Objektum** menüpontjával választhatjuk ki a Microsoft Equation [ikvesőn] egyenletszerkesztő programot (Microsoft Word).

Keresés és csere

Hosszabb dokumentumok esetén egy adott szó megkeresése pusztán a szöveg elolvasásával igen nehézkes. Még időrablóbb, ha egy adott szót vagy jelet egy másikra kell cserélnünk. Például ma már nem fogadható el, ha az ü betű helyett az û jel szerepel, ezért ha egy ilyen (évekkel ezelőtt beírt) szöveg kerül a kezünkbe, akkor azt cserélnünk kell.

Adott szótöredék, szó vagy kifejezés keresése és cseréje alapvető szolgáltatás minden szövegszerkesztő programban. Többnyire a Szerkesztés menü megfelelő pontjával érhetjük el. A szövegszerkesztő programok kereséskor a *kis- és nagybetűket nem különböztetik meg*, de a megjelenő ablakban ezt a funkciót is bekapcsolhatjuk. Mindig lehetőségünk van annak beállítására is, hogy *csak teljes szavakra keressünk*: ilyenkor a szoftver a *vas* szót nem találja meg a *szarvasmarha* szóban. A szövegszerkesztő programok általában támogatják adott formátumú részek keresését és cseréjét is.



Az adott karakter minden előfordulását lecserélhetjük a **Mindent cseréli** gombbal (OpenOffice.org). Mire kell ügyelnünk, ha ezt a funkciót magyar szavak esetén használjuk?

A legtöbb szövegszerkesztő további lehetőségeket is kínál. Például lehetővé tesz hasonló hangzású szavak vagy speciális jelek keresését is.

Sajnos a magyar nyelv ragozó jellege miatt a keresés funkció körültekintést igényel. Az *iskola* szót például kereséskor ragozott alakjaiban (pl. *iskolában*) a program nem fogja megtalálni; ilyenkor az *iskol* szórészt érdemes keresnünk. A ragozás csere esetén is gondot okozhat. Például, ha a *pénz* szót akarjuk lecserélni a *valuta* szóra, akkor a *pénzet* szóból *valutaet* lesz.

Másolás és mozgatás

A kijelölt szövegrész átmásolása vagy áthelyezése többnyire a vágólap segítségével történik. A szükséges műveleteket a Szerkesztés menüben is megtaláljuk, de kényelmesebb a billentyűzet használata. A kijelölt rész vágólapra másolása általában a *CTRL C*, vágólapra helyezése pedig a *CTRL X* kombinációval történik. A vágólap tartalmát ezután a kurzorhoz a *CTRL V* kombinációval illeszthetjük be. A *vágólap** segítségével akár egy másik dokumentumból vagy egy másik alkalmazásból is átvihetünk egy szövegrészt vagy egy ábrát a készülő dokumentumunkba.

Sok szövegszerkesztő lehetővé teszi azt is, hogy a kijelölt szöveget az egér segítségével húzzuk az új helyére. Ha nem áthelyezni, hanem másolni szeretnénk, akkor az egér húzása közben nyomnunk kell egy gombot is a billentyűzeten (általában CTRL).

Nyelvi ellenőrzés

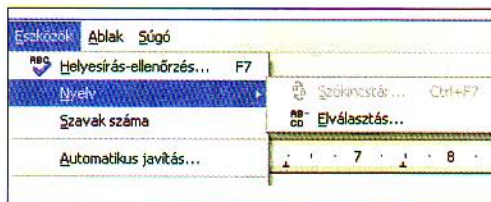
A szövegszerkesztőktől a felhasználók jogosan várják el a nyelvi ellenőrzést, hiszen ennek ma már technikai akadályja nincs. A szövegszerkesztő ilyenkor ellenőrzi a helyesírási hibákat, és gyakran nyelvi ellenőrzést is végez (pl. a vesszők helyes használata). Fontos, hogy a nyelvi ellenőrző programok nem készülhetnek fel minden lehetőségre, a javasolt módosításokat ezért nem minden esetben kell elfogadnunk.

A hibák javítására többféle módszer alakult ki, ezeket gyakran kombináltan alkalmazzák. A felhasználó számára legkényelmesebb, ha a szövegszerkesztő már beírás közben felhívja a figyelmet a hibákra, pl. piros hullámos vonallal aláhúzza a helyesírási és zölddel a nyelvhelyességi hibákat. Logikus megoldás az is, ha a folyamatos ellenőrzést kikapcsoljuk, és csak a munka végén hajtunk végre egy teljes nyelvi ellenőrzést, ekkor jobban tudunk a hibákra koncentrálni.

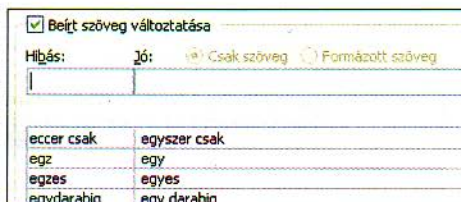
Érdekes megoldás az automatikus javítás. Ilyenkor a szövegszerkesztő bizonyos hibákat rákérdezés nélkül, a szó beírása után azonnal javít, például az „egz”-t „egy”-re.

Ugyancsak hasznos az elválasztás. Alkalmazásánál azonban célszerű résen lenni, ezek a modulok ma még néha tévednek.

A számítástechnika fejlődésével a szövegszerkesztő programokban újabb nyelvi elemek is megjelennek. Ilyen például a szinonima szótár, melynek használatával választékosabbá tehetjük dokumentumunkat.



A nyelvi eszközöket általában az **Eszközök** menüben érhetjük el (OpenOffice.org Writer).



Az automatikusan javítandó szavak listáját magunk is bővíthetjük (Microsoft Word).

Fájlműveletek: megnyitás, mentés, nyomtatás

A fájlok megnyitásával, mentésével és nyomtatásával már többször találkoztunk, ezért most csak a szövegszerkesztőkre jellemző, különleges beállításokat tekintjük át.

Fájlformátumok

A szövegszerkesztők általában a saját formátumukon kívül más fájlformátumokat is képesek kezelni. Érdekes néhányat közelebbről is megismerni.

A *TXT* formátumot gyakorlatilag minden alkalmazói program ismeri. Hátránya, hogy a formátumokat nem tárolja, így használatakor az összes formázási beállítás elvész.

Az *RTF* (*Rich Text Format* [rics tekszt format]) **formátum** is széles körben elterjedt. Előnye, hogy a **legtöbb szövegformázási lehetőséget megőrzi, miközben nem tartalmazhat futtatható kódot, így makróvírust sem**. Emiatt sok esetben ebben a formátumban helyezik el a dokumentumokat az interneten.

A *PDF* formátummal korábban már találkoztunk. Előnye, hogy platformfüggetlenül támogatja a dokumentumok megjelenítését, miközben a dokumentum lényegében már nem szerkeszthető. Ma még nem minden szövegszerkesztő támogatja az előállítását.

A *HTML* az internet terjedésével került előtérbe. A mai szövegszerkesztők képesek a dokumentumot weblapként is menteni, így akár webszerkesztőként is használhatók.

A sokféle formátum egységesítésére vezették be a szoftvergyártók a szoftver- és adatfüggetlen *XML*-t, amely hamarosan az adatcsere általános szabványává válhat.

(A *HTML*- és az *XML*-formátummal részletesebben a webszerkesztéssel foglalkozó fejezetben fogunk találkozni.)

Automatikus mentés

Hasznos szolgáltatás az **automatikus mentés**. Ez azt jelenti, hogy a szövegszerkesztő program bizonyos időnként automatikusan menti a dokumentumot a háttérben. Áramkimaradás vagy hardver- esetleg szoftverhiba esetén így csak az utolsó pár perc munkája vesz kárba.

Fájlnev:	Végvári Internet Szövetség
Fájl típusa:	Microsoft Word 97/2000/XP (.doc)
	OpenDocument-szöveg (.odt)
	OpenDocument-szövegsablon (.ott)
	OpenOffice.org 1.0-szövegesdokumentum (.odt)
	OpenOffice.org 1.0-szövegesdokumentumsablon (.ods)
	Microsoft Word 97/2000/XP (.doc)
	Microsoft Word 95 (.doc)
	Microsoft Word 6.0 (.doc)

Mentéskor megadhatjuk a kívánt fájlformátumot (OpenOffice.org Writer). Mikor érdemes az *RTF* formátumot használni?

Alapértelmezett megnyitási formátum:
Minden dokumentum (*.aww;*.doc;*.rtf;*.txt;*.sam)
Alapértelmezett mentési formátum:
Ability Write dokumentum (*.aww)
<input checked="" type="checkbox"/> Automatikus mentés minden: 10 percen

Az **Eszközök** menü **Beállítások** (vagy **Opciók**) pontjával megadható az automatikus mentés időintervalluma (Ability Write).

Oldalbeállítás és nyomtatás

A **nyomtatás** szinte valamennyi alkalmazói program esetében a **Fájl** menü **Nyomtatás** parancsával érhető el. Itt értelemszerűen megadható a **használandó nyomtató**, illetve hogy **mely oldalakat és hány példányban** nyomtassunk.

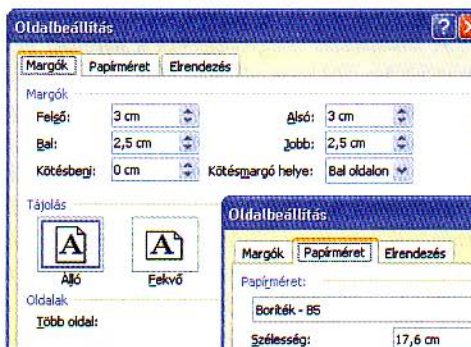
Az **oldal** paramétereinek megadása a **Fájl** menüben (pl. Microsoft Word) vagy a **Formátum** menüben (pl. OpenOffice.org Writer) található meg.

A szövegszerkesztőknél **fontos a margók* megadása**. Legegyszerűbb esetben ez a lap négy oldalán lévő margót jelenti, ha azonban a lapokat bekötjük, akkor erre külön kell gondolnunk („kötésmargó”).

Kétoldalas elrendezés esetén a **margókat tükrözni kell**, így nem bal és jobb, hanem külső és belső margót kell megadnunk (pl. egy könyvben).

A **szöveg tájolása*** lehet **álló** vagy **fekvő**. Ez utóbbit kisebb tájékoztató táblák vagy táblázatok esetén célszerű használni.

Végül fontos lehet a **papírméret megadása** is, ha nem a szokványos A4-es lapra, hanem pl. borítékra nyomtatunk.

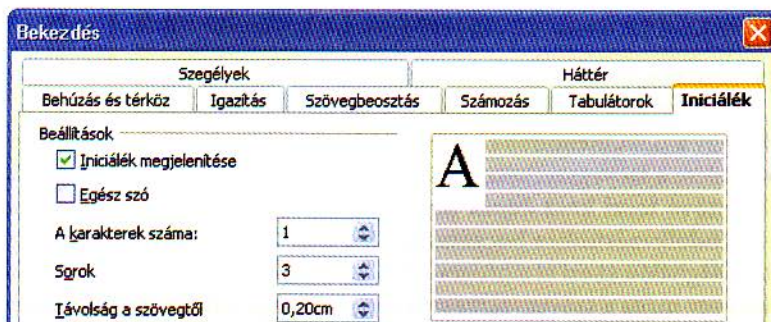


A **Fájl** menü **Oldalbeállítás** pontjával adhatjuk meg a margókat, a tájolást, illetve a papírméretét (Microsoft Word).
Olvassuk le az ábráról a felhasználó által használt beállításokat!

Grafikus elemek

Iniciálé

Az **iniciálé** a **bekezdés díszes kezdőbetűje**. A legtöbb szövegszerkesztőben csak annyit adhatunk meg, hogy **hány soros legyen**, milyen legyen a betűtípusa és besüllyedjen-e a szövegbe. További díszítőelemeket már általában nem támogatnak.



Iniciálé beillesztése (OpenOffice.org Writer)

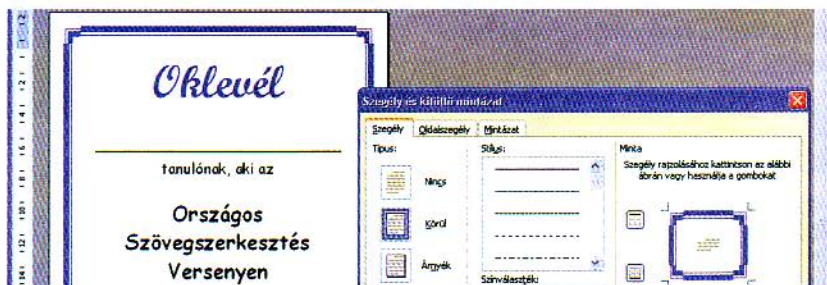
Melyik betű szerepel leggyakrabban iniciáléként a magyar nyelv esetén? Miért?

Szegély és mintázat

A **szegély*** a szavakat, bekezdéseket vagy az egész oldalt keretező díszes vonal. Ennek megfelelően három változata van. *Folyószövegben* ritkán használjuk, általában az elhunyt személyek nevét szokták vele tiszteletből kiemelni. *Bekezdések szegélyezésének* tipikus alkalmazása a figyelemfelkeltő vagy tájékoztató feliratok készítése, pl. „Vigyázz, vészfék!”. Az alkalmi dokumentumok, például oklevelek, emléklapok esetében a *lapszegély* a dokumentum gyakori díszítőeleme.

A **mintázat*** segítségével a kijelölt szövegrész, bekezdés vagy oldal **hátterét állíthatjuk be**. Jól használható címek kiemelésére, űrlapok esetén a kitöltendő részek megjelenésére vagy díszítőelemként. Érdemes figyelni arra, hogy *a mintázat előtt a szöveg olvasható maradjon*. Például, ha sötét mintázattal világos betűket használunk, akkor célzerű vastagabb vonalú, talp nélküli betűtípust választani.

A szegély és a mintázat (vagy háttér) beállítási lehetőségét általában a *Formátum* menüben találjuk: vagy külön menüpontként, vagy a betű, bekezdés, oldal beállítására szolgáló pont részeként. A **szegély beállításánál megadhatjuk a vonal vastagságát, színét és stílusát** (szimpla, dupla, szaggatott, stb.). A **mintázat esetében megadhatjuk a háttérszínt**, de szövegszerkesztőtől függően további lehetőségeink is lehetnek. Így például a háttér sraffozása (Microsoft Word) vagy háttérkép beszúrása (OpenOffice.org Writer).



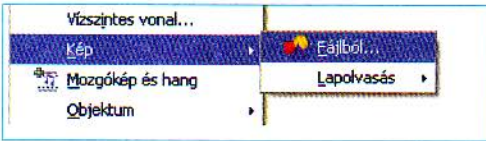
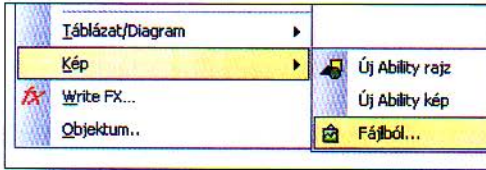
Oldalszegély használata oklevél tervezésekor. A szegélyt általában a **Formátum** menü segítségével érhetjük el (Microsoft Word). Készítsük el az ábrán látható oklevelet!

Képek beillesztése

Gyakori, hogy a dokumentumba színesítésként képeket, ábrákat szeretnénk elhelyezni. Ezek bevitelére a szövegszerkesztő programok igen sok lehetőséget kínálnak. Minden program **lehetővé teszi kész képek beillesztését** fájlból, **sőt gyakran külön gyűjteményt is a rendelkezésünkre bocsátanak**. Ma már az is előfordul, hogy a gyűjtemény egy részét a világhálón át éri el a program (pl. Microsoft Wordben a *ClipArt* [klipárt]).

Általában vagy maga a szövegszerkesztő program, vagy a hozzá tartozó irodai csomag tartalmaz *rajzok készítésére alkalmas* komponenseket (pl. rajz eszköztár). Képek bevitelére a legtöbb program támogatja a *lapolvasóról történő* beolvasást.

A programok a fentiekén túl gyakran *további grafikus objektumok* készítését és bevitelét is támogatják. Ilyenek például a szervezeti diagram, grafikonok stb.



A **Beszúrás** menü megfelelő pontjával szűrhatunk be a dokumentumba egy képet vagy más alakzatot (felül: Ability Write, alul: OpenOffice.org Writer).

A Microsoft Word munkaablakában kulcsszó megadásával kereshetünk a ClipArt képek között.

A kép formázása

A képet beszúrása után az *egérrel húzhatjuk a megfelelő helyre*. Átméretezése a kép négy sarkában és az oldalfelező pontokban megjelenő *nyolc méretező pont* húzásával történik. Gyakran lehetőségünk van a kép *elforgatására* is.

A ClipArt-használata

A ClipArt előre elkészített képek, filmek, hangok gyűjteménye. Egy részét együtt kapjuk az Office programcsomaggal a telepítő CD-n, de beletelhetnek a gépünkön, akkor a Microsoft webhelyén.



A kép egérrel húzható a helyére, és a nyolc méretező ponttal méretezhető át. A kép fölött lévő zöld pont mozgásával a képet akár el is forgathatjuk. Kettőt kattintva a képre a megjelenő **Kép formázása** ablak **Elrendezés** fülén állíthatjuk be a körbefuttatást (Microsoft Word).

A kép és a szöveg egymáshoz való helyzete sokféle lehet. A képet **elhelyezhetjük a szöveg mögött, a szöveg előtt, körbefuttathatjuk a szöveggel, vagy beszúrhatjuk karakterként is.**

A körülfuttatás sokféle lehet. Legegyszerűbb esetben a *kép mellett nincs szöveg*, gyakoribb azonban, hogy a szöveg a képet tartalmazó *téglalaphoz* illeszkedik. Látványos, ám sok tördelési problémát okoz a kép kontúrjához illeszkedő *szoros* körülfuttatás. Érdemes

azonban arra odafigyelni, hogy a látványos körülíratásnál fontosabb az olvashatóság. Például, ha a kép nem a szöveg szélére kerül, hanem a közepére, akkor az előtte kezdődő és utána folytatódó sorokat igen nehéz szemmel követni.

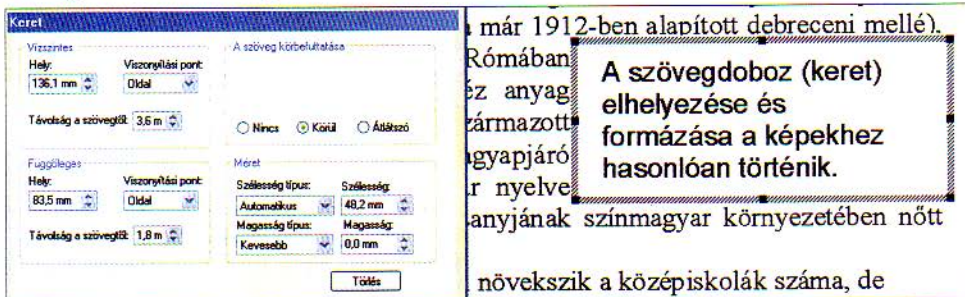
Ha a képet karakterként szöveggel egy sorba szúrjuk be, akkor az ténylegesen karakterként is viselkedik. Így pl. a betűkhöz hasonlóan emelhetjük vagy süllyeszthetjük az alapvonalhoz képest, viszont ekkor nem forgathatjuk el.

Ha a kép széléből szeretnénk levágni, vagy színkorrekciót szeretnénk végezni, akkor ezt a legtöbb szövegszerkesztő program ugyan lehetővé teszi, célszerű azonban inkább grafikus programra bízni. A háttérbe helyezett képek esetén fontos lehetőség a *halványítás*, illetve előtérbe helyezett képek esetén az *átlátszóság* beállítása.

A szövegdoboz (keret)

A *szövegdoboz* (keret)* a folyószövegtől formailag független objektum, amely egy önálló szövegrészt tartalmaz. Ilyen például az újságokban gyakori „keretes cikk”, amely tartalmát tekintve ugyan kapcsolódik a folyószöveghez, formailag azonban önálló elem.

A szövegdobozt általában szegély veszi körül, amelynek vastagságát, színét stb. beállíthatjuk. Mivel a szövegdoboz tartalmát ez a szegély határolja, gyakran szükségünk lehet a szövegdoboz belső margóinak a módosítására is.



Szövegdobozt (keretet) általában a **Beszúrás** menüvel hozhatunk létre. Formázása a képhez hasonlóan történik (Ability Write). Mondjunk példákat a szövegdoboz felhasználására!

Listakezelés

*Lista** alatt összetartozó elemek sorozatát értjük (pl. névsor). Megjelenítéstől függetlenül lehet *felsorolás* vagy *számozás* (számozott lista). *Felsorolás** esetén valamennyi listaelemet azonos szimbólum vezet be, *számozás** esetén pedig számok vagy más rendezett szimbólumok (betűk, római számok stb.).

Ha az elemek nem egyenrangúak, hanem hierarchikus felépítésűek, akkor azt *többszintű listaként* valósíthatjuk meg. Például egy feladatgyűjteményben a számmal jelölt feladatoknak gyakran betűvel jelölt részei is vannak.

A felsorolást és számozást az *eszköztár* segítségével vagy a *Formátum* menü megfelelő menüpontjával formázhatjuk meg.

A listajelek formátuma

Hagyományosan a felsorolás elemeit gondolatjel vezeti be, ezt régebben *bajusznak* neveztek. Használták a tömör kör alakú *diszpontot* is. Ma a szövegszerkesztő programok többségében *tetszőleges szimbólumot*, sőt gyakran *képet* is megadhatunk listajelként.

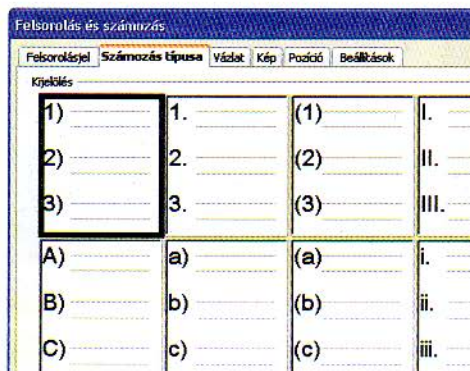
A számozott listákat hagyományosan a következő módon formázták. Ha a lista elemeit *arab szám* vezette be, akkor *utána pontot* tettek, és gyakran félkövérre formázták. Ha a lista elemeit *betűk* vezették be, akkor utána *csukó zárójelet* tettek, és *dőltre* állították:

- | | |
|-------------|------------|
| 1. hardver | a) hardver |
| 2. szoftver | b) szofver |
| 3. menver | c) menver |

Ma a szövegszerkesztő programok ettől eltérő megoldásokat is támogatnak.



A felsorolás szimbólumának megadása (Ability Write)



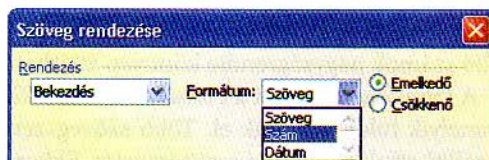
A számozás módjának kiválasztása (OpenOffice.org Writer)

Rendezés

A lista elemeinek *rendezését** minden szövegszerkesztő program támogatja. Ha például szeretnénk készíteni egy névsort, akkor elegendő beírni a neveket (mindegyiket önálló bekezdésbe), majd kiválasztani a rendezés menüpontot. Ez általában az *Eszközők* vagy a *Táblázat* menüben található.

Rendezéskor általában **két dolgot kell megadnunk: a rendezés irányát (emelkedő, csökkenő) és a rendezendő adatok típusát (szám, szöveg, dátum)**. Fontos tudnunk, hogy míg számként 9<10, addig szöveggént a 10 megelőzi a 9-et.

Ha a lista első eleme címet tartalmaz, akkor azt *rovatfejnek* nevezik („*Névsor*”). Általában megadhatjuk, hogy van-e *rovatfej*, mert ha van, akkor azt a program nem rendezi be az elemek közé.



Lista rendezése (Microsoft Word)

Az adatok táblázatos elrendezése

Adatok táblázatos elrendezésére a szövegszerkesztő programok kétféle megoldást is kínálnak: a tabulátort és a táblázatot.

Tabulátorok használata

A billentyűzetten lévő **tabulátor*** gomb hatására a **kurzor** gyakorlatilag minden alkalmazói programban **adott pozícióval jobbra lép**. A Jegyzettömbben pl. 8 karakterrel, a Microsoft Wordben 1,25 cm-rel, az Ability Write-ban 1,7 cm-rel.



Az írógép felső sorában lévő második gomb lenyomásával megjelölhetjük az aktuális oszloppozíciót. Később, ha a sor elején lenyomjuk az első gombot, a kocsi automatikusan az adott pozícióra ugrik. A többi jel rendre egy, két, három ... betűhellyel előbb állítja meg a kocsit, hogy a számokat helyi érték szerint egymás alá lehessen írni.



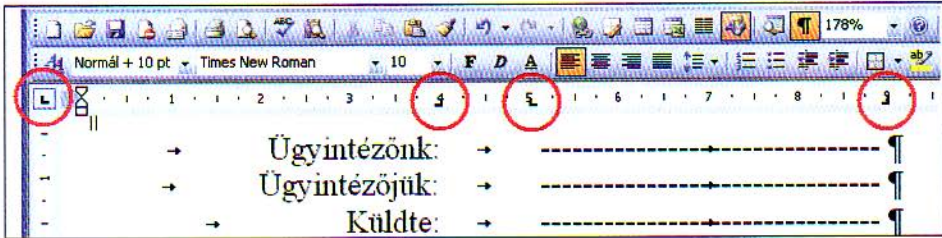
A tabulátorpozíciókat a **Formátum** menü **Tabulátor** pontjával állíthatjuk be számszerűen (Microsoft Word). Milyen beállítást látunk az ábrán?

A szövegszerkesztő programok mindig lehetőséget adnak a tabulátorpozíciók módosítására is, vagyis a **tabulátorpozíciók* helyzetét számszerűen megadhatjuk**. Mivel a **tabulátor bekezdésformátum**, ezért a tabulátor beállítások csak az adott bekezdésre vagy a kijelölt bekezdésekre vonatkoznak.

A szövegszerkesztő programoknál azt is megadhatjuk, hogy az adott pozíció a szöveg bal szélét, jobb szélét vagy közepét jelezze-e. Ennek megfelelően a **tabulátor lehet balra zárt, jobbra zárt vagy középre zárt**. Számok beírásához pedig a **decimális** tabulátort is meg szokták valósítani, ekkor az adott helyre a tizedesvessző kerül, így az egymás alatt álló számok nagyságrendje könnyen átlátható.

A tabulátorbeállítást a **Formátum** menü **Tabulátor** pontjával vagy a **Bekezdés** pont valamelyik fülén érhetjük el. Több szövegszerkesztő esetében a pozíciókat kényelmesen beállíthatjuk a **vonalzó segítségével** is. Ekkor a tabulátor igazításának kiválasztása után a vonalzón a megfelelő helyre kell kattintanunk.

Megadhatjuk azt is, hogy a szövegszerkesztő program *összekösse-e a tabulátorpozíciókat pontsorral vagy vonallal*. Ez jegyzékben vezeti a szemet, űrlapoknál pedig jelzi, hová kell a szükséges adatokat beírni.

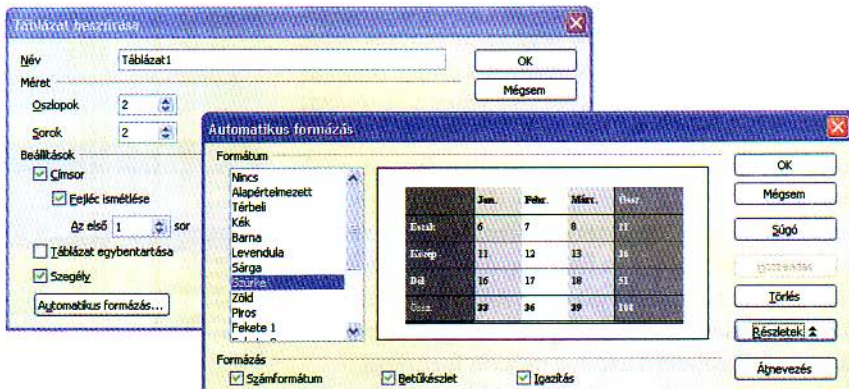


A vonalzó bal szélén lévő ikonnal kiválasztjuk, hogy milyen tabulátort szeretnénk definiálni, majd az adott pozícióra kattintunk a vonalzón (Microsoft Word).

Soroljuk fel a pirossal bekarikázott tabulátorpozíciók típusát és helyét! Melyiknél van kitöltés is?

Táblázat készítése

Az első szövegszerkesztő programoknál a táblázatot még vonalanként kellett megrajzolni, a mai programoknál már a **táblázat* önálló objektum**. Ennek megfelelően **létrehozásakor elegendő megadni a sorok és oszlopok számát**. A cellák mérete valamilyen alapértéket vesz fel, vagy igazodik a beírt tartalomhoz. Létrehozásakor maga a táblázat is kap valamilyen egyszerű formátumot.



A **Táblázat** menü megfelelő pontjával szűrhetünk be egy új táblázatot. Megformázása legegyszerűbben az automatikus formázás lehetőséggel történik (OpenOffice.org Writer).

A táblázat megformázásához a szövegszerkesztő programok nagyon sok lehetőséget kínálnak. Ezek közül a legegyszerűbb az **automatikus formázás**, amikor előre elkészített formátumok közül választhatunk.

Ha a táblázatot önállóan szeretnénk megformázni, akkor lehetőségünk van a **sorok magasságának, az oszlopok szélességének megadására, cellák összevonására és felosztására**, valamint **szegélyek és háttérszínek** beállítására. A cellák tartalmát **vízszintesen és függ-**

gőlelesen is a cellához igazíthatjuk, de felhasználhatjuk a bekezdésformátumokat is. Minden szövegszerkesztő támogatja a sorok vagy oszlopok beszúrását és törlését is.

A **táblázat sorait** a megfelelő menüponttal (*Rendezés*) akár **rendezhetjük is**. Ilyenkor a listánál tanultak szerint kell eljárunk, csupán **meg kell adni, hogy a rendezés melyik oszlop szerint történjen**. Az oszlop azonos értékei esetén általában két újabb oszlopot választhatunk ki mint második, illetve harmadik rendezési szempontot.

A formázás gyorsítása: stílusok, sablonok

A stílus

A szövegszerkesztők és kiadványszerkesztők világában *kétféle stílust* szoktak definiálni.

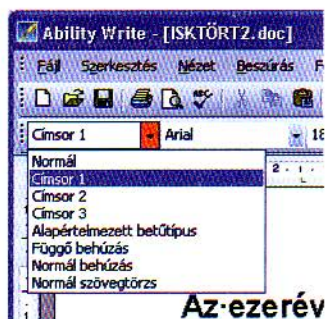
A **bekezdésstílus** vagy egyszerűen **stílus*** egy adott bekezdésre vonatkozó beállítások összessége, melynek nevet adunk és elmentünk. Például, ha egy hosszabb dokumentumot készítünk, akkor elég egyszer megformázni egy címet. Ezt a formátumot elmentjük pl. *főcím* néven, a későbbiekben egy másik cím esetén már nem kell minden beállítást előről kezdenünk, hanem elég azt is *főcím* stílusúra beállítani.

Stílus minden bekezdésnek van. Az alapértelmezett stílus általában a *normál*, míg az ugyancsak előre definiált *címsor1*, *címsor2* ... stílusokat a címek megformázására használják. Fontos speciális stílus a *Formázás törlése*, amely nevének megfelelően valamennyi beállítást töröl, így „tisztá lappal” kezdetjük a formátumok módosítását.

A **karakterstílus** a karakterek formai jegyeit tartalmazza (ne tévesszük össze a félkövér, dőlt stb. betűstílusokkal). Nem minden szövegszerkesztő program ismeri. Szerepe a kiemelésekben van. Egy tankönyvben például a tudósok nevét célszerű azonos módon formázni.

Természetesen a felhasználó hozhat létre *új stílusokat* is, de akár módosíthatja a meglévőket is. A stílusok igazi erőssége az, hogy *ha egy stílust módosítunk, akkor az összes adott stílusú bekezdés ennek megfelelően változik meg*.

Új stílus létrehozásakor fontos beállítás, hogy belekerül-e a sablonba. Ha ezt is előírjuk, akkor a beállítások nemcsak az adott dokumentumra, hanem a később létrehozandó új dokumentumokra is érvényesek lehetnek.



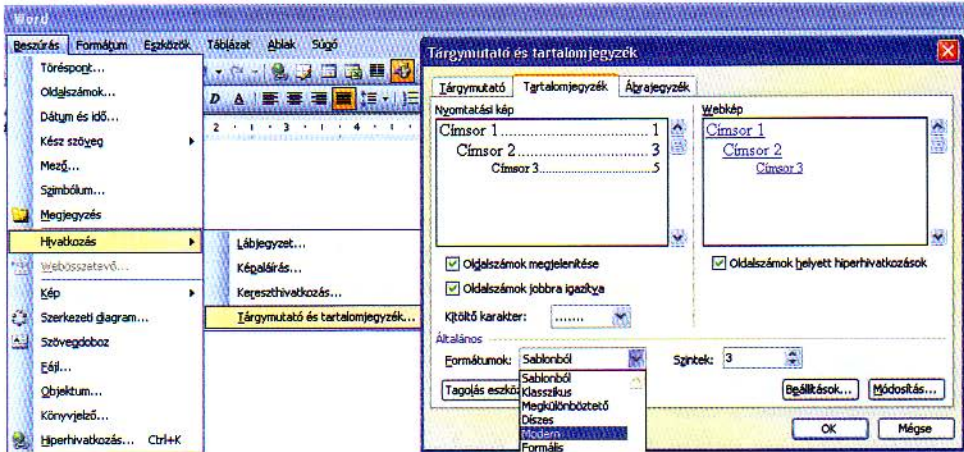
A bekezdés stílusát megváltoztathatjuk az eszköztáron is (Ability Write). Mit jelent a *normál* és a *címsor 1* stílus?



A stílusok módosítását a **Formátum** menüben kezdeményezhetjük (Ability Write).

A tartalomjegyzék

Ha egy dokumentumot címsor stílusokkal formázunk meg, akkor a **szövegszerkesztő programok a címsor stílusok alapján képesek automatikusan elkészíteni a tartalomjegyzéket is**. Ehhez csupán arra kell vigyáznunk, hogy a legmagasabb rendű cím a *címsor1*, a következő a *címsor2* stb. legyen.

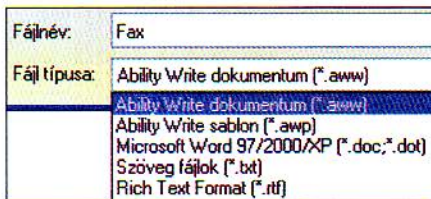


A **Beszűrés** menü **Hivatkozás** menüpontjában kezdeményezhetjük a tartalomjegyzék létrehozását (Microsoft Word). Mely bekezdésekből készíthető el automatikusan a tartalomjegyzék?

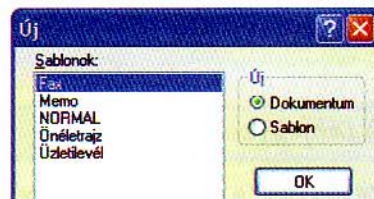
Sablonok

Gyakran előfordul, hogy olyan dokumentumot kell készítenünk, amelynek formai jellemzőit már mások kialakították. Így például egy *önéletrajz* esetében sokszor nemcsak azt szokták elvárni, hogy mely adatok szerepeljenek benne, hanem azt is, hogy milyen módon tagolódjon. Hasonló módon egy cég esetében a *céges dokumentumok megjelenésére* egységes fejlécut, betűtípust stb. szoktak előírni.

A **sablonok* a dokumentumokra vonatkozó formai előírásokat tartalmazzák**, így tartalmazhatnak szövegeket, ábrákat, előre definiált stílusokat, kitöltendő rubrikákat stb.



A dokumentum mentése sablonként (Ability Write)



A használandó sablon megadása új dokumentum készítésénél (Ability Write)

Szövegszerkesztés

Minden dokumentum valamilyen sablon alapján készül. Ha mást nem adunk meg, akkor az új dokumentum egy alapértelmezett sablont használ, ami többnyire *normál* néven szerepel.

A szövegszerkesztők már eleve több sablont is tartalmaznak, de mi ezeket tovább *módosíthatjuk*, vagy magunk is *készíthetünk újakat*. Új sablon készítése a legtöbb esetben igen egyszerű: elkészítünk egy dokumentumot, amelyben minden előírást beállítottunk, majd *elmentjük sablonként*.

Ha egy új dokumentumot egy adott sablonra kívánunk alapozni, akkor legegyszerűbb azt az *új dokumentum létrehozásakor kiválasztani*.

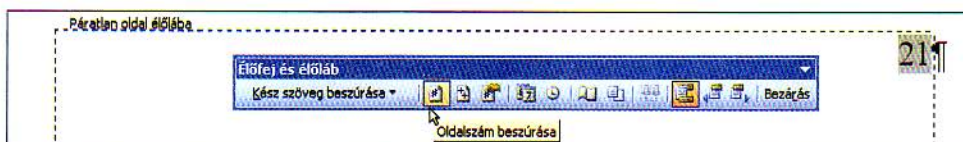
Nagyobb dokumentumok formázása

A nagyobb lélegzetű dokumentumokban (pl. szakdolgozat, hírlevél) újabb problémák jelentkeznek. A dokumentum mérete miatt egyfelől az olvasót plusz információkkal kell ellátnunk (fejléc, lábléc, oldalszámozás, lábjegyzet, tartalomjegyzék), másfelől gyakran célszerű többszörös elrendezést alkalmazni.

Élőfej és élőláb

Élőfejnek* (vagy **fejlécnek**) nevezik a **felső margón**, míg **élőlábnak*** (vagy **láblécnek**) az **alsó margón megjelenő szöveget vagy ábrát**. Élőfejbe vagy élőlábba szokás tenni például az oldalszámot, fejezetcímet vagy pl. telefonkönyvnél a lapon szereplő első és utolsó előfizető nevét.

A szövegszerkesztők esetében az élőfej, ill. élőláb **eltérő lehet az első oldalon vagy a dokumentum páros, illetve páratlan oldalain**. Egy fejezet első oldalán ugyanis nem szokás az élőfejben is megismételni a címet, míg a páros és páratlan oldalak elrendezése általában egymás tükörképe.

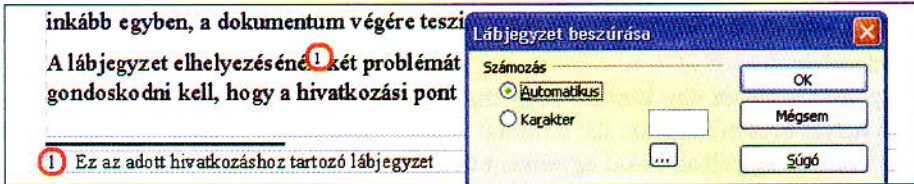


A Microsoft Wordben a **Nézet** menü **Élőfej és élőláb** pontjával formázhatjuk meg a fejléceket és lábléceket. Itt szűrhetjük be az automatikus oldalszámozást is. Azt, hogy páros vagy páratlan oldalakon legyen élőfej illetve élőláb, a **Fájl** menü **Oldalbeállítás** lehetőségével adhatjuk meg.

A lábjegyzet és végjegyzet

A **lábjegyzet*** a dokumentum egy adott pontjához fűzött, az olvasó számára **szolgáló megjegyzés**, amit általában a lap alján helyeznek el. Szakirodalom esetén gyakori, hogy a megjegyzéseket egyben, a dokumentum végére teszik (**végjegyzet**).

A lábjegyzet elhelyezésénél két problémát kell a szövegszerkesztő programnak megoldania. Egyfelől gondoskodnia kell, hogy a hivatkozási pont és a lábjegyzet *egy oldalon maradjon* a szöveg átirása esetén is. Másfelől több hivatkozás esetén *segíteni kell az olvasót* (pl. számozással) a megfelelő lábjegyzet megtalálásában.



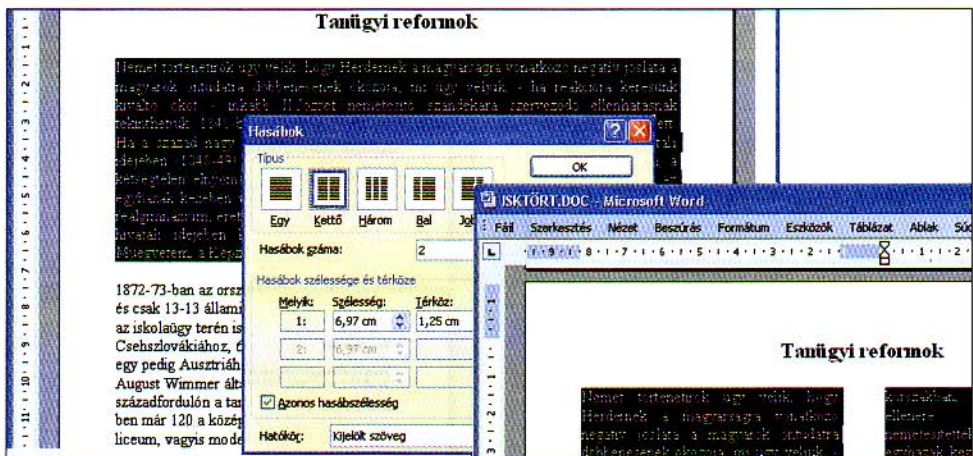
Lábjegyzet a **Beszúrás** menü megfelelő pontjával helyezhető el (OpenOffice.org Writer). Mi történik, ha a lábjegyzet olyan hosszú, hogy nem fér ki a lap aljára? Próbáljuk ki!

Többhasábos rész

Ha egy sorba viszonylag sok betű kerülne (pl. szórólapok, újságok esetén), akkor célszerű az olvashatóságot úgy javítani, hogy a **szöveget több hasábra tördeljük**. Ezt a legtöbb szövegszerkesztőben úgy érhetjük el, hogy a megfelelő szövegrészt kijelöljük, majd a menü vagy eszköztár segítségével megadjuk a hasábok számát.

Ha nem az egész dokumentum, hanem csak *egy belső része lesz többhasábos*, akkor a szövegszerkesztők a dokumentumot három szakaszra bontják. Egyik a többhasábos rész előtti, másik a többhasábos rész, harmadik a többhasábos rész utáni szakasz. Mivel a szakaszokban az oldal jellemzőit: margók, oldalszámok stb. külön-külön is beállíthatjuk, az ilyen dokumentumok kezelése nagy odafigyelést igényel.

Többhasábos dokumentumban a *hasábok közé is tehetünk képeket vagy szövegdobozokat*. Ilyenkor az olvashatóságot javítja, ha ezek legfeljebb egy-egy hasáb felét foglalják el.



Többhasábos szakasz létrehozása. Kijelölés után a hasábok számát a **Formátum** menü **Hasábok** pontjánál adhatjuk meg (Microsoft Word). Milyen más módon hozhatunk létre hasábszerű elrendezést?

Körlevél készítése

A körlevél lényege a következő. Ugyanazt a levelet sok címzettnek, de személyre szólóan szeretnénk elküldeni. A levelek tehát lényegében azonosak, csupán a címzett személyes adataiban és néhány más információban térnek el. Ilyenek például a névre szóló reklámlevelek.

Hagyományosan ez úgy készült, hogy megírták a levélmintát, amelyben a személyes adatok helyét üresen hagyták. Ezt a mintát ezután kellő számú példányban sokszorosították, végül az egyedi adatokat egyenként beírták.

A szövegszerkesztő programok ezt a folyamatot egyszerűbbé teszik, a kész leveleket pedig szebbé varázsolják, mivel a személyes adatokat minden levélbe „belesimítják”.

