

A kibicnek semmi sem drága!

Sokaknak ismerős lehet az a csalódottsággal teli, nyomasztó érzés, ami akkor tölti el az embert, amikor ez a találó mondás kívánczik a nyelvére. Az igazságérzet és bizonyítási vágy fűtötte indulatot, mégis gyakran lehűti a kiábrándultság és a kiszolgáltatottság. A mondat talán már el sem hangzik. Minden alkotó ember tudja: érvnek ez bizony elég kevés! A szerződés kötelez, a megrendelő diktál hiszen ő fizet a vásárlónak pedig mindig igaza van!

Mielőtt azonban az önsajnáltagadhatatlanul kényelmes, de zsákutcát jelentő állapotába lovalnánk magunkat, be kell látnunk: alkotni nem kötelező! Ha az embert büszkeséggel tölti el, hogy vizionlátja ötleteit a képernyőn, megjelenik a neve egy CD borítón, esetleg egy újság vagy könyv lapjain, akkor ennek bizony ára van. Jó érzés végigtekinteni az elkészült programon, íráson, grafikán és belegondolni, hogy ez részévé válik sok általunk nem ismert ember munkájának, szórazásának. De botorság lenne azt hinni, hogy ez megtörténhet kompromisszumok nélkül. Sőt gyakran éppen a nehézségek és akadályok kényszerítik ki azokat a bizonyos „nagy ötleteket”! Bár erre mindig csak utólag jövünk rá.

Jól ismerjük ezeket a „legendás” kompromisszumokat a nagy művészekről szóló anekdotákból. Szinte minden híres festőről, szobrászról, íróról, költőről keringenek olyan történetek, amelyek az alkotó elme győzelmét hirdetik a körülmények, az ellenlábások vagy a középzszerűség felett. A híressé vált tudósoknak szinte „védjegyévé” vált ez a képesség, hiszen a szürke hétköznapiakon végzett „szellemi favágás” a kívülálló számára érdektelen, az eredményre pedig nincsen 100%-os garancia.

De hova soroljuk be a számítástechnika szellemi alkotásait? Kézenfekvőnek tűnik, hogy a tudományok családjában keressük a „rokonságot”. Nem is tévedhetünk nagyot, hiszen magának az eszköznek a számítógépnek a létrejötte és fejlődése tudós koponyák egész hadát igényelte és igényli ma is. Ezzel azonban látszólag már el is zárkózunk a többi nagy „családtól”, mint a művészet, a kultúra vagy éppen a filozófia. A mindennapi példák viszont éppen azt mutatják, hogy a számítástechnika legújabb nemzedéke már elutasított „házasodik be” hol jól, hol rosszul ezekben a „családokba”. Nem is olyan régen még heves, érzelmektől sem mentes vitákat váltott ki, ha egy hagyományos művészet neve előtt megjelent a „számítógépes” jelző. (Na nem mintha ez új lenne a nap alatt! Jó száz éve még minden „épeszű” ember számára nyilvánvaló volt, hogy a benzínmotoros autó soha nem veheti fel a versenyt egy megbízható hátsaslóval, a fényképezés pedig, mint művészet nem említhető egy lapon a festészettel!) A történelem ugyan nem azt mutatja, hogy az emberiség túl sokat tanulna a korábbi hibáiból, azért a számítástechnika (ahogy például az elektromosság vagy vegyészett) lassan mégis csak „beilleszkedett”. Ma már a többség nem kételkedik abban, hogy egy számítógéppel készített grafikának is lehet (persze nem automatikusan) művészi értéke és egy zenész sem válik művészből tudóssá csak azért, mert számítógép segítségével komponál.

