

1. Kommunikáció és a Web szerverek

A WWW kommunikációját a HTTP (HyperText Transfer Protocol) kommunikációs protokoll valósítja meg. A kliens-szerver rendszerek esetében a felhasználói igényeket a kliens programok közvetítik a szerver felé. A böngésző az URL-ben meghatározott protokollt használja. Webszerverekhez kapcsolódás esetén a HTTP protokollt kell alkalmaznia (FTP szerverhez az FTP protokollt stb.).

A **kliens program** feladata, hogy elérje az URL-ben meghatározott protokoll segítségével a szintén az URL-ben megadott szerveret, és a szerveren lévő objektumot (HTML oldalt, fájlt stb.), valamint megjelenítse azt. Ezek a kliens programok a böngészők.

A **web szerver** valamely (általában) távoli számítógépen futó program, mely kiszolgálja a kliens igényeit, és itt található meg a HTML dokumentumok

2. A HTML szerkesztés alapjai

A World Wide Web-en megjelenő dokumentumok leíró nyelve a HTML. A **HTML szerkesztéséhez szükség van az ahhoz kapcsolódó alapfogalmakra:**

Tag: a tag-ek a nyelv elemei, ilyen elem pl. a bekezdés, a kép, a link stb., ezeket a $\langle \rangle$ jelpár közé kell írni.

Form (Űrlap): speciális HTML tag, mely adatbevitelt tesz lehetővé, az azon belül elhelyezett további elemek segítségével (ilyen elemek pl. a beviteli mezők, a nyomógombok, a választónégyzetek, a rádiógombok stb.), magyar megnevezése űrlap.

Frame (Keret): előfordul, hogy egy oldalt több felé osztanak, ezek a frame-ek (keretek), melyeknek lényege, hogy külön-külön scrollozható a tartalmuk (ha engedélyezzük); minden frame-nek külön HTML forrása van, mivel egymástól teljesen különálló egységek.

Script:script-eket használnak arra, hogy valamilyen eseményhez egy tevékenységsorozatot rendeljenek. Ez azt jelenti, hogy egy esemény bekövetkezésekor az ahhoz rendelt scriptben leírt tevékenységek végrehajthatók (ilyen lehet pl. ha egy CD-kereskedés weblapján a felhasználó kitölt egy CD megrendelő lapot, majd a "Rendelés elküldése" nyomógombra kattint és ennek hatására megjelenik a CD-kért fizetendő összeg, valamint a megrendelés a szerveren tárolásra kerül). A script egy olyan programkód, melyet direkt módon futtat a böngésző, tehát nincs szükség a fordítására sem.

CGI:a CGI (Common Gateway Interface) írja le azt az eljárást, ahogyan a script-ek a szerverrel kommunikálnak, valamint adatfeldolgozást végeznek.

Dinamikus és statikus weboldalak

Statikus weboldalak: egyszerű, előre elkészített weboldalak amelyek változtatás nélkül, csak olvasható formában található a webszerveren. (HTML oldalak).

Dinamikus weboldalak: képesek bizonyos programnyelvek, script-ek valós időben történő futtatására is. CGI programozás (DHTML, Java applet, PHP, perl, pascal, C, stb)

3. A lap környezete

Mint bármilyen egyéb kiadványnak, a weblapnak sem mindegy, milyen környezetben kerül elhelyezésre. A helyzet mindig tartalmaz olyan korlátokat, amit nem lehet figyelmen kívül hagyni egy weblap megtervezésekor.

Az ideális környezet:

- a vonalak torlódásmentesek és mindig hozzáférhetőek;
- az olvasók közel azonos böngészőfelületet használnak;
- a tárolókapacitás és a gépteljesítmény elegendő;
- a hozzáférési jogok a weblap készítő(i) számára megfelelőek;

- a fejlesztési eszközök adottak és használhatóak;
- a rendszer védett és megbízhatóan működik.

A weblap készítőjének fel kell készülni arra, hogy mindezen feltételek közül jó néhány nem adott, és esetleg hosszas utánajárással sem lehet a helyzeten javítani.

Lehetőségek a nehézségek elhárítására, vagy megkerülésére:

3.1 Szűk sáv szélesség

- A lapok rövidek, kevés és tárméretben kicsiny képet tartalmaznak, a nagyobb méretű képek letöltését külön kérésre biztosítják
- Megoldás lehet, hogy külön "lassú" és "gyors" változatot állít elő a webszerkesztő, ám ez a karbantartásnál a munka többszöröződését vonja maga után.

Magyarország egyelőre ott tart, hogy a többség számára túl lassú a hozzáférés.

3.2. Eltérő böngészők

A lapkészítő alapvetően háromféle típussal találkozunk:

- az általa preferált böngészővel;
- az ezzel nem kompatibilis (és kisebb tudású) böngészőkkel;
- karakteres böngészőkkel.

A weblapok alapvetően grafikus környezetre lettek kitalálva. A klasszikus kiadványszerkesztéshez képest két komoly eltérést kell figyelembe venni:

- a szerkesztési lehetőségek erősen korlátozottak;
- a lap képe az olvasónál más lehet, mint az alkotónál.

Ezt az utóbbit kivédeni tökéletesen tulajdonképpen nem is lehet. A lapot ezért több elterjedt böngészővel, eltérő képméretű és -geometriájú gépeken és kikapcsolt grafikával is illik ellenőrizni.

Vigyázni kell a legújabb lehetőségekkel is. Ilyen most a keretek és a Java használata. Ha nem adunk alternatívát az ilyen lehetőségeket nem ismerő böngészők számára, akkor esetleg egy szép üres oldalon kívül semmi sem jelenik meg az olvasónál.

