**Farkas Károly:**

**Jegyzetpótló a Webprogramozás alapjai tantárgyhoz műszaki menedzserek számára**

**Óbudai Egyetem**

**2011. október**

**BEVEZETÉS**

Mi az a HTML?

A weboldalak szerkesztésük során nem többek, mint egyszerű szöveges fájlok, amelyek ***.html*** vagy ***.htm*** kiterjesztést kapnak. (Természetesen a végeredmény messze túlmutat egy szövegszerkesztővel létrehozott dokumentumnál!)
A szerkesztéshez használhatjuk a Windows Jegyzettömbjét (Notepad), de manapság már vannak HTML kódszerkesztők is, pl.:**EditPlus**, Notepad++, Context, **Komodo Edit**, stb. amelyek nagyon sok mindenben megkönnyítik a szerkesztést.

A HTML dokumentum tehát egy standard szövegfájl, amely speciális formázó utasításokat tartalmaz. Megjelenítésére leggyakrabban az Internet Explorert vagy a Mozilla Firefoxot használjuk, de a standard HTML kódot sok más internetes böngésző is értelmezni tudja, például Google Crome, Opera, Konqueror, Safari, Netscape.

Ez a jegyzet azért készült elsősorban, hogy bepillanthassunk a honlapkészítés mikéntjébe. Áttekintjük, hogyan és milyen utasítások kellenek ahhoz, hogy saját magunk erejéből készítsük el honlapunkat.

**Kellemes időtöltést és jó munkát kíván
a szerző**

**SZERKEZETI FELÉPÍTÉS**

Ebben a jegyzetben az EditPlus HTML-szerkesztőt használjuk.

Lássunk neki az első lépéseknek! Hamarosan tapasztalhatják, hogy néhány alapvető utasítás ismeretében képesek elkészíteni saját internetes oldalukat.

1. lépés

Az EditPlus indítása után a következő ablakkal találkozhatunk, ha új honlapot akarunk készíteni:



Elemezzük ki a kódunk egyes részeit!

**<HTML>
</HTML>**

Ez az elem jelzi mindenféle böngésző számára, hogy a dokumentum HTML nyelven íródott. A HTML szerkezeti elemeit általában **< >**jelek közé tesszük, és általában párosával használjuk azokat. Van egy nyitóelem, és az estek többségében meg kell adni hozzá a záróelemet is, amelyet a **/** jellel különböztetünk meg a nyitóelemtől. Az oldalunk teljes leírása e két elem közé kerül.

Minden HTML dokumentumban van egy ***fejléc*** (<HEAD>).

**<HTML>
<HEAD>
</HEAD>
</HTML>**

A fejlécen belül érdemes megadni a ***dokumentum címét*** (<TITLE>), amely ugyanakkor a böngészők ablakcímében is megjelenik. Éppen ezért érdemes rövid (64 karakternél rövidebbet), de találó címet keresni. A <TITLE> szerkezeti elem tagjai között csak szöveg szerepelhet.

Ha szeretnénk a megszokott környezetben, magyar ékezetes betűkkel dolgozni, mellőzve az azokat helyettesítő, kicsit nehézkesen használható különböző karaktersorozatokat, a fejlécbe be kell iktatnunk egy újabb sort, amelyben jelezzük, hogy a közép-európai karakterkészlettel dolgozunk.

**<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">**

A ***charset=iso-8859-2*** jelzi, hogy közép-európai karakterkészletet alkalmazunk. További karakterkészleteknek szakkönyvekben lehet utánanézni.

A HTML dokumentum legnagyobb egysége a ***szövegtörzs*** (<BODY>). Itt adhatjuk meg a dokumentum tartalmára vonatkozó összes információt - a szöveget, képeket, kapcsolódó linkeket, hivatkozásokat, stb.

A <HEAD>...</HEAD> és a <BODY>...</BODY> elemek csak a következő sorrendben ágyazhatók be a <HTML> elemek közé:



A későbbiekben ismertetésre kerülő utasítások, HTML elemek is egymásba ágyazhatók. Ennek szabálya a következő: mindig azt az utasítást zárjuk be legelőször, amit legutoljára nyitottunk meg.



Ahhoz, hogy megnézzük, hogyan is néz ki valójában első munkánk, nem kell internetes kapcsolattal rendelkeznünk. Elegendő, ha gépünkön rajta van az Internet Explorer, vagy valamilyen más böngésző. Ha az elmentett file-on kétszer kattintunk, máris látható az eredmény. Ha a böngészőnk mégis kérné az internetes kapcsolatot, akkor válasszuk a ***Kapcsolat nélküli munka*** menüpontot a File menüből.

A továbbiakban a szövegtörzsön belüli adatbevitellel foglalkozunk, kezdve a karakterek formázásával.

**KARAKTERFORMÁZÁS**

Mindenféle beállítás nélkül elkezdhetjük a szöveg bevitelét, mert mindegyik böngészőnek van alapértelmezett karakterkészlete, amit meg tud jeleníteni.
Ha ettől szeretnénk eltérni, akkor már szükségünk van a következő utasítás használatára:

**<FONT> ... </FONT>**

Ennek az utasításnak a használatával meg tudjuk változtatni a dokumentumban megjeleníteni kívánt szövegrészek karaktereinek típusát (FACE), színét (COLOR), méretét (SIZE), illetve a betűk stílusát is. Minden alkalommal, amikor a </FONT> elemmel lezárjuk a szerkesztett részt, mindig az alapállapot aktualizálódik.

A FACE paraméterrel nemcsak egy, hanem egyszerre több karakterkészletet is felsorolhatunk, és ezek közül a böngésző azt alkalmazza, amelyiket a felsorolásban előbb ér el. Ezt a módszert akkor érdemes alkalmazni, amikor nem vagyunk biztosak abban, hogy a böngésző rendelkezik-e az adott készlettel. Ha egyiket sem találja, akkor a böngészőben beállított típust jeleníti meg.