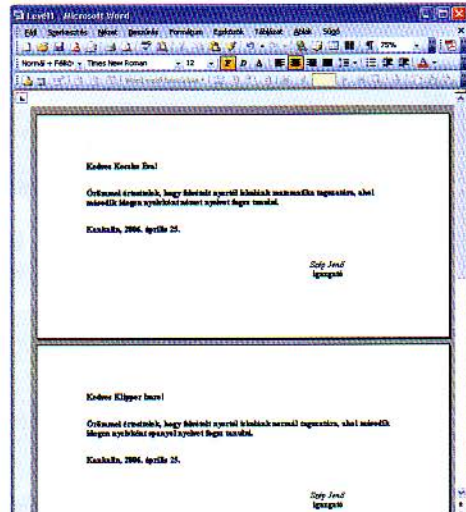
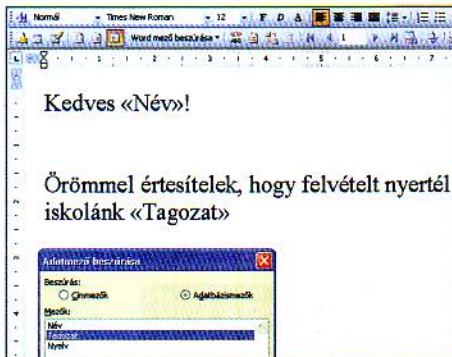
A körlevél* készítése három lépésből áll. Az első az adatok összegyűjtése, vagyis az adatforrás előállítása. A második a törzsdokumentum elkészítése, ez lesz valameny-nyi levél közös formanyomtatványa. A harmadik az adatforrás és a törzsdokumentum egyesítése, ekkor készülnek el az egyes levelek.

Az **adatforrás** legegyszerűbb esetben egy szövegszerkesztővel készült táblázat. Ennek oszlopait **adatmezőknek** hívják, ezek tartalmazzák a beszúrandó adatokat. Az első sorban az adatmezők neve szerepel. (Az adatforrás természetesen készülhet táblázatkezelővel vagy adatbázis-kezelővel is, ezekkel a későbbiekben fogunk találkozni.)

A **törzsdokumentum** a szokásos módon készül. Az egyetlen eltérés az, hogy a szövegszerkesztő megfelelő menüpontjával be kell szűrni az adatmezőket.

A megfelelő menüponttal a két dokumentumot egyesíthetjük. A kész, személyre szóló leveleket az **egyesített dokumentum** tartalmazza, általában külön szakaszokban. Az egyesített dokumentumot a szokott módon nyomtathatjuk ki.

Név	Tagozat	Nyelv
Kecske Éva	matematika	német
Klipper Imre	normál	spanyol
Klumpa Helga	angol	orosz



Körlevél készítésének lépései: 1. Az adatforrás 2. A törzsdokumentum az adatmezők beszúrásával 3. Az egyesített dokumentum. Készítsük el az ábrán látható körlevelet!

Összefoglaló kérdések, feladatok

1. Miben azonosak és miben térnek el a szövegszerkesztő programok a kiadványszerkesztőktől, a prezentációkészítőktől, a webszerkesztőktől és az editoroktól?

2. Soroljuk fel és mutassuk be egy-egy példán a szövegszerkesztő programok grafikus felületének elemeit!

3. Hogyan kell javítani a következő gépelési hibákat: kimaradt betű, véletlenül lenyomott Enter gomb, csupa nagybetűs írás? Több lehetőséget is adjunk meg!

4. Milyen eszközöket kínálnak a szövegszerkesztők a szöveg kijelölésére?

5. Milyen részekre tagolódik egy dokumentum? Miben tér el ez a nyomdai felosztástól?

6. Milyen formai jellemzői vannak egy betűnek? Hogyan csoportosíthatjuk a betűtípusokat?

7. Soroljuk fel a szövegszerkesztésben használt különleges karaktereket, és felhasználási területüket!

8. Milyen bekezdésformátumokat ismerünk? Ne csak az igazítás módjait adjuk meg!

9. Készítsünk szavazócédulát az iskolai diákönkormányzat megválasztásához!

10. Mire kell ügyelni az írásjelek, idézőjelek, gondolatjelek és a zárójelek használatánál? Hogyan kell beírni számokat, képleteket és fizikai mennyiségeket?

11. Készítsünk tájékoztató táblát az igazgatói iroda ajtajára! Mentsük el a dokumentumot legalább háromféle formátumban!

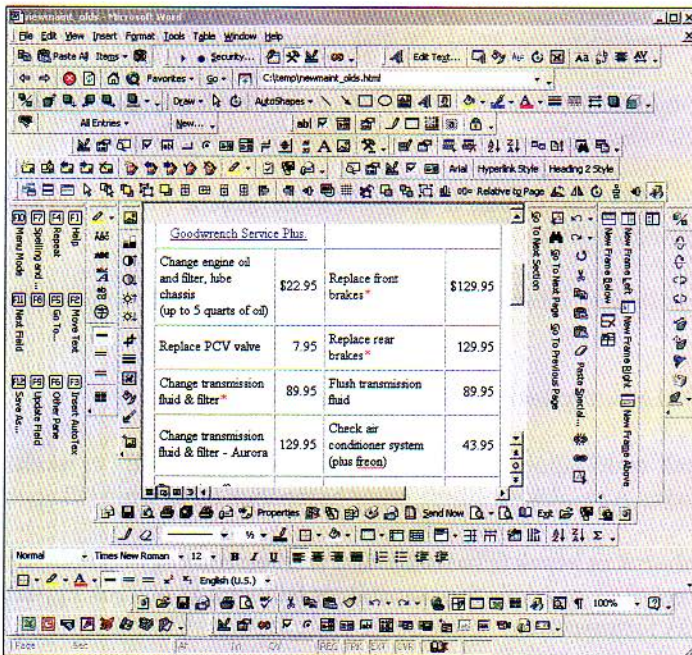
12. Soroljuk fel, milyen lehetőségek vannak a margóviszonyok, a tájolás és a lapméret megadására!

13. Készítsük el a Gittegyulet díszes alapító okiratát!

14. Tervezzünk emléklapot a „Ki mit tud az Európai Unióról?” c. iskolai vetélkedő résztvevői számára! Az emléklap készítésekor használjuk fel az Unió jelképeit!

15. Szövegdobozok és képek felhasználásával szerkesszünk egy B5 méretű szórólapot, melyben népszerűsítjük az iskola focicsapatát, és felhívjuk az olvasók figyelmét az április 1-ji tanár-diák focimeccsre!

16. Soroljuk fel a képek és szövegdoxok formázásának lehetőségeit!
17. Írjuk be és rendezzük névsorba az osztályunkba járó tanulók nevét!
18. Hasonlítsuk össze a tabulátorok és táblázatok létrehozásának, felhasználásának és formázásának lehetőségeit!
19. Készítsük el a büfé árjegyzékét tabulátorok felhasználásával!
20. Készítsük el órarendünket táblázat segítségével!
21. Mi a stílus és a sablon? Hogyan kell tartalomjegyzéket készíteni?
22. Készítsünk önéletrajzt a szövegszerkesztő program sablonjának segítségével!
23. Tervezzünk cégjelzéses sablont iskolánk részére! Írjunk segítségével egy hivatalos értesítést, melyben tájékoztatjuk a szülőket a következő szülői értekezlet időpontjáról!
24. Foglaljuk össze írásban, mit jelentenek a következő fogalmak: élőfej, élőláb, láb-jegyzet, többhasábos szakasz! A dokumentumon legyen élőfej, élőláb, oldalszám, láb-jegyzet, a többhasábos szakaszcól szóló rész pedig legyen kéthasábos!
25. A következő ábra egy szövegszerkesztő eszköztárait tartalmazza. Hány funkciót tudunk beazonosítani?



Dokumentumkészítés számítógéppel

Prezentáció

A multimédia-prezentáció

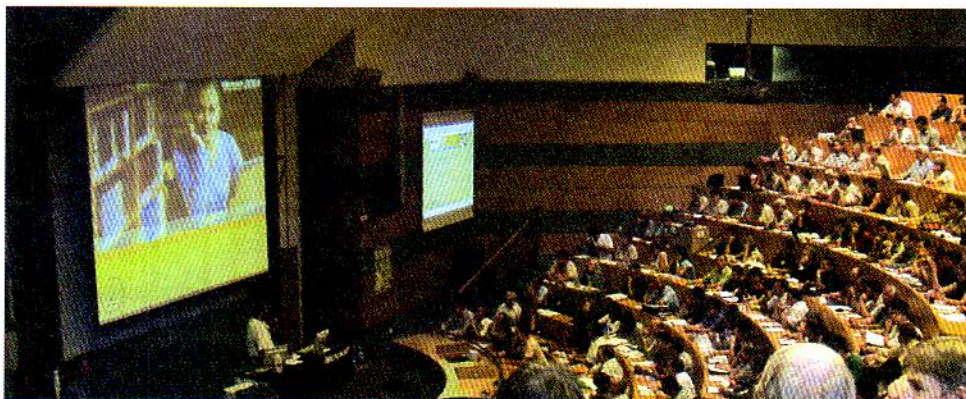
Ha szeretnénk valamit mások számára bemutatni, érdemes azt képekkel, ábrákkal esetleg hangeffektusokkal és animációkkal színesíteni, érdekesebbé tenni. Erre a célra ma egyre inkább számítógéppel készített prezentációt használnak, amely ötvözi a rendelkezésre álló multimédiás anyagokat.

A prezentáció felhasználásának területei

A prezentációt alapvetően két területen használják: reklámanyagok vetítésére, illetve előadások segédanyagainak bemutatására.

Előadások kísérőanyagai. Ha kisebb-nagyobb közönség előtt előadást tartunk, előadásunk vázlatát, a szükséges képeket, táblázatokat stb. célszerű a hallgatósággal megosztani. Ebben az esetben a **prezentáció*** valamennyi segédanyagot egyetlen egységben tartalmazhatja. Így **elegendő a számítógép és a hozzá kapcsolt kivetítő** vagy monitorháló, nincs szükség táblára, írásvetítőre, színes falitáblákra stb. További előnye, hogy az adatok könnyen és egyszerűen aktualizálhatók.

Reklámanyagok. Valamely termék vagy szolgáltatás bemutatása a prezentáció multimédiás eszközeivel egyszerűen kivitelezhető, és látványos reklámforrás lehet.



Előadás közben a prezentációt a terem tetején lévő projektorral vetítik az ernyőre. Kisebb terem esetén egy monitorháló is elegendő lenne.

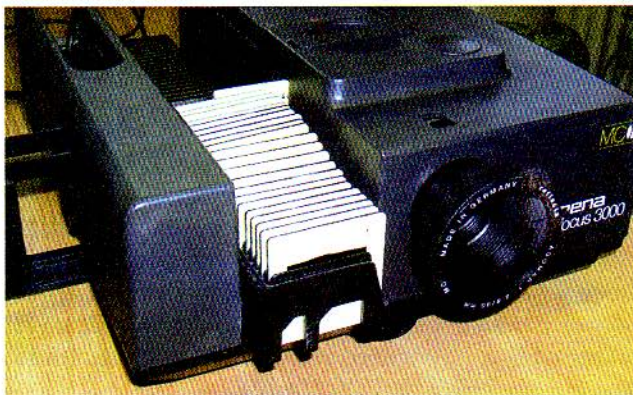
Prezentáció

Ebben az esetben „csupán” gondoskodni kell a számítógép és a kivetítő (esetleg nagyképernyős tévékészülék) megfelelő elhelyezéséről. A prezentáció vetítése előre beállított időzítéssel automatikusan zajlik, illetve ismétlődik.

A prezentáció felépítése

Korábban a bemutatók úgy készültek, hogy a *felhasznált képeket, táblázatokat és egyéb anyagokat diakockákra rögzítették, és ezt diavetítővel kivetítették*. A diakockák közötti váltást kezdeményezhette az előadó, de megoldható volt az időzítés is. (A diavetítő helyett ma már inkább *írásvetítőt* használnak. Ekkor a felhasznált anyagokat fóliára rögzítik, ez történhet pl. tintasugaras vagy lézernyomtató segítségével is.)

A számítógépes prezentációkészítésben mind a mai napig a diasornál megszokott fogalmakat használjuk: a **prezentáció** vagy magyar nevén a **bemutató*** tulajdonképpen **diakockák* sorozata**.



Diavetítő a diakockákkal. Miért kényelmesebb a prezentációkészítő program használata?

A diakockák elsősorban felsorolás jellegű szövegeket, ábrákat, képeket stb. tartalmaznak, egységes megjelenésüket pedig sablonok segítségével valósítják meg. **A diakockák elkészítése emiatt hasonló a szövegszerkesztővel készült dokumentum oldalainak elkészítéséhez.** (Szélsőséges esetben akár szövegszerkesztővel készített dokumentumot is használhatunk bemutatóként.)

A bemutató elkészítésének lépései

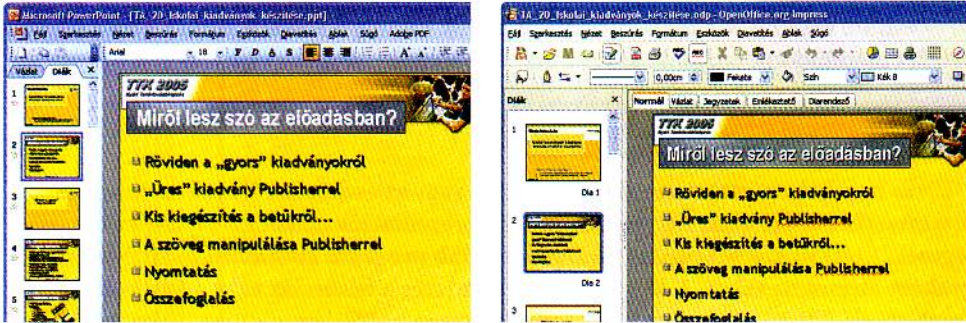
Ha bemutatót szeretnénk készíteni, nagyon **fontos, hogy a szükséges ismereteket még a bemutató elkészítése előtt összegyűjtsük és rendszerezzük.**

Maga a tényleges elkészítés három lépésből áll. Először *elhelyezzük a diákon a szöveg vázlatát* és az összegyűjtött anyagot (képek, diagramok stb.). Ezután *megtervezzük a látványelemeket* (háttér, animáció stb.). Végül *teszteljük a prezentáció megjelenését*, kipróbáljuk és beállítjuk időzítését.

A prezentáció elkészítése nem helyettesítheti a felkészülést, hiszen a bemutató első-sorban a hallgatóság számára készül. *A kész prezentáció birtokában újra át kell gondolni a mondanivalónkat*, nehogy vetítés közben kelljen rögtönöznünk. Ebben segít, hogy a **diákhoz saját jegyzetek is fűzhetők**, amelyek csak az előadó monitorán jelennek meg.

A prezentációkészítő programok

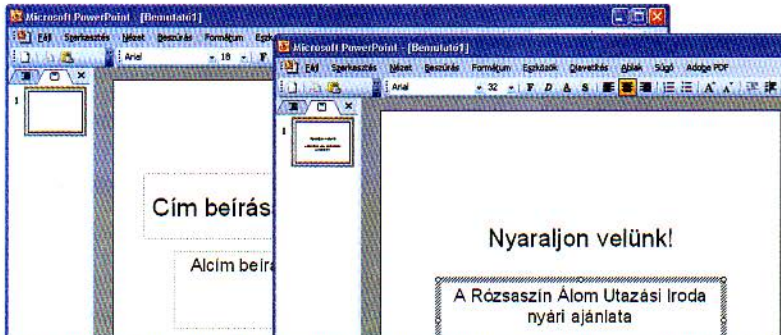
Prezentációkészítő programot a legtöbb irodai alkalmazáscsomag tartalmaz. Ezek felépítése, eszközei sok hasonlóságot mutatnak. Ilyenek pl. a Microsoft PowerPoint [pauerpoint], a Corel Presentations [prezentésnz], az Ability Presentation [prezentésn] vagy az OpenOffice.org Impress [impresz].



A Microsoft PowerPoint és az OpenOffice.org Impress. Hasonlítsuk össze a két program felületét!

A prezentáció létrehozása

Új prezentációt általában kétféleképpen kezdetünk el. Indulhatunk egy üres prezentációból, vagy felhasználhatjuk a program varázslóját, amely végigvezet a diasor készítésén („Előadás-tervező varázsló”, „Bemutatótündér” stb.).



A bemutató első kockája a címdia. Ezt a prezentációkészítő programok automatikusan felajánlják (Microsoft PowerPoint). Készítsük el a diát prezentációnk címdijaként!

Prezentáció

A látványelemek megtervezését általában akkor is **sablonok** segítik, ha üres bemutatóból indulunk.

Ha egy új prezentációt készítünk, akkor **az első kockán a bemutató címét, valamint alcímét vagy készítőjének nevét kell megadnunk**. Ezért a prezentációkészítők általában automatikusan felkínálják első diaként a **címdiát**.

Szöveg formázása

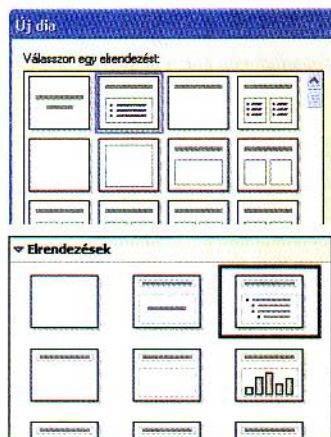
A prezentációkészítők a **beírt szöveg formázására lényegében ugyanazokat a betű- és bekezdésformátumokat kínálják fel, mint a szövegszerkesztők**.

Természetesen előfordulhat, hogy egy-egy lehetőséget nem ott találunk meg, ahol a szövegszerkesztőben. Például a Microsoft Wordben a bekezdések igazítását és a térközt a **Formátum** menü **Bekezdés** pontjánál állíthatjuk be, míg a PowerPointban erre a menüben két menüpontot találunk: az **Igazítást** és a **Sorközt**.

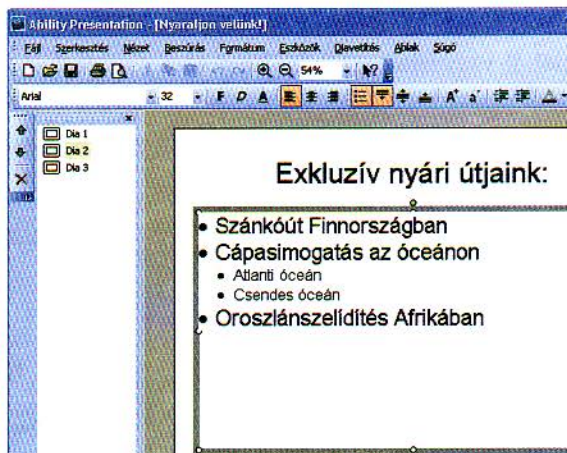
Új dia beszúrása, felsorolás

Új dia beszúrását értelemszerűen a menüből vagy az eszköztárról kezdeményezhetjük. A szoftverkészítők hamar észrevették, hogy a prezentációk diakockái meglehetősen egységesek. Ezért **új dia beszúrása esetén a legtöbb program automatikusan felajánlja a gyakori elrendezéseket**. A leggyakoribb elrendezés a felsorolás vagy a felsorolás és valamilyen objektum, többnyire kép. (Természetesen beszúrhatunk üres diakockát is, ekkor azonban a szövegdobozokat és a képeket nekünk kell feltennünk és elrendeznünk.)

A **második dián általában a bemutató vázlatát szokás feltüntetni**, ezért ez a kocka többnyire egy címet és egy felsorolást tartalmaz. A bemutatóknál igen **gyakori a többszintű felsorolás**. A szintek közötti váltás itt is a szövegszerkesztésben megszokott módon történik.



Az elrendezés kiválasztása
(felül Ability, alul OpenOffice.org)

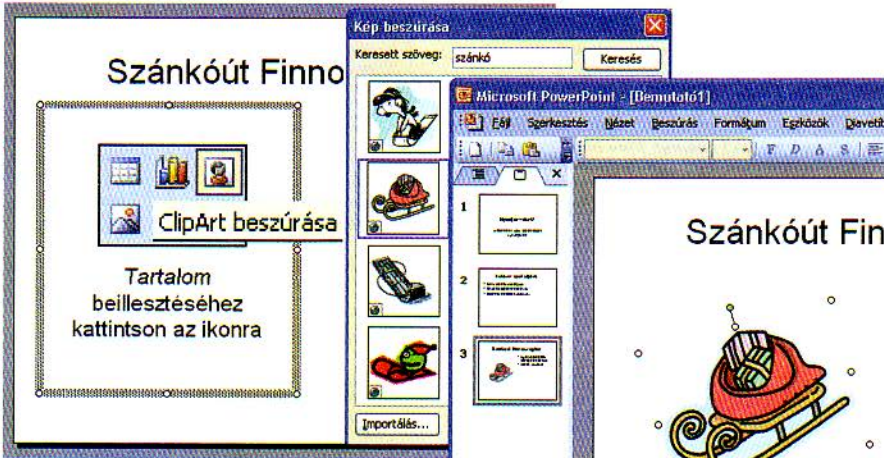


A második dia általában a bemutató vázlatát tartalmazó felsorolás (Ability Presentation). Készítsük el az ábra szerint!

Objektum beszúrása

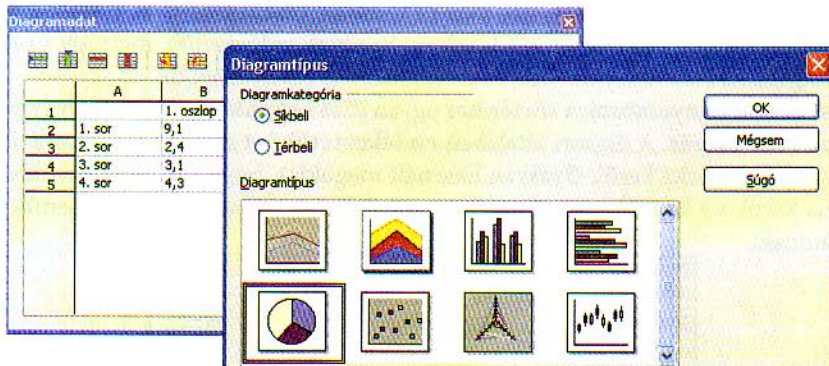
A bemutatók esetében lényegesen nagyobb szerepet kapnak a beszúrható objektumok: a képek, az ábrák, a táblázatok, a diagramok stb.

A **képek és ábrák beszúrása**, illetve a **táblázatok beszúrása** és megformázása a szövegszerkesztésben tanultaknak megfelelően történik.



Ábrát legegyszerűbben a gyűjteményből (pl. ClipArt [klipárt]) szűrhatunk be. A beszúrt ábrát egérrel mozgathatjuk, a nyolc méretező ponttal átméretezhetjük, a zöld ponttal pedig forgathatjuk (Microsoft PowerPoint). Készítsük el a diát prezentációnk harmadik kockájaként!

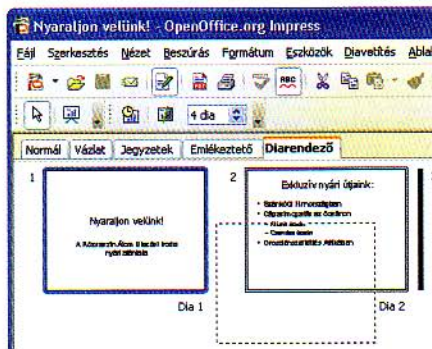
Egy **diagram beszúrása** két lépésből áll. Először egy táblázatban meg kell adnunk a diagramon megjelenő adatokat, majd ezek alapján készül el az ábra. A diagram tényleges elkészítése a táblázatkezelőknél megszokott módon történik.



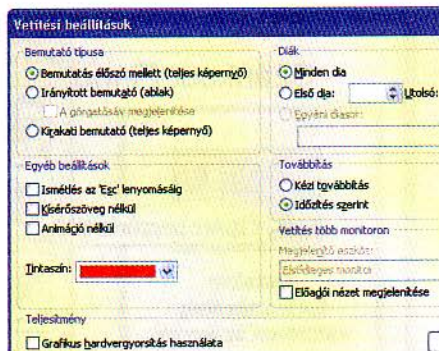
A diagram készítésének lépései: az adatok megadása és a diagramtípus kiválasztása (OpenOffice.org Impress).

A bemutató rendezése és vetítése

A prezentációkészítő programok a diákat többféle elrendezésben tudják megjeleníteni. Tipikusan rendelkeznek egy *szerkesztő* (vagy normál) nézettel, amelyben megtervezhetjük a diakockákat, egy *diarendező* nézettel, ahol a diakockák sorrendjét módosíthatjuk, és egy *vetítő* nézettel a bemutató megtekintéséhez.



Diarendező nézetben egérrel húzhatjuk át például az első diát a második mögé (OpenOffice.org Impress).



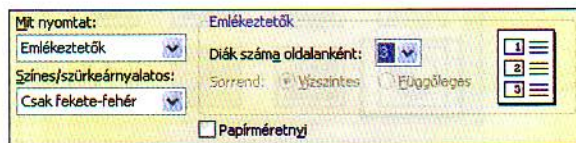
A vetítési beállításokat a *Diavetítés* menüben adhatjuk meg (Microsoft PowerPoint).

A bemutató mentése és nyomtatása

A prezentációt a szokásos módon menthetjük el vagy nyithatjuk meg szerkesztés közben. Érdekes lehetőség, hogy néhány prezentációkészítő a szükséges fájlokból egy *automatikusan induló*, jelszóval védhető CD-t is tud készíteni (pl. PowerPoint).

Jól használható szolgáltatás, hogy a **diákat egyenként, képformátumban is elmenthetjük** (pl. JPG vagy GIF). Mivel a legtöbb prezentációkészítő program tartalmaz vektorgrafikus rajzoló eszköztárat, ezt más célú ábrák készítésére is ki tudjuk használni. Ugyancsak hasznos lehetőség a *HTML-formátumban való mentés*, mert így a prezentációt böngészővel is megnyithatjuk, és a világhálón is elhelyezhetjük.

A **prezentáció nyomtatása történhet ugyan diakockánként is**, azonban ez papír- és festékpazarló eljárás. A diasort általában **emlékeztetőként** szokták nyomtatni, ilyenkor egy lapra több kocka kerül. Gyakran használt megoldás, hogy egy lapra egymás alá három dia kerül, s a kockák mellé a szoftver vonalakat nyomtat, ahová a hallgatók jegyzetelni tudnak.



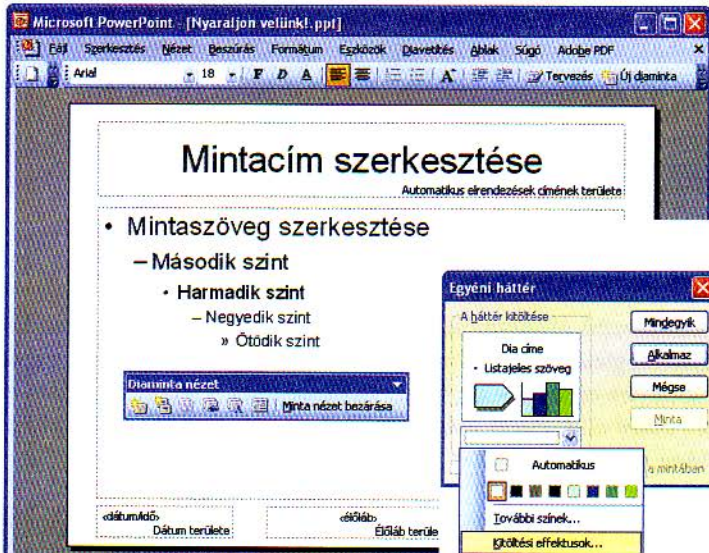
Nyomatáskor célszerű emlékeztetőként kinyomtatni a prezentációt (Microsoft PowerPoint). Mire használják általában ezt az elrendezést?

A prezentáció látványelemei

Egységes megjelenés: diaminta készítése

A bemutatót célszerű egységesre formázni, különben komolytalannak tűnik. Az egységes formátumot magunk is megtervezhetjük, de alkalmazhatunk kész sablonokat is.

Egységes formátumot az adott bemutatóra egy *diaminta** segítségével készíthetünk. Ezen beállítjuk a cím, felsorolás, háttér stb. formátumát. A prezentáció diakockái pedig ezeket a beállításokat automatikusan felveszik.



*A diaminta a **Nézet** menü **Minta** menüpontjából érhető el. Segítségével egységesre formázhatjuk a bemutatót (Microsoft PowerPoint). Mire kell ügyelnünk a betűk típusának és színének megválasztásánál?*

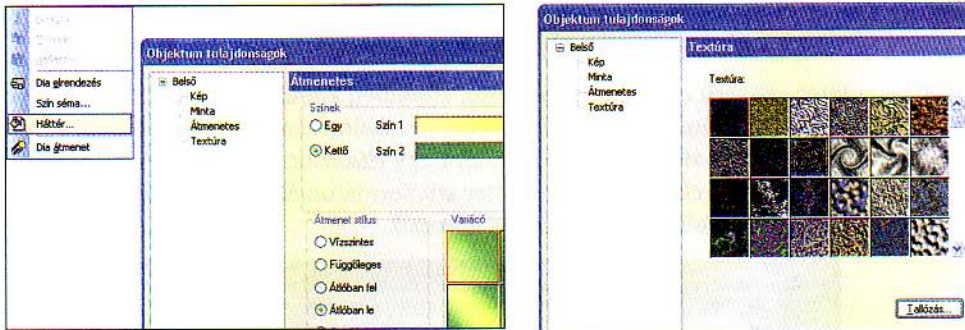
A betűk típusának megadásánál **célszerű a talp nélküli vastagabb betűket választani**, ezek ugyanis a viszonylag kisebb felbontású projektorok esetében is jól olvashatóak (pl. Arial, Helvetica [helvetika]). Ugyancsak fontos, hogy **a betűk mérete ne legyen 16-20 pontnál kisebb**, különben távolról nem olvashatóak.

Természetesen a betűk színének összhangban kell lennie a háttér színével is: világos háttérhez inkább sötét betűt célszerű alkalmazni, és fordítva. Ahogy az öltözködésnél, úgy **a bemutatóknál sem célszerű 3-4 domináns színnél többet használni** egy dián. A színek megválasztásánál érdemes mások tanácsát is kikérni, bizonyos színekombinációk ugyanis a legtöbb ember számára kellemetlenek.

A háttér megadására a prezentációkészítők sokféle lehetőséget kínálnak fel. Ilyenek például a színátmenet megadása, adott mintázat kiválasztása vagy saját kép beszúrása.

Prezentáció

A háttér kiválasztásánál figyelembe kell venni, hogy a **háttér csupán színesíti a bemutatót**. Soha ne válasszunk túlságosan kontrasztos, részletdús megoldást, mert az jelentősen ronthatja a szöveg olvashatóságát.



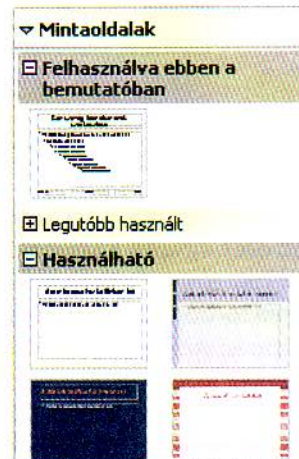
Egyéni háttérrel a helyi menü **Háttér** vagy **Egyéni háttér** pontjával állíthatunk be. Általában lehetőségünk van színátmenet, anyagminta, kép stb. megadására is (Ability Presentation). Mire kell ügyelnünk az egyéni háttér kialakításánál?

Sablonok használata

A prezentációkészítő programok sok olyan sablont bocsátanak a rendelkezésünkre, melyet a gyártó megfelelő szakemberekkel készíttetett, azok vizuális megjelenését tesztelte. Ezért többnyire célszerű ezek közül választanunk. A legtöbb esetben a **kész tervezősablonokat előre elkészített színsémákkal tovább finomíthatjuk**, illetve ízlésünknek megfelelően módosíthatjuk.



Sablonok használata: a tervezősablonról színsémákkal finomíthatjuk (Microsoft PowerPoint).



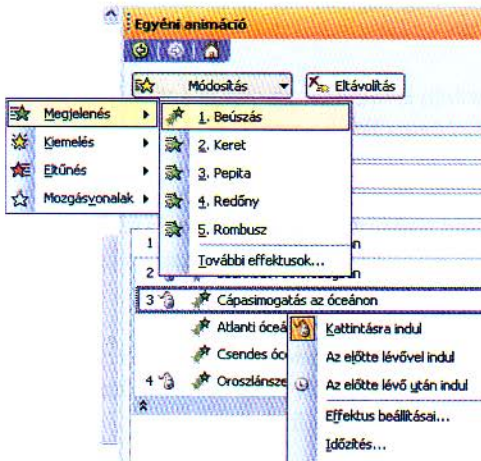
Mintaoldal kiválasztása (Openoffice.org Impress)

Egyéni animáció

A diavetítés egyik leglátványosabb eleme az animáció. Az **animáció*** speciális kép- vagy hanghatás hozzáadása az objektumhoz. Például bizonyos szövegek vagy képek látványosan beúsznak, kiemelkednek, eltűnnek, és ezt esetleg hangeffektusok kíséretében teszik. Az animációk beállítására általában igen sok lehetőség áll a rendelkezésünkre: megadhatjuk az effektusok sorrendjét, sebességét, időzíthetjük őket. Megadhatjuk az összetartozó elemeket is, pl. a bekezdést animálhatjuk egyben, szavanként vagy betűnként.

Az animációk tervezésénél is alapszabály azonban, hogy **a különleges effektusokat csak a hangsúlyos részek kiemelésére** szabad alkalmaznunk. Egyébként elégedjünk meg valamilyen kevésbé látványos megoldással, pl. a sorok beúszásával.

Az animációt általában *a diamintán is beállíthatjuk*, így jelentős munkát takaríthatunk meg.



Egyéni animáció készítésének két lépése. Kiválasztjuk az animációt, majd beállítjuk az időzítését (Microsoft PowerPoint). Próbáljuk ki az általunk használt szoftver minél több animációs lehetőségét!



A diaváltás időzítésének, illetve a diaátmenet (áttűnés) módjának megadása (Openoffice.org Impress). Mikor célszerű beállítani a diák automatikus továbbítását?

Áttűnés, időzítés

Nemcsak a dián lévő elemeket, hanem a **diakockák közötti váltást is animálhatjuk, ez az áttűnés* vagy diaátmenet**. Például megadhatjuk, hogy a vetítés során fokozatosan ússzanak be egymás után vagy menjenek át egymásba a diakockák.

Az áttűnés beállításánál különösen fontos paraméter az **időzítés**. Míg egy előadás során a diaváltást általában az előadó kezdeményezi, addig a reklámcélú prezentációk esetén gyakoribb az **automatikus lejátszás**. Ebben az esetben többnyire perc, másodperc pontossággal adhatjuk meg a diakockák váltását.

Összefoglaló kérdések, feladatok

1. Melyek a multimédia-prezentáció főbb felhasználási területei? Milyen hardvereszközök szükségesek a kivitelezéshez?
2. Ha nem áll rendelkezésünkre prezentációkészítő szoftver, milyen számítógépes és számítógép nélküli megoldásokkal valósíthatjuk meg a bemutatót?
3. Milyen lépésekből áll a prezentáció elkészítése? Mondjunk minél több lehetőséget az anyaggyűjtés formáira!
4. Nézzük végig, hogy milyen elrendezéseket kínál fel az általunk használt prezentációkészítő új dia beszúrása esetén! Nyissunk meg egy teljesen üres diát, és valósítsunk meg rajta egy cím-felsorolás-kép elrendezést!
5. Készítsünk prezentációt, amely népszerűsíti iskolánkat az oda jelentkező tanulók és szülei körében! Alkalmazzunk sablonokat! Mentsük el a prezentációt HTML-formátumban, és a diákat képként is!
6. Tervezzünk két saját diamintát: az egyiket egy tudományos előadáshoz akadémikusok részére, a másikat pedig használt tárgyak reklámozásához az ócskapiac kivetítőjére! Melyik esetben milyen látványelemeket érdemes használni?
7. Készítsünk bemutatót az iskolai menza népszerűsítésére a diákok körében! A prezentáció térjen ki a heti menüre, a kedvezményes árakra és az extra szolgáltatásokra is! Formailag tartalmazzon felsorolást, képet, animációt és áttűnést!
8. Készítsünk prezentációt „Hogyan működik az EU?” címmel az Európai Unió főbb szervezeteinek (Európai Parlament, Európai Unió Tanácsa, Európai Bizottság stb.) bemutatására! A prezentáció színvilága harmonizáljon az uniós zászló színvilágával!
9. Készítsük el az általunk használt prezentációkészítő szoftver grafikus lehetőségeivel *Az analóg jel és a mintavételezéssel kapott digitalizált jel* ábráit *Az informatika alapjai c.* fejezetből!
10. Készítsünk ábrákat a Pitagorasz-tétel bizonyításához! Az ábrák segítségével tervezünk kiselőadást matematikaórára! Vetítés közben használjuk ki az animációs lehetőségeket is a tétel bizonyítása során!

Dokumentumkészítés számítógéppel

Webszerkesztés

A webszerkesztés módjai

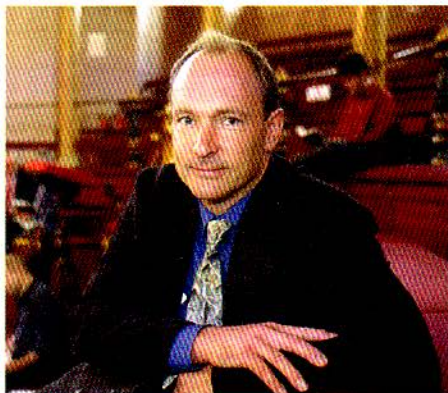
A szélessávú internet terjedésével ma már technikailag és anyagilag is lehetővé vált, hogy a világhálón való keresés mellett akár saját webhelyet is készítsünk. Ebben a fejezetben a kezdő lépéseket tesszük meg ezen az úton.

A World Wide Web kialakulása

1969-ben az IBM felkérésére CHARLES GOLDFARB [csárlyz goldfárb] és két munkatársa ED MOSHER [moser] és RAY LORIE [réj lori] kidolgoztak egy olyan szabványt, amely lehetővé teszi a szöveges dokumentumok belső szerkezetének szoftverfüggetlen leírását. (Ilyen szerkezeti elemek a fejezetek, bekezdések, lábjegyzetek stb.) A nyelvet eredetileg kezdőbetűikből rövidítették GML-nek, később azonban kerestek egy hozzá illő nevet (*Generalized Markup Language* [dzenerelájzd márkáp lengvidzs]). A nyelv továbbfejlesztett változatát ma SGML néven ismerik.

Az SGML-t használta fel 1989-ben a svájci CERN fizikai kutatóintézet két munkatársa TIM BERNERS-LEE [börnerz lí] és ANDERS BERGLUND [anderz berglánd] az intézet dokumentumainak egységes kezelésére. Ők vezették be, hogy az elektronikusan tárolt lapok (a mai fogalmaink szerint) linkekkel hivatkozhatnak egymásra. A kapott leíró nyelvet HTML-nek (*HyperText Markup Language* [hiperteksztt márkáp lengvidzs]) nevezték el. Tim Berners-Lee 1990-ben elkészítette az első webszervert és böngészőprogramot, de tőle származik a WWW (*World Wide Web* [vörlđ vájđ veb]) elnevezés is.

Az SGML az alapja az adatok szoftverfüggetlen leírására létrehozott nemzetközi szabványnak, az XML-nek is (*eXtensible Markup Language* [iksztenziabl márkáp lengvidzs]). Az XML lehetővé teszi, hogy a különböző gyártók különböző szoftverei kölcsönösen „megértsék” egymás adatfájljait.



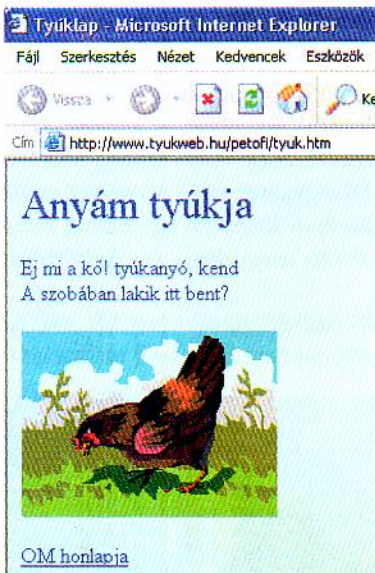
Tim Berners-Lee, aki elkészítette az első webszervert és böngészőt. Erdemei elismeréséül 2004-ben Erzsébet királynő lovaggá ütötte.

A weblapok elhelyezkedése

Mint láttuk, a HTML-kódú fájlok a webszerveren egy mappastruktúrában helyezkednek el. A weblapokon lévő képek, hangok, filmek stb. külön fájlokban vannak, melyekre maga a HTML-fájl csupán egy-egy hivatkozást tartalmaz.

Böngészéskor a webszerver ezt a fájlt az interneten át elküldi számítógépünkre, ahol a böngészőprogram a fájlban lévő formázási utasítások alapján építi fel a képernyőn megjelenő oldalt. Elképzelhető, hogy más böngészőt alkalmazva az oldal megjelenése is részben eltérő lesz.

Az ASP, ASPX, PHP és egyéb kiterjesztések arra utalnak, hogy a webszerver a HTML-kódot *dinamikusan* állítja elő. Például, ha az interneten keresztül vásárolunk, akkor a webszerver a HTML-kódot a folyamatosan változó adatok aktuális értékei alapján generálja (mi van a kosárban, mennyi azok összértéke, stb.). Az ilyen lapok készítéséhez már programozási ismeretek is szükségesek.



```
<html>

<head>
  <title>Tyúklap</title>
</head>

<body bgcolor="lightcyan"
  text="blue">
  <p><font size="6"> Anyám
  tyúkja </font></p>
  <p>Ej mi a kő! tyúkanyó,
  kend<br>
  A szobában lakik itt
  bent?</p>
  <p></p>
  <p><a href="http://www.om.hu">
  OM honlapja </a></p>
</body>

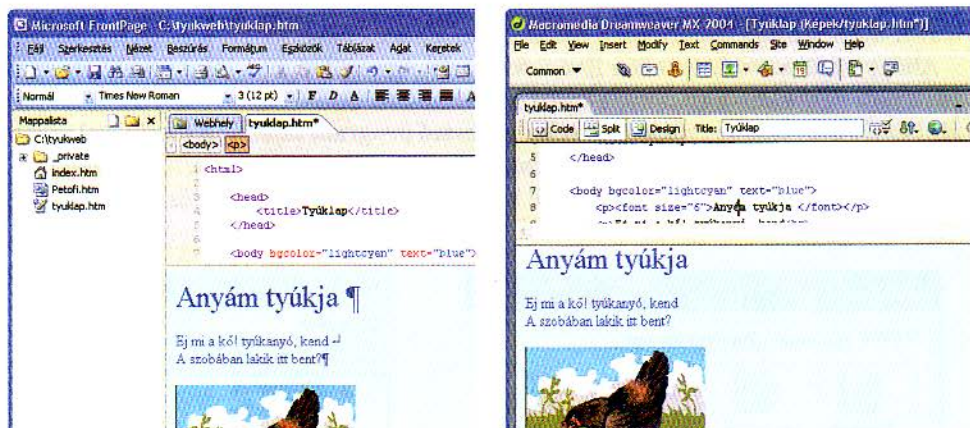
</html>
```

Egy egyszerű weblap és a neki megfelelő HTML fájl.
Elemezzük a HTML-kódot a weblap alapján!