3.3 Tárolókapacitás és gépteljesítmény

Ez akkor merül fel problémaként, ha sokan használják ugyanazt a gépet, illetve nagyon nagy a lap forgalma.

3.4 Hozzáférési jogok

Ha a webszerkesztő egyben a gép rendszergazdája is, akkor bármit bevethet a rendszer teljesítőképességének határáig. Más esetben a lehetőségek eleve kizárnak egyes, általában dinamikus jellegű, változó információkat tartalmazó lapok létrehozását.

Sok esetben, ha nehézségek árán is, de van mód más megoldásokkal az eredeti, vagy ahhoz hasonló elképzelést létrehozni.

3.5 Fejlesztési eszközök

A HTML szerkesztőket inkább kezdőknek lehet ajánlani, weblapot akár egy primitív szövegszerkesztővel és egy jó képmanipulátorral is lehet készíteni.

Maga a HTML "nyelv" igen egyszerű felépítésű, így könnyen és hamar megtanulható. Igazán csak akkor ütközik nehézségbe a programozás, ha egyszerre nagy mennyiségű és nem HTML formátumú dokumentumot kell beilleszteni – ekkor konverterprogramokat kell alkalmazni.

3.6 Megbízható rendszer

A WWW mindig él és feltételezi, hogy ez minden részére igaz. A lapnak tehát olyan gépen kell "laknia", amelynek az elérhetősége a legmegbízhatóbb. Hosszú, vagy gyakori kiesések jelentősen csökkentik a látogatók számát. Az anyagról legyen mindig aktuális mentésünk, gondolva az esetleges rendszerösszeomlásokra, illetve betörésekre.

4. A lap tartalma

A weblap kiadvány. Lehet prospektusjellegű, lehet újsághoz hasonló, de lehet adatbánya is. Ami a lényeg: ott legyen rajta az az információ, amit közölni akarunk, méghozzá könnyen és logikusan kezelhető formában. Ez nem mindig sikerül. Tippek weblapkészítéshez:

- A weblapok új irányzatai a kiadványoknak
- Lehetnek igénytelen "egybeszövegek", ez alig különbözik a gépelt szövegtől.
- Az igazi netsurfer megállás nélkül átvillan az oldalak nagy részén. Kell valami, ami megállásra készíti.
- Törekedni kell arra, hogy a közölni kívánt információ felé megfelelő utat építsünk.
- Logikus linkelés, jól kitalált címek.
- Olvasó gyorsan el tudja dönteni, szüksége van-e valóban erre az információra, ha pár soros rövid leírással, összefoglalóval, esetleg kulcsszavakkal találkozik.
- Amennyiben nem leírójellegű, hanem friss információkat (is) közlő kiadványt kell létrehozni és üzemeltetni, mindenképpen gondoskodni kell a gyors frissítésről. Követni kell, melyik lap mettől-meddig érvényes.
- Hiába a látványos grafika és az érdekes anyag, ha gond van a szóhasználattal, vagy netán a nyelvhelyességgel.

A weblapon szépen, ékes, sőt ékezetes magyarsággal lehet információt közölni.

5. A lap formája

A WWW-n ugyanúgy stílusok alakultak ki, mint a művészetekben. A következő stílusokat különböztetjük meg:

5.1 Hivatalos tájékoztató stílus

- Világos, áttekinthető, egyszerű. Általában egyszínű a háttér, kevés a grafika, az oldalt főleg a dokumentumszerűen szerkesztett szöveg uralja.
- Használhatók a hivatalos iratokban és prospektusokban használt megoldások.
- Karbantartása egyszerű.
- Letöltése szűk sávszélesség esetén is gyors

5.2 Hivatalos névjegy stílus

- Egyéni weblapok közt elég elterjedt.
- Kevés, általában lényegretörő információ, név, cím, egy-két egyéb adat, szinte mindig fényképpel.
- Az ilyen lapot általában egyszer elkészítik, és utána elfelejtik. Gyakran nem is a tulajdonosa dolgozik vele, hanem megkér erre valakit, egyes intézményekben pedig "szabvány" létezik a munkatársak személyes lapjainak megjelenésére és tartalmára.
- Szokásos az ilyen lapokat önéletrajzzal és publikációs, vagy referencialistával bővíteni.

5.3 "Szia, ez vagyok én" stílus

- Ez is névjegy, azaz tulajdonosának bemutatására szolgál, de hangsúlyozottan kerül mindenfajta hivatalosságot. Jellemző a fantázia szabad csapongása és a színvonal jelentős szórása.
- A legtöbb hibás lap itt fordul elő, de az igazán jó ötletek is itt születnek meg.
- A grafika a "mórickarajztól" a háromdimenziós, tükrözőfelületekkel tarkított virtuális valóságig akármi lehet.
- A tartalom és a szerkesztettség ugyanilyen változatos.
- A tehetséges weblapszerkesztők után így, ezeket az oldalakat olvasgatva is érdemes nyomozni.

5.4 „Rulez site”-ok, BBS-ek, adatbányák

- Ezek az előző stíusból kinőtt, már meghatározott tartalommal bíró oldalak.
- Gyakori, hogy valamilyen anyag gyűjtőhelyei (szöveg, kép, hang, programok).
- Léteznek cikkek, írásokat, programozást segítő dokumentációkat, csinos lányokat, rajzfilmfigurákat, vagy madárhangokat gyűjtő helyek, ún. „site”-ok.
- Az általában sok anyagot tartalmazó website esetén a megjelenés vegyes, az első és „bevezető” lapokon igényes, míg az anyag szintjén szerényebb, állomány-jegyzékszerű (ami ez esetben nem baj).
- Érdekesen keveredik ez a stílus a hivatalossal, amikor egy cég vagy szervezet készít magáról burkolt, vagy kevésbé burkolt reklámkiadványt. Ez utóbbi lapokon gyakoriak a „trükkök”: keretek, űrlapok, java alkalmazások, mozgó (animált) gifek.

