

1 BEVEZETÉS

Kellene egy honlap, mert már minden ismerősnek van. De mégis, mi az?

A honlapot és az elkészítéséhez szükséges segédeszközöket először a CERN-ben dolgozták ki. Miért pont az atomtudósok környékén született meg ez a példátlanul népszerű és megállíthatatlanul terjedő eszköz? Ez egy kicsit a véletlen műve.

Valamikor az ősidőkben (vagyis pár éve – a számítástechnikában ez nagy idő) már megalkották a Gophert, mint hatékony és könnyen kezelhető információs rendszert. El is terjedt. Végre elég volt csupán pár billentyű kezelését megtanulni ahhoz, hogy az ember a világ bármely pontjáról információhoz jusson, áttekinthetően, gyorsan. Ekkor már az Internet egységes eszként jelent meg a felhasználó előtt. Az, hogy a kért információ milyen gépen, milyen operációs rendszer alatt és a világ melyik pontján van (az átviteli sebességet leszámítva) érdektelenné vált. Gondoljuk meg, mekkora jelentősége van ennek! Az egész világot "átláthatjuk", nincsenek határok többé!

A CERN tudósai éppoly lusták és válogatósak lehettek, mint bárki más, csak sokkal igényesebbek. A Gophert annak idején Minnesota egyetemén karakteres képernyőkre álmodták meg és csak "unalmas" menüket tartalmazott. Az atomfizikusok meg ábrákat és képeket követeltek. Meg dőlt betűt. S mindenekfelett szívükhöz nőtt a hipertext. Megkapták.

A honlapok a számukra kidolgozott kommunikációs szabványok szerint "közlekednek" az Interneten és ezt értik a **World Wide Web** böngészőprogramok. A lapok szerkesztő- és kiegészítőjelekkel (leírókkal) ellátott szövegállományok. A leírók szabványa a **HTML** (HyperText Markup Language). A HTML egyfajta leírónyelv. Vele a lap megjelenését tudjuk "programozni".

2 MI KELL A HONLAPHOZ?

Először is egy

2.1 GÉP

Ha nekifogunk a honlap készítésének, először is keresnünk kell egy gépet, amin majd dolgozni fogunk. A lap építéséhez megfelel egy "mezei" PC is, kipróbálásához már egy Linuxos vagy Windowsos Netscape, vagy hasonló böngészőprogram szükséges. Amennyiben a lapunkat őrző géppel csak rövid ideig tudunk dolgozni, pl. azért, mert telefonon át tartjuk rendszeresen a kapcsolatot, akkor érdemesebb az otthoni gépen megírni és kipróbálni a lapo(ka)t, majd a kész változatot felküldeni. Ahhoz ugyanis, hogy mások is megtekinthessék a lapunkat

2.2 INTERNET KAPCSOLAT

szükséges. Ezt minden guru **host** néven emlegeti, ez annyit tesz, mint házigazda, s valóban, nála lakik majd a lapunk. Ennek a gépnek az Interneten ismert címe, sőt szinte mindig neve is van. Képzletbeli hostunk hallgasson a www.honlapiskola.hu

névre! Ez a gép tehát rajta van a Neten, méghozzá a nap mind a huszonnégy órájában. Kerüljük el a korlátozott nyitvatartással üzemelő hostokat, mert a Net az egész bolygót magában foglalja és mindig ébren van. Szerény lapunkat pedig Tokiótól Washingtonig bárhonnán meglátogathatják, mert nincsenek távolságok ebben a furcsa világban.

Ahhoz, hogy dolgozni tudjunk, be kell jutnunk a hostra. Számos host biztosít anonim, nyilvános hozzáférést, bizonyos szolgáltatások használatára. Ezzel akár az Internet bármely szegletébe is elvitorlázhatunk, saját lap alapításához azonban nem elég. Kell egy

2.3 ACCOUNT

is, amit postafióknak, hozzáférésnek is próbálnak fordítani, de nem egészen találó. Lényegében arról van szó, hogy a host személyesen ismerjen minket és biztosítson számunkra egy zugot, ahová virtuális ingóságainkkal beköltözhetünk.

Az account megszerzésének rejtelmerei és számos fondorlata romantikus kalandregény témája lehetne, mi most úgy teszünk, mint a matematikusok: *"Vegyünk egy accountot!"*. Az accounttal együtt jár egy **user-név** és egy **password** is. (A password mindig legyen nehezen kitalálható és legalább havonta változtassuk! Az ördög és a hackerek sosem alszanak!)