A webszerkesztő programok csoportosítása

A weblapok elkészítésére alapvetően kétféle eljárás kínálkozik. Az **egyik lehetőség, hogy közvetlenül magát a HTML-kódot írjuk be.** Ehhez egyszerűbb esetben elegendő egy editor program, pl. a Jegyzetomb is. Ha azonban szeretnénk munkánkat kényelmesebbé tenni, akkor inkább egy *forráskódszerkesztő* programot érdemes használni, amely segít a kód bevitelében, ellenőrzésében, a hivatkozások tesztelésében stb.

A másik út, hogy grafikus felületen megtervezzük a weblapokat, magát a HTML-kódot pedig ennek megfelelően a **webszerkesztő program állítja elő**. Az ilyen webszerkesztő programokat (a szövegszerkesztőkhöz hasonlóan) **WYSIWYG** [vizivig] típusúaknak szokták nevezni. A professzionális webszerkesztő programok mindkét eljárást támogatják, sőt gyakori az **osztott képernyős megoldás** is, amikor ugyanazt a lapot mindkét módon láthatjuk és szerkeszthetjük.



A Microsoft FrontPage 2003 [frontpédzs] és a Macromedia Dreamweaver MX 2004 [drimvívör] osztott képernyős nézete. Hasonlítsuk össze a két szoftver felületének elemeit!

Weboldal készítése webszerkesztővel

A továbbiakban azt tekintjük át, hogyan lehet WYSIWYG webszerkesztővel egy webhelyet létrehozni és megformázni. Érdemes időnként átváltani forráskód nézetbe, és megnézni, hogy a grafikusan végrehajtott szerkesztési lépésekből hogyan generálódik a HTML-kód.

Új (üres) webhely létrehozása

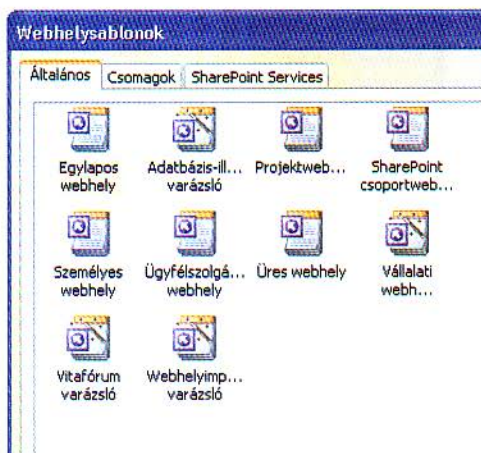
Ha saját webhelyet szeretnénk létrehozni, elsőként *keresnünk kell egy szolgáltatót, amely a szükséges tárhelyet webszerverén biztosítja*. Ez általában könnyen megoldható, ugyanis nagyon sok ingyenes webszolgáltató létezik. Ha olyan internet-hozzáféréseünk van, amely tárhelyet is tartalmaz, akkor érdemes inkább azt választanunk, mert így a technikai problémáinkat segítenek megoldani.

Megfelelő sávszélesség esetén megtehetjük, hogy a webszerkesztő programmal *közvetlenül a szolgáltató által adott tárhelyen szerkesztjük webünket*. Célszerűbb azonban a webhelyet *merevlemezünkön egy saját mappában létrehozni, illetve ott módosítani*. Amikor elkészültünk, a webhelyet a webszerkesztő program megfelelő menüpontjával *„kipublikálhatjuk”*, vagyis elhelyezhetjük a webszerveren. Ilyenkor azok a munkafájlok,

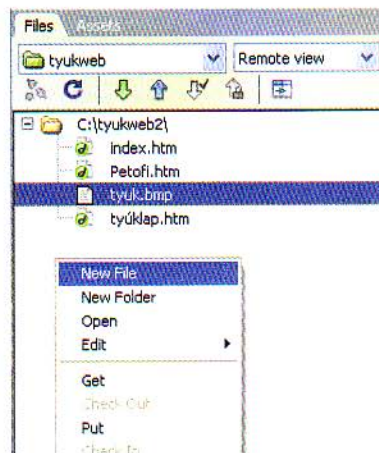
amelyeket a webszerkesztő program saját működéséhez hozott létre, nem kerülnek át a tárhelyre, és azok sem, amelyek nem változtak. A ki publikálás megoldható pl. FTP-programmal is, ekkor azonban tudnunk kell, hogy mely fájlokat kell, illetve nem kell átmásolni a tárhelyre.

A **webhely létrehozását a legtöbb webszerkesztő program varázslókkal vagy sablonokkal segíti**. Ezt a lehetőséget érdemes kihasználni, mert megkönnyíti a kezdeti lépéseket. Ezek az eszközök webszerkesztőnként meglehetősen eltérnek egymástól, azonban az alapfogalmak ismeretében bátran használhatjuk őket.

Ahogy bővül a webhelyünkön lévő információ, úgy szükségünk lehet a webhely mappaszerkezetének módosítására, a fájlok áthelyezésére, átnevezésére stb. Fontos, hogy a **fájlkezelő műveleteket mindig a webszerkesztő program saját fájlkezelőjével végezzük el, mert így a hivatkozásokat a program automatikusan frissíti**. A webszerkesztő program fájlkezelőjét általában úgy kell használnunk, mint az operációs rendszerét, így ez külön tanulást nem igényel.



A webhely létrehozását sablonok segítik, amelyek kész mintákat tartalmaznak (Microsoft FrontPage). Melyik sablon mire használható?



Miért célszerű a fájlkezelő műveleteket a webszerkesztő programmal elvégezni (Macromedia Dreamweaver MX)?

A nyitólap

Amikor böngészünk, akkor a webhely első lapját, a *nyitólapot* nem kell megadnunk. A **nyitólap* neve rögzített, ezt a szolgáltató adja meg**, neve többnyire *index.html*, *index.htm* vagy *default.html* [difólt]. (Saját tárhely igénylésekor feltétlenül meg kell kérdeznünk.) A nyitólap **fizikailag a webet tartalmazó mappaszerkezet gyökerében van**.

Ha a webhelyet webszerkesztő programmal hoztuk létre, akkor a nyitólap általában automatikusan létrejön, és gyakran speciális ikon jelzi. Ha a neve nem megfelelő, a programon belül nevezzük át. Ha nem jött létre, akkor a program saját fájlkezelő parancsaival hozzuk létre a megfelelő néven.

Fontos szempontok a formázáshoz

A weblap elkészítése és formázása sok szempontból hasonló a szövegszerkesztéshez. *Nem tudhatjuk azonban, hogy az, akinek a gépére weblapunk letöltődik, milyen operációs rendszert és milyen böngészőprogramot használ.* Így monitorának felbontása vagy szín-mélysége is jelentősen eltérhet az általunk használttól.

Betű- és bekezdésformázás

A WYSIWYG webszerkesztő programokban a **szöveg beírása, módosítása és formázása a szövegszerkesztő programokéhoz hasonló módon történik.** Általában az eszköztáron vagy a menüben ugyanúgy megtaláljuk az ehhez szükséges ikonokat és parancsokat. Vegyük azonban figyelembe a következőket!

Betűk formázásánál lehetőleg *ne használjunk különleges betűtípusokat*, mert az lehet, hogy nincs a felhasználó gépére telepítve. *A HTML-szabvány szerint a betű méretét csak 7 fokozatban állíthatjuk be.* Ennek ellenére többnyire megadhatjuk a betűk méretét pontokban is, azonban elképzelhető, hogy a böngészőben az mégsem így fog megjelenni.

A betűk színét sok esetben nemcsak kiválasztani tudjuk, hanem megadhatjuk színkóddal vagy a szín (angol) nevével is.

Az alapvető **bekezdésstílusok*** (például címsor1, címsor2, címsor3) a **HTML-nyelvben is léteznek**, így ezeket is nyugodtan használhatjuk. Kezdetben azonban új stílusok létrehozásával vagy meglévő átdefiníálásával ne foglalkozunk.



Betű- és bekezdésformázás a webszerkesztő programban (balra: Microsoft FrontPage, jobbra: Macromedia Dreamweaver MX). Elemezzük a „tyúkanyó” szó betűformátumát!

Kép beillesztése és formázása

A webszerkesztő programokkal a szövegszerkesztő programokhoz hasonlóan szúrhatunk be képeket. Sok webszerkesztő maga is felkínál ábra- és képgyűjteményt a weblap színesítéséhez és formázásához.

A **kép formázásánál*** megadhatjuk:

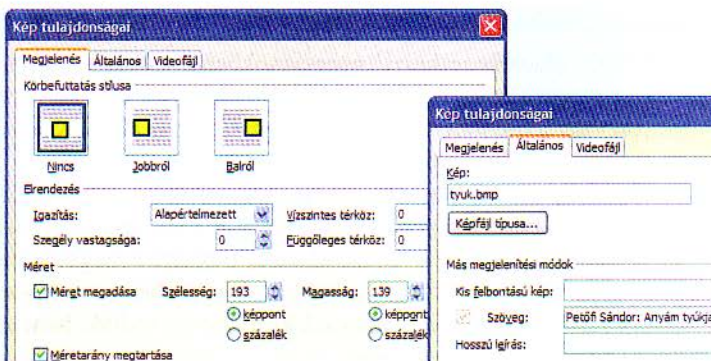
- a körülírtatás módját (körülírtatás balról vagy jobbról, illetve ne legyen),
- a függőleges igazítást (a mellette lévő bekezdés tetejéhez, közepéhez, aljához),
- a kép és szöveg közötti távolságot (pontokban),

- legyen-e a kép határoló szegély, s ha igen, mekkora,
- milyen méretben jelenjen meg a kép (ettől még a fizikai mérete nem változik),
- alternatív szöveg azok részére, akik nem tudják a képet letölteni (kis sávszélesség), vagy megtekinteni. (A látásukban korlátozottak beszélő rendszere képes arra, hogy az így megadott alternatív szöveget felolvassa.)

Ezeket a lehetőségeket a HTML-szabvány tartalmazza. A fentieken túl a webszerkesztő programok gyakran további lehetőségeket is kínálnak a képek vágására, formátumának módosítására, miniatűrre alakítására stb.

Anyám tyúkjá

Ej mi a kő! tyúkanyó, kend
A szobában lakik itt bent?



A kép tulajdonságai megadása (Microsoft FrontPage). Milyen beállításokat olvashatunk le az ábráról a tyúkot ábrázoló képre vonatkozólag?

Hivatkozások beillesztése

A világháló legfontosabb szolgáltatása, hogy a **weblapokat hivatkozásokkal* összekapcsolhatjuk**. A hivatkozás elhelyezése általában két lépésben történik. Először beírjuk azt a szöveget, amit a böngésző meg fog jeleníteni (pl. „OM honlapja”), majd megadjuk, hogy erre kattintva melyik weblap jöjjön be (<http://www.om.hu>).

A weblapra való hivatkozáshoz hasonlóan szűrhatunk be *e-mail címet* is, ekkor a linkre való kattintás a levelezőprogramot indítja el.



Hivatkozás beszúrása. Nemcsak a hivatkozott objektum helyét adhatjuk meg, hanem a Megjelenítési helynél azt is, hogy egy új ablakban nyíljon-e meg (Microsoft FrontPage).

A webhelyen lévő információt célszerű több weblapra bontani, mert a hosszú, sok görgetést igénylő („vécépapír” stílusú) lap kezelése meglehetősen nehézkes. Új weblap egyszerűen beszúrható a webszerkesztő program saját eszközeivel, ezzel azonban még nem ért véget a munkánk. Az új lapot *hivatkozásokkal elérhetővé kell tennünk* a webhely más lapjairól is.

A saját weblapra mutató hivatkozás beszúrása hasonló módon történik, mint a külső weblapé, de legtöbbször a fájl nevét sem kell beírni, hanem elég csupán kiválasztani.

Hivatkozás nemcsak szöveg, hanem más objektum, pl. kép (sőt kép egy részlete) is lehet (*képhivatkozás**). Ekkor a hivatkozott weblap a képre való kattintáskor jelenik meg. Képgalériák esetén gyakori, hogy a weblap csak a miniatűröket tartalmazza. Ilyenkor a kiválasztott kép a kis képre való kattintással egy új ablakban jelenik meg.

Weblap mentése

Mint láttuk, a beillesztett objektumokra, pl. képekre, a HTML-fájl csupán egy hivatkozást tartalmaz, azok a webhely mappaszerkezetében máshol vannak.

Ha a képet nem másoltuk előre a webhely mappaszerkezetébe, hanem csak beszúrtuk egy lapra a webhely készítése közben, akkor a webszerkesztő program rákérdez, hogy a webhelyen hová tegye a képet. Ettől eltekintve a weblap mentése a szokásos módon történik.



A Dreamweaver a kép beillesztésénél, a FrontPage a HTML-fájl mentésekor kérdez rá, hogy hová másolja a webhelyen kívüli objektumokat. Miért kell a képet külön menteni?

A webhely megtekintése

A webhelyet természetesen már szerkesztés közben is érdemes ellenőrizni, amire általában két lehetőségünk van. Egyfelől maga a webszerkesztő program is tartalmaz megtekintési lehetőséget, másfelől megtekinthetjük azt böngészővel is. Ez utóbbi esetben a webhely kezdőlapját kell megnyitnunk, pl. az *index.htm* fájlra való dupla kattintással.

A böngészővel történő ellenőrzésnél két dolgot érdemes megjegyezni. Egyrészt a böngésző csak az elmentett weblapokat tudja megnyitni (tehát a szerkesztés alatt állót ehhez menteni kell). Másfelől a böngésző a megnyitott weblapot nem frissíti automatikusan, ha azt közben módosítjuk. Erről nekünk kell gondoskodnunk, általában a *Nézet* menü *Frissítés* menüpontjával.

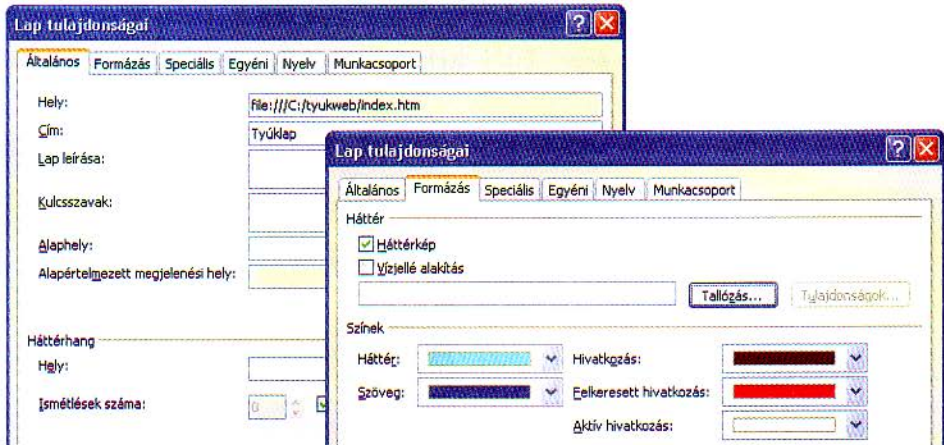
Weblapok formázása

Nemcsak a weblapok tartalma, hanem megjelenési módja is fontos a világhálón. Az egységes formázásra, a látványelemek elhelyezésére nagyon sok módszer alakult ki, alkalmazásuk lassan külön szakmává válik (*webdesign* [webdizájn]).

A weblap címe és háttere

Elegendő weblapunk néhány egyszerű tulajdonságát módosítani ahhoz, hogy máris látványosabban jelenjen meg. Ilyen *a weblap címe*, amely a böngésző címsorában jelenik meg, és ilyen a weblap *háttere* is. Ezek beállításához *a lap tulajdonságait** kell módosítanunk, amit pl. a menüből érhetünk el.

A lap legfontosabb tulajdonsága a weblap *hátterének*, *betűtípusának* és a rajta lévő *hivatkozásoknak* a színe.



A weblap tulajdonságainak megadása (Microsoft FrontPage).
Olvassuk le az ábráról az adott lap beállításait!

Látványos, ha *a háttérben egy képet* helyezünk el. Ilyenkor figyeljünk arra, hogy a kép ne legyen túl kontrasztos, így a szöveg olvasható marad. A kép alapértelmezés szerint *mozaikszerűen* jelenik meg, s így betölti a teljes lap háttérét. Nagyobb kép esetén használhatjuk a *vízjelle alakítás* lehetőséget is, ekkor a lap görgetésekor a háttér helyben marad, vagyis a szöveg tartalma a kép előtt gördül le.

Egy webhelyen célszerű a lapok beállításait egységessé tenni. Erre a legelterjedtebb megoldás a *stíluslapok* készítése. Ezek kész stílusokat tartalmazhatnak nemcsak a bekezdésekre, hanem a többi elemre, például a háttér beállításaira is. Ha egy stílust módosítunk, a webhelyen az összes adott stílusú elem megjelenése ennek megfelelően változik. A stíluslapok készítésével részletesebben nem foglalkozunk.

Táblázat készítése

Mivel a weblap tervezésekor nem tudhatjuk, hogy a böngésző milyen felbontást használ, a lap elrendezésének egyik leghatékonyabb eszköze a **szegélyek nélkül megjelenő táblázat***. Ha a lapon lévő képeket, szöveget és egyéb elemeket egy ilyen táblázatban helyezük el, akkor az elég tág tűréshatárok között az elvárásainknak megfelelően jelenik meg.

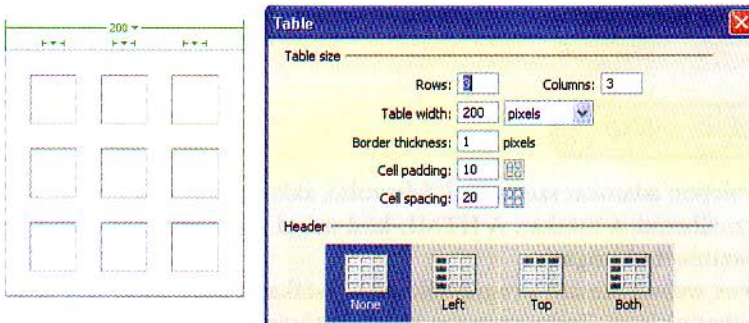
A HTML-kód igen rugalmasan kezeli a táblázatokat. Így például lehetőségünk van *cellák összevonására* (azaz a cellák száma soronként vagy oszloponként eltérő lesz), és arra is, hogy *egy cella újabb táblázatot* tartalmazzon. Igen hatékony, hogy az oszlopok szélességét nemcsak pixeleken, hanem *a megjelenítő ablak százalékában*, tehát a felbontástól függetlenül is definiálhatjuk.

Táblázat segítségével készített elrendezés. A cellahatárokat jelző szaggatott vonalak a böngészőben nem jelennek meg (MS FrontPage).

Készítsük el az ábrán látható táblázatos elrendezésű weblapot az általunk használt webszerkesztő programmal!



A HTML-dokumentumokban a **táblázat cellái önálló dobozként jelennek meg**, ezek távolsága a *cellaköz* (*cell spacing* [szel szpészing]). A cella tartalma és a cella szegélye közötti távolság a *cellamargó* (*cell padding* [szel pading]). A *szegélyméret* (*border thickness* [tiknesz]) megadja a táblázatban használt szegélyek vastagságát, amely vastagabb szegély esetén úgy ad térhatást, hogy a bal oldali és a felső szegély színe (*sötét szegély*) eltér a jobb oldali és az alsó szegély (*világos szegély*) színétől. Ha csupán az objektumok elrendezése a célunk, a szegélyméretet 0-ra kell állítanunk.

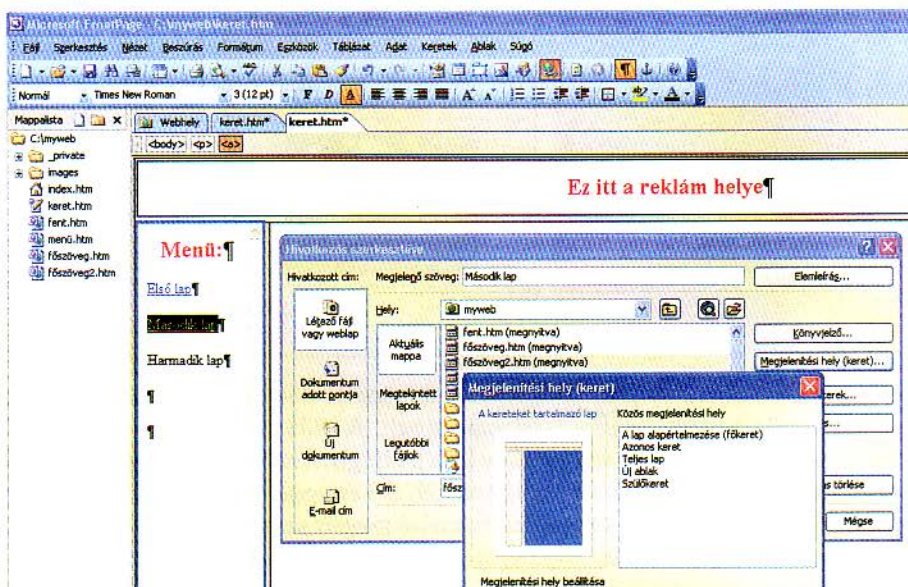


Egy táblázat és formái jellemzői (Macromedia Dreamweaver MX).
Mit jelentenek az egyes beállítások?

Keretes lapok készítése

A lap elrendezésének másik hatékony eszköze a keret. **Keretek segítségével egyetlen képernyőablakban egyszerre több HTML-dokumentum is megjeleníthető.** Gyakori például a következő elrendezés: a lap felső részén lévő keretben a reklám, a bal oldali keretben a menü, míg a főkeretben a menüben kiválasztott lap jelenik meg.

Keretek készítését minden webszerkesztő program támogatja. A keretek elrendezésének megadása után ki kell választanunk, hogy az egyes keretekben mely lapok jelenjenek meg. Ezt követi a menüként szolgáló hivatkozások és az ennek hatására megjelenő lapok összerendelése. Ez a szokásos módon történik annyi kiegészítéssel, hogy most azt is meg kell adni, hogy a kiválasztott lap melyik keretben jelenjen meg.



Keretes webhely szerkesztése. A kereteket és tartalmukat egyben szerkeszthetjük. A menü hivatkozásainak megadásánál fontos a Megjelenítési hely kiválasztása (Microsoft FrontPage).

Listák kezelése

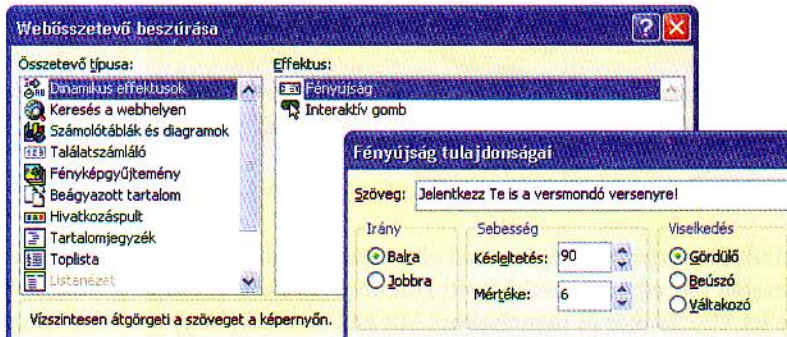
Ha a weblapon adatokat szeretnénk felsorolni, akkor táblázatok helyett weblapok esetén is használhatjuk a listákat. **A HTML-kód mind a felsorolást*, mind a számozott listát* készítését támogatja.**

A grafikus webszerkesztő programokban a listákat a szövegszerkesztésben tanultak szerint hozhatjuk létre. Természetesen itt is lehetőségünk van az alapértelmezett listajelek helyett mást használni: a webszerkesztő programok lehetővé teszik, hogy helyettük grafikus listajeleket, pl. képeket válasszunk ki.

Látványelemek: vízszintes vonal, fényűjság, interaktív gomb

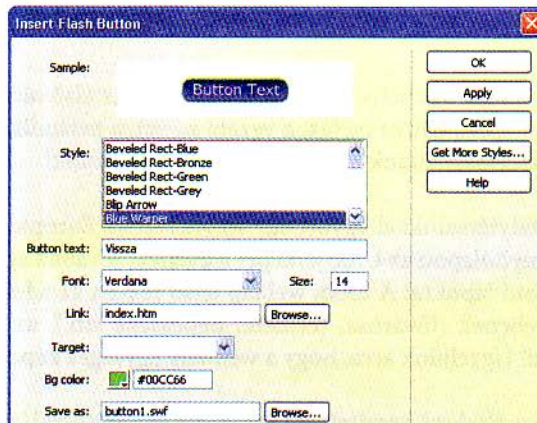
A **vízszintes vonal** az elválasztás szerepét tölti be. Mivel része a HTML-szabványnak, minden webszerkesztő program tartalmaz lehetőséget beszúrására.

A **fényűjság** vagy futófény a hétköznapi utcáin is gyakran használt reklámelem vagy tájékoztató felirat. Mivel a HTML-szabvány része, minden webszerkesztő program támogatja a létrehozását.



Fényűjság beszúrása (Microsoft FrontPage). Milyen beállítási lehetőségeink vannak?

Igen látványos elem az **interaktív gomb**, amely lehetőséget ad a weblapok közötti váltásra (pl. „Vissza” feliratú gomb vagy saját menü gombjai). Létrehozásakor nemcsak a gomb alakját, színét, feliratát, a hivatkozott weblap elérését adhatjuk meg, hanem azt is, hogyan változzon meg, ha a felhasználó fölé tolja az egeret vagy rákattint. Megvalósítása erősen függ a webszerkesztő programtól, mivel programkód beillesztését is igényli. Természetesen ezt a kódot a webszerkesztő program beállításaink alapján előállítja, és a HTML-fájlba is illeszti.



Interaktív gomb beszúrása (Macromedia Dreamweaver MX)

Összefoglaló kérdések, feladatok

1. Foglaljuk össze röviden, milyen célra készült az SGML, a HTML és az XML!
2. Hogyan helyezkednek el a fájlok egy webhelyen? Miért fontos, hogy a fájlműveletek ne az operációs rendszer, hanem a webszerkesztő program eszközeivel végezzük?
3. Hogyan működik a forráskódszerkesztő és a WYSIWYG webszerkesztő program?
4. Milyen lehetőségeket kínál a HTML-nyelv a betű- és bekezdésformázásokhoz? Mire kell tekintettel lennünk a formátumok megadásánál? Miért?
5. Milyen formázási lehetőségeket kínál a HTML-nyelv kép beszúrása esetén?
6. Foglaljuk össze röviden, hogyan kell elkészíteni az általunk használt webszerkesztő program esetén egy egylapos webhelyet! Hol helyezkednek el a fájlok? Mire kell ügyelnünk a lap létrehozásakor és mentésekor? Mit kell tennünk, hogy a weblap megjelenjen az interneten?
7. Hogyan tudjuk egy weblap háttérét beállítani az általunk használt webszerkesztő programban? Mire kell ügyelni, ha képet szeretnénk elhelyezni a háttérben?
8. Készítsünk két weblapból álló webhelyet iskolánk bemutatására! Az első legyen egy rövid ismertető egy képpel és egy hivatkozással a második lapra. A másik lapon mutassuk be röviden osztályunkat! Állítsuk be a két lap háttérének színét, a böngésző címsorát és a szöveg alapértelmezett színét is!
9. Miért fontos elem a webhely megformázásában a táblázat? Milyen beállításokat végezhetünk egy táblázat formázása esetén?
10. Készítsünk kétoldalas webhelyet az iskolai büfének! Az első oldalon szerepeljenek a büfé általános adatai (hely, nyitva tartás, a vezető neve), a második oldalon pedig egy táblázatban a büfé árai! Használjunk fel minél több látványelemet!
11. Készítsünk osztálytársainkkal együtt egy webhelyet az Európai Unió tagországainak bemutatására! A nyitólapon az Unió jelképei legyenek, továbbá egy-egy hivatkozás a tagországokat bemutató lapokra! A többi weblap felső részén az adott tagország általános jellemzői szerepeljenek (fővárosa, területe, népessége stb.), az alsó részén pedig a nemzeti sajátosságai! Ügyeljünk arra, hogy a webhely egységes képet mutasson!
12. Nyissunk meg egy általunk készített HTML-lapot egy editorral (pl. Jegyzettömbbel). Próbáljuk meg értelmezni a megjelenő kódot!

Táblázatkezelés

Táblázatkezelés régen és ma

A táblázatkezelés alapfunkciói

Ha különböző adatokat áttekinthetően szeretnénk tárolni, akkor azokat célszerű táblázatos formában elrendezni. Ezt legegyszerűbben egy kockás papír segítségével tehetjük meg, de mint láttuk, a táblázatok készítését a szövegszerkesztő programok is támogatják.

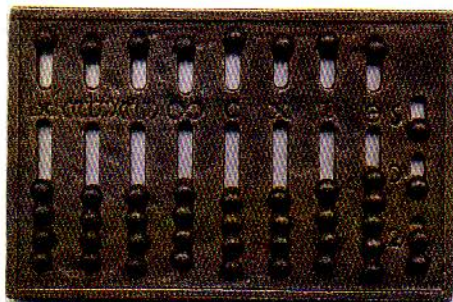
Táblázatok készítésénél a legtöbb esetben nem csupán az adatok egyszerű tárolásáról van szó, hanem azokkal műveleteket is végeznünk kell. Például, ha az osztály féléves osztályzatait egy táblázatban rendezzük el, akkor tanulónként, illetve tantárgyanként átlagokat is szoktunk számolni. A táblázatkezelő programokat azért hozták létre, hogy ezeket a számításokat elvégezzék helyettünk, s ha az adatok módosulnak, akkor automatikusan újraszámolják az eredményeket. A táblázat adatait gyakran célszerű diagramon ábrázolni, mert így az összefüggések sokkal szemléletesebbek, ezért ma már minden táblázatkezelő tartalmaz diagramkészítő modult is.

A táblázatkezelő programok* lehetővé teszik az adatok bevitelét és megformázását, a számítások elvégzését és újrakalkulálását, a diagramok létrehozását. A táblázatkezelő programok ezeken kívül eszközöket tartalmaznak üzleti előrejelzések készítésére, matematikai egyenletek közelítő megoldására, az adatok átrendezésével újabb összefüggések feltárására (kimutatások készítése) stb.

Minden táblázatkezelő támogatja az adatbázis-kezelő funkciókat is, de nagyobb méretű vagy bonyolultabb szerkezetű adathalmaz esetén már célszerűbb adatbázis-kezelő programot használni.



Ügyviteli nyomtatványok: adóbevallás, bevételi-kiadási napló, útnyilvántartás. Miért kényelmesebb táblázatkezelő program használata?



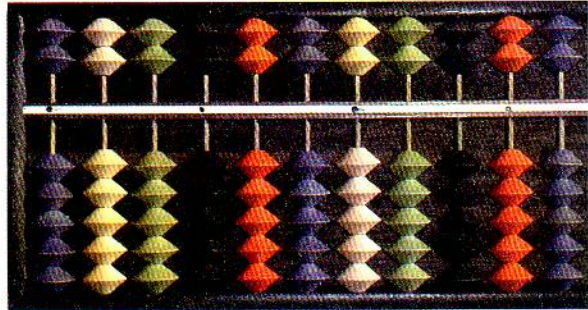
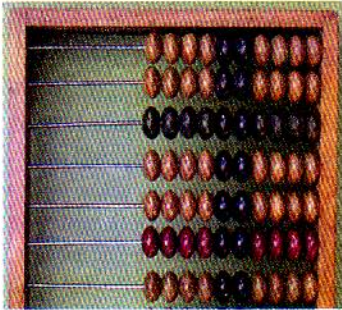
A rómaiak által használt abakusz rekonstrukciója. A vajatokban eredetileg köveket (calculus [kalkulusz]) tologattak.

A táblázatkezelők megjelenése

Szemben a szövegszerkesztéssel, melynek nyomdai és irodatechnikai előzményei voltak, a táblázatkezelés hosszú ideig ténylegesen papíron zajlott.

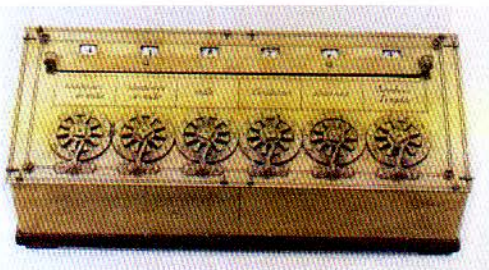
Először a számolást próbálták meg felgyorsítani, ennek egyik első eszköze az abakusz volt. Az abakuszt már az ókori Rómában is használták, bár alakja különbözött a maitól: akkor még kavicsokat (*calculus*) kellett egy kőlap vájataiban tologatni.

A mai abakusznak két változata van. A nálunk is elterjedt „golyós számológép” orosz eredetű (*scsoti*). Az ázsiai változatban (*szorobán*) függőlegesen helyezkednek el a golyók, az alsó részben lévők egyet, a felső részben lévők pedig ötöt érnek.



A mai abakusznak két változata van. A bal oldali az orosz, a jobb oldali az ázsiai típusú. Hogyan lehet az orosz típusú abakusszal összeadni? Keressünk információt az interneten!

Az első mechanikus, fogaskerekekkel és fogaslécokkal működő számológép WILHELM SHICKARD [SIKÁRD] (1592-1635) thübingeni csillagász nevéhez fűződik (1623). BLAISE PASCAL [BLEZ PASZKÁL] (1623-1662) francia matematikus az adóbeszedő édesapja részére készítette el számológépét 1642-ben. A hat számjegyen dolgozó géppel csak összeadni és kivonni lehetett, azonban már sorozatban gyártották. GOTTFRIED WILHELM LEIBNIZ [lájbnic] (1646-1716) mozgatható kocsi segítségével megoldotta a szorzást és az osztást is (1694). Lényegében ezt a változatot használták évszázadokon keresztül, később villanymotor meghajtással.



Balra Pascal számológépe, jobbra az 1970-es években még használt mechanikus számológép. Hol találkozhatunk a matematikában és a fizikában Pascal nevével?

A kétféle eszköz összehasonlítására 1946-ban az amerikai fegyveres erők lapja versenyt rendezett. Ezen egy japán abakuszmaster „mérkőzött meg” egy akkori amerikai villanyszámológépet kezelő katonával. (Az amerikaiak elődöntők során választották ki.) A szorobán az osztás és az összetett műveletek kivételével minden „versenyszámban” győzött.

Az igazi áttörést a személyi számítógépek megjelenése hozta magával. DAN BRICKLIN [den briklin] harvardi egyetemista egyszerűen meguntta, hogy hasonló jellegű pénzügyi számításokat kell többször elvégeznie különböző adatokkal. 1977-ben megtervezte a *VisiCalc* [vizikalk] nevű táblázatkezelő szoftvert, melyet BOB FRANKSTON [frenksztón] készített el 1979-ben.

A táblázatkezelő elvi felépítése azóta sem változott. A később megjelenő táblázatkezelő programok inkább a gép egyre bővülő grafikus lehetőségeit kamatoztatják, illetve újabb funkciókat vezetnek be (Lotus 1-2-3 [lótusz], Quattro [kvatró]). Mivel közös őstől származnak, a mai grafikus felületű táblázatkezelők használatának alapjai megegyeznek (Microsoft Excel [ekszel], Ability Spreadsheet [szpredsít], OpenOffice.org Calc stb.)

	A	B	C	D
1	Bevételek			
2				
3				
4	Termék	Egység	Darab	Összesen
5				
6	Fabátka	1200	12	14400
7	Hajítófa	1560	9	14040
8	Sóhaj	4500	3	
9				

	A	B	C	D
1				
2		Bevételek		
3				
4	Termék	Egység	Darab	Összesen
5				
6	Fabátka	1200	12	14400
7	Hajítófa	1560	9	14040
8	Sóhaj	4500	3	
9				

A bal oldali ábrán a VisiCalc, a jobb oldali ábrán pedig a Microsoft Excel képernyőjét látjuk. A feladat ugyanaz: termékek eladását összesítjük. A VisiCalc mérete 28 kilobájt. Mekkora helyet foglal el az általunk használt táblázatkezelő program?

A táblázatkezelés alapfogalmai

A táblázat részei

A táblázat sorokból és oszlopokból áll. A **sorokat*** egytől kezdődően számokkal, az **oszlopokat*** A-tól kezdődően betűkkel jelölik, a sakkasztalhoz hasonlóan. A Z oszlop után az AA, majd AB, AC ... jön.

Egy sor és oszlop metszéspontjában álló elem a **cella***. Az adatok a cellákban helyezkednek el. Egy cellát az oszlop- és sorazonosítójával adhatunk meg: pl. B6, C9 vagy Z16.

A **cellák téglalap alakú halmazát tartománynak*** vagy blokknak **nevezzük**. A tartományt a bal felső, valamint a jobb alsó sarkában lévő cellák azonosítójával adjuk meg, pl. B2:C9 tartomány. Egy tartományt a mai grafikus felületű táblázatkezelőkben legegyszerűbben **egérrel jelölhetünk ki**, de többnyire billentyűzetkombinációkat is használhatunk. Gyakori, hogy ezt a **SHIFT** folyamatos nyomva tartása mellett a **kurzormozgató** gombokkal tehetjük meg.

Táblázatkezelés

Egy adott sor vagy egy adott oszlop is alkothat egy tartományt. Ennek megadása pl. 2:2 vagy C:C. Kijelölése legegyszerűbben a megfelelő azonosítóra történő kattintással lehetséges, míg több egymás mellett lévő sor vagy oszlop kijelölése az egér húzásával történik. Néha az *egész táblázatot* ki kell jelölnünk. Erre elegáns megoldást kínál a legtöbb táblázatkezelő: kattintsunk az A oszlopazonosító és 1 sorazonosító mellett lévő négyzetre.

A **nem összefüggő tartomány több téglalap alakú cellahalmazból áll**. Egy ilyen tartományt általában a CTRL gomb lenyomása mellett az egér húzásával jelölhetünk ki. Képletben az egyes téglalapok felsorolásával adhatjuk meg pl. B2:C5;E5:E8;F2:G3 a nem összefüggő tartományt. A téglalapok át is fedhetik egymást.

Függetlenül attól, hogy mekkora tartományt jelöltünk ki, **egy táblázatban mindig egy olyan cella van, amibe adatot vihetünk be (aktív cella*)**. Az aktív cellát a táblázatkezelők eltérő színnel vagy inverz módon jelenítik meg.

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								

Nem összefüggő tartomány esetében is pontosan egy aktív cella van, ez a mi esetünkben a B2 (Ability Spreadsheet). Adjuk meg az ábrán látható kijelölt tartományt!

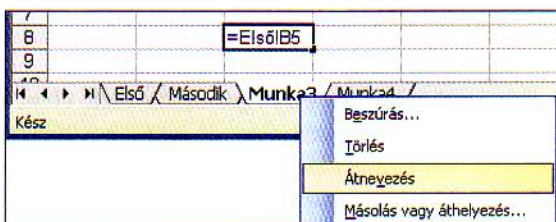
Munkalap és munkafüzet

Táblázatunk a program egy **munkalapján*** (worksheet [wörksít]) helyezkedik el. A munkalap rögzített méretű, pl. 255 oszlopból és 65 535 sorból áll.

A mai táblázatkezelőkben **egy fájl (munkafüzet*** vagy workbook [wörkbuk]) **több munkalapot tartalmaz**. Ha **több összetartozó táblázatot** szeretnénk kezelni, azt célszerű egy munkafüzet különböző munkalapjain elhelyezni.

Például, ha a 9.B osztály tankönyvrendelését táblázattal összesítjük, akkor a matematika-tankönyv árát egy másik táblázatból vesszük ki, amely tartalmazza az iskola által megrendelt valamennyi tankönyv árát.

Ha egy másik munkalap cellájára kell hivatkoznunk, akkor meg kell adnunk annak a munkalapnak a nevét is, ezt nevezik **háromdimenziós cellamegadásnak**. Erre a táblázatkezelők különféle lehetőségeket alkalmazhatnak. Gyakori, hogy a munkalap nevét egy felkiáltójel követi, ezután jön a cella címe: *Munka3!C4* vagy *Első!B5*.



A munkafüzet munkalapjai között az alsó fülek segítségével válthatunk. A munkalapokat a helyi menü menüpontjaival kezelhetjük (Microsoft Excel).

A cella tartalma

Egy adott cella tartalma háromféle lehet: *szöveg*, *szám* vagy *képlet*. Alapértelmezés szerint a táblázatkezelők a *szöveges adatot balra*, a *számokat jobbra zárják*.

Elképzelhető, hogy a cellába írt *szöveg* nem fér el. Ilyenkor általában a *szöveg* a szomszédos cellában folytatódik, ha az még üres.

A táblázatkezelők a *számokat azonos módon tárolják*, de *lehetőséget adnak azok többféle megjelenítésére*. Például a 6,23 megjelenhet pénznemként (6,23 Ft), százaléként (623 %), egy tizedesjegyre kerekítve (6,2) stb.

A *képlet* speciális jellel kezdődik, és a többi cellára vonatkozó műveleti utasítást tartalmaz*. Ez a jel többnyire =, @ vagy valamilyen műveleti jel.

Például, ha az E3 cella tartalma =C3+D3, akkor ez azt jelenti, hogy az E3 cella tartalma a C3 és D3 cella értékének összege. Ha e két cella bármelyike módosul, akkor az E3 is *automatikusan módosul*. A cellában többnyire a képlet eredménye látható, maga a képlet a *táblázat fölötti sorban, a szerkesztőlécen* vagy szerkesztősorban jelenik meg.

fx Σ = =C3+D3				
	B	C	D	E
	Szöveg			
	Már nem▶	2	3	5

A B2 és B3 cella tartalma is szöveg, azonban a B3 tartalma nem fér ki.

A C3 és D3 cella számot, az E3 pedig képletet tartalmaz. A képlet a szerkesztőlécen tekinthető meg (OpenOffice.org Calc).

A képlet és a függvény kapcsolata

Bonyolultabb számítások esetén, amikor a képlet a négy alapművelettel nehezen adható meg, a *képletekben függvények* is szerepelhetnek*. Ilyen függvény szolgál például egy adott cellában lévő szám négyzetgyökének meghatározására vagy egy adott cellatartomány összegének kiszámítására.

	A	B	C
1		1	Osszeg:
2		2	45
3		3	
4		4	Átlag:
5		5	5
6		6	
7		7	2
8		8	Négyzetgy
9		9	1.414214
10			

	A	B	C	D	E
1	1	8			
2	2	9		Osszeg:	105
3	3	10		Átlag:	7,5
4	4	11			
5	5	12		2	
6	6	13		Négyzetgyöke:	
7	7	14			1,414214
8					

C2 tartalma: @SUM(A1...A9)

C5 tartalma: @AVERAGE(A1...A9)

C9 tartalma: @SQRT(C7)

E2 tartalma: =SZUM(A1:B7)

E3 tartalma: =ÁTLAG(A1:B7)

E7 tartalma: =GYÖK(D5)

A képlet felépítése formailag a kezdetek óta nem változott. (Balra a VisiCalc, jobbra a Microsoft Excel.) Készítsük el az ábrán látható táblázatot az általunk használt táblázatkezelő programmal!

A táblázatkezelők alapfunkciói

Tekintsük át a táblázatkezelők alapfunkcióit egy példa segítségével! Példánkban Magyarország kereskedelmi mérlegét, vagyis az exportra és importra vonatkozó adatokat szemléltetjük 1999 és 2003 között. Készítsük el a táblázatot lépésről lépésre!

	A	B	C	D	E	F	G
1							
2			Magyarország kereskedelmi mérlege (millió euróban)				
3							
4		Év	1999	2000	2001	2002	2003
5		Export	24059	31278	34697	36821	38161
6		Import	26102	34457	37193	39024	41132
7		Mérleg					

Magyarország kereskedelmi mérlege 1999 és 2003 között.
A nyers táblázat, amiből kiindulunk (OpenOffice.org Calc).