Összefoglalva: a stílusokat általában a megjelenés formája és a tartalom jellemzi (hol előbbin, hol utóbbin, hol mindkettőn ott a hangsúly), míg az anyag rendszerezettsége és áttekinthetősége a szerkesztőtől függ.

6. A HTML programozási nyelv

ASCII forráskódú, forráskódja bármilyen editorral írható. Kiterjesztése: .html, .htm.

A HTML lap szerkezete

a HTML "nyelv" szabályai szerint így néz ki:

```
<html>
<head>
Fej
</head>

<body>
Törzs
</body>
</html>
```

Mi kerül a fejrészbe?

A fejben a lapra jellemző, rövid, pár szavas leírást kell megadnunk. Ez a szöveg a böngészőprogram ablakának tetején fog megjelenni (a címsorban), illetve a hálózaton üzemelő keresőautomaták is ezt fogják a lap hálózati címe mellé írni. Meg kell tehát gondolni, mi is kerül ide! Ráadásul ékezetet sem ajánlott használni, mert nem mindegyik böngésző- és keresőprogram képes azt helyesen értelmezni.

Mi kerül a törzsbe?

Maga a lap, amit írunk. Az összes szöveg, a képek helye, miegyéb.

HTML parancsok általános felépítése

<nyitótag opció1=paraméter1 opció2=paraméter2...> formázandó szöveg <zárótag>

A tag-ek zárójelként fogják közre a formázandó szöveget. A tag-ek végrehajtásának sorrendje a megadás sorrendjébe történik.

Hibás parancsok

Ha egy tag-et, vagy annak valamilyen opcióját hibás módon adjuk meg, akkor a böngésző nem hajtja végre a parancsot.

HTML oldal címének megadása

A HEAD részbe a <TITLE></TITLE> tag-ek között adjuk meg.

A TITLE elemet arra használjuk, hogy azonosítsuk a dokumentum tartalmát. Mindig pontos, egyértelmű cím, a felhasználó ez alapján tudja eldönteni, hogy a dokumentum tartalma érdekes-e a számára.

A TITLE elem tartalma a *böngésző program* ablakának címsorában jelenik meg.

Weblap szöveg tagolása

<PRE></PRE> Ahogy a forráskódban tagoljuk a szöveget, úgy jelenik meg a böngészőben.

 sortörés

<P></P> bekezdés megadása

opciói: [ALIGN= {left, center, right, justify}] alapértelmezés: left – bekezdés igazítása
Egérkezelő függvények (később)

[ID=NÉV] Az attribútum egy nevet rendel az adott elemhez, ennek az egész dokumentumban egyedinek kell lennie. Kisbetű/nagybetű érzékeny. Hypertext (hivatkozás) célhorgonya.

Ékezetes betűk és speciális karakterek

PC-s magyar Windows-os fájlkészítés esetén nem feltétlenül szükséges rendelkezni az alábbi ismeretekkel, de minden más esetben igen. (Gondoljunk a Mac-ekre az NT-kre a DOS-ra UNIX-ra és külföldi barátainkra). (Külföldi böngészők, és az op.rsz-ek nem ismerik a magyar ékezeteket.) A kis "á" és nagy "Á" úgy állítható elő, hogy az "a" illetve az "A" betűket az ´ szimbólummal kombináljuk (acute az adott ékezet angol neve), ugyanígy készül az "é", "É", "í", "Í", "ó", "Ó", "ú", "Ú".

á= á	Á= Á
é= é	É= É
í= í	Í= Í
ó= ó	Ó= Ó
ú= ú	Ú= Ú

Az "ö", "Ö", "ü", "Ü" előállításakor az ¨ szimbólumot kell alkalmazni.

ö= ö	Ö= Ö
ü= ü	Ü= Ü

Az "ő", "Ő" csak megfelelő karakterkészlet (pl. CE kiterjesztésű fontkészlet) választása esetén lesz "szép". Előállításakor a ˜ szimbólumot kell alkalmazni.

õ= ő	Õ= Ő
-------------	-------------

Azt gondolhatnánk, hogy az "ű", "Ű" ugyanígy készül, de sajnos nem. Ez esetben a ˆ szimbólumot kell alkalmazni.

û= ű	Û= Ű
------------	------------

Természetesen más (Amerikából nézve) egzotikus nyelv speciális karakterei is előállíthatók, amivel nem foglalkozunk most. Nem tűnik nagy igénynek hogy pl. a "<" (lt= less than = kisebb mint) vagy a ">" (gt = greater than= nagyobb mint) szimbólumokat előállítsuk, de ha ezeket begépeljük parancssorként fogja a böngésző értelmezni és nem jelenik meg a

képernyőn. Ebben az esetben az "<", ">" szimbólumok alkalmazása segít. Jó-jó! De hogy állítható elő a "&" szimbólum, hisz a fentiek értelmében az is speciális karakter. Ekkor az "&" (angolul ampersand=és) jelsort kell alkalmaznunk, az idézőjelhez a """ (quotation mark=idézőjel) szimbólumot kell begépelnünk.