<FONT FACE="Arial"> ... </FONT>
<FONT FACE="Arial, Courier, Times New Roman CE"> ... </FONT>

Érdekesebbé, látványosabbá tehetjük oldalainkat, ha színes betűket használunk. A COLOR utasítás segítségével tehetjük meg ezt. A[színek megadhatók RGB-kódjaikkal](http://nemessuli.hu/informatika/html/color216.gif), vagy a nevükkel.
Példánkban a sárga színt állítottuk be, először a nevére hivatkozva, majd RGB-kódjával.

<FONT COLOR="yellow"> ... </FONT>
<FONT COLOR="#FFFF00> ... </FONT>

Bármelyik utasítást is használjuk, az eredmény ilyen lesz.

Megváltoztathatjuk a betűk méretét is. A lehetséges értékek 1-7 vehetők fel, alapértelmezés szerint a betűméret 3, ami a 12-es betűnagyságnak felel meg. A határértékeken nem lehet túllépni, mert a szélsőértéket veszi fel. Relatív értékkel is hivatkozhatunk a betűméretre, ha az aktuálistól való pozitív vagy negatív eltérést adjuk meg.

Lássunk néhány példát!

|  |
| --- |
| **Beállítás nélkül ez az alapértelmezett, 3-as betűméret.****<FONT SIZE=7> 7-es betűméret</font><FONT SIZE="-1"> Eggyel csökkentettük a betűméretet.</font><FONT SIZE=+2> 2-vel növeltük a betűméretet.</font><FONT SIZE=3> Normál betűméret</font>** |

Ha más stílusú szöveget akarunk megadni, akkor a betűk méretével, aláhúzással, döntéssel, vastagítással, illetve a színekkel játszhatunk.

A szöveg stílusának változtatásához elegendő, ha szöveg előtt és mögött a következő utasításokat adjuk meg:

|  |
| --- |
| <b>**Kövér (bold) betűk**</b><i>*Dőlt (italic) betűk*</i><u> Aláhúzott (underlined)betűk </u> |

Természetesen ezek a parancsok kombinálhatók is.

|  |
| --- |
| <b><u>**Kövér, aláhúzott betűk**</u></b><i><u>*Dőlt, aláhúzott betűk*</u></i><b><i>***Kövér, dőlt betűk***</i></b> |

A szavakon belül is megváltoztathatjuk a [méretet](http://nemessuli.hu/informatika/html/piramis.gif):

p i r a m i s

Tovább kísérletezhetünk a [színekkel](http://nemessuli.hu/informatika/html/szines.gif) is:

|  |
| --- |
| **SZ Í N E S** |

További lehetőséget jelenthet, ha például az írógép típusú (<TT>) karaktereket használjuk:

|  |
| --- |
| <TT>Most írógéppel írt szöveget hoztunk létre.</TT>Most írógéppel írt szöveget hoztunk létre. |

**A SZÖVEG TAGOLÁSA**

A szöveget általában valamilyen formában tagoljuk. Ehhez használhatunk címeket, alcímeket, bekezdéseket, stb.

Html szerkesztésben az alcímek (**<Hn>**) használatakor 6 szintet valósíthatunk meg (n=1-6), ahol az 1-es számú a legmagasabb, a 6-os számú a legalacsonyabb szintet jelzi. Az alcímek mindig tartalmaznak egy soremelést is.

|  |  |  |
| --- | --- | --- |
| <H1> | **Íme a legnagyobb méretű alcím.** | </H1> |
| <H3> | **Ez egy köztes méretű alcím.** | </H3> |
| <H6> | **Ez pedig a legkisebb méretű alcím.** | </H6> |

Az alcímeket igazíthatjuk is. Az illesztéshez az **ALIGN** paramétert alkalmazzuk.
A felvehető érték lehet: **LEFT** (balra), **RIGHT** (jobbra) és **CENTER** (középre).
További megkülönböztető jelzéseket is alkalmazhatunk a szöveg hierarchiájának jelzésére - megdönthetjük, aláhúzhatjuk vagy éppen középre is igazíthatjuk az egyes alcímeket.

Lássunk néhány példát!

|  |  |  |
| --- | --- | --- |
| **Nyitóelem** | **Alcím** | **Záróelem** |
| <u><H1> | **Aláhúzott alcím** | </H1></u> |
| <H2 align="left"> | **Balra zárt alcím.** | </H2> |
| <H3 align="right"><i> | ***Jobbra igazított,dőlt betűs alcím.*** | </i></H3> |
| <H4 align="center"> | **Középre igazított alcím.** | </H4> |

A szöveg további tagolására használhatunk bekezdéseket ( **<P>**) is.
A bekezdés szerkezeti elem önmagában is alkalmazható, nem feltétlenül kell alkalmazni a zárótagját </P>. Az utána következő szövegrész nagyobb térközzel tagolódik el az előzőtől, és új sorban kezdődik.

Az igazítást az alcímeknél ismertetett ALIGN paraméter segítségével végezhetjük el: jobbra, balra, középre, kiegészítve a sorkizárással(JUSTIFY).

|  |
| --- |
| **Ötletek a weblapok készítéséhez**Sok különböző vélemény létezik azzal kapcsolatban, hogy milyen egy jó weblap kinézete és felépítése. Sok ötletet, példát és stílusmintát találhatunk a World Wide Weben. Néhány ötlet itt is olvasható, ezek a weblapok többségén jól használhatók.A tartalomnak jól szervezettnek kell lennie. A jól felépített weblap segít elképzeléseink hatékony tolmácsolásában, és segít a felhasználónak navigálni a webhelyen.A weblap szövegének könnyen olvashatónak kell lennie.Ha hátteret adunk a weblaphoz, akkor a háttér és a szöveg színének különböznie kell,a szövegnek pedig kontrasztosan kell megjelennie a lapon.A weblapok a különböző webböngészőkben különbözőképpen jelenhetnek meg. Ajánlatos a weblapokat úgy tervezni, hogy a legtöbb böngésző helyesen jelenítse meg azokat. |