Tegyük fel, nekünk a Jakab név jutott. Jelentkezzünk be! Valami hasonlót látunk majd:

Welcome at www.honlapiskola.hu

login: jakab

password: *****

Ha ezek után a gép morc módon ki nem dobott minket, akkor valóban van élő accountunk. Ide akár levelet is kaphatunk, jegyezzük is meg a címünket:

jakab@honlapiskola.hu

Kérjünk meg egy ismerőst, aki nem erre a gépre kapott accountot, hogy küldjön a címünkre levelet! Ha megkapjuk, a host Internet elérése rendben van. Ezenkívül futnia kell egy

2.4 HTTPD

programnak is. Ezt a host korlátlan ura és parancsolója, a **root** (rendszergazda) helyezi üzembe (azaz mi ezzel ne próbálkozzunk) és ha nem értünk igazán a host **operációs**

rendszeréhez, akkor a legegyszerűbb ellenőrzési mód a host kezdőlapjának hívása. Az általunk használt böngészőbe írjuk be új címként:

```
http://www.honlapiskola.hu
```

Ha erre megjelenik valami szépséges oldal, nyert ügyünk van (majdnem). Ha nem, akkor érdeklődjünk a hoston más userektől, végső esetben a roottól, valaki csak fogja tudni, miért nem működik. Lehet, hogy a httpd program nem fut, de az is lehet, hogy nincs kezdőlap (az elég szomorú) vagy más címen van (nem "www" a host neve vagy nem a szokásos "index.html" a kezdőlap címe).

Ideje csinálni valamit, ha már bejelentkeztünk! Kezdjük az első kapavágással. Hozzuk létre azt a helyet, ahova a lap kerül! Ez

2.5 A PUBLIC_HTML KÖNYVTÁR

vagy directory, vagy folder, kinek hogy tetszik. Egy parancs az egész:
mkdir public_html

Ellenőrizni kell az

2.6 ELÉRÉSI JOGOK

néven ismert szörnyüségét. Arról van szó, hogy a sokak által használt és a hálózatba kapcsolt gépeken ésszerűen korlátozva van, ki mit írhat és olvashat, nehogy vicces kedvű t-bojok mások féltve őrzött állományaiba kotorjanak bele.

Mit is szeretnénk a lapunkkal? Azt, hogy mindenki olvassa, aki csak erre jár. Azt viszont nem szeretnénk, ha át is írják, nem azért vesződünk vele. Tehát írni csak mi írhatjuk. Adjuk ki ezt a UNIX parancsot:

```
ls -al
```

Válaszként valami ilyet kell kapnunk:

```
total 17
drwxr-xr-x  4 jakab  users    1024 Aug 11 00:02 ./
drwxr-xr-x  8 jakab  users    1024 Jul 24 15:26 ../
drwx----- 4 jakab  users     47 Aug 11 00:04 mail/
drwxr-xr-x  4 jakab  users     3 Aug 11 00:05 public_html/
[...]
```

A lényeg az, hogy a három darab **r** és **x** ott legyen elszórva. Ha ez nincs így, akkor újabb parancsot kell adnunk:

```
chmod 755 public_html
```

Az index.html -- és minden "normális" állomány -- más értéket kap (nem tartalmaz x-eket):
chmod 644 index.html

Ha nem vagyunk biztosak a dolgunkban, akkor is kiadhatjuk, baj nem származik belőle. Ha viszont nincs helyesen beállítva, akkor bár ott van a lap vagy a kép, mégis láthatatlan marad. A beágyazott kép helyett egy kérdőjel, vagy valami egyéb jelzés jelenik meg.

Megjegyzem, azért bátorodom itt mindig a UNIX operációs rendszert emlegetni, mert a hostok döntő többsége ezzel, vagy ennek valamely elfajzott rokonával üzemel, alapszinten tehát érteniük kell a "szóból"...

Kész a "telek", kezdhetjük építeni a házat. Először is lépünk be a "telekre":
cd public_html

Ha bent vagyunk, akkor a ház bejárata

2.7 AZ INDEX.HTML

állomány. Egyelőre legyen benne csak annyi, hogy **Under construction!**, azaz magyarul **Fejlesztés alatt!**

Lássuk!

```
echo "Under construction!" > index.html
```

Nocsak, már kész is az első lapunk. Nana! Inkább a nulladik. Ellenőrizzük a hozzáférési jogot (ls -al) és ha megvan a megfelelő számú r és x, akkor

2.8 A LAP CÍME

az Interneten:

<http://www.honlapiskola.hu/~jakab>

Akár meg is nézhetjük.

3 AZ ELSŐ LAP

elkészítéséhez tudnunk kell

3.1 MIVEL KÉSZÍTSÜK?

Nem kell ágyúval verébre lőni, eleinte bármilyen egyszerű karakteres szövegszerkesztő megteszi. UNIX alatt a **pico**, vagy a **joe** (a **vi** az elborult elmék varázseditora, igazi buherátor nem is nyúl másához, de kezdőknek nem ajánlom); DOS alatt a **Norton Editor**, vagy az **EDIT** is megfelel, illetve Windows alatt a **Notepad**. Feltételezem, hogy mindenkinek van kedvence, amit ismer és használ.

Tehát elkezdjük a bejárat, azaz az index.html szerkesztését!

3.2 EGY HTML LAP SZERKEZETE

a HTML "nyelv" szabályai szerint így néz ki:

```
<html>
```

```
<head>
```

Fej

```
</head>
```

```
<body>
```

Törzs

```
</body>
```

```
</html>
```

3.3 MI KERÜL A FEJBE?

A fejben a lapra jellemző, rövid, pár szavas leírást kell megadnunk. Ez a szöveg a böngészőprogram ablakának tetején fog megjelenni, 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.