<= < >= >
&= & "="<>

Alapvető betűstílusok

 félkövér betűk
<I></I> dőlt betűk
<U></U> aláhúzott betűk
 alsóindex
 felsőindex

Fejléc típusú szöveg

<H1></H1> ...
... <H6></H6>

A fejrész megjelenítésére, a törzsszövegtől való elkülönítésére használják, a <H1> a legnagyobb betűméretű, és a <H6> a legkisebb betűméretű.

Opció: [ID=név] hivatkozás célhorgonya

Szöveg formázása

 A szöveg megjelenését formálhatjuk.

Opciói: [SIZE=érték] Betűméret (alapérték:3) megadható abszolút (pl.:size=4), vagy relatív értékkel(pl.:size=+1 alapértékhez adódik)

[COLOR={betűvel, vagy #hatjegyű hexa számmal}] A tag-ek közötti szöveg színét állíthatjuk be. Pl.: COLOR= yellow, vagy COLOR="#FFFF00".

[FACE=betűtípus] A Tag-ek közötti szöveg betűtípusát adhatjuk meg. Kívánatos a TrueType fontok megadása!

Háttérszín, betűszín, háttérkép

- Megváltozik egy oldal arculata, ha megváltoztatjuk a háttérszínét.
- A színekre vonatkozó utasítások közül kevés van, ami a böngészők szinte mindegyikével alkalmazható.
- A háttérszín, háttérképet a dokumentum bevezető részében kell definiálnunk.

<BODY> opciójaként: bgcolor={szín betűvel, vagy „hatjegyű hexa számmal”} háttérszín

background= elérési út/filenév.kit Háttérképet szúrhatunk be

A kép mozaikszerűen jelenik meg (nem méretezhető) megjeleníthető kép kiterjesztése: gif, jpg vagy bmp

text={szín betűvel, vagy „hatjegyű hexa számmal”} A teljes törzsszöveg alapszínét állíthatjuk be.

Listaszöveg készítése

A listakészítéshez több eszköz áll rendelkezésünkre.

- Rendezés nélküli list (Unordered List):

```
<ul>  
<li>első elem</li>  
<li>második elem </li>  
<li>harmadik elem</li>  
</ul>
```

Az jelzi, hogy rendezetlen lista (ul = unordered list) kezdődik, az a lista egy elemét (li = list item) jelzi, az adott elemet tag-gel kell lezárni, a rendezetlen listát egy zárja le. A nem számozott listában az elemeket fekete pont vezeti be. Az tag-be írva a TYPE opciót ezt megváltoztathatjuk kitöltött négyzetre (TYPE="square"), vagy üres karikára (TYPE="circle")

- Rendezett listánál az helyett (ol = ordered list) kerül. A számozott listában az elemeket számok vezeti be. A CONTINUE opciót beírva, egy már létező rendezett lista számozását folytathatjuk.
- Definíciós listát is készíthetünk, ekkor rövid fogalmakat magyarázunk meg. A definíciós lista a <dl> parancs (dl = definition list) és a </dl> parancs közé kerül:

```
<dl>
<dt>fogalom</dt>
  <dd>az első fogalom magyarázata</dd>
<dt>fogalom</dt>
  <dd>a második fogalom magyarázata</dd>
</dl>
```

A <dt> a fogalom (dt = definition term) elé kerül, míg a <dd> a fogalom leírása elé. A definíciós listát a fogalom és a definíció közötti bekezdés és behúzás jeleníti meg.

Táblázatok készítése

Tábla definiálása: <TABLE></TABLE> opciói: ALIGN={left|center|right} – igazítás
BGCOLOR={szín betűvel, vagy „hatjegyű hexa számmal”} – tábla háttérszíne
BORDER=érték – szegély szélessége
WIDTH={érték vagy százalékos érték} – a tábla szélessége. A százalékos érték a képernyő méretéhez viszonyított érték.
CELLPADDING=érték – a cellahatár és az adatok távolságát adja. Alapérték:0-2
CELLSPACING=érték – a cellák távolságát jelöli. Alapérték: 0-1 között

Táblázat elkészítése:

Táblázat sorainak megadása: <TR></TR>(table rows)

Táblázat celláinak megadása: <TD></TD> (table data)

A parancsok záró tag-jeinek megadása nem kötelező!

Opciói: ALIGN={left|center|right} – vízszintes igazítás

VALIGN= {top|middle|bottom} – függőleges igazítás

(TD-nél) WIDTH=érték – a cella szélességét adja meg pixelben

Az opciók TR-nél a teljes sorra vonatkoznak, TD-nél csak az adott cellára.

A tábla első sorába félkövér, középre igazított fejrészt tehetünk, ha celladefiníciónak a <TD> </TD> helyett <TH></TH> tag-eket használjuk.

Tábla címének megadása: TABLE tag után a <CAPTION></CAPTION> tag-ek közötti szöveg adja meg. Megjelenhet a tábla fölött (alapértelmezés), vagy a tábla alatt az ALIGN=bottom opció megadásával.

Cellák

egyesítése:

A táblázat néhány celláját egyesíteni szeretnénk vízszintesen vagy függőlegesen. Erre célra a <TD> tag colspan=érték és a rowspan=érték opcióit kell megváltoztatnunk. (colspan=column span=oszlopfeszítés, rowspan=row span=sorofeszítés).

Ezen változók értéke alaphelyzetben 1. Amennyiben más értéket állítunk be, ennek hatására az adott cella „érték” számú cella helyét foglalja el. Ezek a definiált cellától jobbra, vagy lefelé helyezkednek el.

Képek beillesztése

Általános alakja:

Az IMG elem egy képet ágyaz be a dokumentumba az elem definíciójának a helyére, tartalmi része nincsen, hanem helyette az SRC attribútum által mutatott kép kerül kirajzolásra.