Belátták ezt a software ipar mértékadó képviselői is, így mára szinte természetes, hogy a művészi kreativitást igénylő részfeladatokat valóban művészek végzik el. (Ezt tapasztalhattuk már korábban az autóiparban is, ahol a „nyers” technikát már jó ideje igyekeznek igényes esztétikai kivitelben megjeleníteni.) Továbbra is fennáll azonban a kérdés? Minek is minősül egy számítógép program elkészítése? A közvélekedés szerint leginkább szellemi szakmunka, amely alapvetően a (matematikai) logikából építkezik. El kell ismerni, hogy ez az

esetek elsőprő többségében igaz is. Néha amikor valóban valami új születik a programozás tudománnyá válhat. Ne feledjük: a ma mindennapos, megszokott technika születésekor bizony tudományos vívmányként látta meg a napvilágot! De vajon rejlik-e művészet a számítógép programozásban? És itt most nem az informatikába „ágyazott” művészetéről van szó! Szó sincs design-ról, welcome screen-ekről, intro-król vagy aláfestő zenéről. A kérdés az, hogy van-e a (számítás)technikának esztétikája? Lehet szép egy algoritmus? Létezik csúnya és szép megoldás egy problémára? Lehet egy kevesebbre képes program szebb fejlettebb riválisánál? Nevezhetjük-e Peter Norton legendás Commander-ét például szépnek függetlenül a külső megjelenéstől? Mi lehet az oka, hogy oly sokan használták sőt használják még évekkel az után is, hogy technikai értelemben elszállt felette az idő? Talán ugyanazért, amiért még mindig gurulnak az utakon Volkswagen „bogarok” és Citroën „kacsák” és a gőzmozdony vontatta vonatnak is még van utasa. Persze lehet, hogy mindez csak nosztalgia, de vajon mitől függ, hogy mi iránt érzünk nosztalgiát? Emlékei között kutatva biztosan minden számítógép kedvelő talál néhány olyan berendezést, programot vagy trükköt, ami maradandó nyomot hagyott a gondolataiban.

Mi lehet mégis az oka annak, hogy a számítástechnika művelőit még mindig külön „kasztként” kezeli az átlagember? Miért hisszük el azt a Hollywood által agyonkoptatott képet, ami szerint a számítástechnikus egy sötét, bűzös lukban ücsörgő, szemüveges különc zseni? Talán azért mert zseninek látszani nem is olyan rossz dolog. Nem is kell hozzá sokat tenni, hogy ez a látszat megmaradjon. Néhány elhadart szakkifejezés, egy-két „csípőből” megoldott (ál)probléma és persze a megfelelő pillanatban alkalmazott rejtélyes hallgatás. A gond csak az, hogy ezzel a képpel semmivé foszlik a háttérben húzódó valódi teljesítmény! A sok megoldott probléma, a kitaró szorgalmas munka, a rengeteg tanulás és már-már megszállott alaposság. Elhatározás kérdése személyek és cégek esetén is hogy felépítjük-e a tévedhetetlenség mítoszát. Aki viszont megteszi, ne csodálkozzon a kárörvendő kibicnek hadán, amikor a mítosz mégis „megrogyan”! Senki sem szeret kérkedni a hibáival, tévedéseivel. Érthető, ha valaki nem veri nagydobra a gondokat, csak „csendben” tanul belőle. A problémák létezésének reflexszerű tagadása viszont nem vall nagy bölcsességre. A laikusokat talán meg lehet téveszteni ezzel, ám a kicsit jártasabb embernek ez mindig is „sántítani fog”. Különösen igaz ez a számítástechnika egy olyan szerteágazó területére, mint a multimédia. Aki már kicsit is belekóstolt alkotóként ebbe a műfajba, hajlamos úgy érezni, hogy itt nem csak a „média” „multi” tehát sokszoros hanem a gond, a probléma, a buktató no meg persze a lehetőség is! Valóban lenyűgöző távlatokat nyit meg a technika fejlődése! Nem tévedünk sokat, ha azt érezzük: csak rajtunk áll merre és meddig jutunk el ezzel a lehetőséggel! A gondok és nehézségek rejtegetése itt viszont már valóban reménytelen. Nem is gondolhatjuk komolyan, hogy óramű pontossággal készülhet el egy olyan mű, amelyben igényes grafikai arculat jeleníti meg a szakmailag, nyelvileg magas szintű tartalmat, mindezt a legújabb technika által megteremtett legjobb minőségű hang és mozgókép élménnyel „dúsítva” és persze olcsón, gyorsan, egyszerűen! A követelmények hacsak el nem hagyunk belőle néhányat már önmagukban hordozzák a kompromisszum kényszerét, nem is beszélve arról, hogy mindez csak csapatmunkában elképzelhető, ami szintén nehéz bár sokszor építő jellegű vitákkal jár. De talán mindez nem olyan nagy gond! Talán a problémák, tévedések, nehézségek is vannak olyan érdekesek, izgalmasak ha nem izgalmasabbak mint