3.4 MI KERÜL A TÖRZSBE?

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

Ezek után Jakob számára

3.5 AZ ELSŐ KÉSZ LAP

valahogy így néz ki:

```
<html>
```

```
<head>
```

```
<title>Jakab honlapja</title>
```

```
</head>
```

```
<body>
```

```
<pre>
```

```
Hello!
```

Jakab vagyok, most tanulom a honlapkészítés rejtelseit. Hamarosan ez az oldal tele lesz minden széppel és jóval.

A levélcímem:
jakab@honlapiskola.hu

```
</pre>  
</body>
```

```
</html>
```

Nos, ezzel valóban kész az első lapunk! A szöveg szabadon választott. Lehet játszani. De hamar felmerül a kérdés,

3.6 MI KERÜLJÖN AZ ELSŐ LAPRA?

Amikor az ember honlapot készít, nyilván igyekszik bemutatni magát. Ezt számtalan módon meg lehet tenni, ez a korlátlan önmegvalósítás ideális terepe. Szokásos megoldás egy fényképet kirakni magunkról és pár alapvető adatot, de lehet humoros kisregényt írni magunkról fénykép nélkül, vagy szigorú szakmai önéletrajzot három nyelven, vagy játékos ákombákomokkal teleaggatni az egész lapot. Mindent lehet. Egy dolgot érdemes szem előtt tartani: a lap információs erejét. Érdektelen, nehezen áttekinthető, nehezen letöltődő (azaz nagyon hosszú) lapokról hamar tovalebbről az Interneten bolyongó!

4 OLVASNIVALÓK

Lehet tiszta lappal kezdeni a Hálózaton, de ha már egyszer valamit valamikor leírtunk, nem szívesen pötyögjük be újra. Azaz előbb-utóbb elkövetkezik a

4.1 KÉSZ SZÖVEGEK BEVÁGÁSA

a lapba. A legegyszerűbb, ha van egy "primitív" szövegszerkesztővel elkészített, általában .txt jelzést viselő állományunk. Ha nincs benne ékezet (mert mondjuk angol nyelvű), akkor még jobb. Nincs más dolgunk, mint a szöveg elejére beírni: <pre> és a végére: </pre>. Ezt a törzsbe beillesztjük és kész. Ugye milyen egyszerű?

Olcsó húsnak híg a leve: a **preformatted** (előre tördelt) szövegek csúnyácskák és gyanúsán úgy néznek ki, mint egy program forráslistája egy monitoron. Nem csoda, ezt a megoldást pont erre találták ki! A böngészőprogramok ugyanis egyébként figyelmen kívül hagyják az általunk beírt sorvégjeleket és tördelést, öntik a szöveget, ahogy az az ablakba befér. Próbáljuk ki! Egy hosszabb .txt állományt először zárjuk a <pre> és </pre> jelek közé, s nézzük meg:

```
<pre>
Hol az a dal,
  Ami felkavar?
Hol az a dal,
  Ami vigasztal?
Hol az a dal,
  Ami egyszer észrevétlenül elveszett?
</pre>
```

...majd töröljük ezt és gyönyörködünk!

Hol az a dal, Ami felkavar? Hol az a dal, Ami vigasztal? Hol az a dal, Ami egyszer észrevétlenül elveszett?

Ugye milyen szép? Jó, hogy láttuk, mert később, ha valamiért olvashatatlan betűfolyammá válik a lapunk, gyanakodhatunk arra, hogy valahol valamilyen HTML jelet elfelejtettünk!

Az előre tördelt szövegekkel más baj is van: a böngésző nem mer hozzájuk nyúlni, azaz, ha hosszabb a sor, mint amilyen széles az ablak, akkor nem látjuk az egészet, tologathatjuk a görgetősávot oda-vissza. Korlátozhatjuk persze a szélességét, de valaki bizonyosan annál is kisebb ablakot használ, mint amit mi feltételezünk.

Ha viszont programok forráslistáját kell közölnünk, nehogy mást használjunk! Nincs annál reménytelenebb melő, mint egy precíz bekezdésekkel ellátott forrásszöveget újraszerkeszteni! A böngészők, hogy kihangsúlyozzák, itt valami másról van szó, s nem közönséges szövegről, az előre tördelt részeket más, írógépszerű betűtípussal jelenítik meg.

Az ékezetekről még lesz szó, egyelőre csak annyit, hogy megvert a digitális mindenható minket többféle szabvánnyal is, ezek közül a **Latin-2** megtűrt a weblapokon (éppen csak), minden más olvashatatlan! Van kiút, látni fogjuk.

Végezetül: nem lebecsülésként írtam az elején azt, hogy "primitív" szövegszerkesztő! Ma ugyanis a WinWord is szövegszerkesztő programnak számít. "Primitív" az a szövegszerkesztő, amelyik bárhol olvasható, szerkesztőjelek nélküli, ún. **plain text** állományt állít elő. Azaz az állományban csak a szöveg van, semmi más. Nevezhetjük nyers szövegnek is.