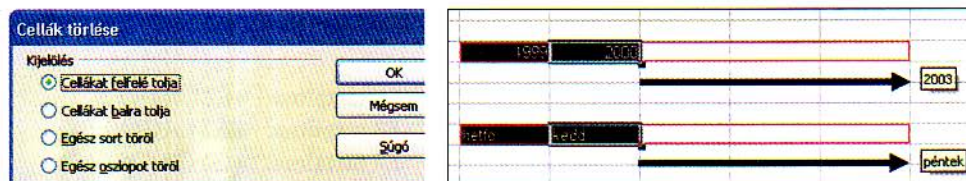
Az adatok bevitele, törlése, javítása

Az adatok beírásakor sokat segít, hogy a táblázatkezelők a számokat automatikusan felismerik és jobbra zárják. Ha egy számot mégis szöveggént szeretnénk bevinni, akkor azt külön jeleznünk kell, például azzal, hogy aposztróffal kezdjük.

A cella tartalmát legegyszerűbben a *Delete* [dilit] gombbal törölhetjük. Ha a cella tartalmát javítani szeretnénk, akkor azt a cella kiválasztása után a *Szerkesztősorban* tehetjük meg. Gyakori, hogy a táblázatkezelő lehetővé teszi közvetlenül a cellában történő javítást is, ha leütjük a megfelelő billentyűzetkombinációt (tipikusan az *F2*-t).

Fölösleges sorokat, oszlopokat vagy cellákat általában a *Szerkesztés* menü megfelelő pontjával törölhetünk. *Cellák törlésénél* meg kell adnunk, hogy a kitörölt cellák helyére az alatta vagy tőle jobbra lévő cellák lépjenek. Új sort vagy oszlopot általában a *Beszúrás* menü megfelelő pontjaival szúrhatunk be.

Sokszor van szükségünk arra, hogy egy sort vagy oszlopot egy sorozat tagjaival töltsünk föl. Például: hétfő, kedd, szerda ... vagy 2001, 2002, 2003 Ezt a táblázatkezelőkben a *kitöltés* funkcióval érhetjük el: **beírjuk a sorozat első egy vagy két elemét, majd a sorozatot az eger húzásával folytatjuk.** A sorozat tagjait vagy képzési szabályát a menü segítségével adhatjuk meg vagy választhatjuk ki.



Balra: Cella törlése, jobbra: kitöltés az eger húzásával (OpenOffice.org Calc)

Képlet bevitelle, másolása

A táblázatkezelők erőssége a képletek alkalmazása. Ha a képletben szereplő cellák bármelyikének a tartalma módosul, akkor a táblázatkezelők a képleteket *automatikusan újraszámítják**. Ennek ellenére a legtöbb táblázatkezelő lehetővé teszi a képletek újrakalkulálását felhasználói utasításra is. (Például a Microsoft Excelben az F9 gomb, az OpenOffice.org Calc esetén a *SHIFT CTRL F9* lenyomásával érhetjük ezt el.)

A grafikus felületű táblázatkezelők a képlet beírásához gyakran adnak segítséget. Például a képletben hivatkozott cellákat színezéssel vagy szegéllyel kijelölik, vagy a függvények paramétereit a képernyő egy adott részén, esetleg buborékokban megjelenítik.

Év	1999	2000	10	20
Export	24059	31278	30	40
Import	26102	34457		
Mérleg	=c5-c6			=Átlag(F4:G6)

ÁTLAG(szám1; [szám2]; ...)

A függvény megadásához szükséges paraméterek buborékokban jelennek meg, a cellahivatkozásokat pedig színezéssel emeli ki a program (Microsoft Excel).

Gyakran előfordul, hogy lényegében ugyanazt a képletet kell alkalmaznunk az egymás alatti sorokra vagy egymás melletti oszlopokra. Ilyenkor hasznos funkció a *képlet másolása**: a képletet az első cellába beírjuk, majd annak tartalmával kitöltjük a szomszédos cellákat. Ekkor alapértelmezés szerint a képletben szereplő cellahivatkozások a másolás irányának megfelelően aktualizálódnak.

Import	26102	34457	37193	39024	41132
Mérleg	-2043	-3179	-2496	-2203	-2971

Cellák másolása
 Cellák másolása
 Kitöltés csak a formátummal
 Kitöltés a formátum nélkül

A képletet a C7 cellába írjuk, majd a cella tartalmával kitöltjük a 7. sort. Az Excelben ilyenkor egy intelligens címke jelenik meg, amellyel megadhatjuk, hogy a formátum ne másolódjon át.

A táblázat formázása

A mai világban nem elegendő, ha a táblázat helyesen működik, a felhasználók azt is elvárják, hogy az esztétikusan legyen megformázva.

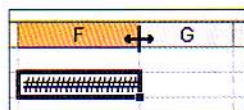
Az oszlopok szélességét általában az egér húzásával szabályozhatjuk, ha a benne lévő adatok nem férnek ki, vagy az oszlop túl széles. Mivel számokból nem mindig lehet egyszerűen levágni a kilógó számjegyeket, ilyenkor a cellában egy speciális jel, például

Táblázatkezelés

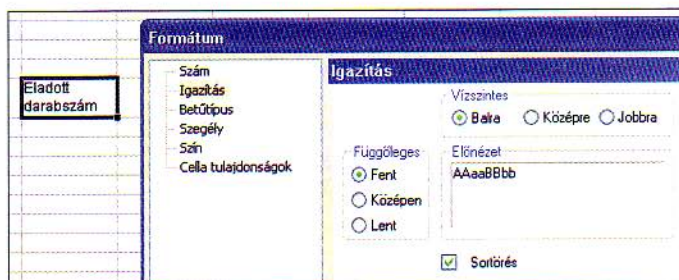
jelenik meg. Az oszlopok szélességét a *menü megfelelő pontjaival*, grafikus felületen pedig *az oszlopokat (jobbról) elválasztó vonal húzásával* is beállíthatjuk.

A *sorok magasságát* hasonló módon, pl. az egér húzásával módosíthatjuk.

A *betű- és bekezdésformázásra* a legtöbb táblázatkezelő a szövegszerkesztésben már megszokott lehetőségeket kínálja. Igazi problémát jelent, ha a cellába *a cella szélességénél hosszabb szöveget* szeretnénk elhelyezni. Erre a táblázatkezelők különféle megoldásokat kínálnak: ilyen például a szöveg *több sorra tördelése*, a szöveg méretének *lekicsinyítése* vagy a szöveg elhelyezése *összevont cellákban*.



Cella szélességének módosítása az elválasztó vonal húzásával (Microsoft Excel)

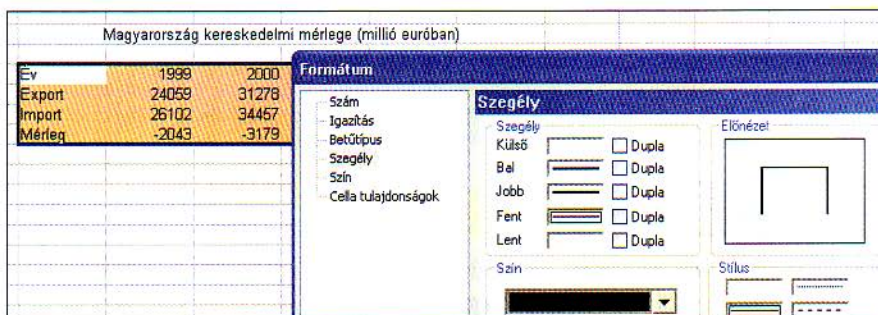


A szöveg igazítása és elrendezése (Ability Spreadsheet). Milyen lehetőségeink vannak, ha nem fér ki a cellába a szöveg?

A *táblázat megjelenését* legegyszerűbben egy *automatikus formátum* kiválasztásával érhetjük el, többnyire a *Formátum* menü *Automatikus formázás* pontjával. Az itt felkínált formátumokat a szoftver készítője általában megfelelő szakemberekkel készítette és vizuálisan ellenőriztette is.

Ha nem szeretnénk ezeket a „konzerv” formátumokat használni, akkor a formátumokat magunk is beállíthatjuk. Ilyenkor megadhatjuk *a cellák szegélyének* típusát és színét, *a cellák háttérének* színét és mintázatát, a cellában megjelenő adatok *betűformátumait*, az adatok *igazítását* stb.

A munkalap megjelenését még látványosabbá tehetjük képek beszúrásával és alkalmas elrendezésével. Ezt a szövegszerkesztésben tanult módon tehetjük meg.



Szegély elhelyezése a kijelölt tartomány köré. Akár határoló vonalanként is eltérhet a szegély színe, stílusa vagy vastagsága (Ability Spreadsheet). Hogyan rejthetjük el a rácsvonalat a képernyő egy kijelölt részén?

Abszolút és relatív cellahivatkozás

A képletben lévő cellahivatkozást *relatív cellahivatkozásnak** nevezzük, ha az a képlet másolásakor a másolás irányának megfelelően módosul. Eddigi példánkban ilyen esetekkel találkoztunk. Ebben az esetben a táblázatkezelők nem a ténylegesen hivatkozott cellacímeket tárolják, hanem azoknak az aktuális cellához viszonyított helyzetét. Előző példánkban a *Mérleg* kiszámításához (C7) tartozó információ (=C5–C6) tehát ténylegesen „a kettővel feljebb lévő cella értéke mínusz az eggyel feljebb lévő cella értéke”. Világos, hogy ez vízszintes irányú másoláskor helyes eredményekre vezet.

(A cellák tartalmának követéséhez a táblázatkezelők lehetővé teszik, hogy a cellákban ne a kiszámolt adatokat, hanem a képleteket jelenítsük meg. Az Excelben például a két üzemmód között az ALT . billentyűkombinációval válthatunk.)

	A	B	C
1	Fagylaltos		
2			
3	1 gombóc	99 Ft	
4			
5	Gombóc	Fizetendő	(Képlet)
6	2	198 Ft	=A6*B3
7	3	0 Ft	=A7*B4
8	4	#ÉRTÉKI	=A8*B5
9	5	990 Ft	=A9*B6

	A	B	C
1	Fagylaltos		
2			
3	1 gombóc	99 Ft	
4			
5	Gombóc	Fizetendő	(Képlet)
6	2	198 Ft	=A6*\$B\$3
7	3	297 Ft	=A7*\$B\$3
8	4	396 Ft	=A8*\$B\$3
9	5	495 Ft	=A9*\$B\$3

Ha egy gombóc ára a B3 cellában van, a gombócok száma pedig az A6-ban, akkor az =A6*B3 képlet másolása helytelen adatokra vezet, mert a B3 hivatkozás is módosul. A B3 hivatkozás rögzítése esetén az adatok helyesek lesznek. (Az A6 természetesen ekkor is módosul.)
Készítsük el mindkét táblázatot!

Gyakran előfordul, hogy egy adott cellában valamilyen állandó érték szerepel, például egy termék egységára. Ha az erre hivatkozó képlet relatív cellahivatkozást tartalmaz, akkor a képlet másolása nemcsak helytelen eredményre, hanem gyakran képlethibára is vezet. Ilyenkor az *abszolút cellahivatkozást* kell alkalmaznunk.

*Abszolút cellahivatkozás** esetén a cella címe másoláskor nem változik, ekkor a táblázatkezelő a cella tényleges helyét tárolja. Az abszolút cellahivatkozásnál a sor-, illetve oszlopkoordináta elé \$ jelet kell tennünk. Pl. az =A6*\$B\$3 képletben az A6 relatív, tehát másoláskor módosul, a \$B\$3 pedig nem, mivel abszolút cellahivatkozás.

*Vegyes cellahivatkozásban** az egyik koordináta abszolút, a másik relatív. Többnyire matematikai táblázatokban használják, pl. \$A2 vagy B\$1.

	B2	=\$A2*\$B\$1									
	A	B	C	D	E	F	G	H	I	J	
1		1	2	3	4	5	6	7	8	9	
2	1	1	2	3	4	5	6	7	8	9	
3	2	2	4	6	8	10	12	14	16	18	
4	3										
5	4										
6	5										

Szorótábla készítése. Mivel az egyik tényező mindig az A oszlopban, a másik mindig az 1. sorban van, az A oszlopra és az 1. sorra abszolút módon kell hivatkozni. A B2 cellába tehát az =\$A2*\$B\$1 képlet kerül. Készítsük el a szorótáblát!

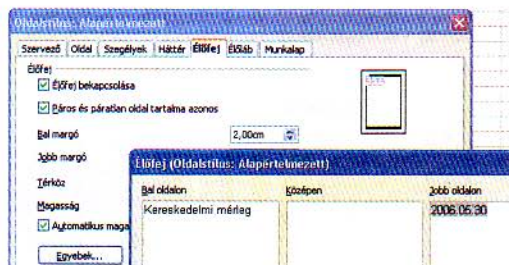
Fájlműveletek: mentés, nyomtatás, megnyitás

A táblázatkezelők alapértelmezés szerint a munkafüzetet *mentik*, vagyis a munkánkat nem munkalaponként, hanem egyben menthetjük. A saját alapértelmezett formátumukon kívül más formátumokat is támogatnak, ilyen pl. a HTML.

Nyomtatáskor az egyes munkalapokat külön-külön nyomtathatjuk, ennek megfelelően a nyomtatásra vonatkozó beállításokat laponként adhatjuk meg. Ezért az oldalbeállítások menüpont vagy a *Fájl* menüben (pl. Excel) vagy a *Formátum* menüben (pl. OpenOffice.org Calc) található. A szokásos beállításokon túl (tájolás, példányszám, margók stb.) a táblázatra jellemző paramétereket is megadhatunk. Ilyen például, hogy milyen sorrendben nyomtassa a lapokat, megjelenjenek-e a cellarácsok a lapon.

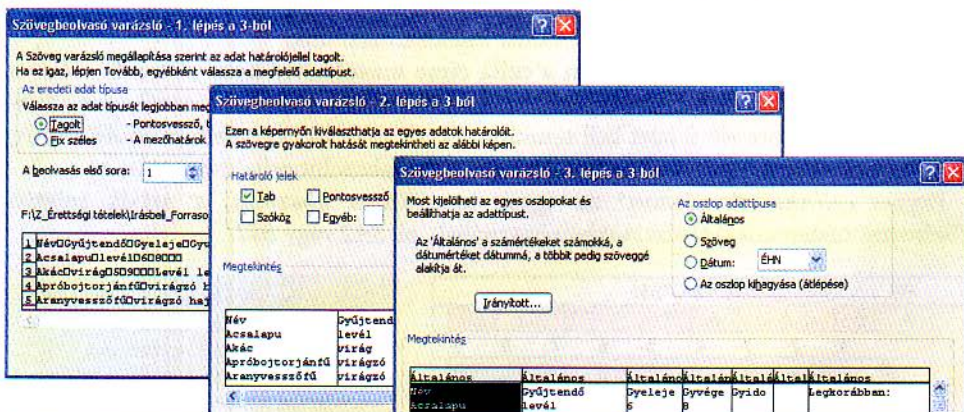
Mivel a munkalapot nem egy adott papírméretre tervezzük, nyomtatáskor lehetőségünk van átméretezésére, sőt, gyakran azt is beállíthatjuk, hogy hány oldal magas illetve széles legyen.

Az *élőfej és élőláb* általában három részre (bal oldal, középső rész, jobb oldal) tagolódik. Ezeket külön-külön, de a szövegszerkesztésben megszokott módon adhatjuk meg.



Élőfej beállítása (OpenOffice.org Calc)

Gyakori, hogy az adatokat **formázatlan szöveges** (text) fájlban kapjuk. Ilyenkor egy adott sorban a cellákba kerülő adatokat valamilyen jel (pl. tabulátor) választja el egymástól. A táblázatkezelők az így kapott adatok beolvasását **varázslókkal segítik**. A varázslók nemcsak az adatok cellákba történő elhelyezését, hanem a típus felismerését is megoldják. Természetesen a táblázatkezelők mentéskor maguk is képesek ilyen formátumú fájl előállítására, de ekkor nemcsak a formázás, hanem a képletek is elvesznek.



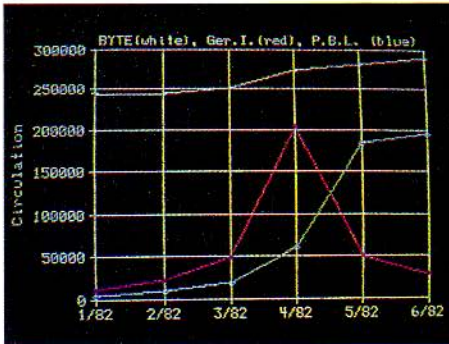
Szöveges fájl importálása varázslóval (Microsoft Excel).

Milyen előnyei lehetnek, ha az adatokat TXT formátumú fájlban tároljuk?

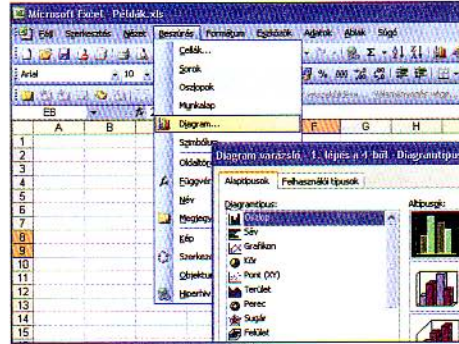
Diagramok készítése

Az automatikus diagramkészítés

A legtöbb ember számára egy adatsor könnyebben áttekinthető, ha a számokat diagramon ábrázoljuk. Diagramkészítő modult elsőként az 1982-ben megjelent Lotus 1-2-3 táblázatkezelő tartalmazott, de ma már a táblázatkezelők alapszolgáltatása.



Elsőként a Lotus 1-2-3 táblázatkezelőben jelent meg a diagramkészítés 1982-ben.



Ma a diagram készítése minden táblázatkezelőben alapszolgáltatás (Microsoft Excel).

A diagram elkészítése automatikusan történik. Az adatok kijelölése után varázsló segítségével vagy a megfelelő menüpontok kiválasztásával adhatjuk meg a diagram* típusát (oszlop, kör, grafikon stb.) és egyéb jellemzőit (feliratok, jelmagyarázat, tengelyek beállítása stb.). Kijelöléskor mindig érdemes a rovatfejet is kijelölni, mert így a tengelyek felirata is automatikusan elkészül (pl. év, Afrika). Később, ha az adatok a táblázatban módosulnak, a diagram automatikusan frissül.

	A	B	C	D	E	F	G	H
1	Év	Afrika	Ázsia	Európa	Latin A.	Észak A.	Óceánia	Összesen
2	1950	221	1 398	547	167	172	13	2 518
3	1960	277	1 701	604	218	204	16	3 020
4	1970	357	2 143	656	285	232	19	3 692
5	1980	470	2 632	692	361	256	23	4 434
6	1990	622	3 168	722	442	284	27	5 265
7	2000	796	3 680	728	520	316	31	6 071
8	2010	984	4 149	720	594	348	35	6 830
9	2020	1 188	4 570	705	659	380	38	7 540
10	2030	1 398	4 887	685	711	408	41	8 130
11	2040	1 608	5 103	661	748	430	44	8 594
12	2050	1 803	5 222	632	768	448	46	8 919

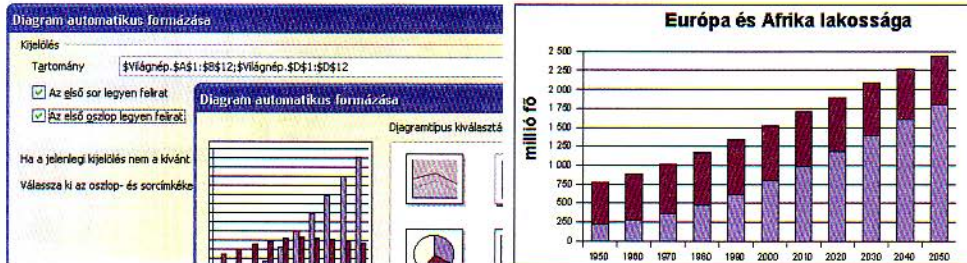
Táblázatunk a világ népességének alakulását mutatja kontinensenként az ENSZ adatai szerint. Milyen képlettel készült az Összesen oszlop?

A diagramok fajtái

A diagram típusának megadásánál azt kell figyelembe vennünk, hogy milyen összefüggést szeretnénk szemléltetni.

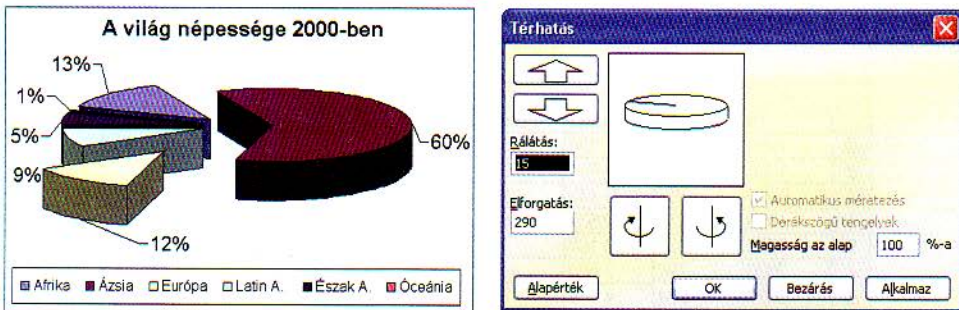
A leggyakoribb típus az **oszlopdiaagram***, amely akár az adatok összehasonlítására, akár időbeli változásának követésére használható. Az összevont oszlopdiaagram e két lehetőséget együttesen szemlélteti. A diagramok színesítésére nemcsak téglalapokkal és hasábokkal, hanem hengerrel, kúppal, piramissal is ábrázolhatjuk az adatokat.

A **sávdiaagram** annyiban tér el az oszlopdiaagramtól, hogy ebben az esetben a kategóriatengely (az x tengely) függőleges, míg az értéktengely (az y tengely) vízszintes.



Oszlopdiaagram készítésének első két lépése és a kész diagram (OpenOffice.org Calc).
Milyen összefüggés olvasható le a két kontinens lakosságának együttes növekedéséről?
Milyen összefüggéseket olvashatunk le Európa és Afrika lakosságára vonatkozóan külön-külön?

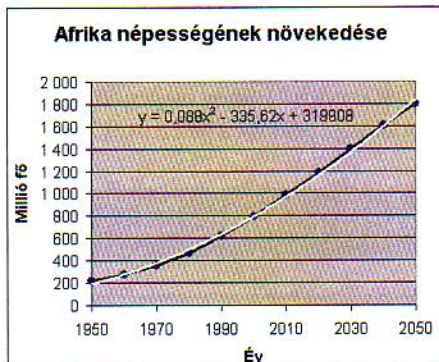
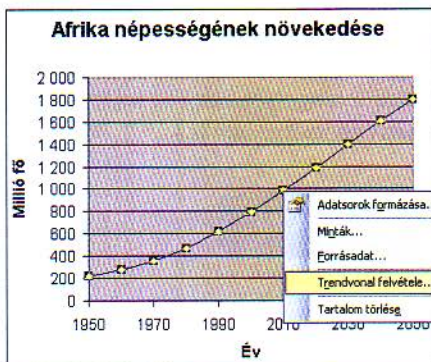
A **kördiagram*** mindig csak egy adatsort ábrázol, ám egyaránt szemlélteti az egyes adatok egymáshoz és az adatok összegéhez való viszonyát is. Lehetőségünk van egy **adat kiemelésére**, illetve az adatok csoportosítására is. A kördiagramokat tértartású, percc vagy „robbantott” változatokkal tehetjük változatosabbá.



Példa „Robbantott torta” kördiagramra és térhatásának beállítására (Microsoft Excel).
Egy számunkra fontos adat (pl. Európa) a körcikk kihúzásával külön is kiemelhető.
Mikor célszerű kördiagramot használni? Adathűen jelennek-e meg az adatok a tortadiagramon?

Adatsorok időbeli változását, illetve egymáshoz viszonyítását a tudományos életben is használt **vonaldiaagramokkal*** szemléltethetjük a legpontosabban. Ezekhez a

táblázatkezelők általában további lehetőségeket is kínálnak, például *trendvonalak* vagy *hibasávok* felvételét.

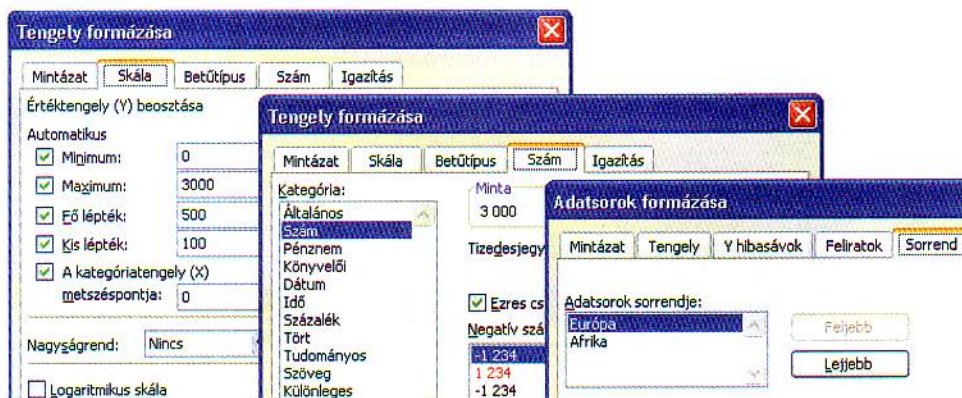


Ha egy másodfokú trendvonalat illesztünk az adatsorra, a kapott görbe egyenletéből további becsléseket adhatunk a lakosság létszámának változására (Microsoft Excel).

A gazdasági és műszaki életben további diagramokat is használnak. Ilyen például a tőzsdei árfolyamok szemléltetésére használt *árfolyamdiagram*, vagy a három változó kapcsolatát visszaadó *buborékdiagram*.

A diagramok utólagos formázása

Elkészítése után a diagram szinte valamennyi komponensét módosíthatjuk. A leglátványosabbak a formai jellemzők: a tengelyek, a háttér, a diagram és a betűk *színe*, *mintázata*, illetve *térhatása*. A diagram használhatósága szempontjából fontosabb a *tengelyek skálájának beállítása*, a *számok formátuma* és az *adatsorok sorrendje*.



Ha a diagram valamely komponensére (pl. tengely, adatsor) kattintunk, akkor módosíthatjuk annak beállításait (Microsoft Excel). Próbáljuk ki, milyen lehetőségeket kínál erre az általunk használt táblázatkezelő program!

Táblázatkezelés

A módosítások egy részét, például a betűk jellemzőit, az *eszköztár ikonjaival* is elvégezhetjük. A diagramok módosításához azonban *külön menü* tartozik, ami gyakran csak akkor jelenik meg, ha a diagramra kattintunk (pl. Microsoft Excel). Gyakori megoldás, hogy egy adott komponens beállításait tartalmazó dialógusdoboz az adott komponensre való *dupla kattintással* is előhívható. Például egy tengely léptékének vagy egy oszlop színének módosításához elegendő kettőt kattintani a tengelyre vagy az adott oszlopra. (Így működik például a Microsoft Excel és az OpenOffice.org Calc is.)

Problémamegoldás táblázatkezelővel

Példa statisztikai számításokra

Az iskolában az osztályfőnökök minden félév végén elkészítik az osztálystatisztikát. Ez tartalmazza a tanulók osztályzatait, a tanulói és tantárgyi átlagokat, a legjobb és legrosszabb tanuló átlagát, a mulasztott órák számát és összegét, a bukások számát stb. Gondoljuk végig, milyen függvényekre van szükségük!

Leggyakrabban egy adott tartományban szereplő *számok átlagát* kell meghatározni, erre a táblázatkezelők **Átlag*** vagy *Average* [evöridzs] néven kínálnak függvényt. A *legmagasabb, illetve legalacsonyabb* tanulmányi eredmény a **Max***, illetve **Min*** függvényekkel választható ki. A hiányzások *összegezéséhez* egy tartomány adatait kell összegezni, erre a **Szum*** vagy *Sum* [szám] függvényt kínálják a táblázatkezelők. Tekintettel a nagyszámú adatra, a cellák tartalmának egyszerű összeadása (C3+C4+C5+...) sok munkát és nagy hibaforrást rejtene magában.

Egy adott érték (esetünkben az elégtelenek) *számának meghatározására* a táblázatkezelők különböző számláló függvényeket tartalmaznak. Ezek között az adatok jellegében és a cellák tartalmában van eltérés. A leggyakrabban használt **Darabtelj*** vagy *CountIf* [káuntif] függvény például meghatározza azoknak a számokat tartalmazó celláknak a számát, amelyek eleget tesznek egy adott feltételnek. Így például a *Darabtelj(B2:E10;1)* megszámolja, hány 1-es van a B2:E10 tartományban.

	A	B	C	D	E	F
1	név	irod	nyelvt	angol	mat	átlag
2	Alma András	3	3		2	2,67
3	Cibere Pál	4	4		3	3,67
4	Ferő Helga	4	2	1	4	2,75
5	Hs Anita	1	1	2	3	1,75
6	Kovács Katalin	5	5	4	4	4,50
7	Magyar Imre	3	1	3	2	2,25
8	Németh Judit	3	4	2	2	2,75
9	Tóth Pál	5	5		3	4,33
10	Vas Mária	4	5	4	4	4,25
11	tantárgyi átlag	3,56	3,33	2,67	3,00	3,18
12						
13	Elégtelenek száma:	4				
14						

Rendezés

Rendezési feltétel: Beállítások

Rendezési szempont: átlag

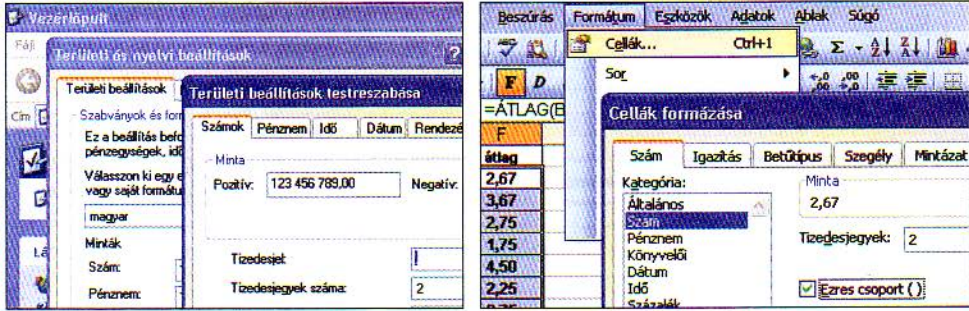
Ázutan: névsor

Növekvő Csökkenő

Részlet a félévi osztálystatisztikából. A tanulókat az elért eredményeik szerint akár rendezhetjük is (OpenOffice.org Calc). Milyen képlet van a B13 cellában?

Számformátumok beállítása

A táblázatkezelőket elsősorban statisztikai, gazdasági, pénzügyi, műszaki stb. *kalkulációkra* használják. Ezekhez a számokat többnyire speciális számformátumban* kell megjeleníteni, jóllehet a táblázatkezelők csupán egyféle formátumot használnak az adatok belső tárolására.



A tizedesjel alapértelmezett beállítása a Windows operációs rendszerben a Vezérlőpulton történik. A számok formátumát a **Formátum** menü **Cellák** pontjával állíthatjuk be (Microsoft Excel). Honnan veszi a táblázatkezelő program az alapértelmezett beállításokat?

A táblázatkezelők gyakran már az eszköztáron is lehetővé teszik a leggyakoribb formátumok beállítását, azonban valamennyi lehetőség eléréséhez érdemes a megfelelő menüpontot használni. Ezt többnyire a **Formátum** menüben kell keresnünk.

Érdeemes tudni, hogy míg a magyar szabványok szerint az ezres elválasztó a szökőz és a tizedesjel a vessző, addig angol nyelvterületen tizedespontot használnak, a vessző pedig az ezres elválasztó. A táblázatkezelők az adott nyelvterületen használt számformátumokat az operációs rendszer területi beállításaiából olvassák ki. Így egy angol nyelvterületen készült táblázatban magyar területi beállítások esetén már a magyar szókásoknak megfelelően jelennek meg az adatok.

Ha egy számot a tényleges pontosságánál kevesebb tizedesjeggyel jelenítünk meg, akkor látszólag a táblázatkezelők kerekítenek, ténylegesen azonban a számot továbbra is az eredeti pontossággal tárolják. Legyen két cella tartalma 1,34 és 2,23. Ha ezeket egy harmadik cellában képlettel összeadjuk, majd a tizedesjegyek számát 1-re állítjuk, akkor a képernyőn 1,3 és 2,2, összegükként pedig 3,6 jelenik meg. Ha az adatokat ténylegesen kerekíteni szeretnénk, akkor ehhez a **Kerekítés** vagy **Round** [raund] függvényt kell használnunk.

Pénzügyi számítások. Pénznem és százalék formátum

Gazdasági és pénzügyi számítások esetében igen gyakran van szükségünk a százalék és a pénznem formátumra.

A **pénznem*** beállításait a táblázatkezelők a területi beállításokból veszik, de ezt a tizedesjegyek számával együtt többnyire módosíthatjuk. A formátumot nem feltétlenül kell külön beállítanunk, ha a pénznem jelét helyesen írjuk be.

Táblázatkezelés

A **könyvelői formátum*** esetében a tizedesjelek egymás alá kerülnek akkor is, ha az egymás alatti számokban a tizedesjegy különböző.

A **százalék formátum*** a szám százszorosát jeleníti meg. Tehát a 12% ténylegesen 0,12, és a táblázatkezelő is ez utóbbival számol. (Az adat így is beírható: 12%, természetesen ekkor nem kell a századrészt venni beíráskor.)

A termék, vagy szolgáltatás megnevezése besorolási száma és egyéb jellemzői	M.e.	Mennyiség	ÁFA nélküli egységár	ÁFA nélküli számlaérték	ÁFA kulcsa	ÁFA összege	Számlaérték ÁFA-val
Kenyér VTSZ 1905	kg	3	70 Ft	210 Ft	12%	25 Ft	235 Ft

A	B	C	D	E	F	G	H	I
1								
2	A termék megnevezése és besorolási száma	M. e.	Mennyiség	ÁFA nélküli egységár	ÁFA nélküli számlaérték	ÁFA kulcsa	ÁFA összege	Számlaérték ÁFA-val
3	Kenyér VTSZ 1905	kg	3	70 Ft	210 Ft	12%	25 Ft	235 Ft
4								

A	B	C	D	E	F	G	H	I
1								
2	A termék megnevezése és besorolási száma	M. e.	Mennyiség	ÁFA nélküli egységár	ÁFA nélküli számlaérték	ÁFA kulcsa	ÁFA összege	Számlaérték ÁFA-val
3	Kenyér VTSZ 1905	kg	3	70	=D3*E3	0,12	=F3*G3	=F3+H3
4								

A felső sorban egy kézi számla részletét látjuk. A második sorban ugyanezt táblázatkezelővel valósítottuk meg. A harmadik sor az egyes cellák tényleges tartalmát mutatja. Készítsük el és formázzuk meg a fenti táblázatot az ábrán látható módon!

Dátum és időkezelés, egyéni számformátumok

A táblázatkezelők minden számot ugyanabban a formában tárolnak, ez vonatkozik a **dátum- és időkezelésre*** is. Hogyan lehet egy ilyen bonyolult rendszert, amelyben a váltószámok sem egységesek, leképezni a valós számok halmazára (év, hónap, nap, óra, perc, másodperc)?

A táblázatkezelők **egységnek a napot tekintik, az időszámítás kezdetének (1-nek) pedig 1900. január 1. 0 óra 0 perccel**. Tehát 1900. január 2-ának 2, 1900. február 1-jének 32, 2004. szeptember 1-jének pedig 38231 felel meg. **Az időt az így kapott szám tizedesjegyei jelentik**. Így 1,5 a „hétköznapi” 1900. január 1. 12 óra 0 perc, míg 38231,3333 jelenti 2004. szeptember 1. 8 óra 0 perccel. Természetesen a táblázatkezelők megjelenítéskor ezeket a számokat az általunk megszokott formára konvertálják.

A dátum kezelésének fenti módja lehetővé teszi, hogy két dátum különbségét a közben eltelt napok számával értelmezzük. Például, ha a mai dátumból kivonjuk a születésnapunkat, akkor megkapjuk, hogy hány naposok vagyunk. Hasonló módon, ha a mai dátumhoz hozzáadunk pl. 100-at, akkor megkapjuk, milyen dátum lesz száz nap múlva.

A két dátum különbsége gyakran nem dátum típusú adat, hanem például *db* (darab). Ilyen számformátum általában ugyan nincs a táblázatkezelőkben, de ez nem baj, mert lehetőség adnak saját számformátum definiálására (ld. ábra).

Saját számformátum létrehozásához általában két dolgot kell megadnunk. Az egyik az, hogy hogyan jelenjenek meg a számjegyek (tagolás, tizedesjegyek száma stb.), a másik pedig az, hogy milyen szöveg vagy jel kerüljön a szám elé vagy mögé.

Érkezés	Távozás	Vendégéjszaka	Napi díj
05.03.16	05.03.22	6 db	5 400 F
05.03.16	05.03.24	8 db	5 600 F

Az E3 cella tartalma a $=D3 \cdot B3$ képlet, a cella formátumát Számra kell állítani (Ability Spreadsheet). Egyéni formátum definiálásával jobban átláthatók az adatok (jobb oldali ábra). Milyen beállítási lehetőségeink vannak, ha saját számformátumot szeretnénk alkalmazni?

Választás két lehetőség között

Az ECDL-vizsgán operációs rendszer ismeretekből az kap megfelelt minősítést, aki a 32-ből legalább 24 pontot szerez. Ilyen esetben két lehetőség között kell döntenünk:

Ha a pontszám ≥ 24 akkor megfelelt, egyébként nem felelt meg.

Ezt a formulát a táblázatkezelők is támogatják, az *If* vagy **Ha*** függvény segítségével. A függvény felépítése a következő:

$=Ha(\text{Feltétel}; \text{Igaz esetén mit kell tenni}; \text{Hamis esetén mit kell tenni})$

A grafikus felületű táblázatkezelők általában képesek arra is, hogy az adott feltételeknek megfelelő cellák tartalmát előírt módon jelenítsék meg (**feltételes formázás**). A logikai feltétel megadását ekkor legördülő listák segítségével választhatjuk ki.

Név	Pontszám	Minősítés
Kiss Pál	22	Nem felelt meg
Nagy Béla	28	Megfelelt

A **Ha** függvény használata (Microsoft Excel)

Feltételes formázás (MS Excel)

Választás több lehetőség közül

Valós problémáknál gyakori, hogy nem két, hanem több lehetőség között kell választanunk. Ilyen például a fizetési sávoktól függő személyi jövedelemadó, a gyerekek számától nem arányosan függő családi pótlék vagy a valuta árfolyama. Ezekben az esetekben használhatnánk több egymásba ágyazott *Ha* függvényt is, azonban a feladat megoldására a táblázatkezelők többféle *keresőfüggvényt* kínálnak.

Táblázatkezelés

A keresőfüggvények *segéd táblázat*ot használnak, ilyen például az *Fkeres** vagy *Vlookup* [vlukáp]. Például, ha a tanulók osztályzatát szeretnénk megadni az elért pontszám alapján, akkor szükségünk van egy átváltási táblázatra, amely megadja, hogy az egyes osztályzatok eléréséhez legalább hány pont szükséges. Az *Fkeres* függvény ekkor az átváltási táblázat első oszlopában megkeresi, hogy a tanuló pontszáma melyik sávba esik, majd a mellette lévő osztályzatot beírja az adott cellába.

	A	B	C	D	E	F	G	H
1								
2		Név	Pontszám	Osztályzat				0 elégtelen
3		Kis Pál	16	jó				8 elégséges
4		Nagy Pál	28	=FKERES(C4;\$G\$2:\$H\$6;2)				12 közepes
5								16 jó
6								20 jeles

Az **FKERES** függvény megkeresi, hogy a C4 cellában szereplő pontszám a \$G\$2:\$H\$6 segéd táblázat melyik sávjába esik, és megadja a segéd tábla 2. oszlopában lévő osztályzatot (Microsoft Excel). Mire kell ügyelni a segéd táblázat elkészítésénél?

A *keresőfüggvények** használatakor lényegében egy összetartozó adatszámítógépes elemének ismeretében keressük a másikat. Az *Index** és *Hol.van** (vagy *Match* [mecs]) függvények segítségével ezt a műveletet két lépésre bonthatjuk. Így sokkal rugalmasabb lehet a keresés, hiszen a segéd tábla helye és felépítése azt kevésbé befolyásolja.

Példaként vegyünk egy táblázatot, amelynek első oszlopa a magyarországi városokat, második oszlopa pedig azok irányítószámát tartalmazza. Legyen feladatunk egy adott város irányítószámának visszakeresése.

A *Hol.van* függvény megkeresi, hogy egy oszlop melyik sorában van a keresett elem. Esetünkben például a *Hol.van*("Kecskemét"; A2:A6; 0) függvény értéke 3, mivel *Kecskemét* a városok A2:A6 oszlopában a harmadik sorban van. (A harmadik paraméter, a 0 azt jelenti, hogy a városok nincsenek névsorba rendezve.)

Az *Index* függvény pont fordítva működik. Megadja, hogy milyen érték van az oszlop egy megadott sorszámú sorában. Például az *Index*(B2:B6; 3) függvény értéke 6000, mivel az irányítószámok B2:B6 oszlopában a harmadik helyen 6000 áll.

Ha a két függvényt kombináljuk, akkor az

$$= \text{Index}(B2:B6; \text{Hol.van}(\text{"Kecskemét"; A2:A6; 0}))$$

képlet belső függvénye megtalálja, hogy *Kecskemét* a 3. sorban van, külső függvénye pedig azt, hogy a 3. sorban lévő irányítószám 6000. (Természetesen a végleges változatba a város neve helyett az azt tartalmazó cella címének kell kerülnie.)

	A	B	C	D	E	F
1	Város	Irsz				
2	Debrecen	4000		Keresett		
3	Szolnok	5000		Város	Irsz	
4	Kecskemét	6000		Kecskemét	6000	
5	Szentendre	2000				
6	Hatvan	3000				

Az **INDEX** és a **HOL.VAN** függvények használata egy város irányítószámának megkeresésére (Microsoft Excel)

Készítsük el az ábrán látható táblázatot!

A függvények összefoglalása

ÁTLAG (*tartomány*): kiszámolja a *tartományban* lévő, nem üres cellák átlagát.

SZUM (*tartomány*): összeadja a *tartomány* elemeit.

MAX (*tartomány*): kiválasztja a *tartomány* legnagyobb elemét.

MIN (*tartomány*): kiválasztja a *tartomány* legkisebb elemét.

GYÖK(*cella*): megadja a *cellában* lévő szám négyzetgyökét.

MA(): kiírja az aktuális dátumot.

DARABTELI (*tartomány; érték*): megszámlolja, hogy hányszor fordul elő az *érték* a *tartományban*.

HA (*feltétel; igaz_ág; hamis_ág*): ha a *feltétel* igaz, akkor végrehajtja az *igaz_ágban* lévő utasítást, egyébként pedig a *hamis_ágban* lévő utasítást.

FKERES (*érték, tartomány, oszlop*): megkeresi az *értéket* a *tartomány* első oszlopában, és visszaadja annak sorából az *oszlopban* lévő adatot. A *tartományt* az első oszlop szerint rendezni kell. Ha az *érték* nem szerepel az első oszlopban, akkor az utolsó, még nála kisebb értéknél áll meg a keresés.

VKERES: az FKERES párja; a keresést nem oszlopokban, hanem sorokban végzi.

INDEX (*tömb; sorszám*): Megadja az adott *tömb* adott *sorszámú* elemét.

HOL.VAN (*érték; tömb; 0*): megadja, hogy az *érték* hányadik eleme a *tömbnek*.