Kép kiterjesztése .gif, .jpg, (.bmp) lehet tömörítettségük miatt.

Opciói a következők lehetnek:

ALT="szöveg" – nevet adhatunk a képünknek. Ez akkor hasznos, ha valaki úgy böngész az oldalunkat, hogy nem nézi a képeket

WIDTH="érték" – a kép szélességét adja meg (pixelben vagy százalékban)

HEIGHT="érték" – a kép magasságát határozza meg (pixelben vagy százalékban)

Érdemes őket akkor is megadni, ha nem akarunk méretet változtatni, mert így a böngésző előre meghatározhatja helyüket és gyorsabbá válik a letöltés.

ALIGN={top|center|bottom|left|right}

Hivatkozások készítése:

A HTML dokumentumok leghasznosabb tulajdonsága a hiperlinkek adta gazdag kapcsolatrendszer. A hiperlinkekre

 SZÖVEG

tipusú formula begépelésével hivatkozhatunk. A "KAPCSOLAT" kifejezésbe írva kell meghatároznunk, hogy a "SZÖVEG" mezőre rákattintva milyen kapcsolat (levél, másik HTML dokumentum, fájl, stb) jöjjön létre. (A "href" a hyperlink reference – hiperkapcsolat referencia rövidítése)

Minden A (anchor = horgony) elem egy horgonyt határoz meg.

1. Az A elem tartalma határozza meg a horgony pozícióját.
2. A NAME attribútum a célhorgony azonosítóját adja meg nulla vagy több link számára.
3. A HREF attribútum a starthorgonyt definiálja egyetlen link számára.

A dokumentumok szerzője olyan linket is létrehozhat, amely nem határoz meg horgonyt (vagyis nincsen megadva a HREF, NAME vagy ID attribútuma). Ezen attribútumok értékeit később a scriptek segítségével be lehet állítani.

A vizuális megjelenítők a linkeket aláhúzással ill. a mutatóeszköz kurzorának megváltozásával jelzik. A link megjelenése attól függően is változik, hogy a felhasználó meglátogatta-e a linket vagy se.

A horgony definiálása után a dokumentum szerzője más dokumentumokból linket irányíthat erre a horgonyra. A hivatkozás lehet relatív vagy abszolút.

A linkek megadása során a következőkre kell figyelni:

- A horgonynevek legyenek egyediek az adott dokumentumon belül. Az olyan nevek használata, amelyek csak a kisbetű/nagybetű használatban térnek el egymástól, nem ajánlott.
- A horgonynevekben lehetőleg csak ASCII karaktereket használjunk.
- A LINK és A elemeket nem szabad egymásba ágyazni!
- Használjuk az ID attribútumot, ha egy elem kezdetéhez szeretnénk ugrani.
- Nem szabad ugyanolyan értékű ID attribútumot adni elemnek és ugyanolyan nevű célhorgonyt megadni egy dokumentumon belül!

A hivatkozások (linkek) színét megadhatjuk a <BODY> tag opciójaként a LINK="szín" (vagy a szín angol neve, vagy a színkód hexa értéke) az oldalon kezdetben megjelenő linknek ad színt. A VLINK="szín" a legalább egyszer már felkeresett hivatkozás színét állítja be. Míg az ALINK="szín" az aktivált link színét adja.

Térképek készítése:

Térképek: Azok a képek, melyek különböző részeire kattintva különböző oldalakra ugorhatunk.

```
<IMG SRC="kep.jpg" USEMAP="#terkep1">
```

A fenti parancsban a kep.jpg nevű képre fogunk kattintani, de a térképutasítások a terkep.htm fájlban a terkep1 nevű térkép-utasításban helyezkednek el.

Tehát az IMG tag egy USEMAP opcióval bővül, ami hozzárendeli a képhez a területek definícióját.

- [USEMAP=#mapnév] Az opció az adott elemhez egy image map-et rendel. Az image map NAME attribútumának meg kell egyeznie a USEMAP attribútum értékével.

```
<MAP NAME="terkep1">  
  <AREA [SHAPE="alak1"] COORDS="x,y,..." [HREF="hivatkozás1"] >  
  <AREA [SHAPE="alak2"] COORDS="x,y,..." [HREF="hivatkozás2"] >  
  <AREA [SHAPE="alak3"] COORDS="x,y,..." [HREF="hivatkozás3"] >  
</MAP>
```

Opciók definíciói:

[NAME=SZÖVEG] A MAP elem által meghatározott MAP-et névvel látja el.

[SHAPE=DEFAULT|RECT|CIRCLE|POLY] Egy régió alakját határozza meg. A lehetséges értékek jelentése:

default

- az egész területet jelenti,

rect

- egy négyszögalakú területet jelöl,

circle

- egy köralakú területet jelöl,

poly

- egy poligon alakú területet jelöl.

- [COORDS=KOORDINÁTÁK] Ez az opció határozza meg a régió pozícióját a képernyőn. A megadott koordináták száma és sorrendje az alaktól függ:

rect

- bal-x, felső-y, jobb-x, alsó-y (azaz bal felső sarok és jobb alsó sarok koordinátája)

circle

- középpont-x, középpont-y, sugár. Ha a sugár értéke százalékban van megadva, akkor az image map-hez hozzárendelt objektum szélességéből és hosszúságából adódik a tényleges méret.

poly

- x1, y1, x2, y2, ... , xN, yN. (az összes csúcs x és y koordinátája egymás után, vesszővel elválasztva)

A koordináták a kép bal felső sarkához relatívak, minden érték pixelben van megadva és az értékeket egymástól vesszővel választjuk el.