maga a megoldás! Ezt persze csak az tudja eldönteni, aki mindkettővel szembesül. Ezért határozta el e cikk szerzője, hogy egy szellemi kalandtúrára invitálja meg a kedves olvasót! Hónapról-hónapra fogjuk végigkövetni egy multimédia CD-ROM „életútját” itt a Codex hasábjain, az ötlet megszületésétől a reménykedve várt megjelenésig. És, hogy mi a garancia a kalandra? Nem „vegytiszta” tanpéldáról van szó. E sorok írásakor a CD-ROM még közel sincsen készen. Hátra van még számos gond és akadály, amelyek gerincét fogják képezni a tervezett cikksorozatnak. Bizonyára meglátogatunk utunk során

PHP kezdőknek I.

Tudom a cím nem túl eredeti, de ez tükrözi talán legjobban ennek a cikksorozatnak a lényegét, és ha valaki most akar megismerkedni a PHP rejtelveivel, akkor úgyis az ilyen című cikkekre veti rá magát. Ebben a sorozatban tehát a PHP-val fogunk megismerkedni.

Ha manapság valaki elkezd foglalkozni a weblapfejlesztéssel, annak szinte kötelező, hogy ismerje a PHP script nyelvet. Ennek oka, hogy ez a script nyelv legalább olyan egyszerű, mint a Java script, de annál sokkal biztonságosabb, és hatékonyabb. A php csak a kiszolgálón futtatható script nyelv, ami azt jelenti, hogy a script a szerveren fut le, és nem pedig a felhasználó számítógépén. Ez azt is jelenti, hogy a felhasználó nem látja a scriptet, ebből fakad a biztonság. Ilyet a Java script is tud, csak sokkal kevesebb funkcióval, és bonyolultabban. Emiatt hatékonyabb a PHP.

Nos akkor nem is húzom tovább az időt mindenféle összehasonlítgatással, hiszen ez nem is igazán lehetséges, és nem is annyira érdekes. Inkább vágjunk rögtön a közepébe, vagy ha nem is pont a közepébe, de legalábbis a lényeges részébe a dolgoknak. Vagyis, hogy hogyan is épül fel a PHP script.

Gondolom, aki már olvasott PHP-vel foglalkozó könyvet, az lehet, hogy felhördül: miért nem a PHP fordító telepítésével kezdem. Nos a válaszom erre az, hogy amikor elkezdtem foglalkozni a PHP-val akkor engem meglepett az, hogy minden könyvben az van leírva először, hogy hogyan telepítsük ezt a fordítót, szerintem ezzel egy kezdőt nem kellene traktálni, hiszen manapság annyi ingyenes és nem ingyenes szolgáltatót találunk, aki nyújt PHP szolgáltatást is (sőt már egyre többen elkezdtek a MySQL szolgáltatást is nyújtani, az ingyenes szolgáltatók közül is). És egy kezdőnek éppen megfelel egy ilyen szolgáltatónál regisztrálva próbálgatni a szárnyait, mintsem az otthoni gépére kelljen felrakni különböző webszerver programokat, és azokkal vesződni, illetve azok beállításával.