(A *tömbnek* nem kell rendezettnek lennie.)

Összefoglaló kérdések, feladatok

1. Milyen esetekben érdemes inkább táblázatkezelőt használni papír helyett? Mondjunk példákat arra is, amikor jobban járunk egy kockás papírral!

2. Nyissuk meg az általunk használt táblázatkezelőt, és jelöljük ki egy munkalapon a B2:C4 tartományt, a 4. sort, a D oszlopot, az egész táblázatot, illetve a B2:C4;D5:H6 nem összefüggő cellatartományt!

3. Mi a különbség a táblázat, a munkalap és a munkafüzet között?

4. Milyen adatok lehetnek egy cellában? Hogyan igazítja ezeket a táblázatkezelő programunk?

5. Soroljuk fel a matematikaórán tanult statisztikai fogalmakat! Milyen függvénnyel lehet ezeket megvalósítani a táblázatkezelőben? Hogyan tudjuk megvalósítani őket függvények használata nélkül?

6. Mi a különbség az abszolút és a relatív cellahivatkozás között? Mondjunk egy-egy példát is!

7. Készítsük el az általunk használt táblázatkezelővel a világ népességét tartalmazó táblázatot, és formázzuk meg! Ábrázoljuk Európa lakosságának változását oszlopdiagramon, illetve vonaldiagramon! Készítsük el a megfelelő trendvonalat is! Ábrázoljuk kör-diagramon a világ népességének várható megoszlását 2050-ben!

8. Mikor célszerű az adatokat oszlopdiagramon, összevont oszlopdiagramon, kör-diagramon, illetve vonaldiagramon ábrázolni? Mondjunk mindegyikre egy-egy példát is!

9. Készítsünk statisztikát arról, hogy az egyes tagországok milyen arányban képviselik magukat az Európai Parlamentben! Vegyük fel egy táblázatba a tagországok nevét, népességét és az adott ország képviselőinek számát! (Ez utóbbit az Európai Parlament hivatalos honlapján megtaláljuk.)

a) Ábrázoljuk kördiagramon, hogyan oszlik meg a képviselők száma országonként!

b) Ábrázoljuk sávdiaagramon, hogy az egyes tagországokban hány lakos jut egy-egy parlamenti képviselőre!

10. Készítsünk táblázatot, amely rögzíti, hogy melyik tanuló melyik hónapban mennyi osztálypénzt fizetett be! Számoljuk ki képlettel, hogy az egyes tanulók összesen mennyi díjat fizettek, és határozzuk meg a havonta átlagosan, illetve az összesen befolyt díj összegét is! Egészítsük ki a táblázatot egy sorral, amely az osztálypénztárban lévő összeget tartalmazza az adott hónapban!

11. Készítsünk táblázatot, amely megadja, hogy hány napot éltünk eddig! Mikor voltunk (leszünk) 1000, 10000, illetve 100000 naposak? Hány nap van még a tanév végéig? Definiáljunk az adatok megjelenítéséhez egy saját számformátumot, amely így jeleníti meg az eredményt: 6 nap!

12. Az iskolai versmondó versenyen háromtagú zsűri mindegyik tagja 1-től 10 pontig értékeli a versenyzőket. Az adatokat táblázattal összesítjük, a megfelelő oszlopok: a versenyző neve, az első, második, illetve harmadik zsűritag által adott pontok. Egészítsük ki a táblázatot egy oszloppal, amely megadja a versenyző összpontszámát! Emeljük ki piros színnel a 10-es pontszámokat! Mely tanulók jutnak tovább, ha a továbbjutás feltétele az, hogy a versenyző legalább 15 pontot érjen el? Mely tanulók jutnának tovább akkor, ha a továbbjutás feltétele az lenne, hogy a versenyző minden zsűritagtól legalább 5 pontot kapjon?

13. Egy táblázat az érettségiző tanulók nevét és matematika-pontszámát tartalmazza. Határozzuk meg, hogy melyik tanuló milyen érdemjegyet kap az érettségi vizsgán! A feladat megoldásához használjunk segéd táblát! (A szükséges adatokat keressük meg az interneten!)

14. Táblázatunk egy cég dolgozóinak éves bérét tartalmazza. Mennyi személyi jövedelemadót kell fizetniük? (Az aktuális személyi jövedelemadó táblát keressük meg az Adóhivatal honlapján!)

Bevezetés az adatbázis-kezelésbe

Adatkezelés táblázatkezelőkkel

Az adattábla

Az *adattábla** az adatfeldolgozás szempontjából azonos módon leírt objektumok tulajdonságait tartalmazza. Például, ha egy áruház részére készítünk adattáblát, akkor annak minden sora egy-egy terméket azonosít (vonalkód, megnevezés, egységár stb.). Ezek között egyaránt lehetnek konzervek és cipők, az áruház szempontjából ezek a cikkek egyformán írhatók le. Világos, hogy a konzervgyárban és a cipőgyárban nem ugyanazokat az adatokat használják, hiszen ott más és más az adatfeldolgozás célja.

Az adattábla *csak az egyes objektumok tulajdonságait* tartalmazhatja (pl. egységár, készlet), de az ezekből számítható összefüggéseket nem (pl. az egységár és a készlet szorzataként kapott összeg). Az adattábla tehát meglehetősen *speciális táblázat*, ezért az adatbázis-kezelésben a táblázatkezeléstől eltérő elnevezéseket használnak.

	A	B	C	D	E
1	Rendszám	Típus	Szín	Gyártév	Érték
2	RTL-001	Lada	Kék	1994	2 345 000 Ft
3	TNT-002	Lada	Piros	1994	45 000 Ft
4	ABC-187	Lada	Piros	1996	100 000 Ft
5	SAT-111	Lada	Piros	1994	356 723 Ft
6	KSJ-736	Lada	Piros	1992	2 342 000 Ft
7	AXW-453	Mazda	Lila	1991	400 000 Ft
8	BUU-234	Mazda	Lila	1992	630 000 Ft
9	ABA-069	Mazda	Piros	1993	780 234 Ft
10	CSE-555	Mazda	Zöld	1994	615 041 Ft
11	VFG-756	Mazda	Zöld	1994	890 000 Ft
12	IIA-221	Skoda	Kék	1994	616 105 Ft
13	ABA-134	Skoda	Kék	1992	780 000 Ft
14	BOR-002	Skoda	Kék	1996	3 452 000 Ft
15	QWR-629	Skoda	Zöld	1994	62 000 Ft
16	VDS-888	Skoda	Zöld	1994	4 678 124 Ft

Autókat tartalmazó adattábla. Az első sor a mezőneveket, a többi pedig egy-egy rekordot tartalmaz.

Az ábrán egy használtautó-kereskedés részére készült táblázatot látunk. Az *adattábla oszlopait mezőknek**, *sorait* (az első kivételével) *rekordoknak** nevezjük. A rekordok egy-egy autót, a mezők pedig egy-egy tulajdonságot adnak meg. A *táblázat első sora a mezők nevét tartalmazza*.

Ha a mezők sorrendjét vagy a rekordok sorrendjét felcseréljük, akkor a tárolt adatok helyessége nem sérül.

Az adattáblát a nevével és a mezők felsorolásával szokták megadni:

Autók (Rendszám, Típus, Szín, Gyártév, Érték).

Azt a mezőt, amely bármely két rekord esetén különböző értéket vesz fel, **kulcsnak*** nevezzük. Autók esetében ilyen mező lehet a rendszám, az emberek esetében a személyi szám, a taj-szám vagy az adószám.

Elképzelhető, hogy a kulcs nem egyetlen mezőből áll. Például egy iskolai tanuló adatait tartalmazó adattáblában a tanuló a neve, a születési helye, a születési ideje és az anyja leánykori neve azonosítja, azaz négy mező. Az ilyen kulcsot **összetett kulcsnak*** nevezzük.

Egy adatbázis általában több adattáblát tartalmaz, amelyek között összefüggések is lehetnek. Például egy iskolai adatbázisnak biztosan tartalmaznia kell a tanulókat, az osztályokat és a tanárok adattábláját is. Világos, hogy ezek nem függetlenek egymástól. A táblázatkezelők azonban az adattáblák közötti kapcsolatokat nem tudják kezelni.

Rendezés

A rendezés gyakori feladat, mivel segítségével könnyen megkereshetünk egyszerű feltételeknek megfelelő rekordokat az adattáblában. Így például az Érték mező szerinti rendezéssel azonnal megkaphatjuk a három legdrágább autót. Ha pedig a Mazdákat szeretnénk leválogatni, akkor elég rendezni a táblát a Típus mező szerint.

A rendezést minden táblázatkezelő támogatja, és általában több rendezési kulcsot is megadhatunk. Ha például az első rendezési kulcs a Típus, a második a Szín, akkor az azonos típusú autókat a táblázatkezelő szín szerint rendezi. Természetesen minden kulcs esetén megadhatjuk, hogy az adott kulcs szerint növekvően vagy csökkenően legyenek rendezve a rekordok.

A rendezés a táblázatkezelőkben a rekordok tényleges (fizikai) átrendezésével jár.

	A	B	C	D	E
1	Rendszám	Típus	Szín	Gyártév	Érték
2	RTL-001	Lada	Kék	1994	2 345 000 Ft
3	TNT-002	Lada	Piros	1994	45 000 Ft
4	ABC-187	Lada	Piros	1996	100 000 Ft
5	SAT-111	Lada	Piros	1994	356 723 Ft
6	KSI-736	Lada	Piros	1992	2 342 000 Ft
7	AXW-453	Mazda	Lila	1991	400 000 Ft
8	BUU-234	Mazda	Lila	1992	630 000 Ft
9	ABA-069	Mazda	Piros	1993	780 234 Ft
10	CSE-555	Mazda	Zöld	1994	615 041 Ft
11	VFG-756	Mazda	Zöld	1994	890 000 Ft
12	IIA-221	Skoda	Kék	1994	616 105 Ft
13	ABA-134	Skoda	Kék	1992	780 000 Ft
14	BOR-002	Skoda	Kék	1996	3 452 000 Ft
15	QWR-629	Skoda	Zöld	1994	62 000 Ft
16	YDS-888	Skoda	Zöld	1994	4 678 124 Ft

Rendezés

Rendezze: Típus Emelkedő Csökkenő

Majd: Szín Emelkedő Csökkenő

Azután: Érték Emelkedő Csökkenő

Az adattartományban: Van sorfejező Nincs sorfejező

Egyebek... OK Mégse

A rekordok rendezésére a táblázatkezelőkben általában három szempontot adhatunk meg (Microsoft Excel). Mire kell ügyelni az adatok kijelölésénél? Miért?

Adatok szűrése

Az adatok szűrése* azt jelenti, hogy megadunk egy logikai feltételt (szűrőfeltétel*), és a program csak azokat a rekordokat jeleníti meg, amelyek ennek a feltételnek eleget tesznek. Például az „1990 után gyártott kék Mazdák” szűrőfeltételt így írhatjuk le:

Gyártév>1990 AND Szín="Kék" AND Típus="Mazda".

A táblázatkezelésben a szűrés legegyszerűbb módja az *automatikus szűrés*, ekkor az egyes feltételeket elegendő csupán kiválasztani.

Az automatikus szűrőt a megfelelő menüponttal kapcsolhatjuk be, illetve ki. Bekapcsolt állapotában a kiválasztott mező neve mellett legördíthető egy lista, ahol megadhatjuk, hogy milyen tulajdonságú rekordokat keresünk. Ha egy adott mező szerinti feltételt beállítottunk, akkor már csak az annak megfelelő rekordok jelennek meg. Ezt tovább szűkíthetjük egy másik mező szerinti feltétel megadásával stb. A lista lehetőséget ad rendezési kulcs kiválasztására és a szűrőfeltétel törlésére is.

Szín	Gyártév	Érték
Kék	(Minden)	2 345
Piros	(Top 10...)	45
Piros	(Egyenl...)	100
Piros	(Üresek)	356
Piros	(Nem üresek)	2 342
Lila	1991	400
Lila	1992	630
Piros	1993	780
Piros	1994	780
Zöld	1996	615
Zöld		890
Kék		616
Kék	1992	780
Kék	1996	3 452
Zöld	1994	62
Zöld	1994	4 678

Egyéni automatikus szűrő

Sorok mutatása melyek értéke ebben az oszlopban:

nagyobb mint 1990

És Vagy

egyenlő
nem egyenlő
nagyobb mint
nagyobb vagy egyenlő mint
kisebb mint
kisebb vagy egyenlő mint
kezdődik

OK Mégsem

Az automatikus szűrő használata (Ability Spreadsheet)
Milyen szűrőfeltételt olvashatunk le az ábráról?

További műveletek

A fentieken túl a táblázatkezelők (általában az *Adatok* nevű menüben) további lehetőségeket is tartalmaznak, most csak kettőt említünk meg.

Gyakran előfordul, hogy a rekordok bizonyos csoportjairól kell statisztikai adatokat megadnunk, például ki kell számolnunk az autók összértékét típusonként. Ezt a műveletet támogatja a *Részösszegek* készítése lehetőség.

Az adattábla statisztikai elemzésével gyakran érdekes új összefüggéseket tárhatunk fel. Például megvizsgálhatjuk, hogy az eladott autók darabszáma hogyan függ a típustól és a színtől. Ehhez készíthetünk egy olyan táblázatot, melynek oszlopfejlécei a típusok, sorfejlécei a színek, a megfelelő cellákban pedig a darabszámok vannak. Az ilyen táblázatot kimutatásnak vagy *Pivot-táblának* nevezik.

Hálózatos környezetben nagy teljesítményű, szerver oldali adatbázis-kezelő rendszereket használnak. Ilyen pl. az *Oracle* [orekl] vagy a *Microsoft SQL* szerver.

A felhasználók igen gyakran nem magával az adatbázis-kezelő rendszerrel, hanem egy azzal készített *célprogrammal* találkoznak. Ilyenek például a számlázó, a könyvelő és a raktárkezelő programok is.

```
. use f:tanulok
. list all for nev>"H" .and. t1>2
```

Rekord#	NEU	SZUL	ANEUE	T1	T2	T3	T4
4	Skrapits Andras	03/03/47	Schull Lidia	5	5	5	4
5	Kosa Emma	07/09/63	Gazdag Erzsebet	4	5	4	4
8	Takfalvi Ibolya	04/02/70	Torma Monika	3	4	4	4
9	Nemeth Lajos	05/05/62	Csintalan Judit	4	5	4	3
13	Lengyel Istvan	06/15/56	Ramos Marta	4	5	4	4

Adatbázis-kezelés utasítások beírásával a dBase-ben.

Az adatbázis-kezelők objektumai

Egy modern adatbázis-kezelő rendszer nemcsak az adattáblákat tárolja, hanem más, az adatok megjelenítéséhez, kereséséhez stb. szükséges **objektumokat is**. Ennek megfelelően, ha megnyitunk egy adatbázist, akkor nem az adatok jelennek meg, hanem egy olyan munkaterület, ahol ezek között választhatunk.

Foglaljuk össze a leggyakoribb objektumokat!

Az adatokat a **táblák*** tárolják, táblázatos formában. Oszlopait mezőknek, sorait rekordoknak nevezzük. A celláknak külön azonosítójuk nincs.

A **lekérdezések*** segítségével leválogathatjuk, törölhetjük, módosíthatjuk az adattáblákban tárolt adatokat. Ehhez általában egy logikai feltételt kell megadnunk.

Az **úrlapok*** az adatok képernyőn való megjelenítését és módosítását, a **jelentések*** pedig az adatok megformázott kinyomtatását teszik lehetővé.

Az adattáblák

Eltérően a táblázatkezelőktől, az adattáblák szerkezetét egy külön táblaszerkesztővel hozhatjuk létre, illetve azzal módosíthatjuk. Az **adattábla létrehozásánál meg kell adnunk a mezők nevét és típusát, valamint megadhatjuk egyéb jellemzőit is**.

A **mezők típusa* két dolgot ad meg: milyen adatokat tárol, és azokkal milyen műveleteket végezhetünk**. A leggyakoribb típusok a következők:

Égész szám*. Nevének megfelelően egész számok tárolására szolgál. Az ábrázolandó számok nagyságától függően többféle lehet (pl. báj, egész, hosszú egész).

Valós szám*. A számokat lebegőpontosan (tehát lényegében normálalakban) tárolja. Az ábrázolandó számok nagyságától függően lehet egyszeres vagy dupla.

Decimális szám*. A számokat fixpontosan tárolja. A számjegyek száma, ezen belül a tizedeshelyek száma rögzített, a számokat lényegében egész számként ábrázolja.

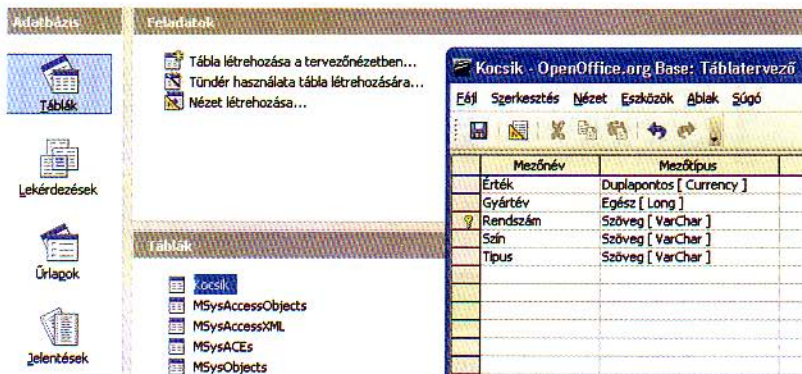
Dátum/idő*. A táblázatkezelésben tanultakhoz hasonló módon tárolja a dátumot és az időt, de az ottaninál nagyobb tartományban.

Igen/nem vagy logikai*. Logikai értékeket tárol, logikai műveleteket végezhetünk vele.

Szöveg*. Szöveget tárol, a karakterek maximális számát előre meg kell adni.

Számláló*. Értéke minden rekord beírásával eggyel nő, értéke kezdetben 1. Főleg kulcs definiálására használják.

A **kulcsot** tartalmazó mezőt külön meg kell jelölnünk, a kialakult gyakorlat szerint általában egy kis kulcs ikon jelzi.



Az adatbázis objektumai közül ikonokkal választhatjuk ki a megfelelőt.

Egy adattábla szerkezetét a táblatervezővel módosíthatjuk (OpenOffice.org Base [béz]).

Készítsünk egy új adatbázist, melybe hozzuk létre a 163. oldalon lévő táblát Kocsik néven!

Mezők indexelése

Érdekes lehetőség a **mezők indexelése***, amely a **rendezést segíti**. A nagyméretű, esetleg több millió rekordot tartalmazó adattáblák fizikai átrendezése ugyanis nem célszerű, helyette a **szükséges mezőkhöz ún. indextáblát készítenek**. Ez az adott mező szerint tartja nyilván a rekordok sorrendjét, listázáskor a program ennek megfelelően jeleníti meg az adatokat. Az indextábla az adatok bevitelekor, törlésekor, módosításakor folyamatosan frissül. Így segítségével akár az is elérhető, hogy a mezőnek *ne legyen azonos értéke két különböző rekord esetén*.

Az indextábla az adatbázis-kezelők egy részénél külön fájlba kerül (pl. dBase), míg másoknál az adatbázist tartalmazó egyetlen fájl része (pl. Access).

Adattáblák készítése

Egyelőre tételizzük fel, hogy a tárolandó mezők leírását, vagy valamilyen formában magát a táblát készen kapjuk. (Az adatbázis megtervezése, a hozzá tartozó objektumok célszerű definiálása komoly előismereteket igényel, ezeket később körvonalazzuk.)

Ha ismerjük a mezőket, akkor azok típusának, méretének és egyéb paramétereinek meghatározásához a következőket kell figyelembe vennünk.

Milyen konkrét adatokat kell majd tárolnunk? Nevek tárolásánál például gondolnunk kell azokra is, akiknek hosszabb a vezetéknevük, vagy viselik házastársuk nevét is, vagy például több utónevük van.

Milyen műveleteket kell majd velük végezni? Például az irányítószámot szöveggént érdemes tárolni, mert nem kívánjuk őket összeadni. Ugyanakkor felmerülhet, hogy az 1-el kezdődőkből szövegműveletekkel kiemeljük a két középső számjegyet. Ezek ugyanis a budapesti irányítószámok, ahol a két középső jegy a kerületet adja meg.

Valamivel egyszerűbb a helyzet, ha az adatokat például egy formázatlan text fájlban kaptuk. Ekkor az adatokat az adatbázis-kezelő rendszer varázslójával célszerű importálnunk a táblázatkezelésben megismert módon. Jóllehet a varázsló sok beállítást elvégez helyettünk, ennek ellenére érdemes azokat alaposan átgondolni.

A *kulcs megválasztására* is figyelniünk kell. Egy közlekedési szabálytalanságokat tartalmazó táblában például a rendszám nem alkalmas kulcsnak, hiszen egy autó többször is szabálytalankodhat. Ha más lehetőségünk nincs, akkor egy számláló típusú mezőt érdemes felvenni azonosítóként.

Műveletek a táblákkal és az adatokkal

A táblákkal végzett műveletek, tehát az oszlopok szélességének beállítása, az oszlopok vagy sorok felcserélése, a rekordok rendezése a táblázatkezelésben megszokott módon történik. Új rekordot általában a tábla végére írhatunk be.

Rekordok törlése és módosítása egyszerűen az adatok átírásával történik. **Ha módosítjuk az adatokat, azok azonnal bekerülnek a fájlba.** Tábla mentésekor csupán a beállításokat mentjük. Az adatbázis helye is fix, így a *Fájl* menüben legfeljebb a másolat létrehozására találunk lehetőséget, a mentésre nem.

Az adatok szűrése

Az adatbázis-kezelés legfontosabb feladata az adott tulajdonságú rekordok listázása. Erre az egyik legegyszerűbb módszer a *szűrés**. A szűrőfeltétel megadását a megfelelő menüpont kiválasztásával kell kezdenünk. Magát a szűrőfeltételt a táblázatkezelésben megismert automatikus szűréshez hasonlóan kell megadnunk. Az egy sorba írt feltételeket itt is a logikai ÉS művelet köti össze. A táblázatkezelők automatikus szűrőjétől eltérően azonban a feltételt pl. fülek segítségével VAGY művelettel is bővíthetjük.

A szűrőfeltétel a tábla beállításaiival együtt menthető. Be- illetve kikapcsolását az eszköztár megfelelő ikonjával végezhetjük.



Szűrőfeltétel megadása Microsoft Accessben. A feltétel a következő:
 (Típus="Skoda" ÉS Szín="Piros") VAGY (Típus="Mazda" ÉS Szín="Lila" ÉS Gyártév>2005).
 Mely rekordok jelennek meg a 163. oldalon lévő adattáblából szűréskor?

A lekérdezések

Adott tulajdonságú adatok listázásának másik módja a lekérdezés. A *lekérdezés** **ese-tében a feltétel nem egy adott tábla tulajdonságaként, hanem önálló objektumként, saját névvel ellátva menthető el.** Így egy táblához több lekérdezés is készíthető; sőt, egy lekérdezés több tábla adatait is megjelenítheti.

Ha kiválasztjuk valamely lekérdezést és azt megnyitjuk, a **lekérdezés** a hozzá tartozó táblákból **az előírt feltételeknek megfelelő adatokat egy táblázatban jeleníti meg.** A lekérdezés tehát úgy viselkedik, *mintha* létrehozna egy új táblát. Ez az ún. *virtuális tábla* csupán a memóriában jön létre, és csak addig létezik, amíg a lekérdezés meg van nyitva. A lekérdezés használatakor viszont a virtuális tábla a többi táblához hasonlóan kezelhető, pl. rendezhető, szűrhető, stb.

A **lekérdezéshez gyakran műveletek is tartozhatnak**, pl. a logikai feltétellel megadott adatok megjelenítése, módosítása, vagy rekordok törlése. Ilyenkor rendre választó, módosító, vagy törlő lekérdezésről beszélünk.

A **lekérdezések előállítására két lehetőség van.** Az egyik a **QBE-rács*** [ejtsd: qbe] (*Query By Example* [kveri báj igzempl]), amikor a feltételt *vizuálisan* állítjuk össze, a másik az **SQL nyelv*** (*Structured Query Language* [sztráksörd kveri lengvidzs]), amikor a feltételt *matematikai logikai úton* adjuk meg. Fontos azonban tudni, hogy a QBE-rács is SQL nyelvű utasítást állít elő.

A **QBE-rács kitöltése** hasonló módon történik, mint a szűrőfeltétel megadása. A szükséges mezőket azonban külön fel kell vennünk, összetett feltétel esetén akár többször is. Lehetőségünk van továbbá annak előírására, hogy a megjelenő lista az adott mező szerint rendezett legyen-e, illetve megjelenjen-e. A mezők a QBE-rácsban feltüntetett sorban fognak megjelenni a virtuális táblában.

Az **SQL a lekérdezések szabványos nyelve.** Amikor pl. böngészőn keresztül egy központi adatbázisból kérdezzük le adatokat, akkor gépünkről SQL nyelven megy el a kérés. Az adatbázis-kezelő rendszer ennek alapján készíti el, majd küldi vissza gépünkre a lekérdezés eredményét egy táblázat formájában. Mivel az SQL nemzetközileg elfogadott szabvány, így különböző rendszerek is képesek rajta keresztül együttműködni.

Mező	Rendszám	Tipus	Szín	Gyártév	Érték
Alias					
Tábla	Kocsik	Kocsik	Kocsik	Kocsik	Kocsik
Rendezés					
Látható	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Függvény					
Feltétel		'Lada'	'Piros'		

```

PirosLada
-----
Eájl Szerkesztés Nézet Beszúrás Eszközök Ablak Súgó
[Icons]
SELECT `Rendszám`, `Tipus`, `Szín`, `Gyártév`, `Érték`
FROM `Kocsik` `Kocsik`
WHERE (( `Tipus` = 'Lada' AND `Szín` = 'Piros' ))
    
```

Autók: PirosLada

Eájl Szerkesztés Nézet Eszközök Ablak Súgó

Rendszám	Tipus	Szín	Gyártév	Érték
TNT-002	Lada	Piros	1994	45 000,00 Ft
ABC-187	Lada	Piros	1996	100 000,00 Ft
SAT-111	Lada	Piros	1994	356 723,00 Ft
KSJ-736	Lada	Piros	1992	2 342 000,00 Ft

Lekérdezés megadása QBE-ráccsal, illetve SQL nyelven. A végeredmény ugyanaz (OpenOffice.org Base). Készítsük el a lekérdezést a Kocsik táblából az általunk használt adatbázis-kezelő rendszerben!

Számítások az adatbázisban

A számításokat az adattáblákban lekérdezések segítségével végezhetjük el. Attól függően, hogy mit szeretnénk meghatározni, két lehetőségünk van.

Az egyik lehetőség, hogy egy vagy több mezőből egy új mezőt kell kiszámítani minden rekord esetén. Ekkor a QBE-rácsban a *Mező* sorába kell írunk a megfelelő képletet.

A másik lehetőség, hogy az adatok bizonyos csoportjaira kell statisztikai számításokat végezni. Ekkor az *Összesítés* (vagy *Függvény*) sorban kell kiválasztanunk a szükséges függvényt, illetve megadnunk, ha az adott mező a csoportosítás alapja.

Mező:	Tipus	Érték
Tábla:	Kocsik	Kocsik
Összesítés:	Group By	Ávg
Rendezés:		Ávg

Az autók átlagértékének megadása típusonként. Az *Összesítés* sor megjelenítését a *Nézet* menüben kapcsolhatjuk be (MS Access).

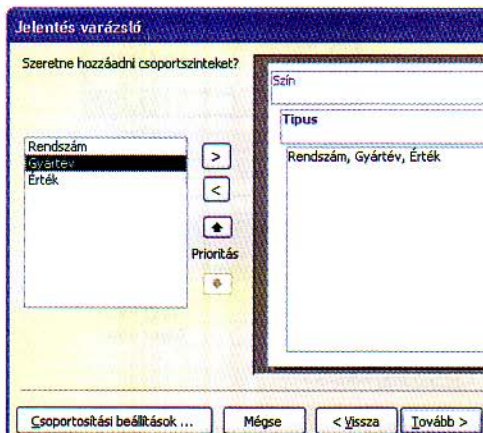
Mező:	megnevezés	készlet	egységár	Nettó érték
Tábla:	árucikk	árucikk	árucikk	
Rendezés:				
Mejelenítés:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

Az egyes árucikkek nettó értéke a készlet és az egységár szorzata. A *Nettó érték* egy új, ún. számított mező lesz a lekérdezésben. A felhasznált mezők nevét szögletes zárójelbe kell tenni (Microsoft Access).

Űrlapok és jelentések

Az *űrlapok** az adatok esztétikus megjelenítésére szolgálnak a képernyőn, a *jelentések** pedig a nyomtatón. Az űrlap az adatbevitel eszköze is lehet. Mindkettő épülhet adattáblára, de akár lekérdezésre is.

Egy űrlap vagy jelentés megtervezése meglehetősen időigényes folyamat, ezért az adatbázis-kezelők *varázslókat* tartalmaznak automatikus elkészítésükhöz. Bevett gyakorlat, hogy az űrlapot vagy jelentést varázslóval elkészítjük, majd utólag módosítjuk.



Kocsik			
Típus		Lada	
Szín	Kék		
	Rendszám	Gyártás	Érték
	RTL-001	1994	2 945 000 Ft
Szín	Piros		
	Rendszám	Gyártás	Érték
	KSJ-736	1992	2 942 000 Ft
	SAT-111	1994	355 723 Ft
	ABC-167	1996	100 000 Ft
	TNT-002	1994	45 000 Ft
Típus		Mazda	
Szín	Lila		
	Rendszám	Gyártás	Érték
	BUJ-234	1992	630 000 Ft

Jelentés készítése *Jelentés varázslóval* és a kész jelentés (Microsoft Access) Készítsük el az ábrán látható jelentést a *Kocsik* táblából!

Kapcsolat a táblák között

Példaként tekintsünk egy könyvtári adatbázist. Ekkor két adattáblánk van: a könyvek adatait tartalmazó Könyv (Leltári_szám, Szerző, Cím, Olvasó) és az olvasók adatait tartalmazó Olvasók (Olvasójegy, Név) tábla.

Alapfeladat, hogy meg tudjuk mondani, melyik könyv kinél van. Példánkban ezt úgy oldhatjuk meg, hogy a Könyvek tábla Olvasó mezőjébe beírjuk az olvasó olvasójegyének számát. Ekkor ez a **közös mező (ún. kapcsolómező*) lehetővé teszi a két tábla közötti kapcsolatot létrehozását**. A kapcsolatot ténylegesen egy külön ablakban hozhatjuk létre,



Az adatbázis-kezelő a kapcsolómezőket vonallal köti össze.

az Access esetében pl. úgy, hogy a Könyvek tábla Olvasó mezőjét egérrel az Olvasók tábla Olvasójegy mezőjére húzzuk. A program a kapcsolatot egy vonallal jelzi.

A kapcsolatnak két szerepe van. Egyfelől **lehetővé teszi, hogy a két táblából egyetlen lekérdezéssel jelenítsük meg az összetartozó adatokat**. (Például készíthetünk egy lekérdezést, amely megadja, hogy melyik könyv kinél van.)

Másfelől előírhatjuk, hogy az adatbázis-kezelő „vigyázzon” a kapcsolatra. Így például ne lehetne addig törölni egy olvasót, amíg van nála könyv, vagy ne lehessen módosítani olvasójegyének számát az Olvasók táblában, míg ez az adat szerepel a Könyvek táblában is stb. Ezt az adatbázis-kezelésben a **hivatkozási integritás** megőrzésének nevezik. A hivatkozási integritást további előírásokkal finomíthatjuk. Például, ha egy olvasót törölünk, akkor a nála lévő könyvek is törlődjenek (**kaszádolt törlés**).

Leltári_szám	Szerző	Cím	Olvasó
A1	Karam Ella	Alfa	B2
A2	Vaj Ákos	Béta	B1
A3	Szemte Lenke	Gamma	
A4	Kala Pál	Delta	B2

Olvasójegy	Név
+ B1	Güzü Géza
+ B2	Gipsz Gábor
+ B3	Hernyó Helga

Mező:	Leltári_szám	Szerző	Cím	Olvasójegy	Név
Tábla:	Könyvek	Könyvek	Könyvek	Olvasók	Olvasók
Rendezés:					
Megjelenítés:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Feltétel:					
vagy:					

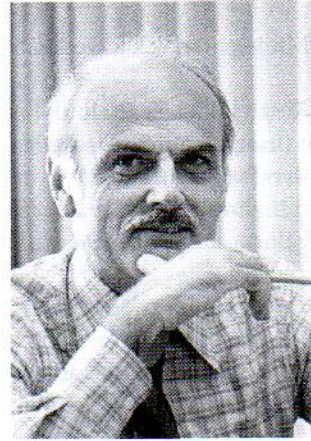
Leltári_szám	Szerző	Cím	Olvasójegy	Név
A2	Vaj Ákos	Béta	B1	Güzü Géza
A1	Karam Ella	Alfa	B2	Gipsz Gábor
A4	Kala Pál	Delta	B2	Gipsz Gábor

A Könyvek és az Olvasók táblát az olvasójegy száma kapcsolja össze. Ez lehetővé teszi, hogy olyan lekérdezést készítsünk, amely egyszerre jeleníti meg az olvasók és a náluk lévő könyvek adatait (Microsoft Access). Kövessük végig az ábrán a Könyvek és az Olvasók adattábla adatainak elemzésével, hogy hogyan jött létre a választó lekérdezés táblázata!

Adatmodellek

Egy „élés” adatbázis több tucat adattáblát és több millió rekordot is tartalmazhat, gondoljunk például a Magyarország lakosságát nyilvántartó adatbázisra. Egy ilyen adatbázis szerkezetének utólagos módosítása rendkívül költséges, ezért **létrehozása tudatos tervezést igényel.**

A tervezés első lépése, hogy szoftverektől és az adatok konkrét értékeiktől függetlenül felállítanak egy adatmodellt. Az **adatmodell** azt tartalmazza, hogy milyen egyedek mely tulajdonságait kell tárolni, és ezek között milyen kapcsolatok vannak. Könyvtári példánkban az egyedek a könyvek, illetve az olvasók. A tárolt tulajdonságok az olvasók esetében az olvasójegy száma és az olvasó neve. A **kapcsolat*** az egyedek közötti viszonyt fejezi ki, példánkban azt, hogy melyik könyv melyik olvasónál van. Csak az adatmodell megtervezése után következhet az adatbázis konkrét megvalósítása valamely adatbázis-kezelő rendszerben.

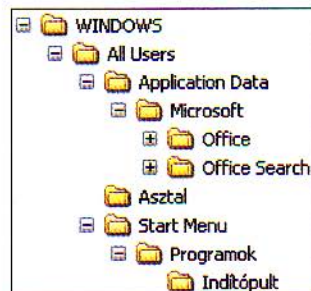


Edgar F. Codd [edgár kodd] (1923-2003), aki az IBM munkatársaként kidolgozta a relációs adatmodellt.

Hagyományosan háromféle adatmodellt különböztetnek meg, ezek a hierarchikus, a hálós és a relációs.

A **hierarchikus és a hálós adatmodell** a kapcsolatokat az adatbázisba beépítve tárolja. Hierarchikus például az operációs rendszer mappaszerkezete, ahol egy fájl vagy mappát a gyökérből kiindulva, egymásba ágyazott mappákon keresztül érünk el (Windows\All Users\Sart menü\Programok\Indítópult). Tehát, ha le akarjuk kérni a fájl vagy mappa tulajdonságait, akkor előbb a gyökérből indulva a hivatkozásokon át el kell hozzá jutnunk. **Hálós szerkezetű például a világháló**, ahol a weblapokat a lapokon lévő linkek kapcsolják össze. Mindkét esetben fizikailag tároljuk a kapcsolatokat, és azt egy adat keresésekor ki is használjuk.

A **relációs adatbázisokban*** a kapcsolatot **külön nem tároljuk.** A Könyvek és az Olvasók tábla nincs közvetlenül hivatkozásokkal összekötve, a **kapcsolómező csupán a kapcsolat lehetőségét teremti meg.** Ennek értékei egyszerűen módosíthatók, amikor az olvasó visszahozza a könyvet, és egy másik elviszi. A relációs adatbázisok általában nagyobb hardverigényt támasztanak a számítógéppel szemben, cserébe nagy szabadságot adnak kapcsolatok igény szerinti kialakításában és módosításában.



A mappaszerkezet hierarchikus felépítésű. Mi ennek az előnye?

Összefoglaló kérdések, feladatok

1. Mit jelentenek a következő fogalmak: adattábla, mező, rekord, kulcs?
2. Készítsük el az általunk használt táblázatkezelővel a fejezet első ábráján lévő adattáblát (hasznáلتautó-kereskedés), és végezzük el vele a következő műveleteket:
 - a) Rendezzük az autókat gyártási évük, azon belül értékük szerint!
 - b) Irányított szűrővel válogassuk ki a piros Ladákat!
 - c) Válogassuk ki az 1994 és 1996 között gyártott piros autókat!
3. Tervezzünk egy adattáblát az osztály félévi jegyeinek összesítésére!
 - a) Válogassuk ki azokat a tanulókat, akiknek az átlaga nagyobb, mint 4,5!
 - b) Rendezzük a tanulókat a matematikaosztályzatok alapján!
 - c) Hogyan alakulnak a kémia-, illetve a fizikaosztályzatok átlagai a matematikaosztályzatok szerinti csoportokban?
4. Hasonlítsuk össze a táblázatkezelőket és az adatbázis-kezelőket a következő szempontok szerint: tábla mérete, rendezés módja, adatokhoz való hozzáférés, táblák közötti kapcsolat lehetősége!
5. Írjuk le a matematikai logika eszközeivel a következő szűrőfeltételeket:
 - a) piros Ladák,
 - b) 10 évnél öregebb, 1.000.000 Ft-nál olcsóbb zöld Mazdák,
 - c) kék Ladák és zöld Skodák!
6. Mit jelentenek a következő fogalmak az adatbázis-kezelésben: indestábla, SQL nyelv, lekérdezés, kapcsolat?
7. Tervezzünk adatbázist EU néven egyetlen táblával (pl. Tagok), amely a tagországok nevét, alapterületét, népességét és a belépés évét tartalmazza!
 - a) Mi legyen a táblában lévő mezők neve és típusa?
 - b) Hogyan határozhatjuk meg az egyes tagországok népsűrűségét?
 - c) Milyen módon tudnánk bemutatni az Európai Unió bővülését?
8. Tervezzünk adatbázist egy iskolában a tanulók és osztályok kezelésére! Milyen egységeket, tulajdonságokat kell tárolnunk, és milyen kapcsolatot kell megvalósítanunk?
9. Készítsük el a fejezetben lévő könyvtári adatbázist! Definiáljuk, majd töltsük fel adatokkal a két táblát! Készítsünk jelentést a könyvtárban lévő könyvekről! Készítsük el azt a lekérdezést, amely megjeleníti, hogy melyik könyv kinél van! Milyen lehetőséget kínál az általunk használt adatbázis-kezelő annak kivédésére, hogy olyan olvasót töröljünk, akinél van könyv?

Algoritmizálás

Programozási előismeretek

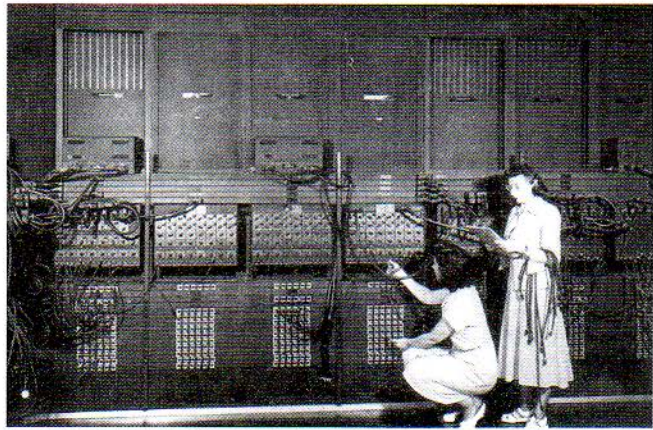
Az első számítógépek programozása

Ahhoz, hogy jobban megértsük a számítógéppel történő problémamegoldás lehetőségeit és határait, meg kell ismerkednünk a programok működésével is.

Mint láttuk, az első programozható számológépet CHARLES BABBAGE [császár bebedzs] tervezte meg a XIX. században. Jóllehet az Analitical Engine [enelitiköl endzsin] nem készült el, Babbage asszisztensnője ADA AUGUSTA LOVELACE [éda augusztalavlész] (1815-1852) megírta hozzá az első programokat. (1979-ben az Egyesült Államok Hadügyminisztériuma egy modern programozási nyelvet fejlesztett ki, melyet tiszteletére Adának nevezett el.)



Ada Augusta Lovelace, Lord Byron [bájrón] lánya, az első programozó



Az ENIAC programozása a korai telefonközpontok kezeléséhez hasonlóan, dugaszolással történt. Milyen előnyei és hátrányai voltak ennek a módszernek a belső programtároláshoz képest?

Az első elektronikus számítógépeknél, így pl. az ENIAC-nál is a végrehajtandó műveleteket egy dugaszolótábla segítségével adták meg. Ezt a megoldást *külső programtárolásnak* nevezik. Neumann János ismerte fel 1945-ben, hogy célszerűbb a programokat az adatokhoz hasonlóan a számítógép memóriájában tartani. A *belső programtárolás* elvét a következő számítógépnél (EDVAC, 1951) már meg is valósították. **Ma gyakorlatilag minden számítógép Neumann-elvű, így a végrehajtandó utasításokat adatként a memóriában tárolja.** (A többi Neumann-elvvel az első fejezetben találkozhattunk.)

Programozási nyelvek és fejlesztői környezetek

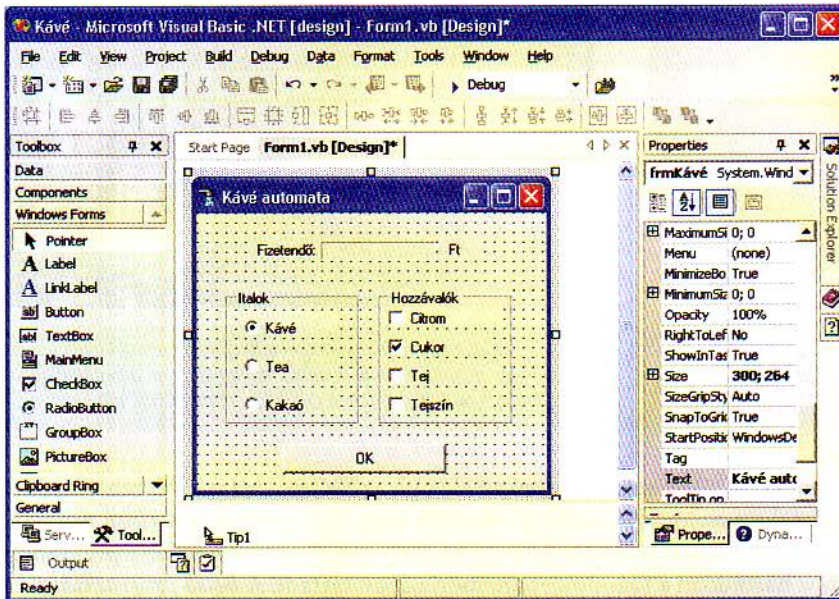
Mint láttuk, a számítógép belsejében a központi feldolgozó egység (CPU) végzi a számításokat és vezérli a számítógép többi egységét. A **központi feldolgozó egység által közvetlenül értelmezhető adatsort gépi kódnak* nevezük**. Mivel a központi feldolgozó egység csak kettes számrendszerben ért, és csak nagyon egyszerű műveleteket ismer, az elvégzendő feladatok megadása gépi kódban nagyon nehézkes.