Amennyiben nem nyers a szöveg, hanem már szépen tördelt valamelyik nagytudású programmal, akkor lépnek színre a

4.2 KONVERTEREK

Annyiféle formátum van már, hogy az elképesztő! Reménytelen vállalkozás tehát univerzális átalakítót keresni vagy írni. Nem is érdemes. Léteznek eredményes kísérletek

olyan állományformátumra, amely az összes elképzelhető szerkesztési lehetőségekre felkészült, ilyen például a **Rich Text Format** (.rtf).

Figyelem, ez a formátum közönséges halandó számára olvashatatlan! Olyan, mint egy lefordított program, csak itt a vastag betű, dőlt betű, ilyen betű, olyan betű van egységes "nyelvre" fordítva. A lényeg: ezt a formátumot a szövegszerkesztő programunknak ismernie illik! (A WinWord pl. ismeri.)

A már megszerkesztett szöveget tehát ebben a formátumban újra el kell mentenünk. Utána pedig keresni kell egy olyan programot, amelyik .rtf állományból .htm(l) állományt készít. Ilyen az **rtftohtm** nevű, FTP archívumokból letölthető shareware program. Használata egyszerű:

rtftohtm valami.rtf

A program elfogadhatóan dolgozik, amíg valami extra nem bukkan fel a szövegben (pl. egy szöveggel körülötvött ábra), ezen biztosan elbukik majd.

Bevallom, ez nem is baj. A HTML leírások ugyanis csak támpontokat adnak a megjelenítésre és nem definiálják azt szigorúan. Azaz:

A weblap megjelenése a böngésző beállításától is függ, semmi garancia nincs arra, hogy más is azt látja, amit mi!

A weblapon nem lehet cél egy profi képesűtség mérethelyes megvalósítása. Kész szövegeink tartalma a lényeg, a formát pedig igazítanunk kell a lehetőségekhez, esetleg

4.3 KÉZZEL SZERKESZTVE

Ha üres lappal indítunk, vagy ha már kész anyagot kell webre hangolni, akkor a megfelelő szerkesztési "utasításokat" magunk is elhelyezhetjük.

Láttuk már, hogy ezeket az "utasításokat" < és > jelek közé zártuk, ez minden esetben így van. Leírásukhoz kis- és nagybetű egyaránt megfelel, a kisbetű talán kényelmesebb, a nagybetű jobban látszik. Vannak jelek, amik párban állnak (zárni kell őket), a bezáró egy / jellel többet tartalmaz.

A szöveg bekezdéseinek elhatárolásához a <p> jelet használjuk. A bekezdéseken belül a szöveget a betűméret és az ablakszélesség figyelembevételével szabadon tördeli a böngészőprogram, ám ennél a jelnél mindenképpen új sort kezd, a két bekezdés közt általában üres sort is hagy.

Ha szükséges, külön is kérhetjük az új sor kezdését (pl. párbeszédeknel) a
 jelet. Ez üres sort nem hagy.

Ha a szövegben élesebb elhatárolást szeretnénk, úgy vonalat is húzhatunk a <hr> jelet.

Egy hosszabb szövegben valószínűleg fejezet és alfejezetcímek is előfordulnak, ezeket a <h1>, <h2>, ..., <h6> jelekkel emelhetjük ki. Figyelem, ezt a jelzést zárni kell! A <h1> párja a cím végén a </h1>. Az egyes a legnagyobb méretű betűt adja, a hatos a legkisebbet. Maga a szöveg a négyessel egyező méretű, így az ötös és hatos "címléíró" inkább speciális célra (pl. copyright jelzésére) használjuk.

Az amerikaiak balra igazítják a címeiket, ha ez nem tetszik, a <center> jellel kérhetjük a középreigazítást. Ezt a jelzést is zárni kell! Összevonható, sőt illik is összevonni a címjelzővel, ekkor így alakul: <h1 align=center> és értelemszerűen elég csak így zárni: </h1>.

A szövegen belül **vastag** és *dőlt* betűs kiemeléssel élhetünk, erre a és az <i> jelek szolgálnak és ezeket is zárni kell. Ez a két jelző azonban egyes vélemények szerint kerülendő, ugyanis a HTML filozófiájának ellentmondanak: szigorú utasítást adnak, fizikai leírók és nem "útmutatók". (Valóban, van olyan eset, hogy nincs mód pl. dőlt betűt írni.) Ezért helyettük más, a logikai jelentést jobban leíró jeleket használhatunk. Megerősíthetjük mondanivalónkat a jelzővel, *hangsúlyt* adhatunk neki az jelzővel, vagy *idézhethetünk* is a <cite>

jelzővel. Levélcímünket pedig illik **<address>** jelzők közé zárni – azaz ez is bezárandó, mint a többi.

