

## HTML/CSS könyv

Írta: TFeri.hu



## HTML-CSS könyv és referencia

## Előszó az új verzióhoz

Még 2004-ben, amikor jelen könyvem eredetijét befejeztem, nem gondoltam volna, hogy ennyire sikeres lett. Sokan és sokszor dicsérték. Köszönet érte! Legtöbben azt írják, hogy egy kifejezetten érthető, jól megszerkesztett és összefogott munka. Nem mellékes, hogy sokan referenciaként alkalmazzák, például az érettségire készüléshoz is. Most, hogy újraírom a könyvet, remélem, hogy legalább ilyen sikeres lesz, mint a régi verzió.

## Jogvédelem

**Jelen mű teljes vagy részleges másolata, annak bárminemű (elektronikus, írásbeli vagy nyomtatott) terjesztése a szerző előzetes írásbeli engedélye nélkül tilos!  
Minden további vita elkerülése végett kérem, keressen meg a <http://tferi.hu/kapcsolat> oldalon leírtak alapján!**

Nos, ennyi bevezető után jöjjön a lényeg: a könyv - kibővített tartalommal!

(c) TFeri.hu, 2012.

## Tartalom

1. fejezet	<a href="#">A HTML logikája</a>
2. fejezet	<a href="#">Betűk formázása</a>
3. fejezet	<a href="#">Fejezetek és formázásaik</a>
4. fejezet	<a href="#">Kapcsolatok</a>
5. fejezet	<a href="#">Listák</a>
6. fejezet	<a href="#">Ékezetek</a>
7. fejezet	<a href="#">Táblázatok</a>
8. fejezet	<a href="#">Képek és videók</a>
9. fejezet	<a href="#">Keretek (frame-ek)</a>
10. fejezet	<a href="#">Egyéb formázások</a>
11. fejezet	<a href="#">Body és Meta</a>
12. fejezet	<a href="#">Térképek</a>
13. fejezet	<a href="#">Kérdőívek</a>
14. fejezet	<a href="#">Egyéb elemek</a>
Melléklet	<a href="#">HTML-színkiosztás</a>
Melléklet	<a href="#">HTML-karakterek</a>
Melléklet	<a href="#">HTML nyelvi kódok</a>

## A tanfolyamon feldolgozott HTML elemek rövid áttekintése

**<!--:** Megjegyzés (Description)  
**<!DOCTYPE:** HTML-dokumentum típusának leírása (DOCument TYPE)  
**A:** Kapcsolat - élő link (Active link)  
**ACRONYM:** Acronim beszúrása (ACRONYM)  
**ADDRESS:** Postacím - kiírásmód (postal ADDRESS)  
**APPLET:** Futtatható elem helye (APPLET)  
**AREA:** Kliens-oldali térkép helyének meghatározása (AREA)  
**B:** Félkövér betű (Bold)  
**BANNER:** Címszalag - nem scrollozható (Banner)  
**BASE:** A dokumentum báziscímének meghatározása (BASE)  
**BASEFONT:** Alapértelmezett betűkészlet (BASic FONT)  
**BGSOUND:** Háttérhang (BackGround SOUND)  
**BIG:** Nagyobb betűméret (BIGger font size)  
**BLINK:** Villogó betűk (BLINK)  
**BLOCKQUOTE:** Szövegblokk (BLOCK QUOTE)  
**BODY:** Az egész dokumentum törzsrésze - NEM táblázat (BODY)  
**BR:** Sortörés (BReak)  
**BUTTON:** Nyomógomb (BUTTON)  
**CAPTION:** Táblázat címe (CAPTION)  
**CENTER:** Középre helyezés (CENTER)  
**CITE:** Idézet kiírása (CITE)  
**CODE:** Programkód kiírásmód (CODE)  
**COL:** Oszlop egy táblázatban (COLoumn)  
**COLGROUP:** Oszlopcsoport (COLoumn's GROUP)  
**COMMENT:** Megjegyzés (COMMENT)  
**DD:** Definíció leírása (Definiton Description)  
**DEL:** Törölt szöveg (DELETED text)  
**DFN:** Definíált kiírásmód (DeFINition)  
**DIR:** Felsorolás-szerű lista készítése (DIRectory)  
**DIV:** Szöveg tagolása (DIV)  
**DL:** Definíciós lista (DL)  
**DT:** Definíciós leírás (Definition Term)  
**EM:** Kiemelt írásmód (EMphasize Text)  
**EMBED:** Dokumentum beszúrása egy másikba (EMBED)  
**FIELDSET:** Mezőelem beszúrása (FIELD SET)  
**FONT:** Betű tulajdonságainak szabályozása (FONT)  
**FORM:** Kérdőívek létrehozása (FORM)  
**FRAME:** Keret beállítása (FRAME)  
**FRAMESET:** Keretezés indítása (FRAMESET)  
**H1:** Fejezetcím - a legnagyobb értékű (Heading 1)  
**H2:** Fejezetcím - 2. legnagyobb (Heading 2)

**H3:** Fejezetcím - 3. legnagyobb (Heading 3)  
**H4:** Fejezetcím - 4. legnagyobb (Heading 4)  
**H5:** Fejezetcím - 5. legnagyobb (Heading 5)  
**H6:** Fejezetcím - legkisebb (Heading 6)  
**HEAD:** Az egész dokumentum fejléce - NEM táblázat (HEADer)  
**HR:** Vízszintes vonal (Horizontal Rule)  
**HTML:** Az egész dokumentum kerete (HyperText Markup Language)  
**I:** Döntött betű (Italic)  
**IFRAME:** Lebegő keretek definiálása (In-line floating FRAMES)  
**IMG:** Kép vagy videó beszúrása (IMaGe)  
**INPUT:** Beviteli mező űrlapokon (INPUT field)  
**INS:** Hozzáadott szöveg beszúrása (INSert text)  
**ISINDEX:** Jelzi a keresőknek, hogy a lap indexelhető (ISINDEX)  
**KBD:** Billentyűzetről bevitt adat (KeyBoarD)  
**LABEL:** Címke definiálása (LABEL)  
**LEGEND:** Megadja a **FIELDSET** elemét (LEGEND)  
**LI:** Lista elem (List Item)  
**LINK:** Kapcsolat más dokumentummal (LINK)  
**LISTING:** Előre listázott szöveg (LISTING text)  
**MAP:** Térkép grafikus elemekhez (MAP)  
**MARQUEE:** Szöveg mozgatása (scrolling MARQUEE)  
**MENU:** Listák menü-szerű felsorolása (MENU)  
**META:** Nem megjelenő, de igen fontos elem (META tag)  
**NOBR:** Sortörés letiltása (NO BReaking)  
**NOFRAMES:** Keretek tiltásakor megjelenő adat (NON-FRAMES version)  
**NOSCRIPT:** Scriptek megjelenítésére alkalmatlan böngészők számára írt rész (NON-SCRIPTing version)  
**OBJECT:** Objektum beszúrása (OBJECT insert)  
**OL:** Rendezett/sorszámozott lista (Ordered List)  
**OPTION:** Kiválasztásos lista eleme kérdőívknél (OPTION)  
**P:** Paragrafus/bekezdés (Paragraph)  
**PARAM:** Változók/paraméterek meghatározása más elemekhez (PARAMeters)  
**PLAINTEXT:** Formázatlan szöveg (PLAIN TEXT)  
**PRE:** Előre formázott szöveg (PREformatted)  
**Q:** Kiemelt szöveg (Quality text)  
**S:** Keresztülhúzott szöveg (Strikethrough)  
**SAMP:** Irodalmi szöveg (SAMPle text)  
**SCRIPT:** Más nyelven írt kódrészlet (script) beszúrása (SCRIPT)  
**SELECT:** Kérdésre való kiválasztási lehetőség kérdőívknél (SELECT)  
**SMALL:** Kiseb betűméret (SMALLer font size)  
**SPAN:** Érvényességi kör (SPAN)  
**STRIKE:** Keresztülhúzott szöveg (STRIKEthrough text)  
**STRONG:** Vastagított szöveg (STRONGer text)  
**STYLE:** Stílus definiálása egy lapon belül (STYLE) - **Csak a folytatásban szerepel!**  
**SUB:** Alsó index kiírás (SUBscript)  
**SUP:** Felső index kiírás (SUPerscript)  
**TABLE:** Táblázat (TABLE)  
**TBODY:** Táblázat törzse (Table BODY)  
**TD:** Táblázat adata (Table Data)  
**TEXTAREA:** Hosszabb szöveg beviteli mező kérdőívekben (TEXTAREA)  
**TFooter:** Táblázat lábléce (Table FOOT)  
**TH:** Táblázat fejléce (Table Head)  
**THEAD:** Táblázat fejléce (Table HEAD)  
**TITLE:** Dokumentum címe - NEM táblázat (TITLE)  
**TR:** Táblázat sora (Table Row)  
**TT:** Írógépszerű kiírás (TeleType)  
**U:** Aláhúzott betű (Underlined)  
**UL:** Nem rendezett lista (Unordered List)  
**VAR:** Változónév (VARiable)  
**WBR:** Sortörés engedélyezése letiltott területen. (WBR)  
**XMP:** Mintaszöveg (eXaMPle)

## 1. fejezet: A HTML logikája

A HTML egy angol mozaikszó. Eredetiben: HyperText Markup Language (=Hipertext jelölő nyelv).

Eredeti verzióját a World Wide Web Consortium **Címük: w3.org** adta ki és a hivatalos szabványt ma is ez a szervezet írja tovább - a gyakorlati technikák alapján. Ezen könyv a HTML 4.01-es szabványát tárgyalja, mely a jelenlegi egyik legerjedtebb a mai interneten. Pontos URL-je:<http://www.w3.org/TR/html401/>.

A HTML nyelv alapja, mint oly' sok minden informatikai apróságnak, a hétköznapi angol nyelv kifejezései. Maga a nyelv "**TAG**"-ekből áll. (kiejtése: "teg"!)

Minta:

```
<EzEgyTag> Ez a közbülső szöveg, amire vonatkozik. </EzEgyTag>
```

Ezek a bizonyos TAG-ek rendkívül sokfélék lehetnek. Szerencsére a nyelv alapjai igen egyszerűek. Egy HTML-lap megírás után egyből megtekinthető bármelyik Internetes böngészőben - akár kapcsolat nélkül is. Weblap szerkesztésére a legegyszerűbb program egy Jegyzettömb. Fontos, hogy a kész file-nak HTM vagy HTML kiterjesztést adjunk és a névben legyenek megfelelően kényesek. Ha grafikus környezethez szoktunk, akkor általában hosszabb és többtagú file-nevekben szoktunk menteni. Az internet erre meglehetősen kényes. Saját érdeklődésünkben és azért is, hogy a file minden féle-fajta környezetben jól megtekinthető legyen az oldal, egyszerűbb, tagolatlan és ékezet nélküli neveket használjunk. A könyvben végig ilyen file-okat használunk!

Most nézzük meg a legegyszerűbb weblapot!

Egyszerű weblap: ([minta1.html](#))

```
<HTML> Szia, itt vagyok! </HTML>
```

A dolog lényege, hogy nem kell tagolni a szöveget, nem kell semmi különös formátumos trükköt ismerni, csak a kész anyagot megnézni és már működik is a weblap. Ráadásul egyszerűbb lapok esetén ez minden egyes gépen ugyanígy néz ki. Nem szabad elfelejteni, hogy egy weblapot mindig <HTML> és </HTML> tagek közé kell tenni. Ami viszont fontos, hogy a HTML-nyelvben lényegtelen a kisbetű-nagybetű különbsége. Tehát a <HTML>, <html>, <Htm> és <HtM> ugyanazt jelentik! Problémát okozhat, hogy a szöveget nem túl egyszerű formázni. Például próbáljuk ki a következő file-t: ([minta2.html](#))

<pre>&lt;HTML&gt; Egyes sor. Kettes sor. Harmas sor. &lt;/HTML&gt;</pre>	<p>Ekkor a következő fog megjelenni egy böngésző ablakában:</p>
--	---



Ahhoz, hogy egy szöveg az eredeti formázás szerint nézzen ki a `<PRE>` tagot kell használni. (PRE=Preformatted - előre formázott)

Az előző file helyesen így néz ki: ([minta3.html](#))

Ha nem vagyunk hajlandók szöveg formázásával szórakozni, akkor egyszerűbb megoldás a formázást a böngészőre bízni. Minden sor végén egyszerűen egy `<BR>` taget kell tenni, melynek hatására a böngésző automatikusan új sort kezd. *Vigyázat!* Ennek a tagnek kivételesen nincsen zárótagja! Tehát a helyes - és egyszerű - forma a következő: ([minta4.html](#))

```
<HTML><PRE>Egyes sor.<BR>Kettes sor.<BR>Harmas sor.</PRE></HTML>
```

Természetesen az egyes sorokat külön bekezdésben is írhatjuk, de így egyetlen sorban is elérjük a megfelelő hatást.