Hamar felismerték tehát, hogy célszerű ún. fordítóprogramot készíteni. A **fordítóprogram* az ember számára olvasható és értelmezhető utasítások sorozatát gépi kódra alakítja át**. Az utasítások megadásához azonban egy új, a használt nyelveknél lényegesen egyszerűbb felépítésű ún. programozási nyelvet kellett létrehozni. A **programozási nyelv* azokat az utasításokat és szabályokat rögzíti, amelyekkel megadhatjuk az elvégzendő műveletsort a fordítóprogram számára**.

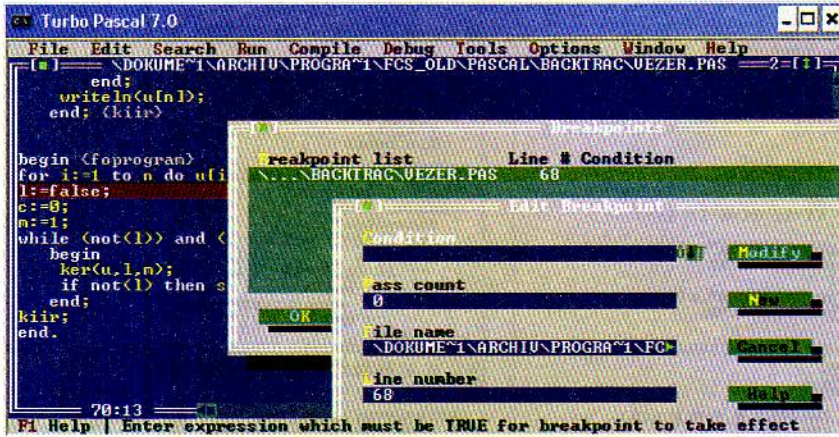
Az első programozási nyelvet, a Fortrant az 1950-es évek elején dolgozták ki. Az első Fortran fordítóprogram 1957-re készült el. Ma a programozási nyelvek száma több ezerre tehető. Néhány elterjedt programozási nyelv: Basic [bézik], Pascal [paszkál], C, Java stb.

A fordítóprogramok szerepét ma már átvették a **fejlesztői környezetek***. Ezek a **fordításon kívül számtalan egyéb szolgáltatást is nyújtanak**. Például szövegszerkesztőt az utasítások beírásához, hibakereső modult, a gépi kód optimalizálását stb.

Ugyanazt a programozási nyelvet akár különböző fejlesztői környezetek is érthetik, és egy adott fejlesztői környezet általában több nyelvet is tud kezelni. Néhány elterjedt fejlesztői környezet: Microsoft Visual Studio [vizsuöl sztyudio], Turbo Pascal, Borland Delphi [delfi].



A Visual Studio modern grafikus fejlesztői környezete a Windows rendszerhez



A nagy klasszikus, a Turbo Pascal fejlesztői környezete a DOS operációs rendszerhez

A mai fejlesztői környezetek nem feltétlenül állítanak elő közvetlenül gépi kódot.

Az *értelmezők (interpreterek*)* a programozási nyelven megírt kódot *soronként értelmezik* és hajtják végre, de gépi kódú (EXE) fájlt nem készítenek. Ezt a megoldást használja például a Visual Basic Script [vizsuöl bézük szkript] vagy a Logo.

Újabban gyakori megoldás az is, hogy nem gépi kódra, hanem *egy köztes nyelvre fordítanak, amit a célszámítógépen egy futtató környezet hajt végre*. Ez lehetővé teszi, hogy egy kész programot különböző platformokon használhassunk. Ilyenek például a Java, a C# [szisárp] vagy a Visual Basic újabb változatai.

A programfejlesztés lépései

A program készítésének első lépése a feladat pontos megfogalmazása, amit *specifikációnak* neveznek. A kész programnak majd a specifikáció szerint kell működnie.

A specifikáció elkészítése után kezdődik az *algoritmus elkészítése*. **Az algoritmus* egy olyan műveletsor, amely a feladatot véges sok lépésben megoldja.** Az algoritmust a programozási nyelvektől függetlenül írják le, így könnyen átvihetők más feladatokra is. Az algoritmus elkészítésére és leírására sokféle módszert dolgoztak ki, mivel ez a programkészítés egyik legkritikusabb pontja. (Gondoljunk arra az esetre, amikor egy nagy alkalmazói programot több száz ember készít!)

Az algoritmus elkészítése után állhatunk neki a *kódolásnak*. Ez az egyes lépések beírását jelenti egy konkrét fejlesztői környezetben egy adott programozási nyelven. **A fejlesztői környezet, illetve a programozási nyelv megválasztása elsősorban a feladattól és a használt operációs rendszertől függ.**

Ritkán fordul elő, hogy a beírt program hibátlanul működik, ezért a programot ellenőrizni kell. A programozási hibák alapvetően kétféleképpen lehetnek. **Szintaktikai hibáról*** beszélünk, ha a programozó a kódot nem a nyelv szabályainak megfelelően írja be. Az ilyen hibát már a fejlesztői környezet is észreveszi és jelzi. **Szemantikai hibáról***


```
#define n 8
#define TRUE 1
#define FALSE 0

int u[9];
int m,l,c;

void kiir(void)
{
    int i;
    for (i=1; i<=n-1; i++)
    {
        printf("%1d",u[i]);
        printf("-");
    }
    printf("%1d",u[n]);
    printf("\n");
}
```

```
SNODEBUG
$STORAGE=2

        INTEGER N,I,C,M,U(8)
        LOGICAL L

        N=8
        DO 10,I=1,N,1
            U(I)=0
10      CONTINUE
        L=.FALSE.
        C=0
        M=1
20      IF ( L .OR. (C.NE.0)) GOTO 30
        CALL KERCU,L,M,N
        IF (L) GOTO 31
        CALL SUMCU,M,C,N
31      CONTINUE
        GOTO 20
```

```
program vezer;
const n=8;
type allas=array[1..n] of byte;
var u:allas;
    l:boolean;
    c:byte;
    i,m:byte;

function nincstes(u:allas; m:byte):boolean;
var i,j:integer;
begin
    nincstes:=true;
    for i:=1 to m-1 do
        begin
            if (u[i]=u[m]) or (abs(u[i]-u[m])=abs(i-m))
                begin
                    nincstes:=false;
                end;
        end;
end;
```

```
* A 8 VEZÉR PROBLÉMA MEGOLDÁSA
*

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. AT-286.
OBJECT-COMPUTER. IBM-PC.
SPECIAL-NAMES. CONSOLE IS CRT.
INPUT-OUTPUT SECTION.
FILE-CONTROL.

DATA DIVISION.
FILE SECTION.
WORKING-STORAGE SECTION.
77 N PIC 9 VALUE 8.
81 ALLAS.
85 U OCCURS 8 PIC 9.
77 L PIC X VALUE "F".
77 C PIC 9 VALUE 0.
```

Az algoritmus ugyanaz, a kód azonban nyelvfüggetlő. A 8 vezér probléma megoldásának egy-egy részlete C, Pascal, Fortran és Cobol nyelven.

beszélünk, ha a program helyesen van beírva, ám nem az előírt feladatot oldja meg. A szemantikai hibák felderítése komoly feladat. Történhet a *program matematikai elemzésével* vagy *teszteléssel**. Teszteléskor azt vizsgálják, hogy több, alkalmas módon megválasztott adatra mit reagál a program. Gyakori, hogy a majdnem kész programot átadják ellenőrzésre a leendő felhasználóknak, és az ő véleményüket is figyelembe véve javítják a hibákat, alakítják ki a program végső változatát.

A program készítése során a fejlesztők folyamatosan dokumentálják az elkészítés egyes fázisait. Ez az ún. *fejlesztői dokumentáció** lesz az alapja a később feltárt hibák javításának, és az újabb verzióknak.

A felhasználók számára a kész programhoz el kell készíteni a használati utasítást tartalmazó *felhasználói dokumentációt**, melyet a programmal együtt kell átadni.

A program életciklusa azonban nem ér véget azzal, hogy átadják a megrendelőnek: a környezet és a fejlesztői igények változásával később is szükségessé válik annak *karbantartása*, illetve *verziókövetése*.

A modern professzionális programok készítéséhez robusztus fejlesztői környezetre és alapos tudásra van szükség. Kezdként is készíthetünk azonban *szkripteket*, amelyekkel egyszerű rutin feladatokat automatizálhatunk, illetve *makrókat*, melyekkel a nagy alkalmazási programokat új funkciókkal ruházhatjuk fel (pl. MS Excel vagy CorelDRAW [koreldró]).

Első programunk

Ahhoz, hogy megtanuljunk programozni, *elsősorban az algoritmizálás elsajátítására kell koncentrálnunk*. A folyamatot azonban akkor értjük meg igazán, ha az algoritmusokat valamely konkrét rendszerben ki is próbáljuk.

Példáinkat két felületen is bemutatjuk. Az egyik a Windows-rendszerbe beépített egyszerű fejlesztői környezet, amellyel pl. Visual Basic szkripteket készíthetünk. Ennek előnye, hogy egyszerűen kezelhető, és bármikor rendelkezésünkre áll. A másik az oktatásban elterjedt „nagy öreg”, a Turbo Pascal [turbó paszkál], amely DOS-ablakban fut.

Az első program algoritmusa

Hagyományosan minden programozó azzal kezdi, hogy első programjával kiírta a képernyőre: „Hello World!”, azaz „Üdvözöllek Nagyvilág!”. Ezzel tudatosítja magában, hogy sikerült elkészítenie első működő programját. Megismerkedett az adott nyelv és fejlesztői környezet alapjaival, így a továbbiakban már a feladatra koncentrálnhat.

Ez az egyszerű program így foglalható össze:

```
Ki: „Hello World!”
```

A *Ki*: tényleges megvalósítása már az alkalmazott operációs rendszer lehetőségeitől, a programozási nyelvtől és persze a programozó elképzeléseitől is függ. Grafikus felület esetén például megjelenhet egy párbeszédpanelben, míg karakteres felület esetén egyszerűen kiírható a képernyőre.

Magát az utasítást általában kézzel kell beírni a fejlesztői környezetbe, például MsgBox (“Hello World!”) [meszidzsboks] vagy Writeln (“Hello World!”) [vrájtájn].

A felület lehetőségeitől függően a megjelenés egyéb jellemzőit – pl. betűtípus, előtér- és háttérszín, a dialógusdoboz címkesora, gombjai – is beállíthatjuk.

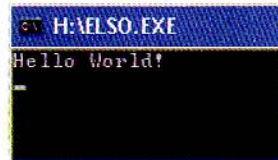
Az adatfeldolgozásban igen gyakori, hogy a *Ki*: utasítás az adatot nem a képernyőre, hanem egy fájlba vagy a nyomtatóra küldi.

```
MsgBox (“Hello World!”)
```



Visual Basicben az adatot legegyszerűbben az **MsgBox** utasítással egy párbeszédpanelben jeleníthetjük meg. Az **OK** gombra kattintva az ablak becsukódik.

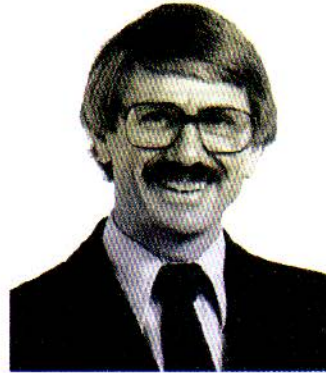
```
Writeln(' Hello World!');
```



Turbo Pascalban a kiírást a **Writeln** utasítás valósítja meg. A szöveg a megnyitott parancssor ablakba kerül, majd a kurzor a következő sorba lép.

Visual Basic szkriptek készítése

A Basic programozási nyelvet 1964-ben KEMÉNY JÁNOS és THOMAS KURTZ fejlesztették ki a Dartmouth College-ban [dartmush kolidzs] (USA) kifejezetten kezdők részére. Igazán elterjedt a mikroszámítógépek megjelenésével vált. Mai grafikus felületű változatát, a Visual Basicet természetesen más-más fejlesztői környezetben, de egyaránt használhatjuk szkriptek, professzionális programok vagy makrók készítésére.



Kemény János (1926-1992, balra) és Thomas Kurtz (1928-, jobbra)
a Basic programozási nyelv megalkotói

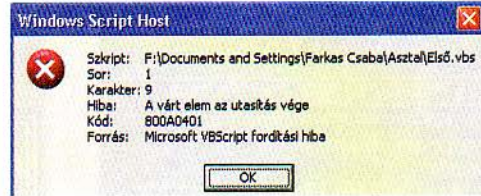
A Visual Basic szkriptet egy tetszőleges editorral (pl. Jegyzetömbbel) írhatjuk be, illetve módosíthatjuk. Lényeges, hogy a fájl kiterjesztése *vbs* legyen. A program indításához egyszerűen kattintunk az ikonjára, ezért célszerű azt az asztalra tenni. Más dolgunk nincs is, mivel a programot egy interpreter (*Windows Script Host*) futtatja, ez viszont része a Windowsnak, tehát nem kell telepítenünk sem.

Mivel a *vbs* fájlok könnyen készíthetők, és futtatásuk is két kattintással történik, sokan visszaélnak a lehetőséggel, és az interneten keresztül terjedő vírusok készítésére használják őket. Az e-mailben kapott *vbs* fájlokat ezért feltétlenül ellenőrizzük!



Első mintaprogramunk elkészítésének lépései (Visual Basic Script):

1. Létrehozunk az Asztalon egy Új szöveges dokumentumot, melyet átnevezünk *Első.vbs*-re.
2. A helyi menü **Szerkesztés** pontjával megnyitjuk *Jegyzetömb*bel.
3. Beírjuk a programkódot és mentjük.



4. A programot két kattintással elindítva megjelenik az eredmény ...
 ... vagy egy hibaüzenet, amely megadja a hiba helyét (sor, karakter) és a hiba valószínű okát.

Turbo Pascal programok készítése

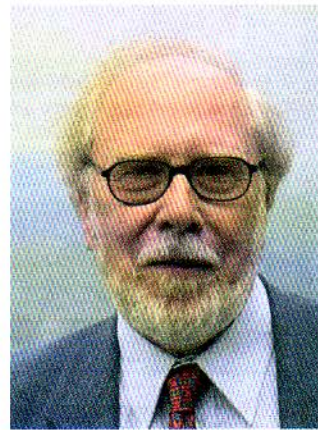
A Pascal programozási nyelvet 1968-ban NIKLAUS WIRTH [niklausz wörsz], a Zürichi Műszaki Egyetem tanára tervezte meg, és azóta több nyelv tervezésének is kiinduló pontjává vált. Igen gyakran használják az oktatásban (*Turbo Pascal*), de létezik Windows alapú professzionális fejlesztői környezete is (*Borland Delphi* [delfi]).

A Turbo Pascal komplett fejlesztői környezet, amelyet telepítenünk kell. A fejlesztői környezet indítása után az alkalmazói programokhoz hasonló menürendszerrel (pl. File menü) jelentkezik be. Mivel *compiler** [kompájler], így *exe* fájlt is készíthetünk vele, amely önállóan futtatható.

Egy egyszerű program elkészítése három fő lépésből áll. Először beírjuk a Pascal nyelvű ún. *forráskódot*. Ezt követi a kód tesztelése, vagyis megpróbáljuk futtatni a programot. Teszteléskor a fejlesztői környezet megadja az általa felismert hibákat. Amikor a tesztelés már sikeres, elkészítjük a *tárgykódot*, vagyis az *exe* fájlt.

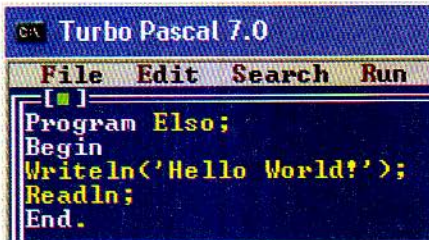
A Pascal nyelvű program kötelezően a program nevét megadó sorral kezdődik, ezt a *Program* kulcsszó vezeti be. A végrehajtandó műveletsort egy *Begin-End* [bigin] utasításpár közé kell beírni. Figyeljünk oda, hogy a Pascal nyelvben a sorokat általában pontosvessző zárja, kivéve a *Begin* és az utolsó *End* utasítást.

```
Program Elso;
Begin
    Writeln(' Hello World!' );
    Readln;
End.
```

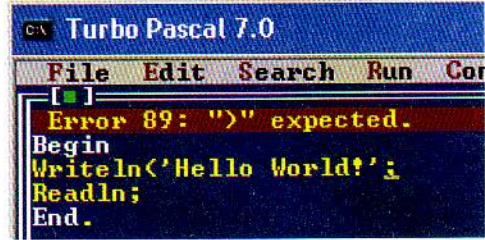


Niklaus Wirth, a Pascal programozási nyelv megalkotója

A *Writeln* a szöveg kiírása után új sort kezd; ha a kiírt szöveg mellé szeretnénk valami mást is írni, akkor helyette a *Write* [vrájt] utasítást kell használnunk. A *Readln* [ridlajn] egy Enter gomb lenyomására vár, addig nem tudunk továbblépni.



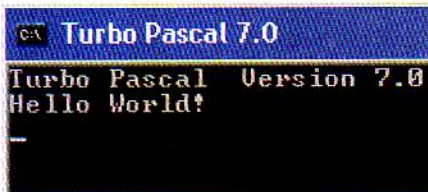
```
CA Turbo Pascal 7.0
File Edit Search Run
[ ]
Program Elso;
Begin
Writeln('Hello World!');
Readln;
End.
```



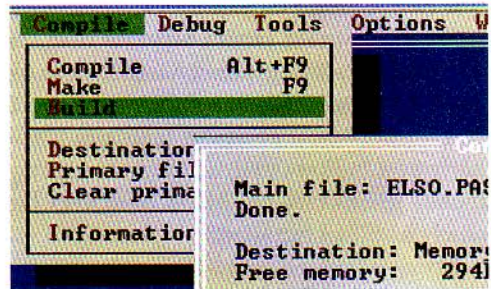
```
CA Turbo Pascal 7.0
File Edit Search Run Cor
[ ]
Error 89: ">" expected.
Begin
Writeln('Hello World!');
Readln;
End.
```

Első mintaprogramunk elkészítésének lépései (Turbo Pascal):

1. Beírjuk a tárgykódot, és elmentjük a **File** menü **Save** [szév] menüpontjával.
2. A programot a **Run** [rán] menü **Run** pontjával teszteljük, az elírásokból adódó hiba okát és valószínű helyét a környezet megjelöli.



```
CA Turbo Pascal 7.0
Turbo Pascal Version 7.0
Hello World!
```



```
Compile Debug Tools Options W
Compile Alt+F9
Make F9
Build
Destination
Primary fil
Clear prima Main file: ELSO.PAS
Done.
Information Destination: Memory
Free memory: 294
```

3. Ha nincs elírás, egy DOS-ablakban látjuk a futás eredményét (ez is lehet hibás).
4. Ha kész a program, akkor a **Compile** menü **Build** [bild] menüpontjával elkészítjük az exe fájlt.

A változó fogalma, típusai

Példa: négyzet területe

Határozzuk meg program segítségével egy négyzet területét, ha ismerjük az oldalát!

Világos, hogy ekkor programunknak három fő lépést kell tartalmaznia.

- Beolvassa az oldal hosszát (jelöljük a -val),
- kiszámítja a négyzet területét (jelöljük t -vel),
- végül kiírja a területet.

Algoritmusunk tehát a következő:

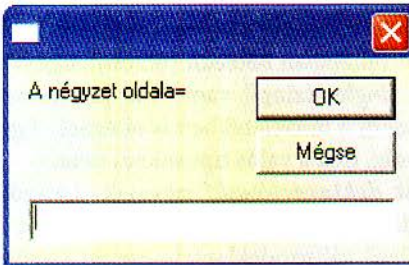
```
Be: a
t = a * a
Ki: t
```


Az adatbevittelt az adatkivittelhez hasonlóan valósíthatjuk meg attól függően, hogy milyen programozási nyelvet, illetve operációs rendszert használunk. Természetesen minden esetben tájékoztatni kell a felhasználót, hogy milyen adatot várunk tőle.

Az adatbevitel megvalósítása kritikus pont, mivel a felhasználó könnyen írhat be hibás adatot. Például, ha 60 helyett azt írja be, hogy *hatvan*, a programunk elszállhat. A grafikus felületek számtalan olyan megoldást tartalmaznak, amelyek a felhasználó számára megkönnyítik az adatok bevitelét, a programozó számára pedig azok ellenőrzését. (Például jelölőnégyzetek, rádiógombok, numerikus léptetők stb.)

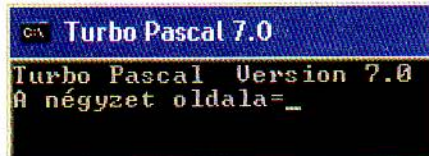
Adatfeldolgozó programok esetében gyakori, hogy a beolvasás nem a billentyűzetről, hanem egy fájlból történik, valamelyik háttértárról.

```
a=InputBox("A négyzet oldala")
```



Visual Basicben az adatot egyszerűen bekérhetjük az InputBox utasítás hatására megjelenő dialógusdobozzal.

```
Write(' A négyzet oldala=' );  
Readln(a);
```



Turbo Pascalban kiírjuk, milyen adatot várunk, majd az adatot beolvassuk a Readln utasítás segítségével.

A változó fogalma

Az algoritmusban szereplő *a* és *t* **változók*** egy-egy memóriaszeletet azonosítanak, ahol az értékeik tárolódnak. Ezt a kapcsolatot a fejlesztői környezet automatikusan kezeli, így a matematikában megszokott módon használhatjuk őket.

Ha egy változó értéket kap, akkor az egyenlőségjel bal oldalán kell állnia. Fontos, hogy ekkor nem matematikai értelemben vett egyenlőségről, hanem értékadásról van szó! Ennek megfelelően az $x=2$ értékadást így célszerű olvasni: „*x* legyen egyenlő 2-vel”.

A fentiek miatt sok programozási nyelv, pl. a Pascal is, értékadásra a „definiáló” egyenlőségjelet ($:=$) használja a „sima” egyenlőségjel (=) helyett.

Gyakori, hogy értékadásnál a változó korábbi értékét módosítjuk, így az akár a jobb oldalon is szerepelhet. Ekkor értelemszerűen a változónév a bal oldalon az új, a jobb oldalon pedig a régi értéket jelenti.

```
alma = alma + 2
```

```
alma := alma + 2;
```

Az alma változó értéke 2-vel nő. Az értékadás Basicben, illetve Pascalban.

Mivel egy bonyolultabb algoritmushoz sok változóra is szükség lehet, megengedett, hogy a változó neve több betűből álljon; sőt kialakult az a szokás, hogy a változó neve utal a benne tárolt adat jelentésére.

A típus fogalma

Mivel a változóknak helyet kell foglalni a memóriában, ezért a legtöbb programozási nyelvben a program elején fel kell sorolnunk, hogy milyen változókat fogunk használni. Ilyenkor többnyire nemcsak a változó nevét, hanem a típusát is meg kell adnunk. A **típus*** két dolgot jelent: milyen értékeket vehet fel a változó, és milyen műveleteket végezhetünk vele.

A típus fogalma végigkíséri informatikai tanulmányainkat. Gondoljunk a mezők típusára az adatbázis-kezelésben vagy a cellák számformátumaira a táblázatkezelésben. Ennek megfelelően a **legtöbb programozási nyelv ismeri a következő típusokat**. **Szöveges típus*** (pl. *string* néven), **logikai típus*** (általában *boolean* [bülön]), **egész típus*** (pl. *integer* [intídzsör]), **valós típus*** (pl. *single* [szingl] vagy *real* [ríl] néven). A számtípusok nemcsak a kezelt számokban, hanem a műveletekben is eltérnek. Egész számok esetében például vizsgálható az oszthatóság, míg a valós típusoknál nem.

A **változó nevének és típusának megadását deklarációnak*** nevezik. Látszólag ugyan kényelmesebb, ha a változókat nem kell deklarálni, a valóságban azonban az elírásból, hibás értékadásból eredő hibákat a kötelező változódeklaráció esetén kiszűri a fejlesztői környezet.

```
Program Negyzet;  
Var a,t:integer;  
Begin  
  Write(' A négyzet oldala=' );  
  ReadLn(a);  
  t := a*a;  
  Write(' A négyzet területe=' );  
  Write(t);  
  ReadLn;  
End.
```

Négyzet területének meghatározása Turbo Pascalban. Pascalban kötelező a **Var** kulcsszóval a változókat deklarálni, ezért a Pascalt erősen típusos nyelvnek is nevezik. Ha a változó nevét elírjuk, vagy olyan műveletet írunk elő, amit az adott típus nem ismer, hibajelzést kapunk.

```
Option explicit  
Dim a, t  
a = InputBox("A négyzet oldala=")  
t = a * a  
MsgBox(t)
```

Négyzet területének meghatározása Visual Basic szkriptel. Eldönthetjük, hogy szeretnénk-e a változókat deklarálni. Ha igen, akkor a **Dim** kulcsszó után kell felsorolni azokat. Ha ekkor a kód elejére beírjuk az **Option explicit** [opsn ekszpliszit] utasítást is, akkor a változó nevének elírása esetén hibajelzést kapunk. A típust külön nem lehet megadni, de ez a szkript nyelvek sajátossága.

Elágazás

Példa: súlyteszt

Felnőtt emberek esetén az elhízás mértékét az ún. testtömegindex segítségével szokták meghatározni. Ha az ember magassága m méter, tömege pedig kg kilogramm, akkor bmi testtömegindexét a $bmi = kg / m^2$ összefüggésből számolhatjuk. Ha ez a szám 25 fölött van, akkor az illető felnőtt túlsúlyos, míg 30 fölött elhízott.

Foglaljuk össze a fenti eljárás algoritmusát!

Be: kg, m

$bmi = kg / (m * m)$

Ha bmi 25-nél kisebb, akkor Ki: Ön nincs elhízva.

Ha bmi 25 és 30 közé esik, akkor Ki: Ön túlsúlyos.

Ha bmi 30-nál több, akkor Ki: Ön elhízott.

Az *elágazás** olyan vezérlési szerkezet, amely az utasítások egy adott csoportját attól függően hajtja végre, hogy egy adott logikai feltétel teljesül-e.

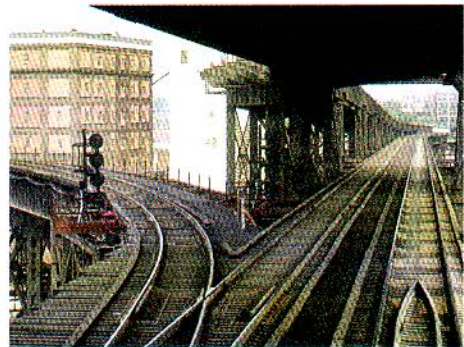
Az elágazás megvalósítása

Az elágazás legegyszerűbb változatát a következő algoritmus adja meg:

Ha *logikai feltétel* akkor
 utasítások

Elágazás vége

Az *utasítások* rész akkor és csak akkor hajtódik végre, ha a *logikai feltétel* teljesül. A fenti szerkezet egy egységet alkot, ennek egy része csupán az *utasítások* blokk. Az *utasítások* blokk beljebb kezdésével ez külön is hangsúlyozható, és jobban áttekinthető maga a programkód is.



A vasúti elágazásnál a váltó dönt

If *feltétel* Then

utasítások

End If

If *feltétel* Then

 Begin

utasítások;

 End;

Az elágazás megvalósítása If... Then [den] szerkezettel Visual Basicben, illetve Turbo Pascalban

Az előző, gyakran kétirányú elágazásnak nevezett megoldás mellett használják az ún. *többirányú elágazást* is. Ekkor *egy kifejezés értékétől függően különböző utasításcsoportok végrehajtására kerülhet sor*. Tipikus alkalmazása, amikor egy kereskedő a megvásárolt árucikkek számától függően különböző mértékű kedvezményt ad. Ezzel az esettel most részletesebben nem foglalkozunk.

Összetett feltétel megadása. Logikai műveletek

Az elágazás kritikus pontja a logikai feltétel helyes megfogalmazása, főleg összetett állítások esetén. Az **összetett állítások képzésére a programozási nyelvek a matematikai logikában megismert logikai műveleteket használják**: többnyire a NOT (negáció), az OR (diszjunkció) és az AND (konjunkció) kulcsszavakkal.

A több relációt tartalmazó összetett állításokat általában a programozási nyelvekben az állítások konjunkciójára kell bontani. Pl. a *bmi értéke 25 és 30 között van* állítás így adható meg: *bmi >= 25 AND bmi <= 30*. A két elemi kifejezést a programozási nyelv szabályaitól függően vagy zárójelbe kell tenni, vagy nem.

```
Option explicit
Dim kg, m, bmi
```

```
kg = InputBox("Tömege (kg)=")
m = InputBox("Magassága (m)=")
bmi = kg / (m*m)
```

```
If bmi < 25 Then
    MsgBox("Ön nincs elhízva!")
End If
```

```
If bmi >= 25 And bmi <= 30 Then
    MsgBox("Ön túlsúlyos!")
End If
```

```
If bmi > 30 Then
    MsgBox("Ön elhízott!")
End If
```

```
Program Sulyteszt;
Var m, kg, bmi:Real;
```

```
Begin
    Write(' Tömege (kg)= ');
    Readln(kg);
    Write(' Magassága (m)= ');
    Readln(m);
    bmi := kg / (m*m);
```

```
If bmi < 25 Then
    Begin
        Writeln(' Ön nincs elhízva' );
    End;
If (bmi >= 25) And (bmi <= 30) Then
    Begin
        Writeln(' Ön túlsúlyos' );
    End;
If bmi > 30 Then
    Begin
        Writeln(' Ön elhízott' );
    End;
Readln;
End.
```

A súlyteszt program kódja Visual Basicben és Turbo Pascalban

A ciklus

Példa: Manhattan szigete

1626-ban az indiánok 24\$ értékű üveggyöngyért eladták Manhattan [menhetn] szigetét a hollandoknak. Ha az akkori törzsfőnök ezt a pénzt betette volna a bankba, mennyit ért volna ez a pénz az ezredfordulón?

Tanulságos végigszámolni, hogyan nő a betét az első pár évben, ha a kamatláb pl. 10%:

év	1626	1627	1628	1629	1630	1631	1632	1633
p	24	26,4	29,04	31,944	35,1384	38,65224	42,517464	46,7692104

Ha a mindenkori összeget a p , az adott évet pedig az $év$ nevű változó tartalmazza, akkor „csak” annyit kell tennünk, hogy elvégezzük a

$$p = p + p * 10\%$$

műveletet mindaddig, amíg az $év$ 1626-tól egyesével 2000-ig nem nő. **Ciklusnak* nevezük az olyan szerkezetet, amikor egy adott műveletsort (ciklusmag) addig kell ismételni, amíg egy adott feltétel (ciklusfeltétel) teljesül.** Az ilyen feladatok gyakoriak a valós életben; a számítógépet eredetileg ezeknek a megvalósítására találták fel.

Programunk algoritmusa a következőképpen fogalmazható meg:

```
p = 24
Ciklus év=1626-tól 2000-ig
    p = p + p*0,1
Ciklus vége
Ki: p
```

A ciklus fajtái

A ciklusfeltétel jellegétől függően a ciklus megvalósítására a legtöbb programozási nyelv többféle megoldást is tartalmazhat.

Gyakran használják a *számláló típusú ciklusokat*, ahol az ismétlések számát egy változó, az ún. *ciklusváltozó* követi nyomon. Előző példánkban ez az $év$, mely 1626-tól 2000-ig „telik”. Ilyenkor az ismétlések számát már előre ismerjük.

```
For cv = kezdőérték To végérték
    utasítások
Next
```

```
For cv := kezdőérték To végérték Do
Begin
    utasítások;
End;
```

A számláló típusú ciklus megvalósítása Visual Basicben és Pascalban. Az algoritmusban cv jelöli a ciklusváltozót.


```
Dim p, ev

p = 24

For ev = 1626 To 2000
    p = p + p*0.1
Next

Msgbox ("Végösszeg=" & p)
```

```
Program Manhattan;
Var p:Real;
Var ev:Integer;

Begin
p := 24;
For ev := 1626 To 2000 Do
Begin
    p := p+p*0.1;
End;
Write(' Végösszeg=' );
Writeln (p);
Readln;
End.
```

Példánk programkódja Visual Basicben és Turbo Pascalban

Kicsit bonyolultabb a helyzet, ha a ciklusfeltételt csak a kiszámított adatokból tudjuk meghatározni. Így például kérdezhetnénk, hogy melyik évben lépi túl a bankban lévő pénz az 1000 dollárt. Most nem tudjuk előre, hogy hányszor kell a műveletet elvégezni, de gondoljuk végig, hogy a feladat megoldása lehet pl. a következő algoritmus:

```
p = 24
év = 1626
Ciklus amíg p <= 1000
    p = p + p*0.1
    év = év + 1
Ciklus vége
Ki: év
```

Ha a ciklusfeltételt a ciklusmag előtt vizsgáljuk, akkor a ciklust előtesztelőnek nevezzük. (Természetesen a ciklusnak van *háttesztelő* változata is, de ezek tényleges megvalósításával bővebben nem foglalkozunk.)

```
Do While ciklusfeltétel
    utasítások
Loop
```

```
While ciklusfeltétel Do
Begin
    utasítások;
End;
```

Az előtesztelő típusú ciklus megvalósítása Visual Basicben és Pascalban

Programozási tételek

A matematikában tételek segítségével fogalmazzák meg a feltárt összefüggéseket (pl. Pitagorasz-tétel), s ezt követően a konkrét példákban már csak hivatkoznak rájuk. (Pl. ha egy derékszögű háromszög befogói 3 és 4 egység, akkor az átfogója a Pitagorasz-tétel alapján ...) Ugyanígy a számítástechnikában is nagyon sok olyan részfeladat van, ami gyakran előfordul, s az évek során ezekre matematikailag is igazolt algoritmust készítettek. Szokás ezeket *nevezetes algoritmusoknak* vagy **programozási tételeknek*** nevezni. Az alábbiak során néhány nevezetes algoritmust tekintünk át egy-egy példa segítségével.

Az összegezés tétele

Határozzuk meg az 1-től n -ig terjedő egész számok összegét!

A feladatot legegyszerűbben a matematikában tanult képlet segítségével oldhatjuk meg. Ennek ellenére hasznos lesz áttekinteni az egyenkénti összegezés lehetőségét, mivel ez az eljárás (*az összegezés tétele**) akkor is alkalmazható, ha az adott lehetőségre vonatkozó képletet nem ismerjük (pl. faktoriális vagy kamatos kamat).

Az eljárás igen egyszerű. A számok összege kezdetben nulla, majd ehhez hozzáadjuk az egyet, aztán az összeghez a kettőt, majd az eredményhez a hármat, és így tovább.

Jelöljük egy adott pillanatban a már összeadott számok összegét s -sel, a következő hozzáadandó számot pedig i -vel. Ekkor látható, hogy a feladat elvégzése során az $s = s + i$ műveletet kell ismételnünk, miközben i értéke 1-től n -ig változik. A feladatot tehát megoldhatjuk egy számláló típusú ciklussal:

```
Eljárás Összegezés
  Be: n
  s=0
  Ciklus i = 1-től n-ig
    s = s + i
  Ciklus vége
  Ki: s
Eljárás vége
```

Hogyan lehet egy ilyen algoritmust felhasználni? Először is tudnunk kell, hogy a fenti algoritmus helyességét matematikailag ténylegesen bizonyították, tehát ha sikerül visszavezetnünk rá egy feladatot, akkor arra is helyes algoritmust kapunk.

A Manhattan szigetére vonatkozó korábbi feladatunk tényleg vissza is vezethető az összegezés algoritmusára. A feladat ugyanis lényegében ugyanez, csak a ciklusváltozónak 1626-tól 2000-ig kell futnia, a művelet során pedig nem a következő számot, hanem a kamatot kell az összeghez hozzáadni.

Hasonló módon, a fenti algoritmus segítségével kiszámíthatjuk az első n pozitív szám szorzatát (n faktoriálisát) is.

Maximum- és minimum-kiválasztás

Válasszuk ki egy előre megadott számsorból a legnagyobb (illetve legkisebb) elemet!

Ez a feladat is igen gyakori, gondoljunk például a sportversenyekre. A feladat megoldásához nézzük végig, hogyan dől el, hogy ki a világbajnok a profi ökölvívók közül! Ha csak egy profi ökölvívó van, akkor nyilván ő a bajnok. Ha megjelenik egy újabb jelölt, annak meg kell küzdenie a korábbi bajnokkal. Ha a régi bajnok győz, akkor megőrizte bajnoki címét. Ha az új győz, akkor ő jobb az eddigi legjobbnál, így tehát az összes többinél is. Az algoritmus így foglalható össze:



Melyik a legmagasabb?

```
Bajnok = első profi ökölvívó
Ciklus amíg van új jelölt
    Ha a jelölt jobb, akkor
        bajnok = új jelölt
Elágazás vége
Ciklus vége
Ki: bajnok
```

Kicsit konkrétabb adatokkal, de a fentiek alapján azt fogjuk meghatározni, hogy mennyi a „legjobb idő” egy úszóversenyen.

Jelöljük az első pályán úszó versenyző idejét $T(1)$ -gyel, a második pályán úszóét $T(2)$ -vel, ... a nyolcadik pályán úszóét

$T(8)$ -cal. A legjobb idő kerüljön a *legjobb* változóba. Ha csak egy pálya lenne, akkor a *legjobb* értéke $T(1)$ volna. Az előzőekhez hasonlóan, ehhez vegyük hozzá rendre a második, majd a harmadik stb. pályán úszó eredményét. Ezúttal az új adatot mindig a korábbi legkisebb adattal kell összehasonlítani. Ha a résztvevők száma n , akkor algoritmusunk a következő:

```
Be: T()
legjobb = T(1)
Ciklus i = 2-től n-ig
    Ha T(i) < legjobb akkor
        legjobb = T(i)
Elágazás vége
Ciklus vége
Ki: legjobb
```

Egy újabb változó bevezetésével nemcsak a legjobb értéket, hanem azt is meghatározhatjuk, hogy ezt melyik sportoló érte el. Természetesen ilyenkor figyelni kell a holtverseny lehetőségére is.

A fenti algoritmus a legkisebb értéket keresi meg, ezért *minimum-kiválasztásnak* nevezzük. Értelemszerűen a *maximum-kiválasztás** csak annyiban tér el, hogy a ciklusban szereplő elágazás feltétele:

Ha $T(i) > \text{legjobb}$ akkor

Furcsának tűnhet, bár rendkívül logikus, az időadatokat tartalmazó $T(1), T(2) \dots$ jelölés. Az adatok ilyen elrendezését gyakorlatilag minden programozási nyelv támogatja, és *tömbnek* nevezik. A *tömb** azonos típusú adatok sorozata.

Rendezés egyszerű cserével (buborékmódszer)

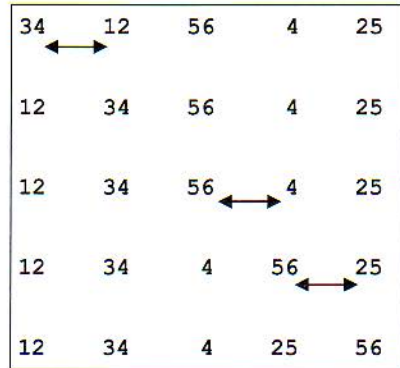
Rendezzük egy adott tömb elemeit növekvő (vagy csökkenő) sorba!

Rendezésre nagyon sok eljárást készítettek, mivel az adatkezelésben igen gyakori feladat. Ezek közül a legközismertebb eljárás a *rendezés egyszerű cserével** más néven *buborékmódszer*. Maga a módszer a többi eljáráshoz képest viszonylag lassú, nagy előnye, hogy könnyen megvalósítható.

A módszer lényege a következő. Hasonlítsuk össze a tömb első és második elemét, s ha az első nagyobb, cseréljük meg őket. Ezt követően hasonlítsuk össze a tömb második és harmadik elemét, s ha a második nagyobb, cseréljük meg őket. És így tovább, amíg végig nem érünk a tömbön. Látható, hogy ekkor a tömb legnagyobb eleme biztosan a helyére kerül, még akkor is, ha ő volt az első.

Sajnos ez a többi elemre nem feltétlenül teljesül. Ahhoz, hogy a második legnagyobb elem a helyére kerüljön, ismét el kell indulnunk a tömb elejéről, és az első elemet össze kell hasonlítani a másodikkal, a másodikat a harmadikkal ... Most azonban elég elmenni az utolsó előtti elemig, hiszen az utolsó már a helyére került.

A következő körben már a harmadik legnagyobb elemet tesszük a helyére, és így tovább. A ciklust tehát lényegében annyiszor kell lefuttatnunk, ahány eleme van a tömbnek.



A buborékredezés első menete

Eljárás Buborékredezés

Ciklus $i = n$ -től 2-ig, -1-esével

Ciklus $j = 1$ -től $i-1$ -ig

Ha $v(j) > v(j+1)$ akkor Csere ($v(j), v(j+1)$)

Ciklus vége

Ciklus vége

Eljárás vége

Összefoglaló kérdések, feladatok

1. Soroljuk fel a program életútjának lépéseit!
2. Mi a fejlesztői környezet és a programozási nyelv kapcsolata?
3. Szeretnénk felcserélni egy szöveges dokumentumban az a és a b betűket. Készítsünk algoritmust a probléma megoldására!
4. Készítsünk algoritmust a kör kerületének és területének meghatározására!
5. Hogyan számítják ki a személyi jövedelemadót? Keressük meg a hatályos adótáblát az interneten, majd készítsük el az algoritmust!
6. Magyar nyelvi tanulmányaink alapján mondjunk példát szemantikai hibára és szintaktikai hibára!
7. Készítsünk algoritmust, amely egy pozitív egész számról eldönti, hogy prímszám-e!
8. Készítsünk algoritmust, amely megszámolja, hogy hány prímszám van 1 és n között!
9. A 30 elemű $M()$ tömb azt tartalmazza, hogy az osztály egyes tanulói hány pontot értek el a matematika érettségi vizsgán. Készítsünk algoritmust, amely
 - a) kiszámolja az átlagos pontszámot,
 - b) eldönti, van-e 0 pontos,
 - c) meghatározza a legmagasabb pontszámot,
 - d) pontszám szerint növekvő sorba rendezi a tömböt!
10. A rendezetlen $N()$ tömb azt tartalmazza, hogy mennyi az egy főre jutó nemzeti jövedelem az Európai Unió tagállamaiban. Legyen a tömb első eleme Magyarország egy főre jutó nemzeti jövedelme. (A szükséges adatokat keressük meg az interneten!)
 - a) Hány olyan ország van, amelynek egy főre jutó nemzeti jövedelme nagyobb, illetve kisebb, mint hazánké? Készítsünk algoritmust a feladat megoldásához!
 - b) Ki lehet-e számolni ezekből az adatokból, hogy mennyi az egy főre jutó „uniós” jövedelem a teljes Európai Unióra vonatkoztatva?
11. Fogalmazzuk meg, hogyan szoktunk neveket papíron névsorba rendezni! Működik ez az algoritmus számokra is?
12. Ma a gyerekek egészen korán megtanulják a számok osztását papíron. Fogalmazzuk meg akkor jelenlegi tudásunk alapján az osztás algoritmusát!

Szoftverismeret

A szoftverek csoportosítása funkciójuk szerint

A korábbiakban nagyon sok programmal ismerkedtünk meg, és még többet csupán megemlítettünk. Tekintettel a programok nagy számára, célszerű őket rendszerezni, hogy eligazodjunk közöttük.