A fenti példában a következő utasításokat használtuk a szövegrészek elrendezéséhez.

|  |
| --- |
| **<u><H2 align="center"> Ötletek ... <H2></u>** |
| **<p align="justify"> Sok különböző ...** |
| **<p align="left"> A tartalomnak...** |
| **<p align="center"> A weblap szövegének...** |
| **<p align="right"> A weblapok...** |

Ha a bekezdéstől függetlenül szeretnénk új sorban folytatni a szöveget - sor kihagyása nélkül -, akkor a <BR> utasítást kell alkalmaznunk.

|  |
| --- |
| Zelk Zoltán: Gida<p>Láttatok már kisgidát?**<br>**Ugye, milyen kedves,**<br>**olyat ugrik, szinte száll,**<br>**szárnytalan is repdes. |

Természetesen arra is van lehetőség, hogy letiltsuk a sortörést egy szövegrészben. Használjuk ilyenkor a <NOBR> ... </NOBR>utasítást!

Ha a soron belül szeretnénk egy szóköznél nagyobb helyet kihagyni, akkor nem elegendő a megfelelő számban leütni a SPACE billentyűt, mert az eredmény csak egy karakternyi hely lesz.
A következő kód alkalmas arra, hogy egy újabb karakternyi üres helyet illesszünk be a sorba:

&nbsp;

Most 5 karakternyi helyet hagyunk ki:     és itt folytatódik a sor.

A fenti példa megvalósítása így néz ki:

... ki:&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;és itt ...

**SZÍNEK HASZNÁLATA**

Színekről már ejtettünk szót a [karakterformázás](http://nemessuli.hu/informatika/html/betuk.html)nál, de természetesen máshol is alkalmazhatjuk azokat.

Nagyon gyorsan és egyszerűen megváltoztathatjuk egy oldal arculatát, ha megváltoztatjuk annak háttérszínét (<BGCOLOR>). Ezt az utasítást a böngészők szinte mindegyike képes kezelni. A háttérszínt a szövegtörzs elején kell definiálni, a <BODY> szerkezeti elem paramétereként:

<BODY BGCOLOR="#CCCCCC">

|  |
| --- |
|  |

A színek meghatározása a már említett módon történik, vagy az [RGB-kód](http://nemessuli.hu/informatika/html/color216.gif)jával, vagy a nevével.

Minden olyan utasításnál (pl.: TABLE, HR, FRAME, stb.), amelynek paraméterei között szerepel valamilyen színbeállítási lehetőség, ugyanígy kell eljárni.

A színes hátteret tartalmazó weblapok olvasásának megkönnyítéséhez a színeket állítsuk kontrasztosra.

|  |
| --- |
| **Például használjunk világos szöveget sötét háttéren,** |
| **vagy fordítva.** |

A következőkben a hyperlinkek adta lehetőségekről olvashatnak.

**HYPERLINKEK**

Nagyon hasznos dolog a hyperlinkek által kialakítható kapcsolatrendszer a HTML dokumentumokban. Segítségével összeköthetjük a dokumentum egyes részeit, más dokumentumokat érhetünk el egyetlen kattintással, kinagyíthatunk egy-egy képet, vagy esetleg e-mailt küldhetünk ismerősünknek, stb. Mindezt anélkül tehetjük, hogy keresgélni kellene a gépünkön, vagy az interneten a megfelelő program után.

Ezt a megoldást a következő utasítással valósíthatjuk meg:

**<A> ... </A>**

A hyperlinkes kapcsolatot jelölő szöveg aláhúzással jelzi a képernyőn ezt a lehetőséget, míg az ugyanilyen kapcsolatot jelölő kép keretezett. (Ettől az alapértelmezéstől természetesen el is lehet térni.)

Az utasításnak több paramétere is van. Mi ezek közül a **HREF**-fel, és a **NAME**-mel ismerkedünk meg. A HREF a hypertext URL kapcsolódási pontját, helyét ada meg, míg a NAME egy dokumentumon belüli cimke, vagy könyvjelző.
Ez utóbbit akkor érdemes alkalmazni, ha hosszabb dokumentumról van szó, és a szövegen belül lehetővé szeretnénk tenni a tallózást.

Lássuk, hogyan is néz ez ki a gyakorlatban! Erről a pontról szeretnénk a fejezet elejére [visszaugrani](http://nemessuli.hu/informatika/html/link.html%22%20%5Cl%20%22eleje).

A megvalósítása így néz ki:

**<A NAME="eleje">**
<center> <font color="teal" size="5" face="Times New Roman CE"> <b> HYPERLINKEK </b> </font> </center>
......
Erről a pontról szeretnénk a fejezet elejére **<A HREF="#eleje"> visszaugrani </A>.**

Megoldásként alkalmazhatunk képet is, valahogy így:



Utasításunkat így módosítottuk:

<p align="center"> **<a href="#eleje"><img src="fel.gif"></a>**

Ha zavaró a keret a kép körül, akkor egészítsük ki az utasítást így:

<p align="center"> **<a href="#eleje"><img src="fel.gif" border=0></a>**

A border=0 utasítás jelzi, hogy nem akarunk keretet a kép köré, és az eredmény így fog kinézni:



Jöjjön a következő alkalmazás!
Weboldalunkról szeretnénk [másik html dokumentumot megnyitni](http://nemessuli.hu/informatika/html/masik.html).

A megvalósítása a következőképpen történt:

Weboldalunkról szeretnénk **<A HREF="masik.html">** másik html dokumentumot megnyitni **</A>.**

Ezt a megoldást akkor tudjuk alkalmazni, ha a két dokumentum ugyanabban a könyvtárban található. Különben a DOS-ban elsajátított módszert alkalmazva kell a helyes elérési útvonalat megadni.