## 2. fejezet: Betűk formázása

A dolog igen egyszerű, hiszen lehet egy betű **félkövér** (Bold = B), aláhúzott (Underlined = U), illetve *döntött* (Italic = I)

A megfelelő tagek ezek szerint: `<B>` Bold `<U>` Underlined `<I>`, illetve `<I>` Italic `</I>`.

Az előbbi parancsok persze kombinálhatók is: `<B><I>` **Félkövér Döntött** `</I></B>`

Egyetlen dologra kell vigyázni csupán: A sorrendet mindig be kell tartani. Gyakorlatban ez azt jelenti, hogy mindig a legutóljára kinyitott tag zárótagját kell leírni. Nézzük a következő - természetesen betűformázási - példát: ([minta5.html](#))

<pre>&lt;HTML&gt; &lt;B&gt;Vastag. &lt;I&gt;Ráadásul aláhúzott&lt;/I&gt; Megint simán vastag&lt;/B&gt; &lt;/HTML&gt;</pre>	<pre>&lt;HTML&gt; &lt;B&gt;Vastag.   &lt;I&gt;Ráadásul aláhúzott&lt;/I&gt; Megint simán vastag&lt;/B&gt; &lt;/HTML&gt;</pre>
Íme a file, ahogy az összetartozó tagek láthatóak.	

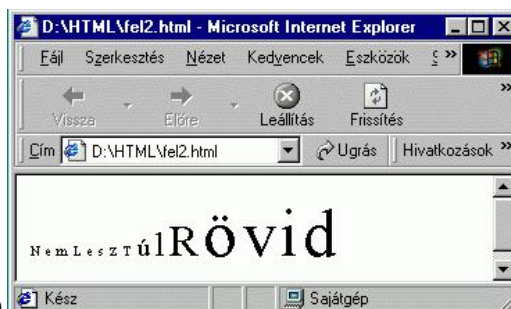
**Kicsit pontosabban fogalmazva:** ez olyan, mint egy csomó, egymásba csomagolt doboz. Mivel minden doboz be akarunk zárni, ezért nyilvánvalóan a legkisebbel kell kezdenünk, majd csak utána haladhatunk fokozatosan a nagyobbak felé. Persze, egy nagyobb dobozban lehet több, kisebb doboz is, de itt is érvényes, hogy előbb a kicsiket kell lezárni és csak utána a nagyot.

1. feladat: Írjon olyan fájlt, amelyben aláhúzott és döntött szöveg is van! ([fel1.html](#))

Ennél eggyel bonyolultabb a betűk méretének változtatása. Természetesen itt is lehetséges a szövegszerkesztőknél megszokott az abszolút méretezés, de szerencsésebb használni a relatív méretezést, ami egy közepes beállításához képesti eltérést tesz lehetővé. A közepes méret a 0-ás: ([minta6.html](#))

```
<HTML>
<FONT SIZE=0> Alapméret </FONT>
</HTML>
```

A skála -7-től +7-ig terjedhet.



2. feladat: Írjon olyan fájlt, amelyben a betűméretek fokozatosan növekednek! ([fel2.html](#))

Bár nem szerencsés, de a betűk típusa is változtatható. Ehhez szintén a `<FONT>` tagre van szükség, de ezúttal a paramétere a `SIZE` helyett a `FACE`-re lesz szükség, de szeretném kihangsúlyozni, hogy ez nem minden esetben szerencsés, mivel lehet, hogy a honlap készítőjének gépen éppen megvan az a betűtípus, de a leendő felhasználóknak nincsen! Mintapélda legyen a következő: ([minta7.html](#))

```
<HTML>
<font face="Arial">Arial</font>
<font face="Times New Roman CE">Times New Roman CE</font>
</HTML>
```

Persze, ha valamelyik betűtípus hiányzik, akkor az alapértelmezett betűtípust kapjuk helyette.

Ha már a betűk formázásánál tartunk, akkor nem árt tisztázni, hogyan lehet az eddig meglehetősen egyhangú dokumentumainkat kiszínezni. Első körben a betűknek adjunk új színt. Alapértelmezésben minden weblapnak vannak beépített színei, de ezeket nagyon egyszerű felülbírálni! A színezéshez a `COLOR` paramétert kell használni! A színek nevei a legegyszerűbb angol nevek lehetnek, például: `black`=fekete, `green`=zöld, `blue`=kék, `gray`=szürke, `yellow`=sárga, `purple`=bordó,...

Persze a fenti paramétereket lehet keverve is használni, ráadásul a sorrend is lényegtelen! Nézzük a következő példát:

**Ez igen!** A fenti példa egy igen rövid kóddal hozható létre: ([minta8.html](#))

```
<HTML>
<font face="Arial" size="+3" color="yellow">Ez igen!</font>
</HTML>
```

Tehát a szöveg stílusa "Arial" mérete a normál- (szabványos-)hoz képest 3-mal nagyobb és zöld színű!

Ha a hagyományos színek kevésnek bizonyulnak, akkor lehetőségünk van mesterségesen kevert színek előállítására is. A módszer lényege a három alapszín (R=red; G=green; B=blue) keverése. Bármelyik szín erőssége 0-tól 255-ig állítható, sajnos hexadecimális kóddal. (00-tól FF-ig) A kódsor elejére egy '#' jelet kell rakni. Néhány hagyományos és érdekes szín kódja

a következő:

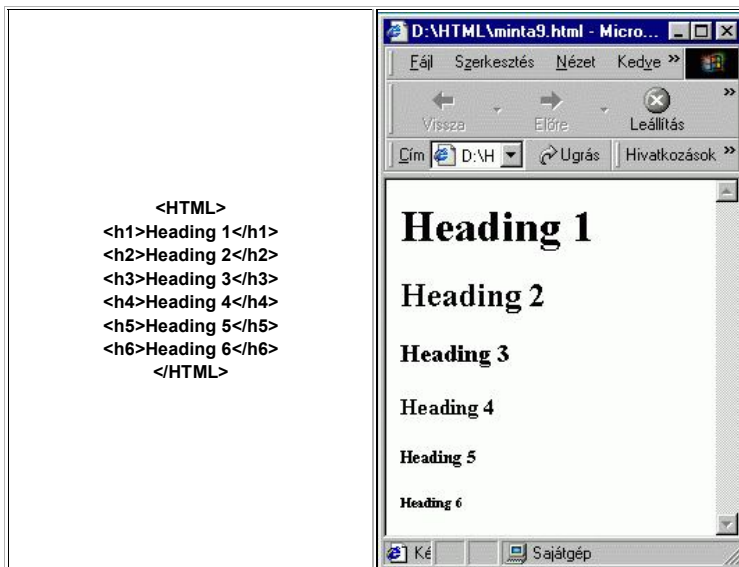
black = "#000000" = fekete	green = "#00FF00" = zöld	blue = "#0000FF" = kék
gray = "#C0C0C0" = szürke	red = "#FF0000" = vörös	yellow = "#FFFF00" = sárga

Javaslom, hogy bátran kísérletezzen mindenki! Mindezek ellenére a könyv *referenciájában* található egy sokkal részletesebb szinkiosztás!

### 3. fejezet: Fejezetek és formázásaik

A böngészőkben vannak eleve beépített stílusok, amit érdemes ki is használni. Ezek a címfarmátumok, pontosabban - jobb magyar kifejezés híján - a "Heading"parancsok. 6 eleve beépített formátum van, sorszám szerint 1-től 6-ig. Az 1-es a legerősebb bekezdés-formátum, míg a 6-os a legkisebb. Apró megjegyzés: a gyakorlatban 2-3 szintű fejezetcímnél csak ritkán alkalmazunk többet!

Nézzük az alábbi (bal oldali) példa megjelenését (a jobb oldalon). ([minta9.html](#))



Ha egy cím túl hosszúra sikeredik, de semmiképpen nem szeretnénk, ha megtömre a böngésző, akkor érdemes használni a **NOWRAP** paramétert is,

pl.: `<H1 NOWRAP> Ezt nem szabad megtörni! </H1>`

Különleges lehetőség a `<BANNER>` is, mivel ez egy soha nem scrollozható címszalagot takar! `</BANNER>` Ezzel a tulajdonsággal nem szabad visszaélni, mert kisebb felbontású monitorokon lehet, hogy nem marad scrollozható rész! Csak Netscape-ben!

Másik lehetőség a `<DIV>` használata, amely különböző részekre (fejezetekre, bekezdésekre) tagolja a szöveget. `</DIV>` Legfontosabb paraméterei: **ALIGN="center/left/right"** - igazítás; **DATAFORMATAS="text/html"** - adatformátum sima szöveg vagy html-kód; **TITLE="mit írjon ki"** az egér fölé érkezésekor.

Az automatikus sortörelés lelítható a `<NOBR>` és a `</NOBR>` parancsokkal. Ez feloldható menet közben a `<WBR>` taggal, de ennek nincsen zárótagja.

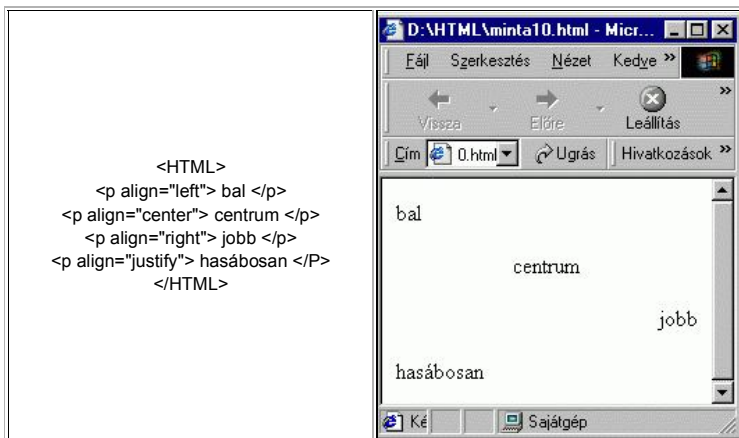
Fontos megjegyzés, hogy soha ne egyetlen képernyőméretre optimalizáljuk a megjelenést (pl.: 1024\*768), mivel mindig sokféle monitor van egyszerre a piacon!

Ezeket az eleve beépített mintákat persze módosíthatjuk is, de ez csak pár fejezettel később kerül sorra!

Ha már egy résznek címet adtunk, akkor érdemes megtölteni tartalommal! Egy-egy külön gondolatot érdemes egy-egy külön bekezdésbe írni. Új sort már tudunk kezdeni a `<BR>` tag segítségével, de ez valamivel több, mint egy egyszerű új sor, mivel itt egész bekezdéseket lehet írni, ráadásul rendezni is lehet! Minden hagyományos szövegszerkesztő és táblázatkezelő programban megszokhattuk, hogy lehet balra (left), jobbra (right) és középre (center) rendezni. Az internet eleve a legkülönbözőbb típusú gépekbe van bekötve, ezért igen gyakran előfordulhat, hogy a legkülönbözőbb képernyőfelbontású monitorokon jelenik meg az a - számunkra kiválóan elrendezett - szöveg, amit mi meg szeretnénk jeleníteni. Ekkor lesz értelme, hogy bekezdésekben gondolkozzunk, mivel minden egyes paragrafust másként tudunk elrendezni. A paragrafust keretbe záró tag a `<P>` és a `</P>`. Itt az új sorral ellentétben **van** lezárójel is! Automatikusan minden paragrafus balra zár, de az **ALIGN="hogyan"** paraméter segítségével lehet másféleképpen is.