- [NOREF] Ha ez a logikai opció meg van adva, akkor az azt jelenti, hogy az adott régióhoz nincsen link rendelve.

Keretek (Frame)

A HTML frame-ek lehetővé teszik, hogy a szerző egyszerre több dokumentumot jelenítsen meg egymástól független ablakokban vagy ablakrészekben. Ez lehetővé teszi azt, hogy míg egy nézetben állandó információ van, addig másokban görgetés vagy más információk megjelenítése zajlik.

Az olyan HTML dokumentum szerkezete, amely leírja a frame elrendezést, különbözik a hagyományos HTML dokumentumok szerkezetétől. Egy normál HTML dokumentum egy HEAD és egy BODY elemet tartalmaz. Egy frame-leíró dokumentumban a BODY előtt helyett a FRAMESET elem szerepel.

A FRAMESET elem által közrefogott rész a felhasználói program fő ablakában lévő frame-eket írja le. A FRAMESET elem tartalmazhatja a NOFRAMES elemet, amellyel a frame-eket meg nem jelenítő programok számára lehet helyettesítő információkat megadni.

A BODY elembe elhelyezhető elemeket TILOS az első FRAMESET elem elé írni, ugyanis ekkor a FRAMESET elemet a programok nem veszik figyelembe.

FRAME-dokumentum felépítése

```
<HTML>
<HEAD>
</HEAD>
<FRAMESET rows="50%,50%" border=1>
  <FRAME SRC="html1.htm">
  <FRAME SRC="html2.htm">
</FRAMESET>
</HTML>
```

FRAMESET elem opciói:

```
<FRAMESET ROWS=HOSSZOK COLS=HOSSZOK BORDER=ÉRTÉK>
```

...

```
</FRAMESET>
```

- [ROWS=HOSSZLISTA] Ez az attribútum a vízszintes frame-ek méreteit adja meg. Értéke, egymástól vesszővel elválasztott hosszértékek pixelekb, százalékban vagy relatívan megadva. Az alapértelmezése 100%, ami egy egész sort jelent.
- [COLS=HOSSZLISTA] Ez az attribútum a függőleges frame-ek méreteit adja meg. Értéke: egymástól vesszővel elválasztott hosszértékek pixelekb, százalékban vagy relatívan megadva. Az alapértelmezése 100%, ami egy egész oszlopot jelent.
- [BORDER=ÉRTÉK] Ez az attribútum a frame keretéről ad információt. A lehetséges értékek jelentése:

1 (vagy más érték) - Ez az érték utasítja a felhasználói programot, hogy frame és szomszédos frame-ek közé keretet rajzoljon, alapértelmezésű érték. Ha nagyobb az értéke, akkor vastagabb lesz ez a keret.

0 - A felhasználói program nem rajzol keretet az adott frame és szomszédjai közé, kivéve, ha az valamelyik szomszédnál meg van adva.

A FRAMESET elem a főablak négyzet alakú területekre való felosztását határozza meg. A frame-eknél az oszlopok balról jobbra, a sorok fentről lefelé jönnek létre.

FRAME elem opciói:

```
<FRAME NAME=NÉV SRC=URL MARGINWIDTH=PIXELEK MARGINHEIGHT=PIXELEK
NORESIZE SCROLLING=GÖRGETÉS>
```

Opciók leírása:

- [NAME=SZÖVEG] Ez az opció egy nevet rendel az adott frame-hez, ezt a linkeknél lehet felhasználni.
- [SRC=URL] Ez az opció a frame kezdeti tartalmának helyét adja meg.
- [NORESIZE] Ha ez a logikai opciót megadjuk, akkor a frame ablaka nem méretezhető.
- [SCROLLING=AUTO|YES|NO] Ez az opció a görgetési információt határozza meg a frame számára. A lehetséges értékek jelentései:

auto

- Ez az érték azt jelzi a felhasználói program számára, hogy csak a szükséges esetben bocsásson görgetőeszközöket a felhasználó rendelkezésére, alapértelmezésű érték.

yes

- Az érték hatására mindig megjelennek görgetősorok.

no

- Az érték hatására soha nem jelennek meg görgetősorok a frame-ben.

[FRAMEBORDER=1|0]

- [MARGINWIDTH=PIXELEK] Az érték a frame tartalma és a bal és jobb margó közti távolságot határozza meg, értékének nagyobbak kell lennie mint egy. Alapértelmezésű értéke a felhasználói programtól függ.
- [MARGINHEIGHT=PIXELEK] Az érték a frame tartalma és az alsó és felső margó közti távolságot adja meg, az értékének nagyobbak kell lennie mint egy. Alapértelmezésű értéke a felhasználói programtól függ.

Az SRC attribútum azt a dokumentumot határozza meg, amely a frame kezdeti tartalmát írja le. A következő példában minden frame-be más dokumentum töltődik:

```
<FRAMESET cols="25%,50%,25%">
  <FRAMESET rows="*,200">
    <FRAME name="balfelso" src="frame1_tartalom.html" scrolling="no">
    <FRAME name="balalso" src="frame2_tartalom.gif"
      marginwidth="10" marginheight="15" >
  </FRAMESET>
  <FRAME name="kozepso" src="frame3_tartalom.html" frameborder="0">
  <FRAME name="jobboldal" src="frame4_tartalom.html" frameborder="0">
</FRAMESET>
</HTML>
```

Cél-frame megadása:

Cél-frame-t a hivatkozás definíciójában adhatok meg a TARGET opcióval

- [TARGET=FRAME_CÉL] Ez az attribútum annak a frame-nek a nevét adja meg, amelyekben a dokumentumot meg kell nyitni.