Hát ez sok volt így egyszerre! Pedig van még... De inkább nézzük, Jakab lapja hogyan néz ki, feltéve, hogy szereti Koncz Zsuzsát hallgatni:

```
<html>
```

```
<head>
```

```
<title>Jakab honlapja V2.0</title>
```

```
</head>
```

```
<body>
```

```
Hello Mindenki!<p>
```

```
Ez itt
```

```
<h1 align=center>Jakab honlapja</h1>
```

```
Jakab vagyok, most tanulom a honlapkészítés rejtelseit.  
Hamarosan ez az oldal tele lesz minden széppel és jóval.
```

```
<p>
```

```
A <strong>kedvencem</strong> sorait idézem:
```

```
<p>
```

```
<cite>
```

```
Ma már a szívünk helyén kompjuter dobol,<br>
```

```
Kiszámított a szenvedély.<br>
```

```
Egy új világ helyett csak látványt tervezünk.<br>
```

```
A lelkünk hallgat, s csak a pénz beszél.
```

```
</cite>
```

```
<hr>
```

```
A levélcímem:<br>
```

```
<address>
```

```
jakab@honlapiskola.hu<br>
```

```
</address>
```

```
</body>
```

```
</html>
```

4.4 ÉKEZETEK

A magyar nyelv ékezteit **majdnem** szabályosan meg tudjuk adni a HTML karakterleíróival. A majdnem azt jelenti, hogy a készletben nincs hosszú ö és ü. Helyette ô és û használt, ezek azok a "kalapos" betűk, amelyeket kis jóindulattal elfogadhatunk (a képernyő és a nyomtató karakterkészleteinek magyarrá cserélésével a kalapok helyett a rendes hosszú ékezetek is megjeleníthetők). A lélek nyugalma itt sem teljes, ugyanis a kalapos ô helyett a hullámvonalas is elterjedt – miért pont ebben lenne egység? (Halkan merem csak megjegyezni, hogy ha már szegény ű kalapos, logikusabbnak tűnik az ő-re is kalapot húzni...)

Minden további szószaporítás helyett itt a táblázat, amellyel akár ékezetmentes billentyűzeten is leírhatjuk azt, hogy:

árvíztűrőtükörfúrógép

ÁRVÍZTŰRŐ TÜKÖRFÚRÓGÉP

Ezt a szöveget érdemes megjegyezni! Az összes magyar ékezetes magánhangzót tartalmazza.

Magyar ékezetes betűk HTML leírói:

á	á	Á	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ö	ö	Ö	Ö
ő	ô	Ő	Ô
õ	õ	Õ	Õ
ú	ú	Ú	Ú
ü	ü	Ü	Ü
û	û	Û	Û

Ékezetes szöveget így írni hosszú távon fárasztó, ezért érdemes valami konvertálót beszerezni (pl. az rtfotm is tudja ezt). Vagy írhatunk **Latin-1** vagy **Latin-2** karaktereket közvetlenül is (ha tudunk), bár ez nem "sportszerű" (pl. e-mailben elküldve egy ilyen szöveg bizonytalan végeredményt ad, és a keresőprogramok sem mind birkóznak meg vele).

5 KÉPEK

Egy kép felér ezer szóval, tartja a mondás. Ilyen az agyunk, jobban szereti látni a dolgokat. Tudták ezt a HTML kidolgozói is, így lehetőségünk van "feldobni" a lapunkat valamilyen vizuális sokkhatással. Egy weblap kép nélkül (ma már) elég szegényesen hat.

5.1 KÉPFORMÁTUMOK

Képfarmátumból is akad pár tucat, szerencsére itt látszik a kiút. A két legelterjedtebb (és melleleg elég tömör) tárolási szabvány az elfogadott a HTML világában: a **GIF** és a **JP(E)G**. (A UNIX miatt az **xbm** is, de ez lényegtelen.)

A **GIF** az "öregebb". Annyit érdemes tudni róla, hogy a tömörítő eljárása veszteségmentes, azaz a képből nem jelennek meg hibák a konverzió során. Ebből persze következik, hogy tömörségének határai vannak. Mégsem ez a legnagyobb baj, hanem az, hogy csak 256 színt tud. Ezt ugyan nagyobb palettából választhatja ki, de egyszerre akkor is csak ennyi szín lehet fent -- s ez kevés lehet egy finom színátmenetnél... Ábrákhoz, rajzokhoz viszont kitűnő.

A JPEG kimondottan fényképekhez van idomítva. Nagyon tömör, annyira, hogy csal: nem egészen ugyanazt a képet tárolja, mint amit megadtunk neki. Az emberi szem ugyanis toleráns és becsapható. Bizonyos hibákat egyszerűen nem lát meg, ezeket tehát büntetlenül el lehet követni. A JPEG cserébe kis méretet és 24 bites valóság-hű színvisszaadást biztosít. Jobbat, mint amit a legtöbb monitor valaha is tudni fog... (Rossz paraméterezéssel persze el lehet rontani szemmel láthatóan is a képet.) A JPEG nem jó a rajzokhoz, ott csúnyán hibázik és ehhez képest keveset karcsúsít az állomány méretén (esetleg még növeli is azt).

A trükkös **animated GIF** az a fajta móka, amikor az állókép megmozdul. A GIF89 tudja ezt, több képet egymásba fonva öt-tíz fázisú "animáció" hozható létre, egyszeri lefutással, vagy végtelen ciklusban (utóbbtól némelyik böngésző összeomlik egy idő után).