Ez például balra van zárva. Alapértelmezés. Kódolása: <code>&lt;P ALIGN="left"&gt; duma &lt;/P&gt;</code>	Ez például középre van rendezve (zárva). Kódolása: <code>&lt;P ALIGN="center"&gt; centrum &lt;/P&gt;</code>	Ez például jobbra van zárva. Kódolása: <code>&lt;P ALIGN="right"&gt; jobb &lt;/P&gt;</code>
---	---	---

Bár az internetről sokáig idegen volt, de megjelent a hasábszerű elrendezés. (`ALIGN="justify"`) Ezt is lehet alkalmazni, de ettől sajnos relativ lassabb lehet a weblap megjelenése egy kisebb számítási képességű lapon. Az egész file (és képe) egybefüggően a következő lesz: ([minta10.html](#))



Szintén javasolt néha napjainkban alkalmazni a `<no>` és a `</no>` tag-eket, melyek segítségével megakadályozhatjuk, hogy egy fontos szöveg (például név) belsejét megtörje a böngésző. Szintén lehet alkalmazni a vízszintes vonalakat. A legegyszerűbb `<HR>` tagnek nincsen záróeleme, mivel csak egy szimpla vízszintes vonalat húz. Paraméterei lehetnek többek között:

**COLOR="színkód"**. (vonal színe) - Csak az Explorerben! Egyes alternatív böngészők már használják.

**ALIGN="left/center/right"** (merre rendezze a vonalat)

**NOSHADOW** (ne legyen 3-dimenziós árnyéka a vonalnak)

**SIZE=n** (milyen vastag legyen a vonal - pixelben; alapértelmezés: 1)

**TITLE="kiírás"** (mit írjon ki a böngésző, amikor a vonal fölé kerül az egér)



WIDTH="n" (milyen hosszú legyen a vonal százalékban vagy pixelben. Alapértelmezés: 100%)

Természetesen vannak még betűformázások, de az legyen kicsit később!

#### 4. fejezet: Kapcsolatok

Az egész internet lényege a kapcsolatok, avagy a hiperlinkek megléte. Ez a gyakorlatban azt jelenti, hogy egy sima egy objektumhoz (sima szöveg, kép vagy részlete, nyomógomb, stb) hozzárendelhetünk az egész internetről egy másik objektumot. Ha például a saját honlapomra akarunk írni egy hivatkozást, akkor a következőt kell beírni a weblap megfelelő helyére: ([minta11.html](#))

```
<A HREF="http://www.tferi.hu/"> Hivatkozás </A>
```

Megjelenéskor ez a "Hivatkozás" nevű szöveg általában aláhúzva jelenik meg és a már jól ismert Windows-cursor  átalakul mutató kézzé: 

Tehát, mint a példán is látható, az-> tag hivatkozást jelent. Ha simán egy egész webkikötőre szeretnénk hivatkozni, mint a fenti példában is, egyszerűen be kell írni a TELJES nevét. (Általában működik http és www nélkül is, de lehet, hogy némileg lassabban! Célszerű minden esetben a teljes verziót beírni, mivel így szinte biztosan jó lesz.) Némileg módosul a helyzet, ha a webkikötő belső felépítését is ismerve nem az egész iskolára szeretnénk hivatkozni, hanem például a "Számítástechnika kezdőknek" könyv kezdetére. Ekkor a helyes hivatkozás a következő lesz: ([minta12.html](#))

```
<A HREF="http://www.tferi.hu/konyv5/tftanul.html"> Könyv lelegeje </A>
```

Ha azon az adott könyvtáron belül szeretnénk hivatkozni, ahol a kiindulási file található, akkor elegendő a file nevét megadni. Például, ha az alábbi [minta13.html](#)-ből szeretnénk hivatkozni az előző fejezet végén lévő [minta10.html](#)-re, akkor a következő kódot kell használni: ([minta13.html](#))

```
<A HREF="minta10.html"> Azonos könyvtárban </A>
```

Egy adott (nagy méretű) file-on belül is tudunk ugrani, de itt picivel bonyolultabb a dolog, ugyanis a file-on belül kell létrehozni egy "címkét". Ez tulajdonképpen egy láthatatlan könyvjelző, amit bármikor fel tudunk lapozni. Kódja: `<A NAME="könyvjelzőnév"></A>` Nézzünk egy példát: ([minta14.html](#))

```
<HTML>
  <a name="teteje">Ez a file teteje.</a>
  <BR>Itt szöveg van.
  <BR>Itt is.
  <BR>Meg itt is.
  <a href="ftp://ftp.microsoft.com/">Microsoft FTP-szerver.</a>
  <a href="mailto:hivatalos@tferi.hu">E-mail a szerzőnek.</a>
  <A href="#teteje">Itt lehet a tetejére ugrani.</A>
<A href="http://www.tferi.hu/konyv5/html/minta14.html#teteje">Ez picit bonyolultabb!</A>
</HTML>
```

Természetesen nem csak `http`- szabvány szerinti file-ra utalhatunk, hanem `ftp`-re, illetve `E-mail`-re is. Ezen kívül ha ismerjük az adott szerver DOS-struktúra szerinti felépítését, akkor hivatkozhatunk rá a jó öreg DOS szerint is, mint a `cd`-parancsnál. Persze még van számos kisebb szabvány is (gopher, news, ...), de ezek mostanra már szinte teljesen kikoptak a hétköznapi használatból.

3. feladat: Írjon egy olyan fájlt, amelyben link van a szerző honlapjára ([tferi.hu](#)), a Microsoft-ra ([microsoft.com](#)) és a Netscape-re ([netscape.com](#))! ([fel3.html](#))

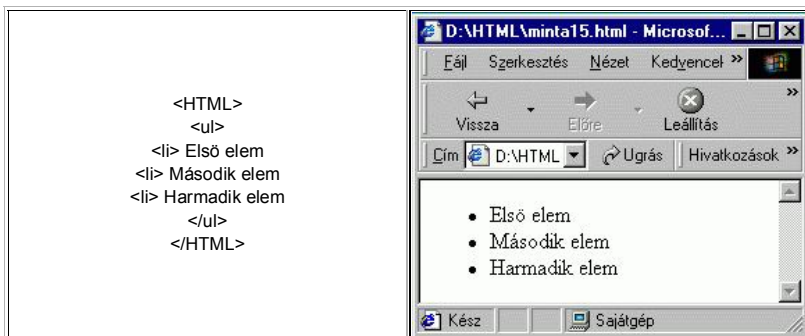
4. feladat: Írjon egy olyan fájlt, melyben hivatkozik a Netscape FTP-re, valamint a szerzőnek küld E-mailt ([hivatalos@tferi.hu](mailto:hivatalos@tferi.hu))! ([fel4.html](#))

#### 5. fejezet: Listák

Az Internet kezdetekor szöveges volt szinte minden hozzáférés, éppen ezért az áttekinthetőség kedvéért rendkívüli fontossággal bír mindenféle lista és felsorolás.

A legegyszerűbb a rendezés nélküli lista. (**Unordered List = UL**) Ennek minden egyes eleme a listatag. (**List Item = LI**) Ennek a tag-nek nincsen zárórésze.

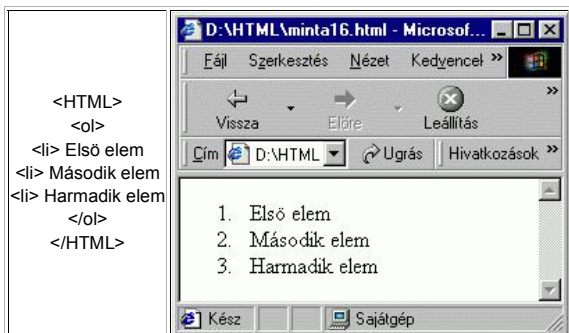
Lássuk a következő példát: ([minta15.html](#))



The screenshot shows a web browser window titled "D:\HTML\minta15.html - Microsof...". The browser's address bar shows "D:\HTML". The main content area displays an unordered list with three items: "Első elem", "Második elem", and "Harmadik elem". To the left of the browser window, the corresponding HTML code is shown:

```
<HTML>
<ul>
<li> Első elem
<li> Második elem
<li> Harmadik elem
</ul>
</HTML>
```

Természetesen lehet rendezett is a lista (**Ordered List = OL**). Ha az előbbi file-ban kicseréljük az `<ul>` taget `<ol>`-re, akkor a következő lesz a lista! ([minta16.html](#))



The screenshot shows a web browser window titled "D:\HTML\minta16.html - Microsof...". The browser's address bar shows "D:\HTML". The main content area displays an ordered list with three items: "1. Első elem", "2. Második elem", and "3. Harmadik elem". To the left of the browser window, the corresponding HTML code is shown:

```
<HTML>
<ol>
<li> Első elem
<li> Második elem
<li> Harmadik elem
</ol>
</HTML>
```

Rendezett lista esetén az `<li>` egyik paramétere megszabja a lista típusát.

**Type = 1** - a listaelemek számok.

**Type = a** - a listaelemek kisbetűk.

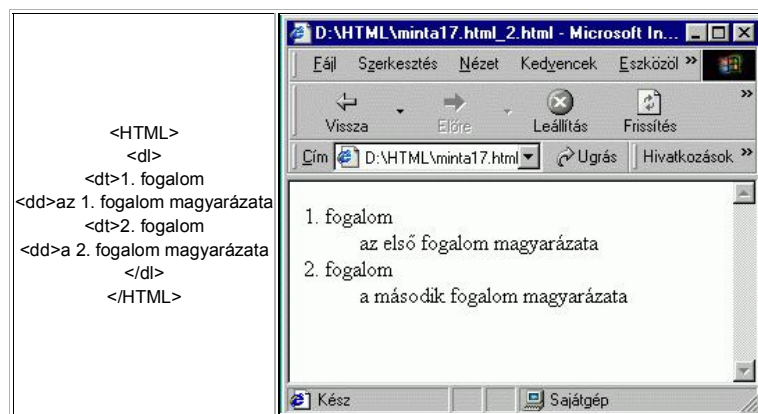
**Type = A** - a listaelemek nagybetűk

**Type = i** - a listaelemek kicsi római számok.

**Type = I** - a listaelemek nagy római számok.

Ugyanezen paraméterek megjelenhetnek az <OL> elemnél is, ráadásul lehet egy **START=n** elem is, ami a kezdőértéket adja meg!

Definíciók listát is készíthetünk, ahol egyszerűbb fogalmakat lehet megmagyarázni. (Definition List = DL). Az egyes elemeket és fogalmakat <DT>-vel jelöljük, (Definition Term) míg magyarázatukat <DD>-vel. (Definition Description) ([minta17.html](#))



Természetesen ezeket a mintákat némi kiegészítő tudással kicsit át is variálhatjuk, (lásd **CSS**-file-ok), de ez nem egy ilyen alapfokú tanfolyam dolga!

## 6. fejezet: Ékezetek

**Némi kiegészítés az eredeti verzióhoz:** amikor a könyv eredetijét írtam, akkor ez a fejezet még kiemelt fontosságú volt. Azóta ez jócskán csökkent, ugyanis a kódlapok (később lesz szó róluk) olyan mértékben elterjedtek, hogy gyakorlatilag megkerülhetetlenek. Óriási előnyük: immáron már nem kell szenvedni az egyes ékezetes karakterek kódjaival. Így ezt a fejezetet inkább csak, mit érdekességet ajánlom!

Az ékezetek nem mindig és könnyedén használhatóak a HTML-nyelvben. Eleve a dolog azért problémás, mert rengeteg nemzeti nyelv van a világon, viszont az internet alapvető nyelve a meglehetősen kicsi karakterkészlettel rendelkező angol. Az egyik lehetséges megoldás az, hogy mindenki csak a saját nemzeti nyelvén készült honlapokat láthatja. Nyilvánvaló, hogy ez mereven ellentmond a web nemzetköziségének. Ráadásul eleve lehetetlenné teszi a nyelvtanulást.