Ezzel az opcióval olyan frame-ekre tudunk hivatkozni, amelyekhez a NAME attribútum segítségével nevet rendeltünk. A TARGET attribútumot olyan elemeknél állíthatjuk be, amelyek linkeket hoznak létre, image map-eknél és form-oknál.

Űrlapok (Form)

Az űrlap (form) segítségével adatokat továbbíthatunk. Az űrlap kitöltése önmagában nem elegendő, az adott szerveren egy programnak (többnyire cgi-scriptnek) kell kezelnie a kitöltött űrlapot. Ezen ok miatt mi csak azzal foglalkozunk, hogyan lehet kitölteni egy űrlapot, használni valaki csak akkor tudja, ha gépének rendszergazdájával konzultál (vagy ha elküldi magának emailbe).

Az űrlapot a <form> </form> jelek közé kell tennünk. Elsőként tisztáznunk kell a módszert (method), ahogy a szerver kezeli az űrlapot. A POST (postázás a leggyakoribb), az ACTION

(akció) azt a programot jelzi, amelyik az űrlapot fogja feldolgozni, vagy azt az email címet, melyre az űrlapot küldhetjük..

egyszerű adatbeviteli mező: az <input> jelek közé helyezzük. Meg kell adnunk a nevét (name), méretét (size), és még megadható a mezőbe írható karakterek maximális számát (maxlength), kezdőértékét (value). <INPUT NAME="nev" SIZE="20" MAXLENGTH=15 TYPE="text" VALUE="Ide írj!">

Az INPUT elem segítségével a következő űrlapelemeket lehet előállítani a TYPE opció értéke szerint:

text - Egy egyszerű egysoros szövegbeviteli mezőt hoz létre
password - Hasonló mint a "text", de a beírt szöveget "*" karakterekkel helyettesíti a kijelzésnél, ezzel megakadályozva az esetleges szemléltető a jelszó elolvasásában .
checkbox - Egy checkboxot hoz létre (legördülő lista)
radio - Egy rádiógombot hoz létre (választókapcsoló)
submit - Egy *submit* gombot hoz létre. (elküldés)
image - Egy grafikus *submit* gombot hoz létre. Az SRC attribútum értéke határozza meg azt a képet, amely a gombot fogja dekorálni. Tekintetbe véve a fogyatékos felhasználókat, használjuk az ALT attribútumot is!

Ha egy mutatóeszközt használunk a gomb aktiválására, akkor nemcsak a form adatai kerülnek elküldésre, hanem a mutatóeszköz koordinátái is a kép bal felső sarkától számítva a "gomb_név.x=x_érték" és "gomb_név.y=y_érték" formában.

Ha a feldolgozóprogram egyes területekhez különböző tevékenységeket rendel, akkor ez hátrányosan érinti a nem grafikus böngészővel rendelkező felhasználókat. Ezen okból a következő alternatív megoldásokat részesítsük előnyben:

- Használjunk több pozícionált *submit* gombot, vagy
- használjunk kliens oldali image map-eket a scriptekkel egyetemben!

reset - Egy *reset* gombot hoz létre. (alapérték visszaállítására)

button - Egy nyomógombot hoz létre. A gomb feliratát a VALUE attributum adja meg.

hidden- Egy nem megjelenő kontrollt hoz létre.

file - Egy fájlkiválasztó kontrollt hoz létre. Az alapértelmezett fájlnevet a VALUE attribútumnál lehet beállítani.

Szövegterület kialakítása:

Hosszabb szövegek bevitelére az input helyett a TEXTAREA tag-et használjuk. Ezzel kisebb memoritereket írhatunk.

```
<TEXTAREA NAME=NÉV ROWS=SOROK COLS=OSZLOPOK DISABLED READONLY >  
Textarea kezdet értéke </TEXTAREA>
```

Attribútumok definíciói

- [NAME=NÉV] Megadja a kontroll nevét.
- [ROWS=SZÁM] Megadja az egyszerre látható szövegsorok számát. A felhasználó természetesen több sort is beírhat ennél az értéknél és ekkor görgetőeszközök jelennek meg a kontroll szélein.
- [COLS=SZÁM] Megadja az egy sorban látható átlagos szélességű karakterek számát. A felhasználó természetesen hosszabb sorokat is begépelhet és esetleg görgetőeszköz jelenik meg vagy a program szóhatáron megtöri a sort.

Választóelemek készítése:

Kiválasztó elem definiálására használható. Ezek közé tesszük a választási lehetőségeket.

```
<SELECT NAME=NÉV SIZE=MÉRET MULTIPLE DISABLED > ... </SELECT>
```

Attribútumok definíciói

- [NAME=SZÖVEG] Ez az attribútum egy nevet rendel a kontrollhoz.
- [SIZE=SZÁM] Ha a SELECT elem egy görgethető *listboxként* jelenik meg, akkor ez az érték a listában egyszerre látható elemek számát adja meg.
- [MULTIPLE] Ez a logikai attribútum megengedi vagy megtiltja a listában egyszerre több elem kiválasztását.

Választási lehetőségek megadása:

A SELECT elem egy menüt hoz létre. Minden választási lehetőséget egy OPTION elem valósít meg és legalább egy darabnak lennie kell.