Ha egy másik weboldalt szeretnénk megnyitni, például a [Hotmail szolgáltatásait](http://www.hotmail.com/), akkor ez a megoldás:

**<A HREF="http://www.hotmail.com">** Hotmail szolgáltatásait**</a>** ...

A HREF levélküldésre is alkalmas:

<A HREF=**"mailto:anonym@valahol.hu"**>Írj nekem!</A>

Az anonym@valahol.hu az az e-mailcím, ahová elküldhetjük levelünket. Formailag így néz ki:

Írj nekem!

Képről képre is hivatkozhatunk:

**<A HREF="nagykep.gif"><IMG SRC="kiskep.gif"></A>**

És most kattintson a képre!



A következőkben a képek beillesztéséről, illusztrációról lesz szó.

|  |  |
| --- | --- |
| **KÉPEK,****ILLUSZTRÁCIÓ** | http://nemessuli.hu/informatika/html/html1.gif |

Ha igazán látványossá szeretnénk tenni az oldalt, akkor használjunk képeket!

Olyan képet válasszunk, amelynek a kiterjesztése ***.gif*** vagy ***.jpg***. Erre azért van szükség, mert az ilyen típusú képek foglalják a legkisebb helyet, az adatátvitelnél ugyanis fontos szempont a méret.

Az **<IMG>** szerkezeti elem teszi lehetővé, hogy képet szúrjunk be a szövegbe. Zárótag nélkül használjuk.

A fenti két képet a következőképpen jelenítettük meg (az elhelyezkedésüket most nem taglaljuk):

**<img src="html1.gif" border=0>
<img src="html2.gif" border=0>**

A border paraméter a kép körüli keret vastagságát állítja be, jelen esetben a 0 a keret hiányát jelzi.

Nézzük még néhány példát!

A keret így változtatja meg a kép egészét - a keretek vastagsága: 0, 2, 4, 8.

      

Célszerű megadni a képek méretét is, mert gyorsabb lesz a letöltés, ha a gép ismeri a pontos méretet. A **WIDTH**-tel a szélességet, a **HEIGHT**-tel a magasságot lehet megadni képpontokban. Ha az eredeti képtől eltérő méreteket akarunk megjeleníteni, akkor is ezt a két utasítást kell használni.

Lássunk néhány példát!

Ez a kép eredeti méretben jelent meg:

**<img src="tanulj.gif" width=310 height=255>**



Most módosítjuk a szélességét:

**<img src="tanulj.gif" *width=150* height=255>**



Most módosítjuk a magasságát:

**<img src="tanulj.gif" width=310 *height=105*>**



Most méretarányosan változtatjuk mindkét adatot:

**<img src="tanulj.gif" *width=150 height=123*>**



Egy képet, egy logot választhatunk háttérnek is.

[Lássunk egy példát](http://nemessuli.hu/informatika/html/hatter.html), hogyan is néz ki, ha a következő ábra töltené ki a weboldalunk hátterét:



**<body background="puzzle.jpg">**

Ebben a formában a háttér együtt mozog a szöveggel, az oldalon lévő adatokkal.

Ezzel szemben, ha a következővel egészítjük ki az utasítást, akkor rögzíthetjük a hátteret, és csak a [szöveg fog mozogni](http://nemessuli.hu/informatika/html/hatter2.html).

<body background="puzzle.jpg" **bgproperties="fixed">**

Ezt a lehetőséget csak a MS Internet Explorer támogatja, a Netscape-nél a háttér továbbra is együtt mozog a szöveggel.

Ha hátteret használunk, akkor óvatosan kell kezelni azt, nehogy elnyomja a lényeget, a mondanivalót.

Következik a táblázatok használata.

**LISTÁK**

**Rendezett lista**, angolul "ordered list". Innen ered az elem neve is: **<OL>**. A számozott lista sorait a **<LI>** elem után, külön sorokba kell írnunk:

**<OL>**

**<LI>**Előétel
**<LI>**Főétel
**<LI>**Desszert

**</OL>**

A számozás módját a **TYPE** paraméterrel adhatjuk meg. Lehetséges értékei:

1 - számok, 1-től kezdve
A - nagybetűk, A-tól kezdve
a - kisbetűk, a-tól kezdve
I - nagy római számok, I-től kezdve
i - kis római számok, i-től kezdve

A listákat egymásba is lehet ágyazni. Ekkor érdemes az egyes szintekhez különböző típusokat rendelni:

|  |  |
| --- | --- |
| **Példa** 1. **Előétel**
	1. **Sajttál**
	2. **Gyümölcstál**
	3. **Hideg töltött uborka**
2. **Főétel**
3. **Desszert**
 | **Forráskód**http://nemessuli.hu/informatika/html/kod.png |

**Rendezetlen lista**, angolul "unordered list", **<UL>**. Felépítése hasonló a rendezett listához, annyiban különbözik attól, hogy az egymásba ágyazott szintek automatikusan más jelölést kapnak:

|  |  |
| --- | --- |
| **Példa** * **Opel**
	+ **Meriva**
	+ **Astra**
	+ **Corsa**
		- **3 ajtós**
		- **5 ajtós**
* **Audi**
* **BMW**
 | **Forráskód**http://nemessuli.hu/informatika/html/kod2.png |

**Definíciós lista**, angolul "definition list". Jelölése: **<DL>**.
A listán belül egy "szakasz" mindig két részből áll: a meghatározásból ("definition term", azaz **<DT>**) és a leírásból ("definition description", azaz **<DD>**).
Ezt a típust akkor használjuk, ha kifejezéseket szeretnénk megmagyarázni, vagy egy-egy rövidebb szöveghez hosszabb megjegyzést kívánunk fűzni. Szerkezete a következő:

|  |
| --- |
| **Példa****HTML****HyperText Markup Language, egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki****CSS****Cascading Style Sheets, stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le** |
| **Forráskód**http://nemessuli.hu/informatika/html/kod3.png |

**TÁBLÁZATOK**

A táblázatok segítségünkre vannak abban, hogy egyfajta egységet teremtsünk internetes oldalainkon. Létrehozásuknál a következő utasítást alkalmazzuk:

**<TABLE> ... </TABLE>**

A táblázatnak vannak sorai: **<TR>**... **</TR>**, és a sorokon belül, a cellákban szerepelnek az adatok **<TD>** ... **</TD>**.