Másik verzió, hogy eleve minden nemzeti nyelv összes karakterét beépítjük a böngészők összes verziójába, de ezzel a képirásos keleti nyelvek (kínai, japán, koreai, stb.) hatalmas mennyiségű jelkészlete miatt a méret drasztikus növekedését érzük el. Lehetséges középút, hogy minden alapvető (angol) karakternek van egy egyszerű (1 byte-os) kódja (aki ismeri: ez az ASCII kód). Gyakorlatilag ez a sima, egyszerű "ekezet ne'ku:li i'ra'smo'd". Ezt persze érdemes repülő ékezetekkel kombinálni, mint az előző példában. Ha ennél többre vágyunk, akkor trükközni kell. Az ékezetes betűk viszonylag egyszerűek, mivel az alapvető (angol) karakter köré az **&xacute**; kombináció kell írni, ahol az "x" az adott karaktert jelöli. De itt van egy táblázat is - a jobb érthetőség kedvéért!

Á = &Aacute;	É = &Eacute;	Í = &Iacute;	Ó = &Oacute;	Ú = &Uacute;
á = &aacute;	é = &eacute;	í = &iacute;	ó = &oacute;	ú = &uacute;

Mint a fentiekből is látható itt kivételesen FONTOS a kisbetű-nagybetű különbsége!

Ha rövid dupla ékezet kell, akkor a **&xuml**; kombinációt kell alkalmazni, míg az ő karakterhez a **&otilde**; kombináció kell vagy az ű-höz az **&ucirc**;

Ö = &Ouml;	Ü = &Uuml;	Ő = &Otilde;	Ű = &Ucirc;
ö = &ouml;	ü = &uuml;	ő = &otilde;	ű = &ucirc;

A legjobb jóindulat ellenére is előfordulhat, hogy nem minden karakter jelenik meg pontosan, mivel egyes Windows betűkészletekben nem feltétlenül találhatók meg (az USA-ból nézve) igencsak egzotikus magyar ékezetes karakterek. Természetesen vannak még speciális karakterek, például a "<" és a ">" jeleket helyettesítő kódok is.

< = &lt; (less than)	> = &gt; (greater than)	= &nbsp; (space)	¢ = &cent; (cent-jel)	£ = &pound; (font-jel)
¥ = &yen; (yen-jel)	& = &amp; (angol és-jel)	© = &copy; (copyright)	® = &reg; (registered)	ä = &auml; (umlautos a)

Ha valakinek eddig gondjai támadtak az ékezetes karakterekkel, akkor remélem, hogy ez a pár mondat megoldja! **5. feladat: Írja le a következő ékes magyar kifejezéseket - HTML kóddal: hosszúságú cipőfűző, illetve árvízű tükörfűrógép.**

Még egy apró megjegyzés, ami később nyer fontosságát. Az előző módszerrel gyakorlatilag bármelyik nyelv karakterét le tudjuk írni, ha az bele van építve a HTML-szabványba. Ha nincsen, akkor a <HTML> tag után közvetlenül be kell gépelnünk a következő részletet - PONTOSAN! ([minta18.html](#)) Magyarázat: ez a magyar nyelvű betűkészletet jelöli, így a gépelés közben nem kell fiyelnünk az ékezetes karakterekre.

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

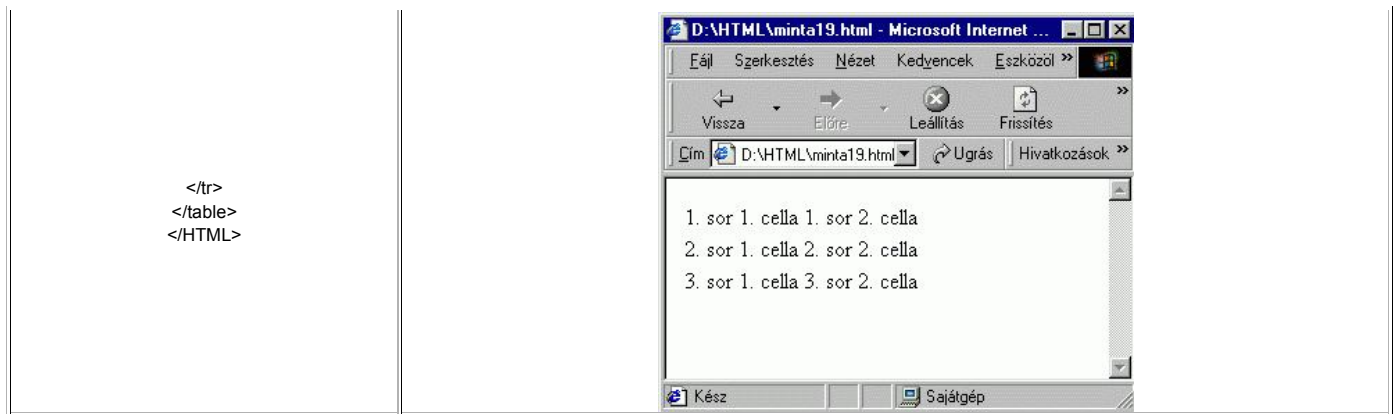
Eme segítséggel már nyugodtan gépelhetünk, mint az a 18. mintából is látható! A pontos jelentésről majd (sokkal) később, mivel most fontosabb az alapok folytatása.

**5. feladat: Gépelje be a fenti két szöveget! (fel5.html)**

## 7. fejezet: Táblázatok

Nos, elértük a HTML nyelv egyik legjobban megjeleníthető objektumát: a táblázatokat. Maga a táblázat tagje igen egyszerű: **<TABLE>** és **</TABLE>**. Ezen belül viszont sok minden lehet. Minden egyes sort **<TR>** és **</TR>** közé kell zárni, míg minden egyes cella tartalmát **<TD>** és **</TD>** közé. Ha semmi különös bonyolítást nem szeretnénk, akkor már el is készíthetjük az első táblázatot! ([minta19.html](#))

<pre>&lt;HTML&gt; &lt;table&gt; &lt;tr&gt; &lt;td&gt;1. sor 1. cella&lt;/td&gt; &lt;td&gt;1. sor 2. cella&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;2. sor 1. cella&lt;/td&gt; &lt;td&gt;2. sor 2. cella&lt;/td&gt; &lt;/tr&gt; &lt;tr&gt; &lt;td&gt;3. sor 1. cella&lt;/td&gt; &lt;td&gt;3. sor 2. cella&lt;/td&gt; &lt;/tr&gt;</pre>	Íme az eredmény:
--	------------------



Ahhoz, hogy kicsit szebbé tegyük a táblázatot, érdemes **<TABLE>** tagbe különböző paramétereket beírni.

A legegyszerűbb a táblázat igazítása: **<TABLE ALIGN="left">** Ez az alapeset. **<TABLE>** Persze lehet középre (**center**), illetve jobbra (**right**) is igazítani.

A második lehetőség a keret vastagsága. **<TABLE BORDER=0>** Ez az alapeset. **<TABLE>** A vastagság képpontokban mérendő és egész szám lehet. Szokásos érték: 1 és 5 között.

A harmadik lehetőség látszólag felesleges, mivel érdemes az oszlopokat megszámlálni és ezt a táblázat fejlécébe beírni. **<TABLE COLS=3>** 3 oszlop lesz!**</TABLE>**

Negyedszerre érdemes figyelembe venni, hogy a táblázatot lehet (és érdemes is) színezn. **<TABLE BGCOLOR="red">** Ez a **hagyományos** színnel vagy kódokkal történik **<TABLE>**.

Lehet egy kép is a háttér. Lásd: **<IMG SRC="képpurl">**

Persze nem csak a táblázat háttérét, de a kereteket is lehet színezn! **<TABLE BORDERCOLOR="yellow">** Így zöld lesz a keret színe. **<TABLE>**. Saját tapasztalat, hogy a színekkel mértékkel kell bánni!

Az egyes cellákat nem kell összezsúfolnunk, mivel a cellákat ki lehet bélelni. **<TABLE CELLPADDING=2>**. Ez a cellahatár és az adat távolságát mutatja meg. Alapeset a 0. **<TABLE>**.

Persze az egyes cellákat sem kell mindig összezsúfolni, így a **CELLSPACING** használatával nagyobb lehet a cellahatárok közti távolság. Alapeset: 0.

Az egyik legfontosabb lehetőség maradt a legvégére: a táblázatnak nem kell kitöltenie az ablak egészét, hanem a **<TABLE WIDTH="50%">** paraméter segítségével megadható, hogy hány százalékot töltsön ki. **<TABLE>** Ha csak simán egy szám szerepel a **WIDTH** után, akkor százalék helyett képpont lesz az értelmezés.

A fenti paraméterek persze kombinálhatóak! A következő táblázat például jobbra igazítva, 2-es keretvastagsággal, 2 oszloppal szürke háttérrel és piros keretekkel készült a böngészőablak 60%-ára. A cellák bélése 2, akárcsak a cellák közti távolság is. ([minta20.html](#))

Paraméter angolul	Magyarul	Lehetséges értékek
ALIGN	igazítás	left/right/center
BORDER	keretvastagság	0, 1, 2, 3, ...
COLS	oszlopok száma	1, 2, 3, 4, ...
BACKGROUND	háttérszín	szín neve vagy kódja, esetleg kép URL-je
BORDERCOLOR	keretszín	szín neve vagy kódja
WIDTH	táblázat szélessége	pixelszám vagy %

```

<html>
<table ALIGN=RIGHT BORDER=2 CELLSPACING=2 CELLPADDING=2
COLS=3 WIDTH="60%" BGCOLOR="#FFFF00" BORDERCOLOR="red">
  <tr>
    <td><b>Param&eacute;ter angolul</b></td>
    <td><b>Magyarul</b></td>
    <td><b>Lehets&eacute;ges &eacute;rt&eacute;k&eacute;k</b></td>
  </tr>
  <tr>
    <td>ALIGN</td>
    <td>igaz&iacute;t&aacute;s</td>
    <td>left/right/center</td>
  </tr>
  <tr>
    <td>BORDER</td>
    <td>keretvastags&aacute;g</td>
    <td>0, 1, 2, 3, ...</td>
  </tr>
  <tr>
    <td>COLS</td>
    <td>oszlopok sz&aacute;ma</td>
    <td>1, 2, 3, 4, ...</td>
  </tr>
  <tr>
    <td>BACKGROUND</td>
    <td>h&aacute;t&eacute;rsz&iacute;n</td>
    <td>sz&iacute;n neve vagy k&oacute;dja, esetleg k&eacute;p URL-je</td>
  </tr>
  <tr>
    <td>BORDERCOLOR</td>
    <td>keretsz&iacute;n</td>
    <td>sz&iacute;n neve vagy k&oacute;dja</td>
  </tr>
  <tr>
    <td>WIDTH</td>
    <td>t&aacute;bl&aacute;zat sz&eacute;less&eacute;ge</td>
    <td>pixelsz&aacute;m vagy %</td>
  </tr>

```

```
</tr>
</table>
</html>
```

6. feladat: Készítsen néhány nyomtatott magyar napilapról egy táblázatot, mely tartalmazza a lap címét és a webcímét! A táblázatnak legyen kerete! ([fel6.html](#))

#### További lehetőségek

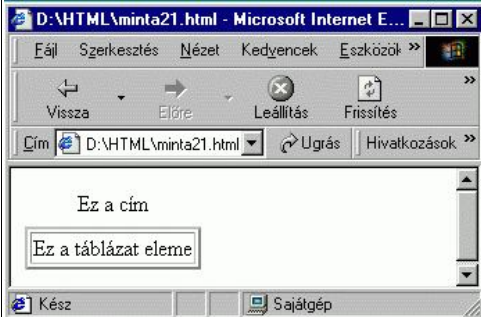
Egy táblázatnak címet is adhatunk. Ezt a **<TABLE>** szimbólum után kell írni a következő minta szerint: ([minta21.html](#)) Szabályozható, hogy hová kerüljön a cím **<CAPTION ALIGN="TOP">** felülre kerül - alapértelmezés. **<CAPTION>**.

További lehetőségek: bottom = alulra és középre, center = középre, left = balra, right = jobbra.

Függőleges helyzetet szabályozhatjuk a **VALIGN** paraméterrel. Minta: **<CAPTION VALIGN="TOP">** - Alapértelmezés = A táblázat címe a táblázat fölé kerül. Másik lehetséges érték: **BOTTOM** = a cím alulra kerül.