Funkciójuk szerint a szoftvereket három fő csoportba soroljuk: lehetnek rendszerprogramok, fejlesztői környezetek vagy alkalmazói programok.

A rendszerprogramok

A **rendszerprogramok*** a számítógép üzemeltetéséhez szükséges programok; ezek is több csoportra tagolhatók.

Operációs rendszer* nélkül a számítógép nem használható.

Az operációs rendszer egy olyan programcsomag, amely:

- kapcsolatot tart a felhasználóval,
- felelős a programok indításáért, futtatásáért és befejezéséért,
- kezeli az erőforrásokat,
- végzi a hibák és kivételek kezelését,
- kapcsolatot teremt a futó alkalmazások között, stb.

Az operációs rendszerek általában további programokat is tartalmazhatnak. Ilyenek például a szövegszerkesztő és a tömörítő programok, de gyakori egy beépített programozási eszköz is.

A PC-s operációs rendszerek közül a legelterjedtebbek a *Windows* és a *Linux* különböző változatai.

A **rendszerközeli programok*** könnyebbé teszik az operációs rendszerek kezelését (*shell programok* [sell]), vagy bővítik annak szolgáltatásait (*utilityk* [jutiliti]). Tipikus shell program pl. a *Windows Commander*, míg pl. a *Nero utility*.

A **hálózati programok*** gépek összekötését és együttműködését teszik lehetővé. Ilyenek például a szervereken futó kiszolgáló programok, ezek kezelése általában komoly szakmai felkészültséget igényel. (Ez a *rendszergazdák* feladata.) Ma a legelterjedtebb PC-s hálózati operációs rendszerek a *Windows Server*, a *Novell Netware* [netver] és a *Linux* különböző változatai.



TUX, a Linux kabalafigurája

A Microsoft Windows Server 2003 emblémája.

Milyen további hálózati programokat tudunk felsorolni?



A fejlesztői környezetek

A *fejlesztői környezetek** segítségével új programokat készíthetünk. Ilyen esetben speciális, úgynevezett programozási nyelven kell leírunk a program működését. Új programok készítése komoly tervezési és tesztelési munkát igényel.

Néhány elterjedt programozási nyelv: C, C++, C# [szísárp], Visual Basic, Pascal, Delphi, Java, Ada, Modula, Clipper.



A Java nyelv emblémája

Az alkalmazói programok

Az *alkalmazói programok** konkrét felhasználói területekhez kapcsolódó szoftverek, amelyek egy-egy feladat vagy feladattípus elvégzéséhez készültek. Az alkalmazói programok lehetnek egyediek vagy általános célúak.

Az *egyedi programokat egy konkrét feladat elvégzéséhez írják*, maximálisan az adott feladat sajátosságaira koncentrálva. Mivel felhasználási területük szűk, ezért csak kevés példányban adhatók el, így előállításuk költséges. Ilyen például egy adott banki rendszerben az ügyfelek pénzforgalmát kezelő szoftver.

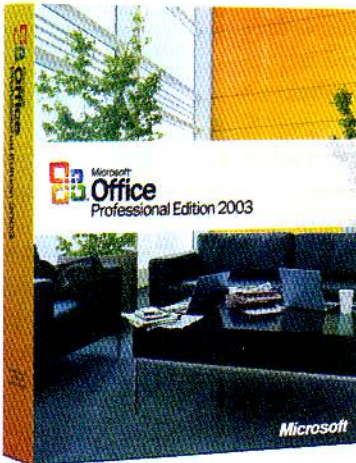
Az általános célú szoftverek fajtái

Az *általános célú programok egy adott feladattípus teljes lefedéséhez készülnek*. Hátrányuk, hogy a felhasználók a funkciók jelentős részét nem fogják kihasználni (mindenki más-más részét nem használja ki), és betanulásuk is hosszabb időt vesz igénybe. Az ilyen programok viszont a nagy példányszám miatt olcsók.

A korábbiakban részletesen megismerkedtünk az ún. irodai („office” [ofisz]) alkalmazásokkal: a **szövegszerkesztőkkel**, a **táblázatkezelőkkel**, a **prezentációkészítőkkel** és az **adatbázis-kezelőkkel**. Bár ezek külön-külön is beszerezhetők, a gyártók többnyire együtt, egy programcsomagban kínálják őket.

Az egyik legismertebb programcsomag a *Microsoft Office*, de számos más gyártó is fejleszt irodai programcsomagokat. Így pl. az *Ability (Ability Office)*, a *Corel (WordPerfect Office [vördperfekt])*, vagy a *Sun Microsystems [szán májkroszisztemsz] (StarOffice [sztárofisz])* is.

Az irodai programcsomagok piacán is megjelentek a nyílt forráskódú, ingyenes megoldások, mint pl. az *OpenOffice.org*. Fontos, hogy ezekhez nem jár terméktámogatás, míg a nyílt forráskódú programból kifejlesztett irodai programcsomagokhoz már igen (pl. *Magyar Office* vagy *EuroOffice*). Természetesen ezek már nem ingyenesek.



Egy szoftver „dobozos” változata. Soroljuk fel egy tipikus irodai programcsomag részeit!

A **grafikus alkalmazások** rajzok, képek elkészítését, feldolgozását teszik lehetővé. Lehetnek pixelgrafikusak (pl. *Adobe Photoshop*, *Corel Paint Shop Pro*) vagy vektorgrafikusak (pl. *CorelDRAW* [koreldró], *Adobe Illustrator*).

A **kiadványszerkesztők** kiadványok, szórólapok, plakátok és egyéb nyomdai munkák készítését segítik (pl. *QuarkXPress* [kvarkixszpressz], *Adobe InDesign* [adobi indizajn], *Corel Ventura*, *Microsoft Publisher* [pábliser]). A legegyszerűbb kiadványszerkesztő rendszerek megvalósíthatók egy jó minőségű személyi számítógéppel is, ezt nevezik **DTP**-nek (*DTP = DeskTop Publishing* [deszktop páblising]).

A gyártók az összetartozó grafikus alkalmazásokat és a kiadványszerkesztőket is gyakran forgalmazzák együtt, egy programcsomagként. Ilyen pl. az *Adobe Creative Suite* [kriétív szjuit] vagy a *Corel Creative Collection* [kriétív kolleksn].

A fentieken túl gyakorlatilag minden felhasználási területhez kész programokat vehetünk. Ezek megvásárlása előtt mindig célszerű alaposan tájékozódni: legjobban a különböző próbaváltozatok letöltésével és kipróbálásával járunk (*trial* [trájl] *verzió*).

Egy szoftver 30 napos próba-verziója. A figyelmeztetés szerint már csak 9 nap van hátra, utána vagy törölnünk kell, vagy meg kell vásárolni. A funkciók kipróbálása azonban ingyenes volt. Milyen célra készült ez a program? Nevezünk meg más hasonló célra készült szoftvert is!



About Activation

Activation is completely anonymous: No personal information is required.

You can activate your software any time within the first 30 days after you first launch it. Not ready to activate yet? Use the Trial mode.

Welcome to Dreamweaver

I have a serial number and want to activate Dreamweaver.

Enter your serial number to automatically activate Dreamweaver; it takes only a few seconds.

Need a serial number?

[Buy one online.](#)

I want to try Dreamweaver. (9 days remaining)

There are 9 days left in your trial period. When it ends, you must activate Dreamweaver to continue using it.

A szoftverek csoportosítása jogi szempontból

A csoportosítás másik szempontja jogi eredetű: milyen módon rendelkezett a szoftver készítője annak felhasználási lehetőségeiről.

A szoftver tulajdonjoga

A **szoftver szellemi termék**, amelyben gyakran sok ember több éves munkája fekszik. Ennek megfelelően a program **készítőjét**, vagy ha alkalmazottként írta, akkor az őt alkalmazó céget, a **szerzői jog* védi**.

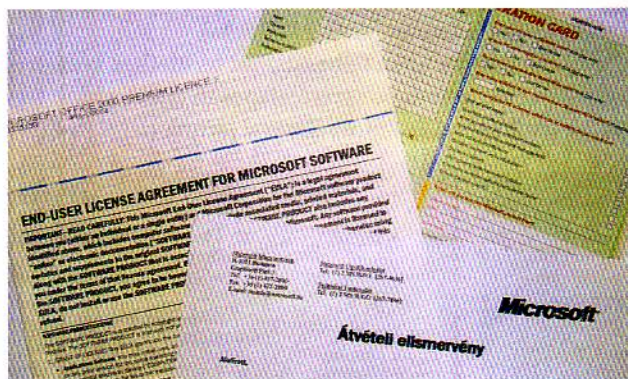
Fontos tudni, hogy **amikor szoftvert vásárolunk**, akkor nem a szoftver tulajdonjogát szerezzük meg, hanem **csupán korlátozott jogokat szerzünk a szoftver felhasználására**. Ezeket a **jogokat a licencszerződés rögzíti**, melyet többnyire a szerzői jog tulajdonosa szab meg. Azzal, hogy a szoftvert elkezdjük használni, a jog szerint ezeket a feltételeket elfogadtuk (*ráutaló magatartás*).

A licenceknek néhány jól elkülönített változata alakult ki. Jogi szempontból tehát a szoftvereket aszerint csoportosítjuk, hogy milyen korlátozásokat írt elő a szerzői jog birtokosa.

A tulajdonosi szoftverek

A **tulajdonosi szoftverek*** kódját nem fejthetjük vissza és nem módosíthatjuk. Attól függően, hogy a tulajdonos milyen egyéb korlátokat írt elő, a tulajdonosi szoftver lehet kereskedelmi, freeware [fríver] vagy shareware [server] program.

A **kereskedelmi programokat*** a licenclben rögzített példányszámban **korlátozás nélkül használhatjuk**, azonban **nem ingyenesek**. Az ilyen szoftverek nem adhatók tovább, és csak egy biztonsági másolatot készíthetünk róluk.



Származási bizonylat, licenc, regisztrációs kártya: a szoftver legális. Melyik dokumentum milyen célra szolgál?

Ügyviteli szoftverek esetében gyakori megoldás, hogy a vevő *bérl*i a szoftvert a készítőjétől, viszont így mindig a legfrissebb verziót használhatja. A felhasználó vagy adott időtartamonként, vagy adott számú bizonylat kinyomtatásáért fizet. (Ilyenek például a számlázó vagy a raktárkezelő programok.)

A magyar jogszabályok szerint **egy kereskedelmi szoftver akkor legális, ha licenccel és származási bizonylattal is rendelkezünk**. A licenclben a gyártó feljogosít a szoftver használatára, és ez tartalmazza a felhasználás korlátait is. A származási bizonylat a megszerzés módját dokumentálja, lehet például számla, adásvételi szerződés, átvételi elismervény. A dobozban gyakran megtaláljuk a *regisztrációs kártyát* is, ennek segítségével megadhatjuk adatainkat a szoftvergyártónak. Visszaküldése nem kötelező, viszont ha megtesszük, tájékoztatást kaphatunk a program újabb változatairól, a feltárt hibákról és azok javításáról, a gyártó akcióiról stb.

A **freeware programok*** is **korlátozás nélkül használhatók**, azonban **ingyenesek és szabadon terjeszthetők**. Gyakori, hogy egy szoftver freeware programként vált népszerűvé, majd újabb verziója üzleti szoftver lett.

A **shareware programok*** annyiban térnek el a freeware szoftverektől, hogy **ingyenesen csak korlátozásokkal használhatók**. Ez általában időkorlát (pl. 30 nap), de gyakori, hogy bizonyos funkciók le vannak tiltva, vagy reklámképernyők jelennek meg indításakor, illetve használata közben. A korlátozások feloldásához már fizetnünk kell.

A szabad szoftverek

A *szabad szoftverek** is a szerzői jog védelme alatt állnak (tehát a szerzők arról nem mondtak le), **azonban a szerző nem írt elő korlátozásokat**. Ez a következőket jelenti. A program ingyenesen beszerezhető, bármilyen célra korlátlanul használható és korlátlanul terjeszthető. A szabad szoftver **kódját a felhasználónak jogában áll megismerni és akár módosítani is**.

A szabad szoftvert *szabad pénzért is árulni*, de akkor a terjesztőnek valamilyen szolgáltatást még hozzá kell adnia, különben miért fizetnének érte? Ez lehet szoftverek válogatása, dokumentáció készítése, támogatás biztosítása. (Ha kéri, akkor a felhasználónak a programkódot ebben az esetben is ingyenesen kell megkapnia.)

A jogi szabályozás általában azt is megoldja, hogy a módosított programot se lehessen üzleti szoftverként értékesíteni.

További kategóriák

A *félszabad szoftverek* esetében a felhasználás módjától függ, hogy a szoftverért fizetni kell-e: pl. otthoni felhasználásra ingyenes, üzleti felhasználásra már nem.

Végül érdekes kategória a *public domain* [páblik domén] *szoftver*; amikor a szerző véglegesen lemondott szerzői jogáról. A fentiekől eltérően a program ebben az esetben nem áll szerzői jogvédelem alatt, így készítőjének később sincs joga a szoftvert átsorolni a szabad vagy a tulajdonosi szoftverek kategóriájába.

A programok telepítése

A telepítés menete

Amikor valaki új helyre költözik, nem elegendő pusztán „behurcolkodni” az új házba, hanem be kell jelentkezni az önkormányzatnál, a villamosműveknél, kérnie kell telefonvonalat stb. Hasonló módon, **ha egy új alkalmazói programot szeretnénk gépünkön elhelyezni, akkor nem elegendő azt pusztán felmásolni. Meg kell teremteni közte és az operációs rendszerünk között a kapcsolatot, hozzá kell szabni gépünk hardver- és szoftverfeltételeihez. Ezt a folyamatot nevezik telepítésnek***.

A telepítés a CD/DVD behelyezésével többnyire automatikusan elindul. Ha nem, akkor azt az adathordozón lévő **INSTALL** vagy **SETUP** [szetáp] nevű programmal kell elkezdenünk. Ezt követően a folyamatot általában a *telepítőprogram* automatikusan elvégzi. Kérdéseit, javaslatait többnyire csupán el kell fogadnunk, mégis feltétlenül olvassuk el, mielőtt lenyomjuk a *Tovább* gombot. Ilyen például, hogy a szoftver mely komponensei települjenek, illetve a merevlemezen milyen alkönyvtárba kerüljön. Üzleti szoftverek telepítése során meg kell adnunk a szoftver eredetiségét ellenőrző *termékkulcsot*. A folyamat végén a telepítő bejegyzi az operációs rendszer nyilvántartásába a program futtatásához szükséges adatokat (Windows esetén ezt *regisztrációs adatbázisnak* nevezik).

Aktiválás és regisztráció

A legtöbb alkalmazói program a telepítés végén felajánlja a termék regisztrációját és aktiválását. Fontos tudnunk, hogy a két fogalom teljesen különböző dolgot jelent.

Az aktiválás célja az illegális szoftvermásolás megakadályozása. Ha egy szoftver esetében a gyártó kötelezően előírja az aktiválást, akkor az aktiválás nélkül csak korlátozott feltételek mellett használható. (Ilyenek pl. a Microsoft Office 2003, az Adobe PhotoShop CS vagy a QuarkXPress 6.0.) Az aktiválás internet-hozzáférés esetén automatikusan, rövid idő alatt lezajlik. *Számítógépünk elküldi a termékkulcsból és a hardver bizonyos paramétereiből képezett telepítési azonosítót a gyártónak, az pedig visszaküldi a korlátozások feloldásához szükséges visszaigazolási azonosítót.*

A regisztráció célja a felhasználó jobb kiszolgálása, azonban nem kötelező igénybe venni. A regisztráció többnyire interneten keresztül zajlik, de történhet a szoftverhez kapott regisztrációs kártya postán történő elküldésével is.

Aktiválás varázsló

Csatlakozás...



Ha a kapcsolódással probléma van vagy egy percnél hosszabb hogy megfelelően működik-e az internetkapcsolat.

Köszönjük, hogy telepítette a(z) Microsoft Office Professional Edition-t az összes szolgáltatást használhassa, aktiválni kell a telepítést. A varázsló lépéseln.

Hogyan szeretné aktiválni a szoftvert?

- Interneten (javasolt)
- Telefonon

Az internetes aktiválás lényegében automatikusan zajlik. Ha gépünk hardverösszetétele jelentősen módosul, akkor szükséges lehet az újraaktiválás. Mi az aktiválás célja?

Összefoglaló kérdések, feladatok

1. Soroljuk fel a számítógépünkre telepített programokat, és adjuk meg besorolásukat funkciójuk és jogi szabályozásuk szerint!
2. Mi a BSA? Keressünk adatokat az interneten!
3. Mekkora az illegálisan használt szoftverek aránya Magyarországon, illetve az Európai Unió tagállamaiban? Keressünk információkat az interneten!
4. Milyen esetben használunk legálisan egy kereskedelmi szoftvert? Milyen előnyei vannak a legálisan használt kereskedelmi szoftvernek a kalózmásolattal szemben?
5. Foglaljuk össze a kereskedelmi szoftver, a freeware, a shareware, a szabad szoftver és a public domain jellemzőit a következő szempontok szerint. Ingyenes-e, korlátozás nélkül használható-e, szabadon terjeszthető-e, a kód megismerhető-e?
6. Mit jelent a program regisztrálása? Milyen előnyökkel jár a felhasználó számára?

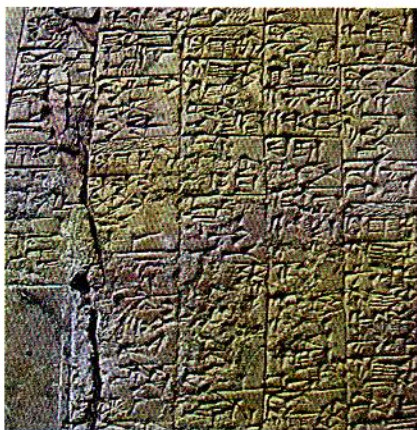
Könyvtárhasználat

A könyvtárak kialakulása

Az írásbeliség kialakulásával és fejlődésével együtt járt a *dokumentumtárak* kialakulása is. A dokumentumok ugyanis sokkal hatékonyabban töltik be szerepüket, ha azokat összegyűjtik és rendszerezik, mivel így sokkal könnyebben és gyorsabban juthatunk hozzá a megfelelő információkhoz. Érthető tehát, hogy az első könyvtárak a *királyi udvarok* és az oktatást is végző *templomok* mellett alakultak ki.

A régészek által megtalált legrégebbi „könyvtár” Kr. e. 2400 körül jött létre: az észak-szíriai *Ebla* város romjai között 1975-ben találták meg a királyi levéltárat. A mintegy 16 000 agyagtábla államközi szerződéseket, törvénykivonatokat, iskolai szövegeket, sőt egy kétnyelvű szótárt is tartalmazott.

Az ókor leghíresebb könyvtára az egyiptomi *Alexandriában* működött, és több mint félmillió papirusztekercset őrzött. A könyvtár teljességre törekedett: a kor valamennyi tudományos és irodalmi művéből igyekeztek egy-egy példányt összegyűjteni. Sajnos a könyvtár Kr. e. 43-ban egy tűzvészben elpusztult.



Ékírásos agyagtábla Ebla városából



Részlet egy corvinából

A *közkönyvtárak* a *rómaiaknál* jelentek meg, a IV. században már 28 működött a városban. Lényegében náluk találkozunk először a könyvkiadással is: a tekercseket kézzel másolták a rabszolgák.

A koraközépkori Európában a könyv szerepe elsősorban az egyházi tanok terjesztésére szorítkozott. A kézzel másolt kódexeket a kolostorokban, székesegyházi és káptalani

könyvtárakban őrizték. Magyarországon például a *pannonhalmi bencés monostort* még GÉZA fejedelem hozta létre 966-ban, az apátság könyvtárában a XI. század végén már mintegy 80 kötetet (kb. 200 művet) tartottak nyilván.

A társadalmi fejlődés Európában csak a XIII-XIV. században hozta létre az első egyetemeiket, és velük együtt az *egyetemi könyvtárakat* (Sorbonne [szorbon]: 1257, Oxford: 1345, Prága: 1366). A reneszánsz idején megjelentek a gazdag főúri és királyi magánkönyvtárak is. MÁTYÁS KIRÁLY híres könyvtára például mintegy 2500 kódexet tartalmazott, sajnos ebből csak 52 kötetet őriznek Magyarországon. Az egyetemi és a reneszánsz magánkönyvtárak már nagy részben világi témájú könyveket tartalmaztak.

A polgáriasodás szükségessé tette az iskolai oktatás kiterjesztését, így a *városokban* hamarosan megjelentek az első könyvtári kezdeményezések. Ennek technikai hátterét a *könyvnyomtatás* felfedezése tette lehetővé, így a könyveket sokkal olcsóbban lehetett sokszorosítani. A folyamatot erősítette a reformáció, illetve az ellenreformáció is. Magyarországon ekkor jöttek létre a *protestáns iskolai könyvtárak* (Sárospatak: 1531, Debrecen: 1547), illetve a jezsuiták által alapított *Egyetemi Könyvtár* (Nagyszombat: 1635). A rend feloszlata után 1773-ban az egyetem és az Egyetemi Könyvtár állami intézménnyé vált, és hamarosan Pestre költözött.

A megnövekedett igény hatására a XVII. században Európa-szerte újra megjelentek a *közkönyvtárak*. A folyamatra a francia forradalom tette fel a koronát azzal, hogy kimondta: *a könyvtárak létesítése és fenntartása közérdek*, tehát állami feladat. A párizsi királyi könyvtárat nemzeti könyvtárrá nyilvánították, és számos egyházi könyvtárat is állami tulajdonba vettek.

A magyar nemzeti könyvtárat GRÓF SZÉCHÉNYI FERENC (1754-1820) hozta létre 1802-ben, amikor a nemzetnek ajándékozta mintegy 15 000 kötetes könyvtárát. Az *Országos Széchényi Könyvtár* állománya ma már meghaladja az 5 millió kötetet.



Széchenyi Ferenc



Az Országos Széchényi Könyvtár

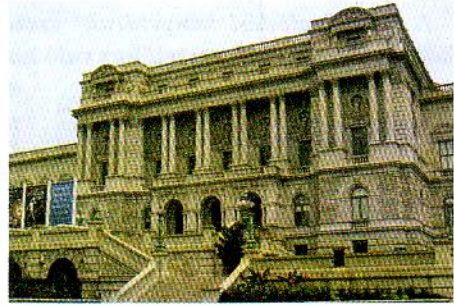
A tudományos kutatások támogatására GRÓF TELEKI JÓZSEF 1826-ban a Széchényi István által alapított Magyar Tudományos Akadémiának ajánlotta föl könyvtárát, így jött létre az *Akadémiái Könyvtár*. Hamarosan megalakultak tudományos és szakkönyvtáraink, így az Országgyűlési Könyvtár, az Ipartanodai Könyvtár és a Technológiai Könyvtár.

(A két utóbbi könyvtár utódja a BME Könyvtár és Tájékoztatási Központ, illetve az Országos Műszaki Információs Központ és Könyvtár 2001-ben *BME OMIKK* néven egyesült.)

A XIX. században új könyvtártípus jelent meg, a közművelődési könyvtár, melynek célja az, hogy a könyvtári szolgáltatások mindenkihez eljussanak. Magyarországon SZABÓ ERVIN 1904-ben alapította meg a Fővárosi Nyilvános Könyvtárat, a mai *Fővárosi Szabó Ervin Könyvtár* elődjét.



A párizsi Francia Nemzeti Könyvtár
(Bibliothèque De France [biblioték dö fransz])
modern épületegyüttese



A washingtoni Kongresszusi Könyvtár
(Library of Congress [lájbróri of kongresz])
28 millió kötetet őriz.

A könyvtár

A könyvtár fogalma, típusai

A könyvtár nem egyszerűen könyvek halmaza. A *könyvtár** olyan intézmény, amely a könyveket, folyóiratokat és más anyagokat gyűjti, rendszerezi, és az olvasók rendelkezésére bocsátja. A könyvtárak a kölcsönzésen és a helyben történő olvasáson túl további szolgáltatásokat is nyújtanak. Helyet adnak a tanulásnak és az oktatásnak, biztosítják a jogszabályokhoz való hozzáférést, sőt újabban lehetőséget biztosítanak az internetezésre is. Állományuk ma már nemcsak a nyomtatott anyagokra korlátozódik, hanem gyakran zenei CD-ket vagy filmeket is tárolnak.

A nyilvános könyvtárak működtetése, mint láttuk, az állam és a helyi önkormányzatok feladata. A könyvtárak igen nagy értéket tárolnak, ezért működésük rendjét jogszabály rögzíti. (Magyarországon jelenleg az 1997. évi CXL. törvény szabályozza a nyilvános könyvtári ellátást.) A könyvtárak nem függetlenek egymástól, a különböző könyvtártípusok egységes rendszert alkotnak.

A könyvtárakat hagyományosan a következő csoportokba soroljuk:

- nemzeti könyvtár,
- közművelődési könyvtárak,
- szakkönyvtárak,
- felsőoktatási könyvtárak,
- iskolai könyvtárak.

A **nemzeti könyvtár*** feladata minden magyar vagy magyar vonatkozású anyag gyűjtése a teljesség igényével. A magyar nemzeti könyvtár az Országos Széchényi Könyvtár, amely a Budavári Palotában működik. Gyarapodását jogszabály biztosítja: minden Magyarországon kiadott műből adott számú példányt kell ellenszolgáltatás nélkül kapnia, ezt *köteles példánynak* nevezik. Mint az ország központi könyvtára, a többi könyvtár részére szolgáltatásokat nyújt: központi katalógust vezet, szervezi a könyvtárak közötti kölcsönzést, a nemzetközi könyvcserét stb.

A **közművelődési könyvtárak*** szerepe, hogy mindenki számára biztosítsák a könyvekhez, az információhoz való hozzáférés jogát. A helyi könyvtárak fenntartása



A Fővárosi Szabó Ervin Könyvtár központi épülete

a települési önkormányzatok feladata, együttműködésüket a megyei könyvtárak szervezik. Gyakori, hogy a hozzánk legközelebb eső könyvtár egy nagyobb könyvtár fiókkönyvtára. Például a budapestieket ellátó Fővárosi Szabó Ervin Könyvtár mintegy 60 fiókkönyvtárat üzemeltet.

A **sakkönyvtárak*** egy adott tudományág vagy szakmai terület dokumentumait gyűjtik. Feladatuk kiterjed az adott szakterülethez tartozó információk szolgáltatásának biztosítására is. A korábban már megismert BME OMIKK, Országgyűlési Könyvtár, Akadémiai Könyvtár mellett ilyenek még az Országos Mezőgazdasági Könyvtár, az Országos Pedagógiai Könyvtár és Múzeum, a Hadtörténelmi Intézet Könyvtára stb.

A **felsőoktatási könyvtárak*** az adott egyetem vagy főiskola oktató munkáját és tudományos kutatásait segítik. Ezek a könyvtárak gyakran több száz évesek, és hatalmas szakirodalommal rendelkeznek. Gyakran sakkönyvtári feladatokat is ellátnak.

Az **iskolai könyvtárak*** értelemszerűen az iskolai oktatást szolgálják. Fontos feladatuk a tanulók felkészítése az önálló könyvtárhasználatra, illetve a tanítók, tanárok számára a szükséges pedagógiai irodalom biztosítása. Kisebb települések esetén az iskolai könyvtár a közművelődési könyvtár szerepét is betöltheti.

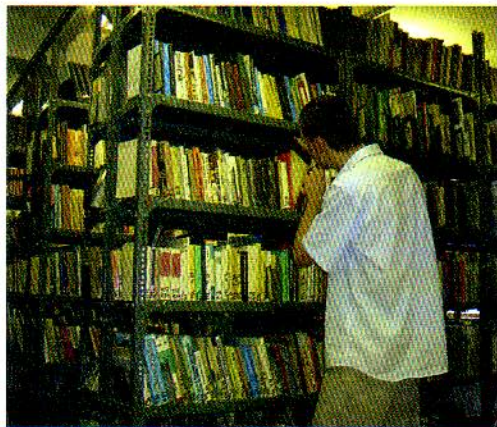
A könyvtár funkcionális részei

Először foglaljuk össze röviden, mit látunk, ha belépünk az iskolai könyvtárba!

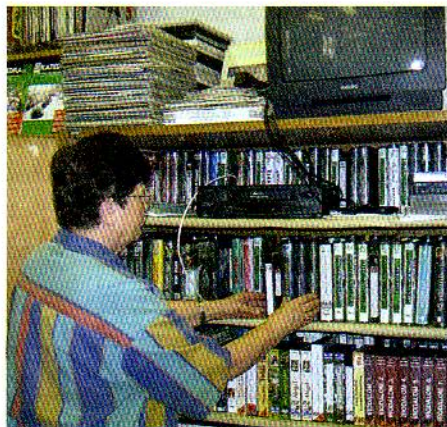
A **bejárati térség** a könyvtár legzajosabb egysége. Szerves része a ruhatár, de itt helyezik el a közérdekű információkat (prospektusok, hirdetések) is, sőt gyakran kisebb kiállításokkal is találkozhatunk itt.

Az iskolai könyvtár viszonylag nagyobb egysége a **csoporthoz tartozó tanulási és foglalkoztató terület**. Ez az asztalokkal és székekkel berendezett rész lehetővé teszi a könyvek és folyóiratok olvasását, a tanulást vagy csoportos foglalkozások tartását. Gyakran helyezik el itt a videót vagy számítógépet egy nagy képernyős televízióval vagy projektorral, hogy csoportok számára is be lehessen mutatni filmeket, multimédiás anyagokat.

A könyvtár másik nagy térsége a **könyvkiválasztó övezet***. Ez általában **szabadpolcos*** rendszerű, vagyis a polcokon elrendezett könyvek között közvetlenül válogathatunk.



Szabadpolcos könyvkiválasztó övezet. Mik az előnyei és hátrányai a zárt raktárhoz képest?



Milyen célt szolgál a multimédiás övezet az iskolai könyvtárban?

A **nyilvántartási, ellenőrzési övezet*** több részre tagolódik.

Nem messze a könyvtár bejáratától helyezkedik el a **kölcsönzőpult**. A könyvtárosok itt végzik a kölcsönzéssel kapcsolatos adminisztrációt, segítenek az eligazodásban, válaszolnak kérdéseinkre.

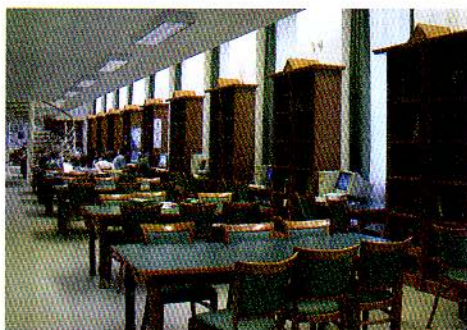
Többnyire a kölcsönzőpult közelében található a könyvek keresését segítő **katalógusszekrények**, illetve az iskolai számítógép-hálózatba kötött **könyvtári számítógépek** is.

A **folyóiratolvasóban** az ismeretterjesztő, szakmai és egyéb folyóiratokat találjuk, ezek általában nem kölcsönözhetők.

A folyamatos munkához szükségünk lehet lexikonokra, enciklopédiákra, szótárakra, bibliográfiákra stb. Ezeket a könyveket a **kézikönyvtár** tartalmazza, s mivel bármikor szükség lehet rájuk, természetesen nem kölcsönözhetők.

FOLGALTATÁS

- e-Mapa ország
- Kereséssel kezelhető
- DVD-kölcsönzés
- CD-kölcsönzés
- Kötte-kölcsönzés
- Audiókasszeta-kölcsönzés
- Internet-hozzáférés
- Nyomtatás
- Szkennelés
- Hízhozzáférés
- Idegen nyelvi dokumentum
- Könyv-kölcsönzés
- CD-ROM-kölcsönzés
- Videó-kölcsönzés
- Fénymásolás
- Internet-tanfolyam
- Országos könyv számítógép-használat
- Diafilmek
- Gételem
- Gyermekrészleg
- Grafika-kölcsönzés
- Európai Információs Pont
- Leállítás, értékelés
- Kölcsönzés
- Felolvasóterem
- Térkép
- Folyóirat-kölcsönzés
- Olvasóterem
- Hangdokumentum-kölcsönzés



A könyvtárak a kölcsönzésen és a helyben olvasáson túl nagyon sok további szolgáltatást is nyújtanak. Soroljunk fel néhányat az ábra alapján!

Az Eötvös Loránd Tudományegyetem Bölcsészkarának olvasóterme. Mi az olvasóterem szerepe?

Könyvtárhasználat

Nagyobb könyvtárak esetén, amelyek esetleg több millió könyvet is tartalmaznak, a szabadpolcos rendszer nem oldható meg. Ekkor a könyveket **zárt raktárakban** tárolják, az olvasó pedig a katalógus segítségével válogathat a könyvek között. Gyakori a **vegyes rendszerű** megoldás is: a könyvek egy része szabadpolcon található, míg a ritkábban keresett könyvek zárt raktárban vannak.

A nagy könyvtárakban sok olyan könyv van, amely nem kölcsönözhető. Ennek többféle oka lehet, például hogy a könyv nehezen pótolható ritkaság, különösen értékes, vagy nagy iránta a kereslet, ugyanakkor kevés példány van belőle. Az ilyen könyveket az **olvasóteremben*** olvashatjuk.

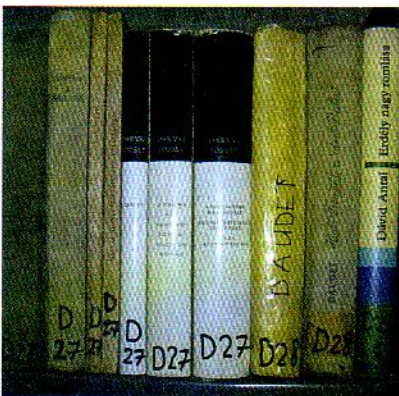
Keresés a könyvtárban

Szépirodalmi művek elrendezése

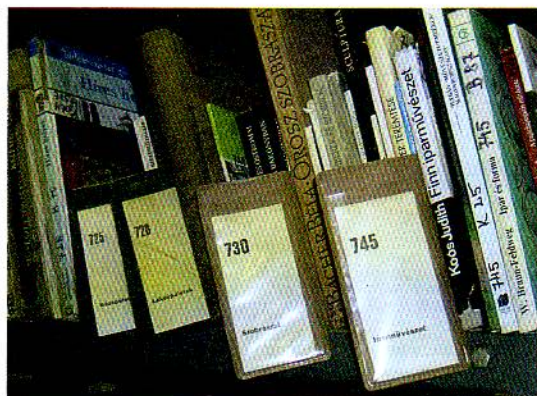
A szépirodalmi műveket a polcokon a **szerző vezetékneve alapján, szoros ábécé-rendben helyezik el.** A rendezés alapja a **könyvtári ábécé.** Ennek lényege, hogy nem tesz különbséget a rövid és a hosszú magánhangzók között, illetve nincsenek benne ket-tős betűk. Tehát az A, illetve az Á betűvel kezdődő nevek egy csoportba kerülnek, csakúgy, mint a Z-vel és a ZS-vel kezdődőek.

A könyv helyét a könyv gerincén is feltüntetik. Ez az azonosító az ún. **Cutter-szám** [kátter] vagy **betűrendi jel.** A **Cutter-szám*** egy betűből és egy kétjegyű számból áll. **A betű az író nevének kezdőbetűje, a szám pedig a szerzők nevének ábécé szerinti sorrendjében nő.** Természetesen a szám minden betűnél újratekődik.

Ha egy könyv több szerző művét tartalmazza (pl. verseket vagy novellákat tartalmazó antológiák), akkor a besorolás alapja a könyv címe. Ha egy szerző több könyvet is írt, akkor azok a mű címe szerinti szoros ábécé alapján következnek. Ilyenkor azonban a cím elején lévő névelőket nem veszik figyelembe.



A szépirodalmi könyvek között a Cutter-szám teremt rendet. Hogyan képezik?



Az ismeretterjesztő könyveket ETO-szám, azon belül Cutter-szám szerint rendezik. Mi az ETO-szám szerepe?

Az ismeretterjesztő művek elrendezése

Az ismeretterjesztő műveket témájuk szerint csoportosítják. A csoportosítás alapja a nemzetközileg elfogadott Egyetemes Tizedes Osztályozás, röviden ETO vagy *szakjelzet*. **Az ETO* a könyveket 10 főcsoportba sorolja, ezeken belül 10 csoportba, ezeken belül 10 alcsoportba stb.**

Azonos ETO-szám esetén a könyveket még kiegészítik a Cutter-számmal is, így a könyv tényleges helyét az ETO-szám és a Cutter-szám együttesen határozza meg.

- 0 Általános művek
- 1 Filozófia, pszichológia, erkölcs
- 2 Vallástudományok
- 3 Társadalomtudományok
- 4 (üres)
- 5 Természettudományok
- 6 Alkalmazott tudományok, technika
- 7 Művészetek, sport, játék
- 8 Nyelv- és irodalomtudomány
- 9 Földrajz, történelem, életrajzok

Az ETO főosztályai

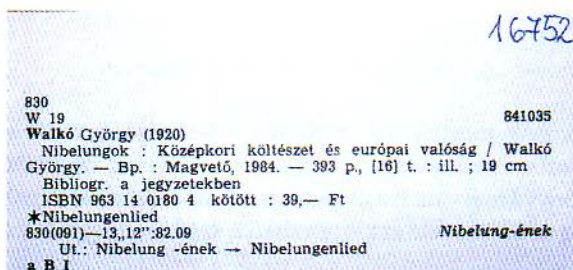
- 5 Természettudományok
- 51 Matematika
- 52 Csillagászat
- 53 Fizika
- 531 Általános mechanika
- 532 Folyadékok mechanikája
- 532.1 Hidrosztatika
- 532.11 Nyomás
- 532.12 Összenyomhatóság
- ...

Példa az alosztályokra

A katalógusok

Ha egy adott könyvet keresünk, de nem tudjuk a helyét, vagy a könyvtár egy része zárt raktárban van, akkor a katalógusok segítségével kell igénybe vennünk.

A **katalógusok*** a katalóguscédulákat tartalmazzák különböző szempontok szerint elrendezve. **Minden katalóguscédula* egy-egy könyvet azonosít**, és tartalmazza a könyv legfontosabb adatait (szerző, cím, kiadó, kiadás éve, Cutter-szám, ETO-szám, a könyv méretei, oldalainak száma stb.).



Katalóguscédula. A tárgyszó a jobb alsó sarokban van. Hol találjuk a Cutter-számot és az ETO-számot?



A katalóguscédulák a katalógus-szekrény fiókjaiban vannak elhelyezve.

A könyvtárakban általában többféle katalógus is szerepel, így egy adott könyvet többféle szempont szerint is kereshetünk. Ugyanarról a könyvről tehát több katalóguscédula is készül, melyet a számunkra kedvezőbb módon találhatunk meg.

A leggyakoribb a **szervi betűrendes katalógus***. Ha a könyvnek több szerzője is van, akkor értelemszerűen a katalóguscédulát minden szerzőhöz be kell illeszteni. Ismeretterjesztő művek esetében gyakran célszerűbb inkább az **ETO-szám szerinti katalógus*** választani.

Gyakori a **tárgyszó szerinti katalógus*** is. A tárgyszavak a könyv tartalmát egy-egy rövid kifejezéssel azonosítják, s a könyveket az így kapott „lexikonszerű” katalógusban helyezik el. Természetesen előfordul, hogy egy könyvet több tárgyszó szerint is besorolhatunk, ekkor a tárgyszó szerinti katalógusban többször fog szerepelni.

A számítógépes katalógus

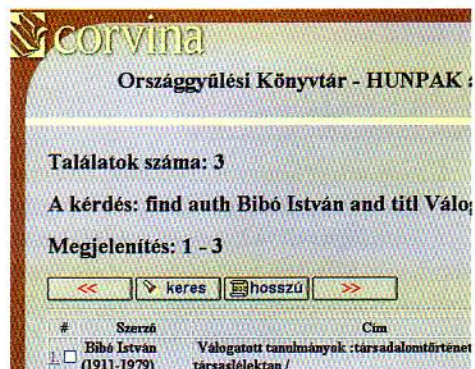
A katalóguscédulák és a katalógusszekrény szerepét egyre több könyvtárban veszi át a számítógépes katalógus. A **számítógépes katalógus a könyvek adatait egy számítógépes adatbázisban tartja nyilván**. A számítógépes adatbázisokban való keresés nemcsak több szempont szerinti keresést tesz lehetővé, hanem a keresési feltételek is összetettebbek lehetnek. Ráadásul ezeket az adatbázisokat az interneten keresztül ma már akár az otthonunkból is elérhetjük.

Mivel a számítógépes adatbázis kialakítása időigényes, ezért sok könyvtárban a két rendszer még együtt él; a könyvek egy része csak a hagyományos módon érhető el.



The screenshot shows the search interface of the Országgyűlési Könyvtár - HUNPAK. It features a search form with the following fields and options:

- Search term: Bibó István
- Field: mnt: Szerző
- Field: Válogatott
- Field: mnt: Cím
- Field: mnt: Tárgyszó
- Buttons: Keresés, Törölés
- Mezők: Megjelenési idő, Nyelv: összes, Típus: összes
- Field: Gyakorlott keresés:



The screenshot shows the search results page of the Országgyűlési Könyvtár - HUNPAK. It displays the following information:

- Találatok száma: 3
- A kérdés: find auth Bibó István and titl Válogatott
- Megjelenítés: 1 - 3
- Navigation buttons: <<, keres, hosszú, >>
- Table of results:

#	Szerző	Cím
1	Bibó István (1911-1979)	Válogatott tanulmányok : társadalomtörténeti társaslélektan /

Keresés az Országgyűlési Könyvtár katalógusában (www.ogyk.hu).

Végezzük el az ábrán lévő keresését egy másik nagy könyvtár elektronikus katalógusában is!

A számítógépes könyvtári adatbázisok megjelenése nemcsak a keresést, hanem a **könyvtár teljes működését** felgyorsította. A könyvtáros nemcsak a könyveket tarthatja nyilván, hanem azt is, hogy melyik könyv kinél van. Ha új könyvet vásárolnak, az azonnal megjelenik a keresőrendszerben; míg az olvasó felé azt is jelezheti a szoftver, hogy nincs bent az általa keresett könyv.

Gyakori, hogy a számítógépes rendszert további elektronikus eszközökkel egészítik ki. Például az olvasójegyeket vagy a könyveket vonalkóddal látják el, így a kölcsönzés, illetve a visszahozás adminisztrációja jelentősen csökkenthető.

Az elektronikus könyvtár és az elektronikus könyv

Az internet megjelenése nemcsak azt tette lehetővé, hogy otthonunkból keressünk a könyvtár katalógusában, hanem azt is, hogy a könyvekhez digitálisan jussunk hozzá. **Az elektronikus könyvtárak* a könyveket digitális formában tárolják**, tehát a monitoron olvashatjuk vagy esetleg ki is nyomtathatjuk őket. Ma már több magyar nyelvű elektronikus könyvtár is található az interneten (pl. mek.oszk.hu).

Az elektronikus könyvtárak segítséget nyújtanak a fogyatékkal élők számára is, hiszen az interneten át otthonukból hozzáférhetnek a könyvekhez. A látásukban korlátozottak számára pedig hasznos szolgáltatás, hogy a könyveket gyakran akár meg is hallgathatják. (Ilyen például a magyar nyelvű Világhalló elektronikus könyvtár is.)

A „hagyományos” könyvek digitalizálása mellett egyre több olyan könyv van, amit csak elektronikusan adnak ki. A **digitális könyv (e-book (fbuk))** felépítése hasonló a papír alapú társához, azonban csak elektronikus formában, pl. PDF fájlban szerezhető be.



Felül a Magyar Elektronikus Könyvtár, alul pedig a Világhalló elektronikus könyvtár nyitólapja.