Van **interlaced GIF** is, ez azt jelenti, hogy a kép először csak úgy nagyjából jelenik meg, aztán finomodik. Ha épp sietünk, a durva "vázlatból" is eldönthetjük, érdemes-e várunk a képre – és ez jó!

Érdekes és gyakran használt lehetőség a **transparent GIF**, azaz a háttérszín átlátszóvá tétele. Ha a semleges, egyszínű hátteret átlátszóvá alakítjuk, a kép még inkább része lesz a lapnak, a keret nem árulkodik bántón.

Ha már kiválasztottuk a képe(ke)t, két lényeges dolog tisztázandó:

5.2 A MÉRET ÉS AZ ELHELYEZÉS

A kép sem tárolás, sem megjelenés szempontjából ne legyen túl nagy. Egy irgalmatlan "*borzalom x rettenet*" méretű kép egyszerűen nem fér el az ablakban – akkor meg mit érünk vele? A legtöbb megjelenítő legfeljebb 640x600 pixel nagyságú (azaz nagyjából az SVGA monitor méretének megfelelő, annál kicsit keskenyebb) ablakkal dolgozik.

A képet "meg kell komponálni", azaz el kell beszélgetnünk önmagunkkal, hogy kell-e ez a kép egyáltalán, és ha kell, ekkorában kell-e, mert esetleg kisebbben is jó. Nem könnyű döntés, a kicsinyítés ugyanis (sajnos) minőségromlással jár. A kép minősége lehetőleg haladja meg a Junoszty ipari szabvány követelményeit – mi tudjuk, mi van a képen, de hátha más nem! Lehet, hogy vágni kell a képből, lehet, hogy egy kicsit világosítani, vagy sötétíteni, azaz akad vele munka.

A kész képet el kell helyezni a lapon valahogy és valahova. Természetesen csak a törzsben. Figyelem! A kép **nem része** a dokumentumnak, nem belerakjuk, csak jelezzük, hogy ott lesz. Az elhelyezés a böngésző feladata. Van ennek egy kellemetlen következménye: ha el akarunk "lopni" egy oldalt (miért ne tennénk, hiszen ezt szabad, ha a gazdája nem tiltja), akkor annak képeit külön-külön még le kell szedegetni. A Netscape-ben ezt a jobb egérgombbal a képre

kattintva tehetjük meg. Léteznek ügyes segédprogramok, amelyek már képesek minden elemet együtt elmenteni. Mások anyagait azonban nem tulajdoníthatjuk el, azaz egy nekünk tetsző lapot ne tegyünk ki sajátunkként, mert nem illik.

A lapon a képet az `` leíróval jelezzük. Ez persze csak a legegyszerűbb eset, itt a képállomány ugyanott van, ahol a lap (mert lehet más könyvtárban, de más gépen is!) és semmilyen egyéb igényünk nincs.

Pedig lehetne! Tehetjük a képet balra (alapértelmezés) vagy jobbra. Igazíthatjuk a szöveget a felső széléhez, az aljához, vagy középre. Ez lényeges, ugyanis a kép **egy szövegsor magas** a böngésző logikája szerint. A kép mellé több sort írni tehát első nekifutásra nem lehet! (Azért van rá megoldás.)

Az elhelyezésre vonatkozó utasítást az `img` után elhelyezett `align=` mögé írjuk, mint az angolok: jobbra **right**, balra **left**, föl **top**, le **bottom**, középre **middle**. Jakab képe így válik a lap részévé:

Ez a csúnya ember `` vagyok én.

Adhatunk a képnek más méretet is, mint amekkora valójában, ezt vagy pixelben (akkor mindig ugyanakkora lesz), vagy százalékban (az ablak méretétől függ) adjuk meg. Jakabnak igen nagy az arca ;-), úgyhogy javítunk egy picit:

Ez a csúnya ember ``vagyok én.

Adhatunk a képnek vékony vagy vastag keretet a `border=` beszúrásával. A keret méretét pixelben adjuk meg, a 10 már szép széles.

Vannak, akik az információs szupersztrádán gyalogriksán közlekednek, grafikus felület nélkül. Ők a legszebb képünkől is csak ennyit látnak:

[IMAGE]

No, ez nem sok. Hogy ők se maradjanak ki semmiből és legalább külön letölthessék a képet (erre módjuk van), ha "egyenesben" nem láthatják, az előbbi kiírást módosíthatjuk az `alt=` beszúrásával. Az egyenlőségjel mögé **rövid** magyarázatot írunk:

``

Holtbiztos, hogy letöltik!... ;-)

A képekkel való játék ezzel persze még nem merült ki. Egy fontos és a lap kinézetét döntően befolyásoló tényező

5.3 A HÁTTÉRKÉP (BACKGROUND)

Itt a kép nem valahova kerül a lapon, hanem a lap "papírja" lesz, erre írunk majd. Olyan, mint egy tapéta, csak nincs neki vége.