Hogyan hozhatunk létre egy 2x3-as táblázatot?

|  |
| --- |
| **<TABLE>  <TR>    <TD>I/a</TD>    <TD>I/b</TD>    <TD>I/c</TD>  </TR>  <TR>    <TD>II/a</TD>    <TD>II/b</TD>    <TD>II/c</TD>  </TR></TABLE>** |

És íme az eredmény:

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

Készíthetünk keretet táblázatunknak, ha kiegészítjük az 1. sor így:

**<TABLE border=2>**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

Amint látható a keret a lehető legszorosabban veszi körbe az adatokat. Ezen is változtathatunk, ha megadjuk az adatok és a keret közötti távolság méretét:

**<TABLE border=2 cellpadding=10>**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

Beállíthatjuk a cellák közötti távolságot is:

**<TABLE border=2 cellpadding=10 cellspacing=10>**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

Tegyük láthatóbbá a keretet egy kis színnel (a Netscape nem támogatja):

**<TABLE border=2 cellpadding=10 cellspacing=10 bordercolor="green">**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

A cella tartalma alaphelyzetben balra záródik. Ha középre vagy jobbra akarjuk igazítani, akkor az **ALIGN** paramétert kell alkalmaznunk:

|  |  |  |
| --- | --- | --- |
| **\*\*\*\*\*\*\*\*\*\*** | **\*\*\*\*\*\*\*\*\*\*** | **\*\*\*\*\*\*\*\*\*\*** |
| **bal** | **közép** | **jobb** |

A fenti példát a következő utasításokkal hozhatjuk létre:

|  |
| --- |
| **<TABLE border=1>  <TR>    <TD><b>\*\*\*\*\*\*\*\*\*\*</b></TD>    <TD><b>\*\*\*\*\*\*\*\*\*\*</b></TD>    <TD><b>\*\*\*\*\*\*\*\*\*\*</b></TD>  </TR>  <TR>    <TD><b>bal</b></TD>    <TD ALIGN=center><b>közép</b></TD>    <TD ALIGN=right><b>jobb</b></TD>  </TR></TABLE>** |

Ha hosszabb szöveget viszünk be az egyik cellába, nemcsak vízszintesen, hanem függőlegesen is rendezhetjük a szöveget. Ekkor a**VALIGN** paramétert kell használni, amely alapértelmezés szerint: közép. Az igazítást állíthatjuk felülre (**TOP**) és alulra (**BOTTOM**).

|  |  |  |  |
| --- | --- | --- | --- |
| **1. sor2. sor3. sor4. sor** | **felül** | **alaphelyzet** | **alul** |

Utasításokkal:

|  |
| --- |
| **<TABLE border=1>  <TR>    <TD><b>1. sor<br>2. sor<br>3. sor<br>4. sor</b></TD>    <TD VALIGN=TOP><b>felül</b></TD>    <TD><b>alaphelyzet</b></TD>    <TD VALIGN=BOTTOM><b>alul</b></TD>  </TR></TABLE>** |

Vannak alkalmak, amikor a cellákat célszerű összevonni. Ezt megtehetjük vízszintesen és függőlegesen is. Utasításként a **COLSPAN**, illetve a **ROWSPAN** szolgál erre.

Lássunk példát a cellák vízszintes egyesítésére:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
| **két cella egyben** | **adat** |
| **egy** | **kettő** | **három** |

 |

|  |  |  |
| --- | --- | --- |
| **egy** | **kettő** | **három** |
| **adat** | **két cella egyben** |

 |

Az első táblázai utasításai:

|  |
| --- |
| **<TABLE border=1 cellpadding=10>  <TR>    <TD COLSPAN=2> két cella egyben</TD>    <TD>adat</TD>  </TR>  <TR>    <TD>egy</TD>    <TD>kettő</TD>    <TD>három</TD>  </TR></TABLE>** |

Példa a cellák függőleges egyesítésére:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

|  |  |
| --- | --- |
| **háromcellaegyben** | **egy** |
| **kettő** |
| **három** |

 |

|  |  |
| --- | --- |
| **egy** | **háromcellaegyben** |
| **kettő** |
| **három** |

 |

Az első táblázai utasításai:

|  |
| --- |
| **<TABLE border=1 cellpadding=10>  <TR>    <TD ROWSPAN=3> három<br>cella<br>egyben</TD>    <TD>egy</TD>  </TR>  <TR>    <TD>kettő</TD>  </TR>  <TR>    <TD>három</TD>  </TR></TABLE>** |

A táblázat egészét, valamint a cellákat méretezni is tudjuk a **WIDTH** paraméter segítségével. Az adatokat megadhatjuk százalékban vagy képpontokban is. Méretezzük át a fejezet elején található táblázatunkat!

|  |
| --- |
| **<TABLE border=2 WIDTH=50%>  <TR>    <TD WIDTH=20%>I/a</TD>    <TD WIDTH=30%>I/b</TD>    <TD WIDTH=50%>I/c</TD>  </TR>  <TR>    <TD>II/a</TD>    <TD>II/b</TD>    <TD>II/c</TD>  </TR></TABLE>** |

Amikor a TABLE utasítás után alkalmazzuk a WIDTH paramétert %-os formában, akkor a táblázat szélességét az adott képernyő méreteihez viszonyítja a megjelenítő program.
Ha a TD utasításhoz fűzzük ezt a paramétert, akkor a már adott táblázat méretéhez viszonyítja a cellaszélességet százalékosan.