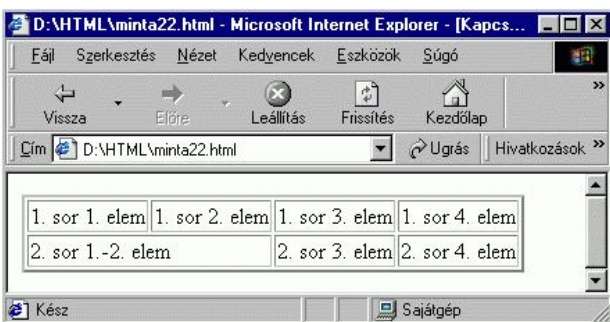
A két paraméter persze együtt is használható!

```
<HTML>
<table border="2">
<caption> Ez a cím </caption>
<tr><td> Ez a táblázat eleme</td></tr>
</table>
</HTML>
```



Szükség esetén két vagy több cellát is összevonhatunk, mint ez a következő példában látható! ([minta22.html](#)) A cellafeszítés (colspan) tipikus használata látható itt:

```
<HTML>
<table border="2" cols=4>
<tr>
<td> 1. sor 1. elem </td>
<td> 1. sor 2. elem </td>
<td> 1. sor 3. elem </td>
<td> 1. sor 4. elem </td>
</tr>
<tr>
<td colspan="2"> 2. sor 1.-2. elem </td>
<td> 2. sor 3. elem </td>
<td> 2. sor 4. elem </td>
</tr>
</table>
</HTML>
```



Függőlegesen ugyanez: **ROWSPAN**.

Minden egyes cella tartalmát darabonként is igazíthatjuk **<TD ALIGN="LEFT">** Balra rendez - alapértelmezés **<TD>** Lehet középre (**center**) vagy jobbra (**right**) is rendezni.

Természetesen egyenként is lehetőségünk van a cellák szélességének állítására is az egész táblázat szélességére alkalmazott **WIDTH** paraméterrel.

Minden egyes cellához lehet rendelni egy címet is, mely akkor jelenik meg, ha az illető cella fölé kerül az egér. **<TD TITLE="cim">** elem tartalma **<TD>**.

Hosszú szövegeknél előfordulhat, hogy egy sornál több is lehet 1-1 cella tartalma. Ilyenkor jól jöhet a **<TD VALIGN="CENTER">** paraméter **<TD>**, ahol **valign** = vertical align = függőleges rendezés. Lehetséges értékek: **center** = közép; **top** = fent; **bottom** = alul; **baseline** = a betűk szintjének aljához igazítja az objektumot.

Ennek némileg ellentmond, hogy ha semmiképpen sem akarjuk, hogy egy cellában a szöveget megtörje, akkor használhatjuk a **NOWRAP=TRUE** paramétert a **TD** tag-ben. Ekkor a cellatartalom nem kerül törésre. Alapértelmezés: **NOWRAP=FALSE**. (azaz a cellatartalom törhető.)


Cellánként is lehetőség nyílik háttérszín (**BGCOLOR**), illetve háttérgrafika (**BACKGROUND**) definiálására, azért csak óvatosan a díszekkel...

Külön lehetőségünk van **<TH>** táblázat fejlécének - table head **<TH>** és **<tfoot>** táblázat láblécének - table foot **<tfoot>** definiálására is, bár ezt a nem-Microsoft böngészők nem tudják igazán megjeleníteni.

Értelemszerűen létezik a táblázatnak a törzse is. Ez a **<tbody>** és a **</tbody>** közé kerülhet.

Ha egy táblázatot 3-dimenziós módban (többfajta keretszínnel) szeretnénk látni, akkor erre is ad lehetőséget a HTML nyelv újabb definíciója, de csak a **<table>** tagben. Ilyen lehetőség a **BORDERCOLORLIGHT** és a **BORDERCOLORDARK**, amint ez a következő példában is látható: ([minta23.html](#))

```
<HTML>
<table border="4" cols="4" bordercolor="red" bordercolorlight="gray" bordercolordark="green">
... A többi lényegében ugyanaz, mint az előző file-ban.
```



Ezek után már bátran neki lehet állni táblázatok formázásának és kitöltésének!

## 8. fejezet: Képek és videók

Valamirevaló weblap semmit sem ér képek és látványosabbnál látványosabb animációk, esetleg videók nélkül. A kép beszúrása is ugyanolyan egyszerű, mint az eddigi legtöbb parancs: **<IMG SRC="képnév.kit">**, ahol **IMG** = image = kép/látvány; **SRC** = source = forrás. Záróelem nincs, akár a **<BR>** tagnél. Ha a forrásfile azonos könyvtárban van a weblappal, akkor simán be lehet írni a file nevét. Ha az adott weblap környezetéből kell vennünk egy forrásfile-t, akkor a jól ismert DOS-struktúra szerint utalhatunk rá; ellenben ha teljes hivatkozás kell URL-lel, akkor a **hiperhivatkozásoknál megismert** módszert kell alkalmaznunk.

Támogatott formátumok - a W3.org szabványügyi szervezet hivatalos forrása szerint:



```
.avi - Audio-Visual Interleaved (AVI)
.bmp - Windows Bitmap (BMP)
.emf - Windows Enhanced Metafile (EMF)
.gif - Graphics Interchange Format (GIF)
.jpg, .jpeg - Joint Photographic Experts Group (JPEG)
.mov - Apple QuickTime Movie (MOV)
.mpg, .mpeg - Motion Picture Experts Group (MPEG)
.png - Portable Network Graphics (PNG)
.wmf - Windows Metafile (WMF)
.xbm - X Bitmap (XBM)
```

A képet követő szöveg helyzetét az **ALIGN** paraméter rendezi. Értékei: **Absbottom** (=abszolút alja), **Absmiddle** (=abszolút közép), **Baseline** (=alapvonal), **Bottom** (=alja), **Left** (=balra), **Middle** (=középre), **Right** (=jobbra), **Texttop** (=szöveg közepére), valamint **Top** (=tetejére).

Hasznos lehet, ha a kép(ek) miatt egy ténylegesen üres sort behúzunk. Ilyen esetekben érdemes használni a jól ismert **<BR CLEAR=ALL>** tag-et, mely után a szöveg csak a kép alatt folytatódik.

Ha túl nagy a kép, akkor érdemes a letöltés előtt egy sima helyettesítő szöveget kiírni az **ALT="text"** paraméterrel, ahol értelemszerűen a text helyett kell beírni a szöveget.

Természetesen meg lehet adni a kép keretének vastagságát is a szokásos **BORDER** paraméterrel - pixelben megadott egész szám.

Érdemes megadni a kép szélességét (**WIDTH**), valamint magasságát (**HEIGHT**) - persze szintén képpontban, hogy a kép letöltésétől függetlenül a böngésző már tudja rendezni az oldalt, de ezt a párost lehet használni a kép átméretezéséhez is!

Néha jól jöhet egy kis helyet hagyni a grafikus elem alatt és felett (**VSPACE**), illetve balra és jobbra. (**HSPACE**).

Gyakori megoldás, hogy a kiváló minőségű (és ezért lassan letöltődő) kép helyett először egy gyenge felbontású, de még éppen, hogy látszó "piszkozatot" löknek ki a weblapra. A **LOWSRC** paraméter már csak azért is hasznos, mert így a nem kívánt kép teljes letöltését nem kell megvárni. Természetesen itt is meg kell adni a teljes forrást.

Egy teljes kép letöltését lehet látni a következő file-ban: ([minta24.html](#))

```
<HTML>

</HTML>
```



Betölthető file-ok az Explorerbe és a Firefox-ba egyaránt: **GIF, JPG, JPEG és PNG**. Ha sima animációs **GIF**-eket szeretnénk betölteni, akkor ez semmi gondot sem jelent. Például a bal oldalt látható "kukac" is így forog. Ellenben a videók már több gondot okozhatnak. Ilyenkor a kezdőkor a forrásfile megadásánál **SRC** helyett **DYNSRC**-t kell használni. (**DYNSRC** = Dynamic Source). Használható file-kiterjesztések az **MPG, MPEG** és az **AVI**, bár ez utóbbival a a régebbi Firefox-oknak gondjai lehetnek.

**E-MAIL** Egy videórészletet persze többször is le lehet játszani a **LOOP** paraméter segítségével. Alapértelmezés **LOOP=1**. Végtelen lejátszást érhetünk el a **LOOP="-1"** (esetleg **LOOP="infinite"**) adat segítségével.

Csak az Explorerben használható paraméter a **LOOPDELAY**, mely ezredmásodpercben szabályozza, hogy két lejátszás között mennyi idő teljen el.

## 9. fejezet: Keretek (frame-ek)

<p>A HTML-oldalak egyik leghasznosabb része a keretek (angolul: <b>FRAME</b>-ek) használata. Ezt a <b>szomszédos dokumentum</b> is bizonyítja. (<a href="#">minta26.html</a>) A felépítése gyakorlatilag igen egyszerű, mivel a két keret külön-külön is felhasználható és minden hivatkozhat mindenre. Amint ez a forráskódban is látható, a kereteket két tag: a <b>&lt;FRAMESET&gt;</b> és a <b>&lt;FRAME&gt;</b>. Természetesen a <b>&lt;FRAMESET&gt;</b>-ben százalékosan, illetve pixelesen is megadható a keretek aránya. Szükség esetén több keret is egymásba ágyazható. A vízszintes felosztáshoz a "rows" szócskát kell használni, míg a függőleges felosztáshoz a "cols" szükséges.</p>	<pre>&lt;HTML&gt; &lt;FRAMESET rows="50%,50%"&gt; &lt;FRAME SRC="fel1.html"&gt; &lt;FRAME SRC="fel2.html"&gt; &lt;/FRAMESET&gt; &lt;/HTML&gt;</pre>	
---	---	--

Egyéb paraméterek is lehetségesek: A **BORDER=n** paraméter a keretek közötti helyet adja meg képpontban (=pixelben). Alapértelmezés: 1.

A keret színét a **BORDERCOLOR="színkód"** határozza meg, amint ezt előbbieken is láthattuk.

Ha nem szeretnénk keretvonalat, akkor a **FRAMEBORDER="no"** paraméter kell. Vigyázat! Így a lapunk könnyen áttekinthetetlen lehet! (Ellentéte: **FRAMEBORDER="yes"**)

Ha szeretnénk felkészülni régebbi böngészőkre is (Explorer 3-asig és Netscape 2-esig bezárólag), akkor érdemes használni a **<NOFRAMES>** taget is, melynek záróértéke természetesen a **</NOFRAMES>**, amint ez az alábbi példában is látható: ([minta27.html](#)). Tény, hogy a mai, modern böngészőkben erre már nem kell felkészülni!

<pre>&lt;HTML&gt; &lt;FRAMESET COLS="100,*"&gt; &lt;FRAME SRC="fel1.html"&gt; &lt;FRAME SRC="fel2.html"&gt; &lt;NOFRAMES&gt; &lt;BODY&gt; Hát ez nem nyert! &lt;/BODY&gt; &lt;/NOFRAMES&gt; &lt;/FRAMESET&gt; &lt;/HTML&gt;</pre>	<p>Ha valaki szeretne felkészülni ilyen igényekre is, akkor kénytelen elkészíteni a keretek nélküli verziót is - sajnos.</p> <p>Még valami: ha nem feltétlenül kell, akkor ne adjuk meg minden frame vastagságát - főként pixeles értékénél tilos ezt tenni! Ilyenkor érdemes a bal oldali példa 2. sorában látható csillagot (*) használni, ami a jó öreg DOS-os hagyományok szerint a maradék részt jelenti, azaz a mindent.</p>
---	--

Persze lehetőség van arra is, hogy az egyes keretekkel külön is trükközzünk. A legegyszerűbb a scrollozás. A paraméter lehetséges értékei: **SCROLLING** = NO, YES, illetve **AUTO**. (**NO** = tilos a scroll; **YES** = mindig megjelenik a gördítőcsík; **AUTO** = Csak akkor jelenik meg a csík, ha az szükséges - ez az alapérték.)