Gondoljuk meg, ez mivel jár! A háttérbe álmódott kép (bizony) nem lehet túl éles vagy kontrasztos. Nem ütheti agyon ugyanis a lapon a szöveget. A böngészők betűi túl könnyen olvashatalanná válnak. Legyen a kép vagy sötét és írjunk rá fehér vagy pasztellszín betűkkel, vagy legyen világos, amin a sötét "tinta" mutat. Kísérletezni kell.

A háttérkép témája alapvetően kétféle lehet: vagy egy elmosódott hangulatfestő fénykép, vagy valami iméltlódó ákom-bákom (pl. buborékok, kockás füzetlap, márvány-utánzat). A böngésző veszi ezt a képet és annyiszor teszi egymás mellé és alá, ahányszor a lapon lévő egyéb anyag (képek, szöveg) megköveteli. Ez fényképnél lehet kínos. Ugyanis (megint csak) nem tudjuk, hogy az aki olvassa a lapunkat, mekkora ablakot használ és mekkora betűket (ha kisebbet, mint mi, lehet, hogy nem is tudja a kép miatt elolvasni!). Az általunk feltett fénykép vagy csak félig-negyedéig látszik, vagy kétszer-háromszor is ott van... Nehéz igazán jó háttérrel kitalálni! Egy tipp: keressünk olyan képet, amelynek szélei illeszkednek. Azaz a képet önmagához illesztve nem üti a szemünket a határolóvonal. Mindez a probléma a tapéta jellegű kisképeknél nem jelentkezik.

Ha merünk vállalkozni rá, így tehetünk háttérrel a lapunk mögé:

`<body background="teglak.gif" text=ffffff link=aaffff vlink=88ffff`

alink=00ffff>

A kép ez esetben is a lapunk mellett lapul ugyanabban a könyvtárban, de lehet ez másképp is.

Igen, adós vagyok több magyarázattal is. A **text=** a szöveg színe, a **link=** a kapcsolódási pontoké (ezt majd hamar megmagyarázom, mindjárt a következő fejezetben), ha még nem néztük meg, a **vlink=**, ha már megnéztük és az **alink=**, amikor épp rákattintunk. A színek megadása a vörös-zöld-kék (**RGB**) sorrendben hexadecimális kétjegyű számokkal lehetséges. (A függelékben egy kis táblázat segít ezeknek a kódoknak a hétköznapi színekkel való összepárosításában.) Háttére egy lapnak egy lehet, ezért került a törzs kezdősorába, a **body** kibővítéseként ez a pár jelző. A háttér lehet egyszínű is, ezt a **bgcolor=** adja meg ugyanitt, szintén RGB kódban. Ez szerepelhet a **background=** helyett, de mellette is! Utóbbi esetben (főleg, ha a kép nagy) a megadott háttérszín megjelenik, majd a kép megérkezésekor a képpel felülíródik.

Végül, mert sajnos idetartozik, néhány szó

5.4 AZ ÁTVITELI SEBESSÉGRŐL...

Nincs az a vonal, ami előbb-utóbb el ne dugulna a forgalomtól. Ekkor pedig egy nagy képet letölteni idegölő dolog. A fejezet elején említett mondás megfordul: ezer szó felér egy képpel, ráadásul gyorsabban átjön a hálózaton.

Kíméljük egymást! Legyen csak annyi képünk, amennyi feltétlenül szükséges (azért a lapunk egyéni arculatát nem kell agyoncsapni!) és ezek legalább tárméretben kicsik legyenek! Ha más megoldás nincs, szedjük több lapra szét az eredetit. Egy lapra 40-60 kbyte-nál több kép csak igen-igen indokolt esetben kerüljön! Számítsunk arra is, hogy (főleg a tapasztalt barangolók) kikapcsolják a képek letöltését -- azaz a lap megtervezésekor fordítsunk gondot arra, hogy képek nélkül is mutasson valahogy. Számos eset van, amikor ez megoldhatatlan, de amikor van mód rá, legalább akkor tegyük meg. Vagy az **alt=** segítségével mellékeljünk magyarázatot a képekhez, vagy készítsünk önálló "**text only**" verziót is a sávszélességben "szegények" számára (ez gyakori megoldás).

Ha valamilyen okból sok kép kell a lapra, akkor is van mód a takarékosagra. Ne magát a képeket tegyük fel, hanem kicsiny **ikonokat**, s ezek mögé pakoljuk be az egész estét betöltő plakátképeinket. Ennek mikéntjéről (is) szól a következő fejezet.

6 LINKEK

Amit eddig tettünk, az alig különbözik egy bármilyen kiadványszerkesztő programmal előállítható laptól – sőt igazából azokhoz képest elég szerények a lehetőségek. Van azonban valami, amitől egy HTML lap életre kel: ezek a kapcsolatok, vagy angol szakszóval a linkek. Lapunkon bárhol elhelyezhetünk linkeket egy-egy szóhoz, mondathoz, akár szövegrészhez vagy képhez kapcsolva, s aki erre rákattint, máris új oldalt kap. Vége a lexikonok "lásd még" utalói után folytatott lapozgatásnak és a szöszedet szerinti keresgélésnek, a "lineáris" olvasásnak – csak egy kattintás az egész. Ez a **hipertext**.