És íme az eredmény:

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

Lássuk hogyan változik az eredmény, ha a cellák szélességét képpontokban adjuk meg:

|  |
| --- |
| **<TABLE border=2 >  <TR>    <TD WIDTH=10>I/a</TD>    <TD WIDTH=20>I/b</TD>    <TD WIDTH=30>I/c</TD>  </TR>  <TR>    <TD>II/a</TD>    <TD>II/b</TD>    <TD>II/c</TD>  </TR></TABLE>** |

Ha kisebb méretet adnánk meg, mint ahogy az adat elférne, akkor az adat hosszával egyező lesz a cella szélessége, lsd. az 1. oszlopot!
A táblázat felett látható vonal 10 képpontnyi hosszú.

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |
| **II/a** | **II/b** | **II/c** |

A táblázatot, illetve az egyes cellákat kiszínezhetjük a **BGCOLOR** segítségével. Ha a TABLE utasítás után használjuk, akkor az egész táblázatot kiszínezhetjük, ha a TD utasítás után, akkor csak az adott cellát színesíthetjük.

Az egész táblázat beszínezése tehát így néz ki:

**<TABLE border=2 BGCOLOR=green>**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |

A cellák más-más színűre festése így oldható meg:

**<TD WIDTH=20 bgcolor=red>I/a</TD>
<TD WIDTH=20 bgcolor=white>I/b</TD>
<TD WIDTH=20 bgcolor=green>I/c</TD>**

|  |  |  |
| --- | --- | --- |
| **I/a** | **I/b** | **I/c** |

**Az alapfokú HTML tanfolyam befejeződött.
Sok sikert a kezdeti próbálkozásokhoz.**



A további vázlatosan közölt ismeretek Nagy Gusztáv Webprogramozás tankönyve alapján készültek:

Amennyiben többször használunk formázó utasításokat, azokat célszerű a html kódban kiemelve írni. Ez a kiemelés a CSS része a html nyelvnek.

**„Mi a CSS?**

A CSS a *Cascading Style Sheets* rövidítése

A stílusok a HTML megjelenítési elemei és attribútumai helyett használhatók, azoknál jóval több lehetőséget biztosítva

A stílusok meghatározzák, hogy hogyan jelenjenek meg vizuálisan a HTML elemek

A stíluslapok segítségével könnyen szét lehet választani az oldal tartalmát annak kinézetétől (a dizájntól)

A stílusokat általában külön állományban tároljuk (.*css* kiterjesztéssel)

Külső stíluslapokkal gyorsítani tudjuk a munkavégzést

Több stílus is hatással lehet egy elem megjelenésére

a *World Wide Web Consortium*40 (W3C), egy non-profit, szabványokat alkotó

szervezet oldotta meg. A HTML 4.0-ás verziójával és a vele párhuzamosan fejlesztett

CSS segítségével létrejött egy jól használható eszközpáros a webfejlesztők részére.

Mára minden jelentősebb böngésző támogatja a CSS-t, bár a támogatottság mértékében

vannak eltérések.

**Lépcsős elrendezés**

Melyik stílus fog érvényesülni, ha több stílust is definiálunk ugyanahhoz a HTML elemhez?

A következő négy beállítás érvényesül egyre nagyobb prioritással (tehát ütközés esetén a

későbbi felülírja az előzőt).

1. a böngésző alapbeállítása

2. külső stíluslap

3. *head* elemben definiált stílus

4. soron belüli stílus

Tehát a soron belüli stílus a legmagasabb prioritású, tehát felülír minden alacsonyabb

szintű formázást.

A nyelvtan három elemet különböztet meg: kiválasztó, tulajdonság és érték:

kiválasztó {tulajdonság: érték}

A kiválasztó legegyszerűbb esetben egy HTML tag, a tulajdonság azt határozza meg,

hogy milyen jellemzőt akarunk módosítani, míg az érték a változást határozza meg. A tulajdonságot és az értéket egy kettősponttal kell egymástól elválasztani, és a kettőt együtt kapcsos zárójelbe tenni.

body {color: black}

Ha az érték több szóból áll, idézőjelbe kell tenni:

p {font-family: "sans serif"}

„

A CSS tehát a html kód része, vagy külön fájlba tett, külön mentett része.

Az eddigi eszközeinkkel statikus honlapokat készíthetünk. Ha mozgóképet szeretnénk alkalmazhatjuk a Flash programot.

Még hasznosabb lehet a változtatható, változó tartalmú honlap, ehhez kell a Java Script kódrész.

**„**

Mi a JavaScript?

Szkriptnyelv (tehát értelmezett, a böngésző értelmezi)

Interaktivitást (működést) ad a HTML oldalhoz

HTML oldalba építhető, vagy attól jobban elszeparálható kód

Események kezelésére alkalmazható

A neve ellenére nincs szoros kapcsolatba a Java nyelvvel

C++ és Java szintaxisra alapoz

A szabvány leírását az ECMAScript specifikáció tartalmazza

**Kód beillesztése a HTML oldalba**

A HTML *script* tagja használható arra, hogy az oldalhoz JavaScript kódot kapcsoljunk. A

*script* tag tartalma közvetlenül nem jelenik meg a böngésző ablakában, hanem a böngésző

fogja azt futtatni. Például a következő kód a HTML aktuális helyére a *Hello Világ!*

szöveget jeleníti meg:

<body>

<script type="text/javascript">

document.write("Hello Világ!")

</script>

</body>

**Megjegyzés**: Ha megnézzük a böngészőben az oldalunk forrását, akkor látható, hogy ott az eredeti, letöltött

verzió látszik, és nem a JavaScript által „módosított”.

Jól látszik a példán, hogy JavaScriptben elfelejthetjük a sorokat lezáró pontosvesszőket.

A *document.write* módszerét a mai korszerű megközelítésben nem illik alkalmazni.

Nagyon egyszerű kódok kivételével nem szokás48 a JavaScript kódot a *body* tagba tenni.