Az **SRC** után természetesen a forrásfile-t kell írni.

A **NORESIZE** (külön érték nélküli) paraméter megakadályozza, hogy a felhasználó átméretezhesse a weblap kereteit. Ellentéte a **RESIZE**.

Itt is van lehetőségünk a keret színének megadására a **BORDERCOLOR="színkód"** (vagy név) segítségével, de véleményem szerint szerencsésebb a **FRAMESET** tagben a szín megadása.

A **MARGINHEIGHT=pixelszám** segítségével az egyes keretek tetejétől és aljától való távolságot lehet megadni.

Hasonlóan a keret bal és jobb oldalától való távolságot a **MARGINWIDTH=pixelszám** paraméterrel állíthatjuk.

A **TITLE="kiírandó duma"** paraméter hatására a beírt szöveg (jelenleg: kiírandó duma) meg fog jelenni a tippek sorban a böngésző legalján, ha az egérrel föléje megyünk.

A legfontosabb paraméter még hátravan. Ez a **NAME="ablak\_neve"**. Sok értéke van, melyek azt határozzák meg, hogy hová kell megnyitnia az új file-t egy rákattintáskor. Figyelem! Ezt csak úgy tudjuk használni, hogy a megnyitásra használt `<A HREF="file neve.html">` tagbe bele kell illeszteni a **TARGET="ablak neve"** paramétert is! Az ablakokat tetszőlegesen el lehet nevezni és érdemes is ezzel élni, ha elég bonyolult a lap hivatkozása. Az ablakok lehetséges - beépített - nevei: "\_self" = önmagába nyitja meg; "\_blank" = egy tiszta, névtelen lapban nyitja meg; "\_parent" = a lapot a közvetlen szülőjében nyitja meg, ahonnan eredtek a linkek; "\_top" = az ablak teljes méretében nyitja meg az új linket. A félreértések elkerülése végett szeretném kihangsúlyozni, hogy a beépített nevek előtt egy alsó vízszintes vonás van! (Általában: jobb SHIFT és a mellette lévő mínusz a billentyűzeten.)

#### Összegezve:

A "\_self" paraméter esetén a megfelelő frame helyébe töltődik az oldal.

A "\_parent" paraméter esetén a megfelelő frame fölé töltődik az oldal.

A "\_top" esetén legfölülre.

A "\_blank" paraméter esetén egy vadonatúj böngésző ablakába.

Az "ablakneve" paraméter használata esetén a megadott nevű keretben nyílik meg a hivatkozás.

A "\_search" paraméternél a link a böngésző kereső-paneljába töltődik be. Minimum Explorer 5-nél használható.

A "\_media" esetén a link a böngésző MediaBar nevű részébe (magyarul: média panel) töltődik be. Minimum Explorer 6-nál használható.

7. feladat: *Készítsen egy ilyen kétoldalú elhelyezkedő keretrendszer. Minta: [fel7.html](#).*

Ahogy haladt előré az élet (meg az egyre újabb böngésző-szabványok), úgy változik a HTML is. Ennek egyik áldozata lett a **FRAME**. Helyette bejött az **IFRAME**. Használata végtelen egyszerű, mivel egy virtuális keretet hoz létre a weblapon belül. Például a következő kódot kell beépíteni a weblapba:

```
<HTML>
...
<IFRAME SRC="http://www.tferi.hu/"></IFRAME>
...
</HTML>
```

Fontosabb paraméterek:

- ALIGN: (left/right/top/middle/button): rendezés a szokott helyre. Elavult. A DIV használata javasolt helyette.
- FRAMEBORDER: (1/0): legyen keret (1) vagy nem (0).
- HEIGHT: (pixel vagy %): A keret magassága képpontban vagy %-ban.
- LONGDESC: megadja azt az oldalt, ahol az iFRAME hosszabb leírása található.
- MARGINHEIGHT: margó magassága képpontokban.
- MARGINWIDTH: margó szélessége képpontokban.
- NAME: a keret neve. Ezzel lehet hivatkozni rá linkeléskor!
- SCROLLING: (yes/no/auto): a keret görgethetősége.
- SRC (URL): megadja az IFRAME keretbe beépített dokumentum címét.
- WIDTH: (pixel vagy %): a keret szélessége képpontban vagy %-ban.

## 10. fejezet: Egyéb formázások

### Szövegek dekorálása

Természetesen egyéb formázási lehetőségeink is lehetnek. Az alábbi néhány sor csak pár lehetőséget mutat be:

**<ADDRESS>**

*Postacímét jelöl ki.*

**</ADDRESS>**

**<BLINK>** Ez itt villog, de csak Netscape-ben! **</BLINK>**

**<BIG>** Nagyobb karakterméret **</BIG>**

**<BLOCKQUOTE>**

Szövegblokk - érdemes kipróbálni!

**</BLOCKQUOTE>**

**<CITE>** *Idézet - ritkán használható***</CITE>**

**<CODE>** Programkód - a más betűstílus miatt érdemes használni**</CODE>**

**<COMMENT>** Kommentálás - semmi hatással nincs a dokumentumra. Nem íródik ki **</COMMENT>**

**<DFN>**Definált szöveg **</DFN>**

**<EM>** *Kiemelés - ritkán használt***</EM>**

**<KBD>** Billentyűzetről begépelhető input írógép stílussal **</KBD>**

**<S>** Keresztülhúzott szöveg**</S>**

**<SAMP>**

Irodalmi szövegrészlet, illetve mintakarakter

**</SAMP>**

**<SMALL>** Kisebb betűméret **</SMALL>**

**<STRONG>** Vastagított megjelenés **</STRONG>**

**<SUB>** Alsó index **</SUB>**

**<SUP>** Felső index **</SUP>**

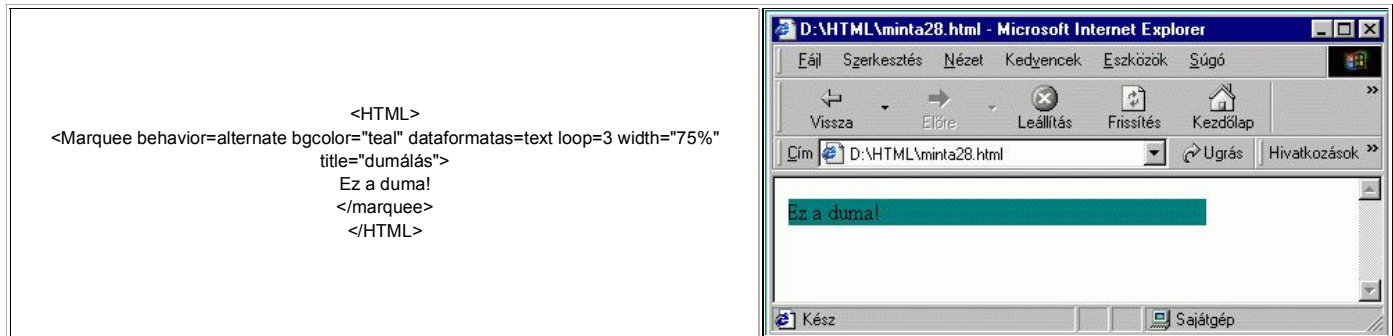
**<VAR>** Változónevet jelöl ki **</VAR>**

**<TT>** Írógépkarakterek**</TT>**

### Marquee

Érdekes, de sajnos csak az Explorerben megjelenő lehetőség a szövegek feliratszerű mozgatása a **<MARQUEE>** és a **</MARQUEE>** között. Előfordulhat, hogy egyes alternatív böngészőkben is megjelenik a hatás kívánásunk szerint, de ez nem túl valószínű!

Általában akkor lehet látványos az effektus, ha az ablak nagyobb részében látható lesz a mozgás, mint a lenti példában: ([minta28.html](#))



```
<HTML>
<Marquee behavior=alternate bgcolor="teal" dataformatas=text loop=3 width="75%"
title="dumálás">
Ez a duma!
</marquee>
</HTML>
```

Pár paraméter:

**BEHAVIOR** = alternate/scroll/slide - mozgásirány = balra-jobbra/átlapozás (scroll) a direction által meghatározott irányba/a szöveg levágása az előző mintájára

**BGCOLOR** = háttérszín kódja vagy neve

**DATAFORMATAS** = text/html - adatformátum - sima szöveg/html

**DIR** = ltr/rtl - szöveg olvasási irányának meghatározása; ltr=balról jobbra (alapértelmezés); rtl = jobbról balra

**DIRECTION** = left/right/up/down - scrollozás iránya - bal/jobbról/fel/le; Alapértelmezés: left

**HEIGHT** = "n" - Hány %-ot, vagy képpontot foglalhat el a HTML-tag a képből.

**HSPACE** = "n" - Mekkora legyen a HTML-tag mellett a szabad hely vízszintesen?

**VSPACE** = "n" - Mekkora legyen a HTML-tag mellett a hely függőlegesen?

**LOOP** = "n" - Hány ismétlés után álljon le a mozgás. (Végtelen = -1)

**SCROLLAMOUNT** = "n" - Hány pixel elcsúszással rajzoljon ismét a MARQUEE.

**SCROLLDELAY** = "n" - Hány millimásodperc után rajolja újra a HTML-taget.

**TITLE** = "cím" - Mi jelenjen meg, ha az egeret fölé visszük?

**WIDTH** = "n" - Az ablak hány százalékában jelenjen meg a MARQUEE.

8. feladat: Készítsen egy jobbról balra 4-szer elhaladó pár szavas szöveget, amely az aktuális ablak 65%-át foglalja el, de az egész alakzat középre legyen igazítva és sárga háttére legyen. (Minta: [fel8.html](#))

9. feladat: Ugyanezekkel a paraméterekkel készítsen egy újabb objektumot, melyben egy **weblap linkje** göndül tova. (Minta: [fel9.html](#))

#### Basefont

Fontos karakterformázási lehetőséget tartalmaz a <BASEFONT> tag, mely egy dokumentum alapvető karakterkészletét tartalmazhatja, amelyhez képest a többi karakter hogyan nézzen ki. Lehetséges paraméterei - főleg az Explorerben: COLOR (szín) és FACE (karakterkészlet neve). A főbb böngészők ismerik a SIZE (méret) paramétert. További segítséget jelenthet a következő file ([minta29.html](#)):

```
<HTML>
<basefont color="blue" face="Arial, Courier" size="4">
Ez lesz az alapértelmezés!
</basefont>
</HTML>
```

#### BGSound

Csak az Explorer által használt lehetőség a <BGSOUND> (háttérhang). Itt hangkártyával rendelkező gép segítségével lejátszathatjuk az oldalhoz rendelt háttérhangot. Választható kiterjesztések: **WAV**, **AU**, **MID** és **CD**. Kötelező paraméter a forrás megjelölése (SRC). Lehetséges paraméter az ismétlések számát szabályozó LOOP=n, melyben n egy sorszám. Abban az esetben, ha n = -1, akkor végtelenszer ismétlődik a háttérhang. Érdekes lehetőség a **BALANCE=n** paraméter, ahol n -10000 és +10000 közötti érték. Ez határozza meg a bal és jobb oldali hangszórók közti hangosság értékét. Alapértelmezés: 0. (Egyenlő hangosság.) További hasonló paraméter a **VOLUME=n**, ami neve alapján sejtethetően a hangerőért felelős. n értéke -10000 és 0 között lehet, ahol 0 a legnagyobb hangerő.