Az OPTION elem SELECTED attribútuma (lásd később) segítségével előre egy vagy több (MULTIPLE attribútum!) listaelemet is kiválaszthatunk.

```
<OPTION SELECTED DISABLED LABEL=CÍMKE VALUE=ÉRTÉK> ... </OPTION>  
<OPTGROUP DISABLED LABEL=CÍMKE> ... </OPTGROUP>
```

Attribútumok definíciói

- [LABEL=SZÖVEG] Ez az attribútum határozza meg az OPTGROUP elem esetén a csoport címkéjét, az OPTION elemnél pedig ha meg van adva, akkor a listaelem (rövidebb) címkéjét az OPTION elem (hosszabb) tartalma helyett.
- [SELECTED] Azt jelzi, hogy az adott opció előre ki van jelölve.
- [VALUE=SZÖVEG] Az OPTION elem kezdeti értékét állítja be, ha nincsen megadva, akkor az OPTION elem tartalma lesz az.

Javascript alapismeretek

A Javascript a WWW-ért született. Jelenleg nem is létezik más implementációja, mint ami a böngészőkben és webservereken fut.

Javascript két helyen futhat: kliensen vagy szerveren.

Témánk néhány kilensoldali programrész lesz. A szerveroldali rész szintaxisa ugyan megegyezik, de programozói interfész (API) teljesen más.

Kliens oldali Javascriptnál a programok forrása a HTML oldalban van. Általában rövid kódok, amelyeket a böngésző értelmez.

A Javascript és a C szintaxisa nagyon hasonlít egymásra, méginkább a Java nyelvé (ami a C++ nyelv egyszerűsített változata) viszont a Javascript nem a Java-ból fejlődött ki, hanem különálló script-nyelvként (beépülő nyelv) fejlesztette a Netscape ún. Livescript nyelvet. A Javat a Sun Microsystems fejlesztette Netrunner néven mikora Netscape úgy döntött neki szüksége van erre a rendszerre, mert sokkal több mindent lehetett vele csinálni mint a Livescript-tel. Így a Netscape 2.0-ba bekerült mind a kettő Javascript néven.

Tehát: JAVA ≠ Javascript

Scriptek használata

A HTML oldal BODY részében a <SCRIPT></SCRIPT> Tag-ben adjuk meg a különböző beépülő nyelvek kódjait. Az adott nyelvet a language opció után adhatjuk meg. Ha nem adunk meg nyelvet, akkor alapértelmezés szerint Javascript nyelv a beállított.

Megadás: <SCRIPT LANGUAGE="VBscript">

...
</SCRIPT>

<SCRIPT LANGUAGE="Javascript 1.1">

...
</SCRIPT>

<SCRIPT LANGUAGE="Javascript">

...
</SCRIPT>

Egy sor kiírása Javascript-tel:

<HTML>

<BODY>

Ez itt egy hagyományos HTML dokumentum.

<SCRIPT LANGUAGE="JavaScript">

document.write("Ez már JavaScriptben íródott!!!
")

</SCRIPT>

Ez a sor újra HTML!

</BODY>

</HTML>

Háttérszín váltása:

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function sarga(){
  document.bgColor="#FFFF00";
}
function kek(){
  document.bgColor="#00FFFF";
}
function eredeti(){
  document.bgColor="#FFFFFF";
}
</SCRIPT>

<P ALIGN="center">
<FORM>
<INPUT TYPE="button" VALUE="Sárga háttér" onClick="sarga()">
<INPUT TYPE="button" VALUE="Kék háttér" onClick="kek()">
<INPUT TYPE="button" VALUE="Eredeti háttér" onClick="eredeti()">
</FORM>
</P>
</BODY>
```

Üdvözlét kiírása beírt névvel:

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function Udvozol(){
  var name = document.forms[0].elements[0].value;
  alert("Szia "+name);
}
</SCRIPT>
<FORM NAME="urlap"> Név: <INPUT TYPE="text" NAME="nev">
<INPUT TYPE="button" VALUE="Írd be a neved, majd kattints rám!"
onClick="Udvozol()">
</FORM>
</BODY>
```

Új, üres ablak nyitása:

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function ujablak2(){
  ablak = open("pelda.htm", "uj_ablak",
  "width=300,height=200,status=no,menubar=no");
}
</SCRIPT>
<FORM>
<INPUT TYPE="button" VALUE="Új ablak" onClick="ujablak2()">
</FORM>
</BODY>
```

Új ablak szöveggel:

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function ujablak3(){
  ujablak = open("", "uj_ablak3",
    "width=400,height=300,status=yes,toolbar=yes,menubar=yes");
  ujablak.document.open();
  ujablak.document.write("<HTML><HEAD><TITLE>Dokumentum készítése JavaScript-
tel</TITLE></HEAD>");
  ujablak.document.write("<BODY><CENTER><H1>");
  ujablak.document.write("Ezt az oldalt teljesen a JavaScript ");
  ujablak.document.write("hozta létre!");
  ujablak.document.write("</H1></CENTER></BODY>");
  ujablak.document.write("</HTML>");
  ujablak.document.close();
}
function bezár(){
  close();
}
</SCRIPT>
<P ALIGN="center">
<FORM>
<INPUT TYPE="button" VALUE="Dokumentum készítése JavaScript-tel"
onClick="ujablak3()">
<INPUT TYPE="button" VALUE="Ablak bezárása" onClick="bezár()">
</FORM>
</P>
</BODY>
```

Köszönés napszaknak megfelelően:

```
<BODY>
<SCRIPT LANGUAGE="JavaScript">
  var ora=(new Date()).getHours();
  if (ora<=6 || ora>=20) {
    document.writeln("Jó estét!");}
  else if (ora<10){
    document.writeln("Jó reggelt!");}
  else {
    document.writeln("Jó napot!");}
</SCRIPT>
</BODY>
```