Összetettebb működést inkább önálló függvények formájában szoktunk elhelyezni a

*head* tagba, vagy még inkább külső JavaScript állományba. Nézzük meg ezek módjait:

A könnyebb áttekinthetőség érdekében a demonstrációs példákban, vagy éppen a fejlesztés közben ezt a szabályt időnként áthágjuk.

JavaScript

<head>

<script type="text/javascript">

function message() {

alert("This alert box was called with the onload event")

}

</script>

</head>

<body onload="message()">

...

</body>

A példán látszik, hogy a *body* tag betöltődésekor (*onload* esemény) fog lefutni a *message* nevű függvény, ami egy figyelmeztető dialógusablakot dob fel. Ha a *message* függvénykódját egy *message.js* állományban helyeztük volna, akkor a head részbe a következő kód lenne szükséges:

<script type="text/javascript" src="message.js"></script>

**Változók**

A változók olyan tárolók, amelyekbe adatokat helyezhetünk. A változó értékét meg tudjuk változtatni a szkript futása során. A változóra a nevével tudunk hivatkozni, és az értékét bármikor lekérdezni vagy módosítani.

A változó nevek (mint ahogy az egész JavaScript nyelv) kis-nagybetű érzékenyek, és betűvel vagy aláhúzás karakterrel kezdődnek.

**Változó deklaráció**

A *var* kulcsszó segítségével a következő módon tudunk változót létrehozni:

var strname = "Hege"

Létre lehet hozni változót *var* nélkül is:

strname = "Hege"

Ahogy a példákban is látszik, egyből értéket is adtunk a változóknak.

„

Az idézetek Nagy Gusztáv jegyzetéből valók

Amennyiben a honlapra írt adatokat tárolni, feldolgozni akarjuk, dinamikus honlapot készítünk a szerverünkön szükséges lesz a php program.

Ennek alkalmazása talán már nem is a webprogramozás alapjait jelenti, hanem haladó szintjét.

A képek kezelésére sokféle képszerkesztő áll rendelkezésünkre. Legismertebb talán a PhotoShop.

A html kódot sok magasabbszintű programnyelv automatikusan elkészíti.

A html nyelv jelenlegi legfejlettebb változata a html 5 verzió

A html programnyelv ismerete egyrészt mellőzhető, hiszen terjednek a tartalomkezelő rendszerek, másrészt a html kód ismerete, használata azért is célszerű, mert ezzel tudunk készíteni hipertext-eket, amelyek a kommunikálásnak újszerű módját teszik lehetővé. Gondoljunk például a tanulásra, amely a hipertext segítségével még inkább a tanulóhoz igazítható tanulási folyamatot tesz lehetővé.

A tartalomkezelő rendszerek használata pedig immáron valóban programozói ismeretek nélkül (másféle informatikai ismeretekkel ) teszi lehetővé a honlapok alkalmazását, a web2 aktív, indukáló használatát.

A honlapok olvasása, a számítógép kezelése gyorsolvasó képességeket kíván és fejleszt. A gyorsolvasásról bőségesen találunk irodalmat és gyakorlatokat az interneten. (Ma már miről nem?)

A gyorsolvasást rég óta kutatom. Néhány tézisemet itt is ismertetem.

Képernyőről egyre többet olvasunk, és ugyanakkor összességében egyre kevesebbet olvasunk, helyette filmet nézünk. Még inkább jellemző ez a fiatalságra. (Sokan, ezt gondolják a nagyobb problémának, de csak azért, mert magunkkal szemben elnézőek vagyunk.)

Képernyőről történő olvasás lassúbb vagy gyorsabb?

Az idősebbek jobban szeretnek még mindig papírról, könyvből olvasni, sőt úgy gondolják, a képernyőről nem is lehet gyorsan olvasni. Igaz volna ez?

**2. Definíciók:**

Először határozzuk meg, mit tekintsünk olvasásnak.

Asimov ezt írta a „Hold tragédiája” című művében:

* „Ott van a baj, hogy félrevezetjük magunkat az olvasás definíciójával. Szinte mindenkit meg tudunk tanítani (ha elég energikusak és kitartóak vagyunk) arra, hogy elolvassa az utcaneveket, megértse az útbaigazításokat és a figyelmeztetéseket a falragaszokon, kiböngéssze a főcímeket az újságban. Feltéve, hogy a nyomtatott közlés rövid, és viszonylag egyszerű, az olvasás indítéka pedig erős, majdnem mindenki tud olvasni.”

A PISA vizsgálat 2001-ben ezt a meghatározást ajánlotta:

* „Írott szövegek - és jelképek, ábrák, képek, képsorok - megértésének, használatának és értékelésének képessége, annak érdekében, hogy az egyén képes legyen céljai elérésére, tudásának fejlesztésére, és a társadalmi életben való részvételre.”

A definíciót 2006-ban így módosították:

* *„írott szövegek megértése, felhasználása és az ezekre való reflektálás annak érdekében, hogy az egyén elérje céljait, fejlessze tudását és képességeit, és hatékonyan részt vegyen a mindennapi életben.”*

Asimov szerint az olvasás, képesség arra, hogy „átrágják magukat ezer egymást követő szón.”

Néha az a gondolatom, hogy napjaink írásbelisége gyakran már az ezer szó terjedelmet meg nem haladó szöveg felfogását tételezi csak fel. (Termékismertetők, help modulok, weboldalak, blogbejegyzések, email-ek, sms-ek, de néha már tanfolyami „jegyzetek” is.) Az igazi olvasás nagyobb terjedelmű dokumentum hatékony feldolgozásának képessége.