A 42 soros Biblia egy oldala. Mikor és ki készítette?

Dokumentumismeret

A dokumentum fogalma, fajtái

A **dokumentum*** az ismereteket rögzítő információhordozó. Mai fogalmaink szerint ide tartozik a könyv, a folyóirat, a napi- és hetilapok, de a zenei CD, a filmszalag, és a számítógép memóriája is stb. Első megközelítésben a dokumentumokat két csoportba sorolhatjuk: a **nyomtatott*** és a **nem nyomtatott*** dokumentumokra.

A dokumentum megjelenési formája az emberiség története során folyamatosan változott. A dokumentumokat kezdetben **kézírással** rögzítették. Mivel ez a folyamat meglehetősen lassú és költséges volt, hatalmas ugrást jelentett Gutenberg találmánya, a szétszedhető és újra összerakható nyomóformát használó **nyomda**. (Az első így előállított könyv a 42 soros Biblia volt, melyet Gutenberg 1454-1455-ben készített.)

A következő lépést a *hang*, illetve a *látvány rögzítése* jelentette. E két terület ma is gyorsan fejlődik. Edison fonográfja helyett ma már CD-lejátszót használunk; a kémiai úton előhívott papírképeket kezdik kiszorítani a digitális kamerák; a filmszalag mellett pedig előbb a videoszalag jelent meg, ma pedig egyre inkább terjed a DVD lemez.

A XX. század második felének találmánya, a *számítógép* nemcsak a „hagyományos” adathordozók digitális változatát hozta magával, de új formákat is teremtett: ilyen például az internet. (A számítógépes dokumentumokkal a korábbi fejezetekben részletesen megismerkedtünk.)

A könyv

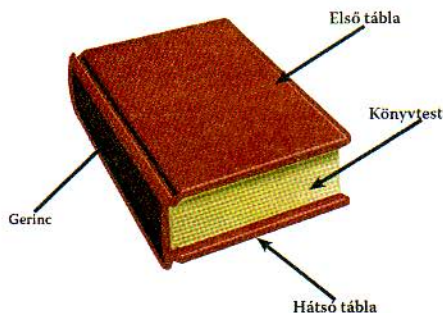
A könyvtárak legmeghatározóbb dokumentuma a könyv. A *könyv** írásjelekkel rögzíti azokat a *szellemi alkotásokat*, amelyeket szeretnénk megőrizni és terjeszteni.

Fizikai megjelenését tekintve a *könyv kétféle lehet: keménytáblás (kötött) vagy puhatáblás (fűzött)*. A *keménytáblás* könyv lapjait cérnával varrják össze, és az így kapott könyvtestet rögzítik a kötéstáblába. Igényesebb esetben a kötéstábla vászonzól vagy bőrből készült borítást is kaphat. A *puhatáblás* könyvek lapjait egymáshoz ragasztják, és a könyvtesthez is ragasztással rögzítik a kartonból készült fedelet. Gyakori, hogy egy könyvet mindkét változatban kiadják, ilyenkor a két változat között a „kötve”, illetve a „fűzve” jelzéssel tesznek különbséget.

A *könyvek nemzetközi azonosítására az ISBN szám** (International Standard Book Number [internetsnőlstenderbuknábör]) szolgál. Az azonosító egyedi, még egy könyv különböző kiadásai is eltérő számot kapnak. A hazánkban kiadott könyvek ISBN száma korábban 963-mal kezdődött, ma ezt megelőzi a 978 vagy 979 számsor is. Ez a szám azonosítja a könyvet a *Magyar Nemzeti Bibliográfiában* is, amely a magyar vonatkozású műveket tartalmazza, s melyet az Országos Széchényi Könyvtár vezet.

Ha belelapozunk egy könyvbe, akkor láthatjuk, hogy a *könyv két jól elkülöníthető részre tagolható*: a *főszövegre* és a *járulékos részekre*. A *főszöveg a szerző érdemi mondanivalóját tartalmazza, a járulékos részek pedig kiegészítik, magyarázzák azt*, vagy segédletekkel látják el a könyvet. Ilyen járulékos részek a *címnegyedív*, az *előszó*, a *tartalomjegyzék*, az *irodalomjegyzék*, a *tárgymutató* stb.

A *címnegyedív általában 4 oldalból áll*. Az első a *szennycímloldal*, amely csupán a



A kötött könyv részei. Miben tér el a fűzöttől?



Az ISBN szám és a vonalkód egy korábban kiadott könyv hátulján. Ma már a két szám azonos.

szerző nevét és a könyv címét tartalmazza. Ezt követi a **sorozatcímoldal**, amely az egy sorozatba illeszkedő könyvek esetén a sorozatra vonatkozó adatokat tartalmazza. A legfontosabb oldal a **címoldal**, amely kötelezően tartalmazza a szerző nevét, a mű címét, a kiadó nevét, a kiadás évét és helyét. A negyedik a **copyrightoldal** [kopirájt], ide kerülnek a szerzői jogokra vonatkozó adatok, a mű létrejöttében szerepet játszó személyek neve és a kiadó adatai.



A sorozatcímoldal és a címoldal Gárdonyi Géza könyvének a két világháború közötti kiadásában. Milyen kötelező adat hiányzik a címoldalról?

Tartalmilag a könyveket két fő csoportba sorolhatjuk: lehetnek ismeretközlő művek vagy irodalmi művek. Ezeket részletesen a következő táblázat tekinti át:

Ismeretközlő művek:

- tudományos könyvek,
- szakkönyvek: gyakorlati ismereteket tartalmaznak,
- ismeretterjesztő könyvek: alap-, középfokon mutatnak be egy adott szakterületet

Irodalmi művek:

- szépirodalom: művészi értékkel rendelkező alkotások
- szórakoztató irodalom: a szórakozást, kikapcsolódást segítő „alkalmazott” irodalom

Mind az ismeretközlő, mind az irodalmi műveken belül megkülönböztetjük a gyermekeknek írt könyveket, a **gyermekirodalmat**.

A könyveken belül külön csoportot alkotnak a **segédkönyvek***, amelyek **más könyvek olvasása közben adnak segítséget**. Tekintsük át a segédkönyvek főbb csoportjait!

• **Lexikonok és enciklopédiák.** Az ismereteket a *lexikonok betűrendben*, az *enciklopédiák pedig tematikusan* csoportosítva dolgozzák fel. Történelmileg talán a leghíresebb a

Nagy Francia Enciklopédia, melyet 1751 és 1776 között 35 kötetben adtak ki. Ma a lexikonok és enciklopédiák gyakran elektronikus úton is megjelennek, sőt van olyan lexikon (pl. *Encarta* [enkárta]), melyet kizárólag elektronikusan készítenek.



Révai Nagy Lexikona (a kép forrása: wikipedia). Melyik volt az első lexikon?



Wikipédia, a szabad internetes lexikon.
Keressük fel (www.wikipedia.org/)!

- **Szótárak.** A szótárak kapcsán általában kétnyelvű szótárak jutnak eszünkbe, ezek két különböző nyelv címszavait rendelik össze. Fontosak azonban az egynyelvű szótárak is, pl. a helyesírási szótárak, az Idegen szavak szótára vagy a Magyar értelmező kéziszótár.

- **Atlaszok.** Céljuk a földrajzi, illetve történelmi tájékozódás, ezért a térképeken kívül bőségesen tartalmaznak adatokat is. Gondoljunk pl. az autóstérképekre.

- **Bibliográfiák.** A bibliográfiák egy-egy témakörben összegyűjtik a kiadott könyvek jegyzékét. Ilyen például a már említett Magyar Nemzeti Bibliográfia is.

- **Adattárak, névtárak, címtárak.** Ezek egy-egy szakterület adatait gyűjtik össze. Ilyenek például a menetrendek, telefonkönyvek vagy akár a Felsőoktatási felvételi tájékoztató is.

Időszaki kiadványok

Jóllehet a középkori kódexekhez képest a nyomtatott könyv előállítása sokkal gyorsabb volt, a XVII. században már ez a sebesség sem volt elegendő. Az információ gyorsabb terjesztése érdekében ekkor jelentek meg az első tudományos folyóiratok, majd ezeket követték az irodalmi, politikai, szórakoztató kiadványok. Magyarországon az első magyar nyelvű hírnap, a *Magyar Hírmondó* 1780. január 1-jén jelent meg Pozsonyban.

Az **időszaki kiadványok* közös jellemzője a folyamatos megjelenés.** Ha ez szabályos időközönként történik, akkor *sajtóról* beszélünk (napilapok, hetilapok stb.), ha pedig nem, akkor *nem folyóirat jellegű kiadványokról* (évkönyvek, almanachok).

Az **időszaki kiadványokat az ISSN (International Standard Serial Number [internesönöl sztenderd szeriöl nábör]) szám* azonosítja.** (Ez az azonosító magára az időszaki kiadványra vonatkozik, nem pedig annak egyes számaira.) Az ISSN szám egy nyolcjegyű szám, melyet általában megelőz az ország kódja, például: HU ISSN 1215-5551.

A **napilapok*** a gyors tájékozódást szolgálják. A legtöbb napilap általános témájú (bár főleg politikai jellegű), de vannak egy-egy témára, pl. a sportra vagy a gazdasági életre szakosodott napilapok is. Főbb jellegzetességeik:

- Címoldal helyett **fejlécük** van, amely tartalmazza a napilap címét, évfolyamát az azon belüli számmal, és többnyire a jellegét (pl. politikai napilap).



Az első magyar hírlap, a Magyar Hírmondó első száma 1780-ból



A legrégebbi magyar napilapot, a Népszavát 1873-ban alapították.

- A főoldalon található meg a **vezércikk**, amely a lap véleményét vagy álláspontját tükrözi egy aktuális témában.
- A belső oldalakon lévő információkat **rovatokba** rendezik (pl. napi hírek, belpolitika, külpolitika, olvasói levelek, sport).
- Általában az utolsó oldalon található meg a **kolofon**, amely tartalmazza a szerkesztők nevét, a szerkesztőség, a kiadó és a nyomda adatait, valamint a terjesztéssel kapcsolatos információkat.



Az egyik hetilap még őrzi a napilapok jellegzetességeit, a másik inkább a folyóiratokhoz hasonló.

A **hetilapok*** témája változatosabb: lehetnek pl. politikai, művészeti, irodalmi, szakmai, szórakoztató jellegűek. Igen népszerűek az általános témájú, képekkel bőségesen illusztrált hetilapok, a *magazinok*. A hetilapok formailag a napilapok rokonai, kivételükben azonban gyakoribb a színes címlappal ellátott, összetűzött vagy ragasztott megoldás.

A **folyóiratok*** **elsődleges feladata a friss tudományos és szakmai információk közreadása**. Ennek megfelelően egy-egy témára szakosodtak, bár vannak általános témájúak is. Formai jellemzőik a következők:

- A címadatokat a többnyire puha kartonból készült **borítófedél** tartalmazza.
- A **lapok számozása** kezdődhet ugyan minden számban egytől is, de gyakori az évfolyam első számától induló folyamatos számozás.

• Számonként **tartalomjegyzéket**, az évfolyam utolsó számában pedig gyakran **összesített név- és tárgymutatót** is közölnek.

• A cikkekről rövid tartalmi kivonatokat készítenek, melyeket régebben több évre visszamenőleg önálló kötetben jelentettek meg (*repertórium*). Ezt ma már inkább elektronikus formában teszik elérhetővé.

A **nem folyóirat jellegű dokumentumok** ugyan szabálytalan időközönként jelennek meg, az egyes számok mégis összetartoznak. Ilyenek például az évkönyvek: az iskolai évkönyvek általában szabálytalan időközönként, például valamilyen eseményhez kapcsolódóan látnak napvilágot.

Nem nyomtatott dokumentumok

A **nem nyomtatott dokumentumok*** közös jellemzője, hogy megjelenítésükhöz általában valamilyen **segédeszköz is szükséges**. A technika fejlődésével folyamatosan új adathordozók jelennek meg, miközben mások elavulttá válnak. Előfordulhat, hogy egy-egy régi dokumentumot azért nem tudunk ma már használni, mert nincs meg hozzá a megfelelő eszköz. Fontos ezért az ilyen információk **migrálása**, vagyis a kiszorulóban lévő hordozóról új hordozóra másolása.

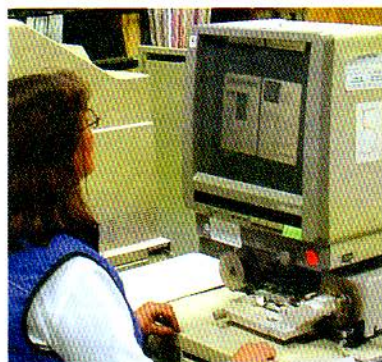
A **mikrofilm** a XX. század második felének jellegzetes adathordozója. Lényege, hogy az értékes, sérülékeny eredeti dokumentumokat lefotózzák, és az eredeti példányok helyett az olvasó a filmet

A Középiskolai Matematikai és Fizikai Lapok 54. évfolyamának tartalomjegyzéke (2004)



Cikkek:	
Kiss Emil - Matusang Tóbiás: Számelméleti feladatok és számelméleti feladatok	37
Csirmerek	42
Dósa Tibor: Két híres parabolikus görbe	45
Frenai Róbert: Széleskörű feladatgyűjtemény	48
Tóbiás Aron: Egy KÖMAL feladat megoldása	51
Tóbiás Aron: H. 3677. feladat megoldása	54
Tóbiás Aron: H. 3677. feladat megoldása	57
Miklós Balázs: Beszámoló	60

Az évfolyam utolsó száma tartalmazza a **KöMaL tartalomjegyzékét**.



Mikrofilm olvasó

tanulmányozhatja. Egy-egy mikrofilmlapra, amelynek mérete kb. az A4-es lap fele, 30-120 oldalnyi hagyományos dokumentumot is rögzítenek, így ezek kis helyen elférnek. A filmlapot *mikrofilm-leolvasóval* lehet megnézni.

Ma már gyakoribb, hogy a dokumentumokat inkább **szkennelik**, és a **számítógép monitorán** tekinthetjük meg őket.

A **hang** rögzítését THOMAS ALVA EDISON oldotta meg a *fonográf* feltalálásával (1877). A hangdokumentumok azonban csak a *hanglemez* megjelenésével terjedtek el. A hanglemezt hamarosan felváltotta a *magnószalag*, napjainkban pedig lényegében áttértünk a *CD lemezekre*.

A **hangdokumentumok*** a következő csoportokba szokták sorolni: zenei felvételek, szépirodalmi művek, gyerekeknek szóló összeállítások, ismeretterjesztő művek (pl. nyelvtanuláshoz), helyismereti dokumentumok.

A számítógép segítségével ma már az elektronikusan rögzített szöveges dokumentumok felolvasása is megoldott. Ezért sok könyvtár a gyengén látó embertársaink részére olyan eszközöket is biztosít (pl. beszélő számítógép, képernyőolvasó), amely a szöveges információt is hanggá alakítja.

A **képi dokumentumok*** két fő csoportja az állóképek és a mozgóképek.

Az **állóképek** rögzítésére a *fénykép* a XIX. század első felében több lépésben alakult ki. Ma az állóképeket nagy mennyiségben nyomdai úton állítják elő, ilyenek például a *képeslapok* és a *poszterek*. Az állóképek speciális változata a csak segédeszközzel megtekinthető *diakép*.

A *digitális fényképezőgépek* terjedésével már az állóképek megtekintésében és feldolgozásában is megjelent a számítógép.

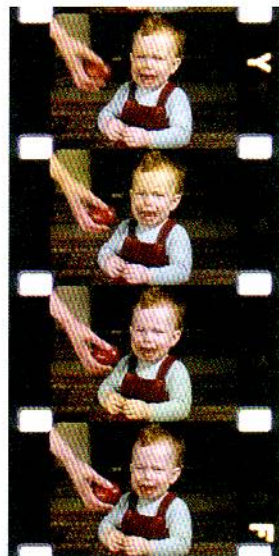
A **mozgóképek** szerepe igen fontos: segítségükkel lehet legjobban bemutatni a természeti folyamatokat, a történelmi eseményeket stb. A *film adatait* annak gyártója a hordozón szokta feltüntetni. A legfontosabb adatok: a film címe, műfaja, a közreműködők neve, a gyártás éve és száma, a gyártó adatai és a technikai paraméterek.

A mozgófilmeket többféleképpen lehet csoportosítani:

- *Célja szerint* lehet reklámfilm, oktatófilm, ismeretterjesztő film, dokumentumfilm, játékfilm stb.
- *A rögzítés módja szerint* lehet filmszalagon, videoszalagon, DVD-n stb.
- *Formai megjelenése szerint* lehet fekete-fehér vagy színes, rövidfilm stb.



Edison fonográfja



Film képkockái

Név- és tárgymutató

Névmutató

Aiken, Howard 20
Arisztotelész 16
Babbage, Charles 19, 175
Berglund, Anders 131
Berners-Lee, Tim 131
Boole, George 16
Bricklin, Dan 145
Codd, Edgar 173
Eckert, J. P. 20
Edison, Thomas Alva 213
Frankston, Bob 145
Gates, William Henry 41
Géza fejedelem 200
Goldfarb, Charles 131
Goldstine, H. G. 20
Gutenberg, Johannes 91, 207
Hess András 92
Hollerith, Herman 12
Jaquet-Droz 19
Kemény János 180
Kurtz, Thomas 180
Lebegyev, Sz. A. 20
Leibniz, Gottfried Wilhelm 144
Lorie, Ray 131
Lovelace, Ada Augusta 175
Mátyás király 200
Mauchly, J. W. 20
Mosher, Ed 131
Neumann János 21, 175
Norton, Peter 53
Pascal, Blaise 144
Shannon, Claude E. 16
Shickard, Wilhelm 144
Szabó Ervin 201
Széchényi Ferenc, gróf 200
Teleki József, gróf 200
Torvalds, Linus Benedict 41
Turing, Alan 20
Wirth, Niklaus 181
Zuse, Konrad 20

A, Á

a: drive 30
ablak* 43
abszolút cellahivatkozás* 151
adat* 9
adatbázis-kezelő* 143, 166, 194
adatfájl* 46
adatforrás 118
adatmodell 173
adattábla* 163, 167
aktív cella* 146
aktiválás 198
aktuális mappa* 48
alaplap* 23
alfanumerikus mód 28
algoritmus* 177
alkalmazói program* 194
alkönyvtár 47
analóg* 9
animáció* 129
antivalencia 17
appendelő vírus* 56
archiválás 31, 51, 55
aritmetikai és logikai egység 24
árnyalat 82
ARPANET 65
árvasor 102
ASCII kódkészlet* 14
asztal* 42
asztali számítógép 22
Átlag függvény* 156, 161
attribútum* 50
áttűnés* 129
automatikusan újraszámítás* 149
azonnali üzenetküldés* 77

B

balra zárt* 100
bájt* 13
Basic programozási nyelv 180
batch 39
baud 33
behúzás* 101
beillesztés* 53

bejáratú térség 202
bekezdés* 96, 100
bemutató* 122
betűméret* 98
betűstílus* 98
betűtípus* 98
bibliográfia 210
billentyűkombináció* 43, 94, 99
billentyűzet* 26
bináris* 10
bit* 10, 12
bittérkép 81
bit/s* 13
BOOT vírus* 56
botkormány 27
böngészőprogram* 68, 132
buborékmódszer 191

C, Cs

CD* 31
CD-írás* 55
cella* 139, 145
ciklus* 187
cilinder 31
címnegyedív 208
ClipArt 108
CMYK-színrendszer* 82
compiler* 181
Cutter-szám* 204
csatolás* 73
csatorna* 11
csevegés* 76
csillagtopológia 64
csomagok 66
csoportos tanulási és foglalkoztatási terület 202

D

*Darabtel*i függvény* 156, 161
dátum- és időkezelés* 158
dátum/idő* (típus) 167
decimális típus* 167
deklaráció* 184
dekódolás* 11
desktop 22
dia 122, 124
diakocka* 122
diagram* 125, 153
diaminta* 127
digitális* 9

digitális fényképezőgép* 27, 85
digitális zoom 85
diszjunkció* 16
disztribúció 41
DNS 66
dokumentum* 207
DOS* 40
dpi 28, 85
DVD* 31
dzsóker karakterek 50

E, É

editor 92
egér* 27, 44
egész típus* 168, 184
egyesített dokumentum 118
egyfelhasználós* 39
egytaskos operációs rendszer* 39
elágazás* 185
elektronikus könyvtár* 207
elektronikus levél* 71
elemi adat* 14
elérési út* 48
előfej* 116, 152
élőláb* 116, 152
első generáció* 34
elválasztás 105
e-mail* 71
e-mail cím 72, 136
emlékeztető 126
érintőképernyő 27
értékkadás 183
eszköztár* 94
ETO* 205
ETO szám szerinti katalógus* 206

F

fájl* 46
fattyúsor 102
fejlesztői környezet* 176, 194
fejlesztői dokumentáció* 178
felbontás* 85
felhasználói név* 7
felhasználói dokumentáció* 178
felsorolás* 110, 140
felsőoktatási könyvtár* 202
feltételes formázás 159
félszabad szoftver 197
fényceruza 27
fényerő* 82, 86

férgek* 56
 fixpontos számábrázolás* 15
 Fkeres függvény* 160, 161
 „Fogd és vidd” 44
 folyóirat* 212
 folyóiratolvasó 203
 fordítóprogram* 176
 formázás 51
 forrás* 11
 forráskódszerkesztő 132
 fórum* 76
 főszöveg 208
 freeware* 196
 FTP* 77, 134
 függvény* 147, 161

G, Gy

game port 33
 gépi kód* 176
 gibibájt* 13
 gigabájt* 13
 gopher* 78
 grafikus alkalmazás 195
 grafikus felület* 37
 grafikus mód 28
 gyökér* 48
 Gyök függvény 161

H

Ha függvény* 159, 161
 hajlékonylemez* 30
 hálós adatmodell 173
 hálózati kártya* 33
 hálózati meghajtó 59
 hálózati program* 193
 hangdokumentum* 213
 hangkártya* 29
 hanyattgér 27
 hasáb 117
 hardver* 22
 harmadik generáció* 34
 háttér 128
 helyi hálózat* 63
 helyi menü* 94
 hetilap* 212
 hierarchikus adatmodell 173
 hisztogram 87
 hivatkozás* 68, 136
 hivatkozási integritás 172
 Hol.van függvény* 160, 161

hordozható számítógép 22
 hozzáférési jogok* 59, 166
 hőnyomtató* 29
 HSB-színrendszer 82
 HTML* 68, 131
 húzás 44

I, Í

időszaki kiadvány* 210
 igazítás* 100
 ikon* 42
 importálás 152, 169
 Index függvény* 160, 161
 indexelés* 168
 indítás 44
 információ* 9
 informatikai biztonság* 58
 iniciálé 107
 intelligens címke 94
 interaktív* 40
 interfész* 32
 internetcím* 66
 interpreter* 177
 intranet* 79
 IP-cím* 66
 írásvédett fájl* 51
 irodalmi művek 209
 ISBN szám* 208
 iskolai könyvtár* 202
 ismeretközlő művek 209
 ISSN szám* 210

J

járulékos részek 208
 jel* 9
 jel/zaj arány* 11
 jelentés* 167, 171
 jobbra zárt* 100

K

kapcsolat* 166, 172, 173
 kapcsolómező* 172
 kapuáramkörök 17
 karakter* 14, 96
 karakterstílus 114
 kaszkádolt törlés 172
 katalógus* 205, 206
 katalóguscédula* 205
 katódsugárcsőves monitor* 28
 két formázása* 109, 125, 135

képhivatkozás* 137
képi dokumentumok* 213
képlet* 147
képlet másolása* 149
képlomás* 53
kereskedelmi program* 196
keresőfüggvények* 159, 160
keretek 140
kettes számrendszer 10
kézikönyvtár 203
kiadványszerkesztő 92, 195
kibibájt* 13
kijelölés 44, 50, 97
kilobájt* 13
kimenetsor 96
kiszegítő lehetőségek 61
kiterjesztés* 46
kitöltés 148
kivágás* 53
koaxiális kábel* 33
kódlapok* 14
kódolás* 11, 14
konfiguráció 24
konjunkció* 16
kontraszt* 86
korrekció* 86, 88
könyv* 208
könyvelői formátum* 158
könyvkiválasztó övezet* 203
könyvtár* 201
könyvtári ábécé 204
könyvtárszerkezet 47
körbefuttatás 109
kördiagram* 154
körlevél* 118
köteget* 39
középre zárt* 100
közlemény* 11
közművelődési könyvtár* 202
központi egység* 23, 24
központi feldolgozóegység* 24
kulcs* 164, 168
kulcsszavas kereső* 70
kvantálás* 10

L

lábjegyzet* 116
lány elválasztójel 103
lapolvasó* 27
lap tulajdonságai* 138

laptop 22
lebegőpontos számábrázolás* 15
legördülő menü* 43
lekérdezés* 167, 170
leütés 96
levelezési lista* 75
levelezőprogram* 72
levélszemét* 74
lézernyomatató* 29
licenc 195, 196
lista* 110
Linux* 40
logikai érték* 15
logikai típus* 167, 184
logikai műveletek 16, 186
lomtár* 57

M

Ma függvény 161
mainframe 22
makró, makróvírus* 56, 178
mappa* 47, 74
mappaszerkezet* 47
margó* 107
második generáció* 34
másolás* 49, 53, 105
másolat* 72
matematikai logika 15, 186
mátrixnyomatató* 28
Max függvény* 156, 161
maximum-kiválasztás tétele* 190
mebibájt* 13
megabájt* 13
meghajtó* 47
mentés 106
menü* 42, 43, 94
merekvelem* 31
mező* 163
mikrofilm 212
Min függvény* 156, 161
miniszámítógép 22
mintavételezés* 9
mintázat* 108
modem* 33
moderált lista 76
monitor* 28, 85
Moore-törvény 35
mozgatás* 50
multitask* 39
munkaablak 94

munkaállomás* 7, 64
 munkafüzet* 146
 munkalap* 146

N

nagy területű hálózat* 63
 napilap* 211
 negáció* 16
 negyedik generáció* 34
 nem nyomtatható karakter* 95
 nem nyomtatott dokumentum* 207, 212
 nem törhető kötőjel 103
 nem törhető szóköz 103
 nemzeti könyvtár* 202
 Neumann-elv* 21, 175
 notebook 22
 nulladik generáció* 34
 nyelvi ellenőrzés 105
 nyilvántartási, ellenőrzési övezet* 203
 nyitólap* 134
 nyomtató* 28, 60
 nyomtatott dokumentum* 207

O, Ö

olvasóterem* 204
 operációs rendszer* 37, 193
 optikai zoom 85
 órajel 24
 oszlop, oszlopazonosító* 145
 oszlopdiaagram* 154
 összegezés tétele* 189
 összetett kulcs* 164
 ötödik generáció* 35

P, Q

párhuzamos port* 32
 Pascal programozási nyelv 181
 PDA 22
 pendrive* 32, 52
 pénznem formátum* 157
 perifériák* 23
 pixelgrafika* 81
 pixelgrafikus programok* 83
 pocketPC 22
 pont* 98
 port 32
 postafiók* 71
 prezentáció* 121
 prezentációkészítő* 92, 123, 194
 programozási nyelv* 176, 194

programozási tételek* 189
 protokoll* 64
 PS/2 port* 33
 public domain 197
 QBE-rács* 170

R

rajzgép* 29
 RAM* 26
 rámutatás 44
 redundancia* 11
 regiszterek 24
 regisztráció 198
 rejtett fájl* 51
 rekord* 163
 relációs adatbázis* 173
 relatív cellahivatkozás* 151
 rendezés* 111, 114, 164, 168
 rendezés egyszerű cserével* 191
 rendszergazda* 7, 59
 rendszerközeli program* 193
 rendszerprogram* 193
 réteg 84
 RGB-színrendszer* 82
 ROM* 25
 router 66
 RTF formátum 106

S

sablon* 115
 saját számformátum 159
 sáv* 30
 segédkönyv* 209
 shareware* 196
 síntopológia 64
 sor, sorazonosító* 145
 sorkizárt* 100
 soros port* 32
 spam 71
 SQL nyelv* 170
 stílus* 114, 135
 switch* 33

Sz

szabad szoftver* 197
 szabadpolcos rendszer* 203
 szakasz* 97, 117
 szakkönyvtár* 202
 szám, számformátum* 147, 157
 számláló* 168

számozott lista* 110, 140
származási bizonylat 196
százalék formátum* 158
szegély* 108
szektor* 30
szemantikai hiba* 177
személyi hálózat* 63
szerkesztőléc 147
szerver* 7, 64, 193
szerzői betűredes katalógus* 206
szerzői jog* 195
szimbólum* 100
színcsatorna 89
szintaktikai hiba* 177
szoftver* 22, 193
szöveg, szöveges típus* 147, 168, 184
szövegdoboz* 110
szövegszerkesztő* 92, 194
Szum függvény* 156, 161
szuperszámitógép 22
szűrés* 165, 169
szűrőfeltétel* 165

T

táblázat* 113, 139
táblázatkezelő* 143, 163, 194
tabulátorpozíció* 112
tájéolás* 107
tálca* 42
talpas betű 98
talp nélküli betű 98, 127
tápegység* 23
tárgy szerinti katalógus* 206
társítás* 47
tartalomjegyzék 115
tartomány* 145, 146
taszk* 39
TCP/IP protokoll* 64
telepítés* 197
telítettség 82
telnet* 78
tematikus kereső* 70
térköz* 101
területi és nyelvi beállítások* 60, 157
tervezősablon 128
testreszabás* 44
tesztelés* 178
TFT-monitor* 28
tintasugaras nyomtató* 29
típus* 167, 184

titkos másolat* 72
topológia* 64
többfelhasználós* 39
tömb* 184
törzsdokumentum 118
trójai faló* 56
tulajdonosi szoftver* 196

U, Ű

UNICODE* 15
USB* 32
UTP-kábel* 33
úrlap* 167, 171

V

valós típus* 168, 184
valós idejű* 40
vágólap* 53, 105
változó* 183
városi hálózat* 63
végjegyzet 116
vegyes cellahivatkozás* 151
vektorgrafika* 81
vektorgrafikus program* 84
veszteséges tömörítés* 54
veszteségmentes tömörítés* 54
vevő* 11
vezérlőegység 24
vírus* 56
vírusirtó program* 57
Vkeres függvény 161
vonaldiagram* 154
vonalzó 101, 112
vörösszem-effektus 89
VPN* 79

W

WAP* 78
webhely* 67, 134
weblap* 68, 132, 138
webszerver* 68, 133
winchester 31, 51
Windows* 40
World Wide Web 67, 131
WYSIWYG* 92, 133

Z

zaj* 11
zárt raktár 204
zip drive 30

Tartalom

(SZ)ÁMÍTÁSTECHNIKA 5

Igaz történetek 5

AZ INFORMATIKA ALAPJAI 7

Az iskolai számítógépek használata 7

Az iskolai hálózat felépítése és 7

használatának szabályai

Érintés- és balesetvédelem 8

Az információ és mérése 9

Információ, adat, jel 9

Az analóg és digitális jel 9

A bináris számrendszer 10

Az információátadás (kommunikáció) 11

folyamata

Az információmennyiség 12

Az információmennyiség mértékegységei 13

Az adatok tárolása 14

A karakterek kódolása 14

A számok ábrázolása 15

Logikai műveletek és kapuk 15

Logikai értékek 15

Logikai műveletek 16

A kapuáramkörök 17

Összefoglaló kérdések, feladatok 18

Az első számítógépek 19

A számítógép felépítése 22

A számítógépek csoportosítása 22

A hardver tagozódása 23

A személyi számítógépek 23

A központi egység 24

A központi feldolgozóegység 24

A központi tár (memória) 25

Beviteli eszközök 26

A monitorok 28

A nyomtatók 28

További kiviteli eszközök 29

Mágneses elven működő háttértárak 30

Optikai elven működő háttértárak 31

A pendrive 32

A számítógépen lévő csatlakozók 32

Hálózati berendezések 33

Számítógép-generációk 34

Összefoglaló kérdések, feladatok 36

AZ OPERÁCIÓS RENDSZER 37

Az operációs rendszer fogalma és néhány jellemzője 37

Az operációs rendszer feladata 37

Az operációs rendszerek csoportosítása 39

A PC-s operációs rendszerek 40

A grafikus felület 41

A grafikus felület elemei 43

Az egér kezelése 44

A grafikus felület testreszabása 44

A fájlrendszer 46

A fájl létrehozása, jellemző adatai 46

Meghajtók és mappák 47

A könyvtárszerkezet kezelése 48

Fájlok és mappák másolása és mozgatása 49

Fájlok és mappák csoportos kiválasztása 50

A fájlok és mappák tulajdonságai 50

A meghajtók kezelése 51

Segédprogramok 52

A segédprogramok jellemzése 52

A vágólap (clipboard) 53

Az adatok tömörítése 54

Adatok archiválása CD-re 55

Vírusok, vírusvédelem 56

Törölt állományok helyreállítása 57

Együttműködés többfelhasználós környezetben 58

Az informatikai biztonság 58

Hálózati meghajtó 59

Hálózati nyomtató csatlakoztatása 60

Egyéni munkakörnyezet kialakítása 60

Összefoglaló kérdések, feladatok 61

KOMMUNIKÁCIÓ HÁLÓZATON 63

A hálózatok áttekintése 63

A számítógép-hálózatok csoportosítása 63

Helyi hálózatok 64

Az internet: a hálózatok hálózata 65

Az internet kialakulása 65

A számítógépek azonosítása az interneten 66

A világháló használata 67

A World Wide Web 67

Tartalom

A világháló működése	68	A bevitel tiltott műveletei	95
A böngészők használata	69	A hibák javítása	96
Keresés a világhálón	70	A szöveg részei	96
Elektronikus levelezés	71	A kijelölés módjai	97
Az elektronikus levél (e-mail)	71	Betűformázás	98
Az elektronikus levelezés működése	71	A betűk formái jellemzői	98
Az elektronikus levelek kezelése	72	A betűtípusok csoportosítása	98
Fájl csatolása	73	A betűformázás lehetőségei	99
A levelek csoportosítása és szűrése	74	Szimbólumok beillesztése	100
Az internet további szolgáltatásainak áttekintése	75	Bekezdésformázás	100
A levelezési lista és a fórum	75	Bekezdésformátumok	100
Csevegés (Chat) és azonnali üzenetküldés (Messenger)	76	A bekezdésformátumok beállítása	101
FTP és gopher	77	Bekezdések laphatáron	102
További internetes szolgáltatások	78	A szöveg helyes bevitele és javítása	102
Összefoglaló kérdések, feladatok	80	Az írásjelek használatának szabályai	102
		Szavak és kifejezések elválasztása	103
		a sorok végén	
		Matematikai, műszaki és pénzügyi jelek beírása	103
		Keresés és csere	104
		Másolás és mozgatás	105
		Nyelvi ellenőrzés	105
		Fájlműveletek: megnyitás, mentés, nyomtatás	106
		Fájlformátumok	106
		Automatikus mentés	106
		Oldalbeállítás és nyomtatás	107
		Grafikus elemek	107
		Iniciálé	107
		Szegély és mintázat	108
		Képek beillesztése	108
		A kép formázása	109
		A szövegdoboz (keret)	110
		Listakezelés	110
		A listajelek formátuma	111
		Rendezés	111
		Az adatok táblázatos elrendezése	112
		Tabulátorok használata	112
		Táblázat készítése	113
		A formázás gyorsítása: stílusok, sablonok	114
		A stílus	114
		A tartalomjegyzék	115
		Sablonok	115
		Nagyobb dokumentumok formázása	116
		Élőfej és élőláb	116
		A lábjegyzet és végjegyzet	116
		Többhasábos rész	117
		Körlevél készítése	118
		Összefoglaló kérdések, feladatok	119

DOKUMENTUMKÉSZÍTÉS 81 SZÁMÍTÓGÉPPEL - GRAFIKA

Grafikus adatok kezelése	81		
Pixel- és vektorgrafika	81		
Színkezelés	82		
Képformátumok	83		
Rajzok, ábrák készítése	83		
Digitális képek kezelése	85		
A digitális képek készítésének és megjelenítésének főbb eszközei	85		
A kép vágása és forgatása	86		
A fényerő és a kontraszt	86		
Színkorrekció	88		
A „vörösszem-effektus”	89		
Képek konvertálása	89		
Összefoglaló kérdések, feladatok	90		

DOKUMENTUMKÉSZÍTÉS 91 SZÁMÍTÓGÉPPEL - SZÖVEGSZERKESZTÉS

A szövegszerkesztő programok jellegzetességei	91		
A szövegszerkesztők előtt	91		
A szövegszerkesztők fogalma	92		
A szövegszerkesztők felülete	93		
A szövegszerkesztők funkcióinak elérése	93		
A szöveg megjelenítése	95		
A szöveg begépelése, részei és kijelölése	95		

DOKUMENTUMKÉSZÍTÉS 121 SZÁMÍTÓGÉPPEL - PREZENTÁCIÓ

<i>A multimédia-prezentáció</i>	121
A prezentáció felhasználásának területei	121
A prezentáció felépítése	122
A bemutató elkészítésének lépései	122
A prezentációkészítő programok	123
<i>A prezentáció létrehozása</i>	123
Szöveg formázása	124
Új dia beszúrása, felsorolás	124
Objektum beszúrása	125
A bemutató rendezése és vetítése	126
A bemutató mentése és nyomtatása	126
<i>A prezentáció látványelemei</i>	127
Egységes megjelenés: diaminta készítése	127
Sablonok használata	128
Egyéni animáció	129
Áttűnés, időzítés	129
<i>Összefoglaló kérdések, feladatok</i>	130

DOKUMENTUMKÉSZÍTÉS 131 SZÁMÍTÓGÉPPEL - WEBSZERKESZTÉS

<i>A webszerkesztés módjai</i>	131
A World Wide Web kialakulása	131
A weblapok elhelyezkedése	132
A webszerkesztő programok csoportosítása	132
<i>Weboldal készítése webszerkesztővel</i>	133
Új (üres) webhely létrehozása	133
A nyitólap	134
Fontos szempontok a formázáshoz	135
Betű- és bekezdésformázás	135
Kép beillesztése és formázása	135
Hivatkozások beillesztése	136
Weblap mentése	137
A webhely megtekintése	137
<i>Weblapok formázása</i>	138
A weblap címe és háttere	138
Táblázat készítése	139
Keretes lapok készítése	140
Listák kezelése	140
Látványelemek: vízszintes vonal, fényűjság, interaktív gomb	141
<i>Összefoglaló kérdések, feladatok</i>	142

TÁBLÁZATKEZELÉS 143

<i>Táblázatkezelés régen és ma</i>	143
A táblázatkezelés alapfunkciói	143
A táblázatkezelők megjelenése	144
<i>A táblázatkezelés alapfogalmai</i>	145
A táblázat részei	145
Munkalap és munkafüzet	146
A cella tartalma	147
A képlet és a függvény kapcsolata	147
<i>A táblázatkezelők alapfunkciói</i>	148
Az adatok bevitelle, törlése, javítása	148
Képlet bevitelle, másolása	149
A táblázat formázása	149
Abszolút és relatív cellahivatkozás	151
Fájlműveletek: mentés, nyomtatás, megnyitás	152
<i>Diagramok készítése</i>	153
Az automatikus diagramkészítés	153
A diagramok fajtái	154
A diagramok utólagos formázása	155
<i>Problémamegoldás táblázatkezelővel</i>	156
Példa statisztikai számításokra	156
Számformátumok beállítása	157
Pénzügyi számítások. Pénz nem és százalék formátum	157
Dátum és időkezelés, egyéni számformátumok	158
Választás két lehetőség között	159
Választás több lehetőség közül	159
A függvények összefoglalása	161
<i>Összefoglaló kérdések, feladatok</i>	161

BEVEZETÉS AZ ADATBÁZIS-KEZELÉSBE 163

<i>Adatkezelés táblázatkezelőkkel</i>	163
Az adattábla	163
Rendezés	164
Adatok szűrése	165
További műveletek	165
<i>Adatkezelés adatbázis-kezelővel</i>	166
Az adatbázis-kezelő rendszerek	166
Az adatbázis-kezelők objektumai	167
Az adattáblák	167
Mezők indexelése	168
Adattáblák készítése	168
Műveletek a táblákkal és az adatokkal	169

Tartalom

Az adatok szűrése	169
A lekérdezések	170
Számítások az adatbázisban	171
Úrlapok és jelentések	171
Kapcsolat a táblák között	172
Adatmodellek	173
Összefoglaló kérdések, feladatok	174

ALGORITMIZÁLÁS 175

Programozási előismeretek	175
Az első számítógépek programozása	175
Programozási nyelvek és fejlesztői környezetek	176
A programfejlesztés lépései	177
Első programunk	179
Az első program algoritmusai	179
Visual Basic szkriptek készítése	180
Turbo Pascal programok készítése	181
A változó fogalma, típusai	182
Példa: négyzet területe	182
A változó fogalma	183
A típus fogalma	184
Elágazás	185
Példa: súlyteszt	185
Az elágazás megvalósítása	185
Összetett feltétel megadása. Logikai műveletek	186
A ciklus	187
Példa: Manhattan szigete	187
A ciklus fajtái	187
Programozási tételek	189
Az összegezés tétele	189
Maximum- és minimum-kiválasztás	190
Rendezés egyszerű cserével (buborékmódszer)	191
Összefoglaló kérdések, feladatok	192

SZOFTVERISMERET 193

A szoftverek csoportosítása funkciójuk szerint	193
A rendszerprogramok	193
A fejlesztői környezetek	194
Az alkalmazói programok	194
Az általános célú szoftverek fajtái	194

A szoftverek csoportosítása jogi szempontból	195
A szoftver tulajdonjoga	195
A tulajdonosi szoftverek	196
A szabad szoftverek	197
További kategóriák	197
A programok telepítése	197
A telepítés menete	197
Aktiválás és regisztráció	198
Összefoglaló kérdések, feladatok	198

KÖNYVTÁRHASZNÁLAT 199

A könyvtárak kialakulása	199
A könyvtár	201
A könyvtár fogalma, típusai	201
A könyvtár funkcionális részei	202
Keresés a könyvtárban	204
Szépirodalmi művek elrendezése	204
Az ismeretterjesztő művek elrendezése	205
A katalógusok	205
A számítógépes katalógus	206
Az elektronikus könyvtár és az elektronikus könyv	207
Dokumentumismeret	207
A dokumentum fogalma, fajtái	207
A könyv	208
Időszaki kiadványok	210
Nem nyomtatott dokumentumok	212
Összefoglaló kérdések, feladatok	214

TÁRGYMUTATÓ 215

TARTALOM 221

Könyvünk az Oktatási Minisztérium által kiadott kerettanterv alapján készült, így megfelel mind a NAT, mind a kerettanterv elvárásainak. Azoknál a témaköröknél, melyek szerepelnek a középszintű érettségi vizsgán is, az ismereteket az érettségi vizsgán elvárt mélységig dolgoztuk fel. Ennek megfelelően mind a gimnáziumok, mind a szakközépiskolák sikeresen használhatják az alapóraszámban tanuló csoportjaik oktatásához.

A könyv elkészítése során az elméleti ismereteket, illetve az egyes szoftverek használatához szükséges háttérismereteket szoftverfüggetlenül fogalmazzuk meg. Az ismeretek alkalmazását a konkrét szoftverekben az illusztrációk, illetve az ábrákhoz tartozó magyarázatok, kérdések, feladatok tartalmazzák.

A könyv feldolgozásához a JO-0162 kiadói kódú Informatikai feladatgyűjtemény c. tankönyvünket ajánljuk.

ISBN 963 87000 2 5



9

789638 700025

www.jos.hu