#### 11. fejezet: BODY és META

Az eddigi képnél kicsit bonyolultabb, árnyaltabb fogalmazás szükséges a teljes értékű HTML lapokhoz. Ezentúl a dokumentum sablonjához a következőt ajánlom: ([minta30.html](#))

<pre>&lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Ez a cím&lt;/TITLE&gt; &lt;META name="description" content="Lap leírása"&gt; &lt;META name="keywords" content="Kulcsszavak"&gt; &lt;META name="generator" content="Mivel lett csinálva"&gt; &lt;META name="author" content="Szerző neve"&gt; &lt;META name="robot" content="index,follow"&gt; &lt;META http-equiv="Expires" content="Mon, 27 Nov 2000 14:25:27 GMT"&gt; &lt;META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-2"&gt; &lt;/HEAD&gt; &lt;BODY BGCOLOR="#FFFFFF" TEXT="#000000" LINK="#0000FF" VLINK="#800080"&gt; Na, és csak itt jön a lényeg! &lt;/BODY&gt; &lt;/HTML&gt;</pre>	<p>Látható, hogy maga a &lt;HTML&gt;-tartomány két részre oszlik. Van egyrészt egy fejléc&lt;HEAD&gt; és &lt;/HEAD&gt; között, valamint egy szövegtörzs a &lt;BODY&gt; és a &lt;/BODY&gt; között. Eddig csak ez utóbbi került sorra. A fejléc megléte nem kötelező, de sokat tud segíteni a következő elemekkel:</p> <p>&lt;TITLE&gt; A Dokumentum címe &lt;/TITLE&gt;</p> <p>A &lt;META&gt; kezdetű elemeknek nincsen zárórésze és nem is jelennek meg, de annál hasznosabbak! Használatuk kizárólag a fejlécben engedélyezett. Főbb paraméterei már a bal oldali kódban is láthatóak.</p> <p>Szókások:</p> <ul style="list-style-type: none"> <li>- <b>Description:</b> pár szavas leírás - ösztönítő!</li> <li>- <b>Keywords:</b> A lapra jellemző kulcsszavak a keresőrobotoknak.</li> <li>- <b>Generator:</b> Milyen programmal lett a lap elkészítve. (Pl. Jegyzettömb)</li> <li>- <b>Author:</b> A szerző neve, esetleg E-mail címe!</li> <li>- <b>Robots:</b> Parancsok a keresőrobotok számára. Lehetséges értékek: "<b>index/noindex</b>" indexelhető vagy nem; "<b>follow/nofollow</b>" követhető a hivatkozások vagy nem. Mivel úgyis mindkettőt használni kell, ezért célszerű helyettük írni a "<b>deny</b>" szót (tilos mindkettőt), illetve az "<b>all</b>" szót, amely mindkét tevékenységet engedi.</li> <li>- <b>Expires:</b> Lejárati ideje. Ha pl. csak minden hónap elsején frissíti a lapját, akkor érdemes a mindig következő frissítési dátumot beírni. Így közvetlenül a Proxy-kból is le tud tölteni a lap. Ha mindig a legfrissebb tartalmat akarja láttatni, akkor időpont helyett egyszerűen a "<b>NOW</b>" (=most) szót kell beírnia.</li> </ul> <p>Valamennyi &lt;META name=...&gt; kezdetű tagben használható a <b>lang="en"</b>, vagy a <b>lang="hu"</b> paraméter, ami a nyelvet jelöli. Indexelni például nem bűn angolul akkor is, ha a lap kizárólag magyar!</p>
---	--

- A **HTTP-EQUIV** egy már lehetőséget rejt. Ugyanis itt megadhatjuk a böngészőnek, hogy az adott dokumentum mikor veszti érvényét. Így ha a böngésző újabb parancsot kap a dokumentum letöltésére, akkor ha a cache-ben eltárolt határidő előtt van még a weblap, akkor nem bajmóldók a letöltéssel, hanem beolvassa magából a gépből és már készen is van. Célszerű egy weblapnál nem túl távoli időpontot megadni!

- A másik sor (<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-2">) pedig a magyar karakterkészlet használatára készíti elő a gépet.

- Lehetőség nyílik arra is, hogy egy dokumentumot állandóan frissítsünk. A <META HTTP-EQUIV="REFRESH" CONTENT="2"> használata például 2 másodpercenként frissít egy weblapot! (Tözsdei árfolyamoknál még jól jöhet.)

- Több értelme lehet egy másik webhely automatikus betöltésének. A <META HTTP-EQUIV="REFRESH" CONTENT="5; URL=http://poszter.ini.hu/"> tag például az URL-ben megadott webhelyt hozza be 5 másodperc elteltével!

- Szintén érdemes használni a <META HTTP-EQUIV="Reply-to" content=" [hivatalos@tferi.hu](mailto:hivatalos@tferi.hu) "> taget is, ami a lehetséges válaszcímet adja meg.

Egy másik fontos elem a <BODY> és a </BODY> közé ékelte szövegtörzs nyitóeleme. Rengeteg lehetőséget tartalmaz és igen sokféleképpen lehet alkalmazni. Lehetséges paraméterei közül csak a lényegesebbek kerülnek itt felsorolásra:

- **BGCOLOR**="színkód" - background color; A háttérszín neve vagy kódja.
- **ALINK**="színkód" - active link; Az éppen aktív link(ek) színkódja vagy neve.
- **VLINK**="színkód" - visited link; A már meglátogatott linkek színkódja vagy neve.
- **LINK**="színkód" - link's color. A még meg nem látogatott linkek színkódja vagy neve.
- **TEXT**="színkód" - text color; A sima (linkek nélküli) szöveg színkódja vagy neve.
- **BACKGROUND**="url" - Háttérkép neve (esetleges útvonallal).
- **BGPROPERTIES**=FIXED - Vízjel megadása. Ez így nem gördíthető!
- **TITLE**="szöveg" - Ez a szöveg lesz kiírva az egérmutató fölé (mintegy segítségként), ha az ablak fölé ér.
- **NOWRAP**="TRUE" - Szöveg nem törhető meg automatikusan. Alapértelmezés: **NOWRAP**="FALSE".
- **SCROLL**="NO" - Eltünteti az oldalsó scrollozó-csíkot. Alapértelmezés: **SCROLL**="YES". Lehetséges érték még: **"AUTO"** (automatikus).
- **BOTTOMMARGIN**=n - Alsó margó pixelben. Felülbírálja az alapértelmezést. Csak egész szám lehet. Alapértéke: 15. Minimum: 0.
- **TOPMARGIN**=n - Felső margó pixelben. Felülbírálja az alapértelmezést. Csak egész szám lehet. Alapértéke: 15. Minimum: 0.
- **LEFTMARGIN**=n - Bal oldali margó pixelben. Felülbírálja az alapértelmezést. Csak egész szám lehet. Alapértéke: 10. Minimum: 0.
- **RIGHTMARGIN**=n - Jobb oldali margó pixelben. Felülbírálja az alapértelmezést. Csak egész szám lehet. Alapértéke: 10. Minimum: 0.

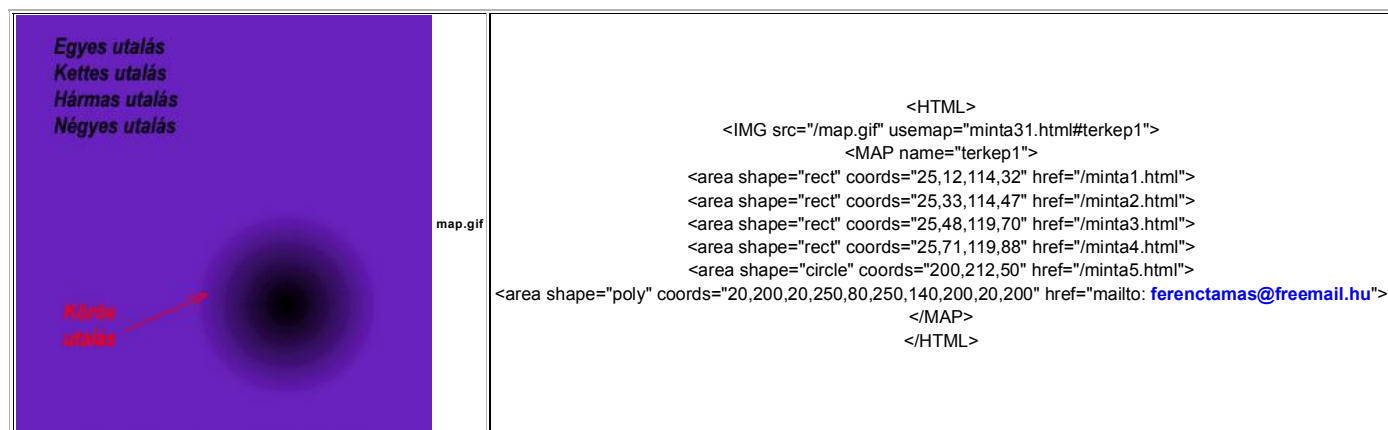
További elemek is lehetnek a weblap fejlécében:

<INDEX> : Nem tartalmaz semmi szöveget és nincsen záróelem sem. A böngésző számára jelzi, hogy kereshető (indexelhető) dokumentumról van szó. Javasolt helyette a **"robots"** META-tag használata.

<LINK> : Jelzi a dokumentum kapcsolatát más dokumentumokkal, stílussal címszalaggal, stb. Kell záróelem. Paraméterei: **DISABLED** - megtiltja az elem használatát; **HREF**="url" - élőkapocs létrehozása; **MEDIA**="SCREEN" vagy **MEDIA**="PRINT" vagy **MEDIA**="ALL" - meghatározza a dokumentum kimenetét, lehetőségek: képernyő, nyomtató, minden (alapértelmezés)

## 12. fejezet: Térképek

Ha egy bonyolultabb képünk van rengeteg utalással és felirattal és/vagy rajzzal, akkor célszerű egy olyan rendszert kialakítani, amelynél egyes részekre kattintva nyitódhat meg egy újabb kapcsolat. Mintaképpen nézzük meg a következő képet és file-t: ([minta31.html](#))



(Megjegyzés: a fenti kép a PhotoShop 5.0 és a CorelDraw 9 programokkal készült!)

A dolog lényege, hogy a térképnek és a feldolgozott utalásoknak nem kell azonos file-ban lennie, de lassabb kapcsolatú gépek miatt célszerű azonos file-ba tenni a két anyagot. Az első fontos utasítás az **IMG SRC**-ben lévő **USEMAP** paraméter. Itt meg kell adni a térképfeldolgozó file, valamint a térkép helyét.

A térkép feldolgozásához kell a **<MAP name="térképneve">** tag. A konkrét élőkapocsokat az **<area>** tag szabályozza. Paraméterei:

- **shape** = rect (téglalap), circle (kör), poly (polygon), default (mindenhol máshol - egyéb helyen)
- **coords** = koordináták. Téglalap esetén két átlellenes saroké, kör esetén a középpont és a sugár, poligon esetén sorban a sarkok (az utolsó legyen azonos az elsővel!).
- **href** = élőkapocs. Teljes utalás is lehet, nem csak egyszerű file.
- **alt** = alternatív szöveg.
- **target** = Megnyitandó file helye (pl.) megosztás esetén.
- **title** = az egérmutató hatására kiírandó szöveg.
- **nohref** = ennek a területnek nincsen élőkapcsa.

Minták: [Kapcsolatok \(4. fejezet\)](#).

10. feladat: Nem feltétlenül könnyű egy ilyen térkép (első) előkészítése, de némi próbálkozás után a fenti minta alapján sikerülhet!

## 13. fejezet: Kérdőívek a HTML dokumentumban

Hadd kezdjem ezúttal egy megjegyzéssel: a kérdőívek helyes összeállítása és megfogalmazása a gyakorlatilag rendelkezőknek nem túl nehéz, csak időigényes feladat. Én magam is úgy szokta elkészíteni őket, hogy a egy weblapszerkesztővel előbb előkészítem a terepet (bocsnát, a kérdőívet), majd utána kezdem el egy kódszintű szerkesztővel a finomhangolást!

Egy weblapon kérdőíveket is közé tehetünk, amelyek használata első pillantásra nem feltétlenül egyszerű!

A **<FORM METHOD="mód" ACTION="mit csináljak vele">** és a **</FORM>** közötti utasítások határozzák meg az űrlapot.

Az **ACTION** határozza meg a feldolgozást végző programot. Ez többnyire a kiszolgáló szerverén a **/cgi-bin/** könyvtárban van, de lehet E-mailben is elküldeni. Ez utóbbi sokkal egyszerűbb és nem kell hozzá semmilyen feldolgozó-program. Olvasása a legegyszerűbb szövegszerkesztővel (értsd: NotePad/Jegyzet) is lehetséges! Ha mindenáron egy feldolgozó-programot kell megcímezni, akkor kiterjesztése többnyire PHP vagy ASP szokott lenni. **Vigyázat!** Az ilyen programok írása nem feltétlenül egyszerű!