A gyorsolvasás az olvasásnak hatékony, mesteri formája. Magam a gyorsolvasást így határoztam meg 2004-ben:

* **az információ felvétel hatékonyabb,**
* **elsősorban jobb megértést eredményező,**
* **a figyelem-összpontosítás mértékét és formáját, valamint az olvasás sebességét, az olvasmányhoz, és az olvasás céljához igazító,**
* **a hagyományosnál rendszerint gyorsabb,**
* **különféle olvasási technikák tudatos kombinációja,**
* **általában többszöri olvasás.**

**3. Mennyire tudunk olvasni?**

A rengeteg vizsgálat közül most csak néhány nemzetközi összehasonlító adatot idézünk:

* IEA 1970/71 „A magyar gyerekek nem tudnak olvasni.”
* OECD 1998. Magyarország 18/22 Tizennyolcadik hely huszonkét nemzet között.
Hazai átlag nem éri el a hármast.
* PISA 2000. Magyarország 23/32 (Huszonharmadik hely, harminckét vizsgált ország.)
A listán minden tényező tekintetében az utolsó csoportba tartoztunk. 5% ötös, 23% egyes!
* PISA 2003. Magyarország 25/40
* PISA 2006. Magyarország 27/57 Ezen belül az OECD országok csoportjában helyezésünk: 22/30 OECD

Az olvasás sebessége csak egy – és nem is feltétlen a legfontosabb – jellemző. Erre a hazai mérési adatokat – amelyekből néhányat sorolok – új újabbal egészíthetem, ki, a közelmúltban a kecskeméti GAMF főiskolán mért értékkel:

* 275 szó/perc Takács Menyhért, 1959, Magyar Könyvszemle
* 90-250 Dezső Zsigmondné, 1967, Gyorsolvasás, KGMTI
* 200 Halász László, 1969, Élet és Tudomány
* 232 Makrai J., 1971, Bevezetés a gyorsolvasásba, ÉGSZI
* 150-180 Katona Zoltán, 1974, A gyorsolvasás, BME MTKI
* 150-250 Szabó Ferenc, 1993, Gyorsolvasás– Hatékony olvasás
* 150-170 Farkas Károly, 1981-1994 (benne GAMF is)
* 206 Farkas – Borsos, 2005, GAMF hallgatók, kb. 20 fő
* 192 Farkas – Versegi, 2005, GAMF Szakkollégium, 20 fő
* 163 Farkas – Gurkáné - Kiss – Lakatosné - Pap-Szigeti, 2006, GAMF hallgatók, 269 fő

(A nevemmel jelzett mérések, azonos teszttel készültek.)

Az olvasás sebessége természetesen erősen függvénye az olvasmánynak. A GAMF-on végzett felmérésben többféle tesztszöveget használtunk. Az a1 jelű az általam évtizedek óta használt egy oldalas (253 szó) levél. Az első ábra a mért adatok egy részlete:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | g | i | m |
| Mérések száma | 269 | 223 | 71 | 28 | 204 | 20 |
| Középérték | 163 | 157 | 187 | 143 | 191 | 175 |
| Szórás | 66,8 | 82,5 | 72,6 | 41,7 | 120,5 | 58,6 |

1. ábra. Olvasási sebességek

Az a1, a2, a3 általános, egyre hosszabb szöveg,

 a **g** gazdasági, az **i** informatikai, az **m** menedzser jellegű szakszöveg.

 (Statisztikai feldolgozás: Pap-Szigeti Róbert)

Az olvasási sebességnél fontosabb mérőszám a megértés foka. A GAMF mérés eredményei közül ezen érték eloszlását mutatja a második ábra.



2. ábra. Megértés foka

4. Papírról vs képernyőről történő olvasás

A kecskeméti felmérés egyik célja volt vizsgálni, hogy mennyire tudnak a hallgatók képernyőről olvasni. Az eredmény sok kolléga számára is meglepő volt. Az első táblázat adatait összevetve láthatjuk, hogy képernyőről való olvasás nem nehezebb a jövő nemzedék számára, sőt – bár nem szignifikánsan, de – eredményesebb volt az olvasás képernyőről.

|  |  |  |
| --- | --- | --- |
|  | Általános szöveg | Szakmai szöveg |
|  | papírról | képernyőről | papírról | Képernyőről |
| Megértés átlagos mértéke % | 57 | 60 | 79 | 81 |
|  | 25 | 28 | 14 | 12 |

1. táblázat. Olvasás papírról vs. képernyőről

Papírról 193, képernyőről 73 fő olvasott. A résztvevők a vizsgálatba önként jelentkezett hallgatók voltak.

**5. Szórakoztató gyakorlatok az olvasás pszichológiájának vizsgálatához:**

A következő két ábra a gyorsolvasás előnyeiről, hátrányairól enged következtetéseket levonni.

* Agyunk kénytelen egyenként egymáshoz kapcsolni az egyes
       

 szavakat, ami nehezebb és fárasztóbb munka számára, mintha
        
* összefüggő szócsoportokból
 --------------
 értelmes gondolategységekből
 ---------------
egyszerre ragadná meg
 ---------------
 ezek teljes értelmét.
 -------



Szórakoztató, olvasási készséget is fejlesztő gyakorlatok találhatók a hivatkozásban feltüntetett webhelyeken is.

**6. Következtetések**

Hazánkban az olvasási képesség átlaga és megoszlása nem éri el a kívánt szintet. (Megengedhetetlenül sok fiatal nem képes olvasni. Másodfajú analfabétizmus meghaladja a harminc százalékot.)

Méréseim jelzésszerűen mutatják, hogy a közelmúltban, felsőfokú oktatásban résztvevők olvasási képessége szignifikánsan nem változott.

Főiskolás hallgatók a képernyőről jobban olvasnak, mint papírról.

A számítógép és kiemelten az internet használata, gyorsolvasó képességeket kíván és fejleszt. (Ennek minden előnyével és hátrányával.)

**7. Irodalom:**

Farkas Károly: Gyorsolvasás. Hatékony olvasás. Gazdaságos informálódás a nyomtatott és elektronikusan megjelenített dokumentumokból. APC Kiadó, Budapest, 2004

http://mwlogo.fw.hu

http://gyorsolvasas.fw.hu