Szóval a PHP scriptet legáltalánosabban a HTML forrásban a következőképpen szoktuk beilleszteni:

```
<?php
...
?>
```

Én is ezt a formát fogom a továbbiakban használni, van más jelzőmód is, de ezek bemutatásával most nem foglalom a helyet, majd egy későbbi cikkben leírom azokat is. Mielőtt még mélyebbre ásnánk magunkat magában a script készítésben, ejtenék pár szót a kiterjesztésről, vagyis hogy milyen kiterjesztésű lehet egy ilyen scriptet tartalmazó HTML oldal. Ezt a kiszolgáló szabja meg, illetve ha valaki magának állít be szerveret, akkor az egy config fájlban állíthatja be, hogy mely kiterjesztéseknél fusson a php fordító. Ennek pontos beállítását a php readme-je tartalmazza. Általában a kiterjesztés *.php illetve egyes szolgáltatók szokták még használni a *.php3 illetve

néhány zsákutcát, teszünk néhány vargabetűt. Talán éppen ettől lesz majd örömteli a most még csak homályosan derengő végcél elérése! A CD alkotóinak kis csapata aggódó, de bizakodó szülőkként követik az első lépéseket. Az ezzel járó felfedezés öröme szeretenék ezúton megosztani a Codex olvasóival. Így erre a hónapra nem is maradt más mondandó. Kalandra fel!

Szűcs Zoltán szucs.zoltan@intermail.hu

a *.html kiterjesztéseket is.

A script lényegében bárhova elhelyezhető a HTML forrásban. Ennyi bevezető után készítsünk egy nagyon egyszerű kis php kódot, amit a <body> </body> közé illesztve máris kipróbálhatunk. A példa egyszerű: írassuk ki a weblapra a php segítségével, hogy Hello Világ! :

```
<?php
    echo('Hello Világ!');
?>
```

Ebből a pár sorból is kitűnik, hogy a php sok mindenben hasonlít más programozási nyelvekre. Azaz a sorvégi ; illetve a zárójelek használata, vagy a szöveget (string) jelölő ' jel, ami helyett lehet használni az " jelet is, de ezt azért nem nagyon ajánlatos használni, mivel ha forrásrészletet szeretnénk beszúrni a php-val, akkor összekeveredhetnek a dolgok. Ha már úgyis itt tartok, akkor legyen a következő példa amelyben egy HTML forrásrészletet szúrunk be a php segítségével a lapra.

```
<?php
    echo('<br><font color="FFFFFF"><b>Hello
Világ!</b></font>');
?>
```

Így most a Hello Világ! szöveg új sorba fehéren, és félkövéren fog kerülni. Ezzel bemutattam az echo parancs lényegét, persze lehet vele változókat is kiíratni, de ez majd később jön. Most előbb bemutatom, hogyan lehet még beilleszteni a forrásba egy másik fájl tartalmát:

```
<?php
    include('szoveg.txt');
?>
```

Az include al is lehet olyan fájl beilleszteni, amiben van html forrás, sőt még php kódot is tartalmazhat. Ezzel nagyban meg lehet gyorsítani a weblapok készítését, hiszen ha egy szépen felépített weblapban csak a szöveg változik, akkor nem kell az egészet újra minden szöveghez megépíteni. Persze - mondhatják azok, akik már eléggé gyakorlottak a HTML készítésben de erre valók a frame-ek. Én viszont úgy gondolom, hogy a frame-ek kezelése bonyolultabb, hosszabb, és persze kevésbé elegánsabb, mint a php, de aki ennek ellenére a frame-eket szereti inkább használni, annak attól még nem kell figyelmen kívül hagynia az include eljárást, mivel ezt sok egyéb másra is lehet használni, ahol a frame-eket nem. Ebben a cikkbe nagyjából ennyi fért, tudom hogy eléggé kevés, és az ember nem szereti az elején abbahagyni, viszont most következne a változók témaköre, ami túl hosszú, és félbehagyni nem igazán jó. Ezért a következő rész nagyjából csak a változókkal fog foglalkozni, abban már sokkal több példa lesz ismertetve.

Csubák Péter Chuby@chello.hu
<http://www.csu-ga.hu/chubyprograms>