Minta levélküldésre: **<ACTION METHOD="mailto: hivatalos@tferi.hu">**

Minta feldolgozó-programra: **<ACTION METHOD="feldolgoz.php">**

A **METHOD** a kitöltött űrlap továbbítási módját határozza meg. Ez lehet **GET** (Hozzáfüzi az űrlapot az url-hez - időnként túl hosszúra nyúlhat az url!), illetve **POST** (adatcsomagban - biztonságos!)

Érdeemes használni **POST** adatküldés esetén az **ENCTYPE="TEXT/PLAIN"** paramétert, mert így tapasztalatom szerint minden levelezőprogram jól megérti a küldött adatokat, még az egyébként igen kényes [www.freemail.hu](http://www.freemail.hu) is!

Az **<INPUT NAME="név" TYPE="típus" ALIGN="igazítás">** utasítással határozható meg egy kitöltendő űrlapmező.

A **NAME** természetesen a mezőnév, amely alapján a feldolgozóprogram azonosítja a bevitt adatot. Ez gyakorlatilag szabadon megválasztható, de javasolt a klasszikus DOS szabványait megtartani és nem túl sokat variálni a nevekkel. Tapasztalatom szerint nem minden E-mail-en elküldött űrlap jön át tökéletesen, ha túl sok a magyar elnevezés.

Az **ALIGN** természetesen igazítást jelent. Lehetséges értékei: left, right, center.

Itt is használható a **TITLE**, mely itt is az egér hatására megjelenő szöveget jelenti. Tapasztalatom szerint bonyolultabb űrlapoknál sok mező rossz kitöltése elkerülhető pár **TITLE** tag-gel.

A **TYPE** pedig az adattípus, melyet vár a beviteli mező. Lehetséges típusok:

**TEXT** -szöveg,

**PASSWORD** - jelszó (nem jelenik meg bevitelkor!),

**HIDDEN**-rejtett (ez sem jelenik meg),

**CHECKBOX** - kapcsoló (több is kiválasztható egyszerre),  
**RADIO** - kapcsoló (egyszerre csak egyet lehet kiválasztani),  
**RANGE** - numerikus adat,  
**FILE** - csatolandó fájl,  
**SUBMIT** - adattovábbító gomb,  
**RESET** - megsemmisítő (inicializáló) gomb,  
**BUTTON** - egyéb nyomógomb.

Az <INPUT> utasításban további opciók is szerepelhetnek, a fő opciók értékeitől függően:

A **VALUE** kiegészítő opcióval megadott értéket veszi fel alapértelmezésként a szöveges vagy numerikus beviteli mező. Értékét nem kell megadni.

**TEXT** típusú mező esetén egy további opció, a **SIZE="méret"** opció határozza meg az ablak szélességét, a **MAXLENGTH="érték"** pedig a maximális hosszt. Lehetséges paraméter a **READONLY**, mely nevéből adódóan csak olvashatóvá teszi a file-t.

A **CHECKBOX** és a **RADIO** típusú mezők további paramétere lehet a **CHECKED** opció, mely bekapcsolja a kapcsolót - alapértelmezésként.

A **RANGE** típusú mező esetén megadható az a tartomány, melybe a bevitt értéknek bele kell esnie, a **MAX="maximum"** és a **MIN="minimum"** további opciókkal.

A **FILE** típusú mezőben megadott fájl az **ACCEPT** kiegészítő opcióval megadott **MIME** módon csatolódik az elküldendő kérdőívhez. (Megjegyzés: Egy Browse nyomógommbal támogatott fájlkereső-ablakból lehet a fájlt kiválasztani.)

A **SUBMIT** és a **RESET** gombokhoz tartozó kiegészítő opció a **VALUE="felirat"**, amely a gombok feliratát jelöli ki. Egyébként a **SUBMIT** gomb lenyomásának hatására küldi el az űrlapadatokat a kérdőív a feldolgozó programnak, a **RESET** gomb lenyomása pedig az alapértékekkel tölti fel a beviteli mezőket.

A **BUTTON** gombot nem feltétlenül kell használni, mivel gyakorlatilag semmilyen szerepe nincsen!

Hosszabb szöveg bevételére alkalmas a <TEXTAREA NAME="név" ROWS="magaság" COLS="szélesség" VALUE="szöveg"> és a </TEXTAREA> utasításpár, amely egy beviteli ablakot nyit megadott szélességben és sorban. A **VALUE** az alapértelmezésként megjelenítendő szöveget adja meg.

Egy kérdésre andandó válasz egyszerű - menüből történő - kiválasztását teszi lehetővé a kérdőívben a <SELECT NAME="név" SIZE="sor"> és a </SELECT> utasításokkal létrehozott kiválasztásos menü, melynek menüpontjait az <OPTION> utasítással adhatjuk meg. Az **OPTION** fontos paramétere a **VALUE="érték"**, ami a szöveges/karakteres mezőknél igen hasznos lehet. Érdemes használni, de nem szabad egy kérdőívben túl sok választást megadni, mert akkor tapasztalatom szerint az internetes felhasználó nem hajlandó végigcsinálni a kérdőívet.

A **SIZE** opció azt határozza meg, hogy hány sorban jelenjenek meg a **SELECT**-ben választható menüpontok egy szkrollozható menüben. Hiánya esetén, hagyományos legördülő menüből lehet választani. A **MULTIPLE** opció esetén több menüpont is kijelölhető egyszerre. Az <OPTION SELECTED> formájú utasítás adja meg az alapértelmezett választást!

Érdekes paraméterként lehet alkalmazni az **ACCESSKEY=gomb**-ot, amelynél a böngészőben lehet megadni az űrlapra ugráshoz használt **ALT+gomb** kombinációt!

Használható a **DISABLED** is, ami az éppen aktuális elem aktív használatát tiltja le.

Érdemes elgondolkozni az **AUTOCOMPLETE="ON"**, illetve **"OFF"** használatán egyes szöveges mezőknél, persze a jelszövegeket kivéve. Nevéből érthetően ez engedélyezi a mező automatikus kitöltését, ha már a felhasználó járt ott!

Lehetőségként felmerülhet, hogy alkalomadtán el kell rejtteni a fókuszt, magyarul azt a helyet, ahová először írni lehet. (A dolog értelme: a felhasználónak muszáj legalább egyet kattintania az űrlap kitöltése előtt!)

Ezt a **HIDEFOCUS="true"** értékkel tehetjük lehetővé. Alapértelmezésben a fókuszt látható, azaz **HIDEFOCUS="false"**.

Itt is felhasználható, igény szerint akár elemenként is a **LANG="nyelvi\_kód"**. A hazánkban használt legfontosabb nyelvi kódok:

```
hu = Hungarian/Magyar
en = English/Általános angol
en-us = English - United States /Amerikai angol
de = German - Standard/Általános német
de-at = German - Austria/Osztrák német
ro = Romanian/Román
ru = Russian/Orosz
cz = Czech/Cseh
```

További nyelvi kódok ebből az angol nyelvű táblázatból szedhetők ki: [htmlang6.html](http://htmlang6.html).

Egy apró tanács a komolyabb kérdőívek kitöltése előtt: mindig többszörösen próbáljuk ki a kész kérdőívet. Lehetőleg ne csak a saját gépünköről, hanem minél több helyről. Ha lehetőségünk van rá, akkor érdemes több verziójú Explorerben, Netscape-ben és minél több operációs rendszerben is kipróbálni. A rendszerek közül ne maradjon ki a Linux sem! Ha saját magunknak ez gondot okoz, akkor kérjük meg ismerőseinket az ellenőrzésre. "Éles" megrendelőívvel SOHA ne kísérletezessünk, csak az összes próba után!

A következő példa egy nem kifejezetten egyszerű kérdőívet mutat be: ([mintta32.html](http://mintta32.html))

```
<html>
<head>
<meta http-equiv="Content-Language" content="hu">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1250">
<title>Tamás Ferenc</title>
</head>
<body>
<form method="post" action="MAILTO: ferenctamas@freemail.hu" enctype="text/plain">
  <h1 align="center">Tamás Ferenc</h1>
  <h2 align="center">A számítástechnika alapjai CD-ROM megrendelőlapja</h2>
  <h3 align="center"><span style="background-color: #FF0000">EZ CSAK MINTA, NEM
  MEGRENDELŐLAP!</span></h3>
  <p align="center">&nbsp;&nbsp;&nbsp;</p>
  <div align="center">
    <center>
      <table border="1" width="75%">
        <tr>
          <td width="60%">
            <input type="checkbox" name="Konyv1" value="ON" checked disabled>
            Tamás Ferenc: A számítástechnika alapjai könyv</td>
            <td width="15%" align="right">4.777 KByte</td>
          </tr>
          <tr>
            <td width="60%">
            <input type="checkbox" name="PPT" value="ON" checked disabled>
            Kiváló minőségű bemutató PowerPoint fájlok</td>
            <td width="15%" align="right">300.000 KByte</td>
          </tr>
        </tr>
      </table>
    </center>
  </div>

```



ALT = szöveg (Helyettesítő szöveg, ha nem sikerül lefuttatni.)

CODE = filenév (A lefordított Java-class neve.)

CODEBASE = url (A bázis-url helye.)

HEIGHT = n (magasság)

HSPACE = n (Az applet vízszintes margója.)

NAME = név (Az applet neve.)

SRC = url (A forrásfile teljes url-je.)

TITLE = cím (A kiírt szöveg, amit az egér felette lebegésekor íródik ki.)

VSPACE = n (Az applet függőleges margója.)

WIDTH = n (szélesség)

BASE: A dokumentum báziscímének meghatározása - keretek (FRAME) alkalmazásakor. (BASE url) Paraméterei: HREF=url és TARGET=ablak\_név | \_blank | \_parent | \_self | \_top

BUTTON: Nyomógomb beszúrása. Használata szinte teljesen azonos a [kérdőívekben](#) megismertével.

CENTER: A nyitó- és zárótag közötti elemet, vagy elemeket középre rendezi.

COL: Táblázatban oszlop definiálása. Ritkán használt!

COLGROUP: Oszlopok közös csoportjának gyűjtője. Ritkán használt!

DEL: Dokumentumból törölt szöveg kijelölése.

DIR: Felsorolásszerű lista készítése. Eleme: [LI](#).

EMBED: Dokumentum beszúrá egy másikba. Fontosabb paraméterei, mint az [APPLET-nél](#).

FIELDSET: Mezőelem beszúrása. Csak LEGEND taget tartalmazhat közvetlenül utána.

IFRAME: Lebegő keretek újabb definiálási lehetősége. Paraméterei lényegében azonosak a [keretekével](#).

INS: A dokumentumhoz hozzáadott mező beszúrása.

LABEL: Címket definiál a kontrolláló elemek számára.

LEGEND: Megadja a FIELDSET elemét. Csak ezen belül lehet használni!

LISTING: Előre listázott szöveg. Kb. azonos a PRE taggel.

MENU: Listák menü-szerű felsorolása. Eleme: [LI](#).

NOSCRIPT: A scriptek megjelenítésére képtelen böngészők számára a HTML-kódot tartalmazza.

OBJECT: Objektum beszúrása a HTML-dokumentumba. pl. kép, dokumentum applikáció, vezérlés, stb. Fontosabb paraméterei, mint az [APPLET-nél](#).

PARAM: Változókat határoz(hat) meg egyéb elemeknek, például: [APPLET](#), [EMBED](#), [OBJECT](#).

PLAINTEXT: Mindenféle formázás nélküli sima szöveg megjelenítése.

Q: Kiemelt szöveg megjelenítése. Ritkán használt!

SCRIPT: Egy más nyelven írt script beszúrása.

SPAN: Érvényességi kör definiálása.

STRIKE: Keresztülhúzott szöveg kiírási mód. Lásd: [S tag](#).

TBODY: Táblázat törzsének definiálása. Ritkán használt!

THEAD: Táblázat fejlécének definiálása. Ritkán használt!

XMP: Mintaszöveg definiálása. Nem használt. Javasolt helyette: [PRE](#) és [SAMP](#).

Módosítás: ( 2012. január 21. szombat, 16:04 )