Persze ezeknek a linkeknek valamilyen értelemmel kell bírniuk.

Technikai oldalról a link nem más, mint egy cím, a kapcsolódó anyag (mert az nemcsak egy másik HTML lap lehet) **URL**-je, azaz hálózati "lelőhelye".

6.1 AZ URL FELÉPÍTÉSE

A legegyszerűbb, ha rögtön egy példával kezdjük:

<http://www.honlapiskola.hu:8080/~jakab/cdlista.html#koncz>

6.2 PROTOKOL

A kettőspontig bezárólag tudatjuk, hogy az általunk elérni kívánt anyagot milyen kommunikációs eljárással lehet elérni.

- **http**
HyperText Transfer Protocol. Ez a honlapok számára kitalált adatcsere mód, szinte mindig ezzel találkozunk majd.
- **gopher**
Gopher. A gopher tekinthető a webvilág őseinek. Itt szigorú menüstruktúrák közt válogathatunk, míg el nem jutunk az anyagunkhoz. Sokkal kevésbé rugalmas, mint a web, ám egy igazán jó gopher még ma is életképes!
- **ftp**
File Transfer Protocol. Ez már özönvíz előtti technológia, de mert az első igazi program- és adatkönyvtárak ezzel kezdték, elterjedt. Igazán jó turkálóhelyek léteznek, sőt most is nyílnak ezzel a protokollal. Böngészővel egész kellemes keresgélni bennük, olyan, mintha egy file-managert (pl. a Norton Commandert) használnánk.
- **file**
Ez ugyanaz, mint az ftp, csak a saját gépünkön.
- **telnet**
Telnet. Ezzel egy gépbe léphetünk be, mintha terminállal dolgoznánk. Van egy-két különleges szolgáltatás, főleg nagy könyvtárak belső keresőrendszerei, amelyet így tudunk elérni, de nagyon idegen a web világtól, inkább csak a teljesség kedvéért került be. (A böngészőprogramok nem is kezelik egyelőre, hanem egy önálló telnet programot hívnak meg.)
- **mailto**
Ez egy hasznos protokoll: ha ide kattint valaki, akkor levelet írhat a megfelelő címre. Ha a saját címünk az, akkor nekünk – egy ilyen illik is feltenni a lapunkra! A folytatása eltér a többitől és a fenti példától, ugyanis a kettőspont után egy e-mail címet kell írunk. Jakabé így néz ki:

A levélcímem:

<address>

```
<a href="mailto:jakab@honlapiskola.hu">jakab@honlapiskola.hu</a>
</address>
```

- **news**

Az Interneten több ezer newsgroup, azaz elektronikus vitafórum létezik. Ezeket is elérhetjük, de igazán ez sem része a webnek. Szintén sajátos megadási módja van:

```
<a href="news:news.group">news.group</a>
```

- **Host**

A hostokat már említettük, itt lakik az egyik hoston a mi lapunk is. Kétféle megadási mód van: név vagy IP szám szerinti.

Név szerint pl. a már ismert **www.honlapiskola.hu**.

IP szám szerint pl. a **193.6.1.39**.

Ez utóbbit kerüljük. Csak akkor szükséges, ha a host-nak nincs bejegyzett neve (gyakori kis vagy frissen telepített gépeknél, tehát rákényszerülhetünk).

- **Port**

Általában elhagyható. Ritkán, de előfordul, hogy a böngészőnk magától nem tudja megállapítani, melyik "ajtón" kell belépnie a távoli gépbe. Ekkor a hostnevet (vagy az IP számot) egy kettősponttal és egy számmal egészítjük ki. Nehéz kideríteni, ha vaktában kell próbálkoznunk (ami elég reménytelen), akkor ezeket a számokat érdemes először megpróbálni: 8000, 8080. A szabványos szám ugyanis a 80-as, de új szervereknél megesik, hogy tesztelés vagy más ok miatt nem ezt használják.

- **Elérési út és állománynév**

Kezdjük azzal, hogy ha ide semmit nem adunk meg, akkor a megcímezett gép legfőbb lapjának kell(ene) megjelennie, amely általában az üzemeltető honlapja. "Társbérlok" egy / után megadott néven érhető el, ha vannak. A felhasználók viszont a ~ jel után a **user** nevük alapján, azaz Jakab kezdőlapja a ~jakab "útvonalon". Ha e mögé állománynévet nem illesztünk, akkor a megadott útvonal egy könyvtárra mutat (ez esetünkben Jakab public_html könyvtára) és abban az index.html állomány az, ami majd megjelenik. Ez az alapértelmezés. Ezért kellett pont ezt a könyvtárat és állománynévet megadnunk a "tanfolyam" elején.

Kínos, de az ellenség keze ide is betette a lábát. Bizonyos jó marketingpolitikával, ám annál (khmm) erőtlenebb operációs rendszerrel rendelkező cégek ránk hagyományozták a 8+3 betűs állománynéveket. Ekkor az index.html index.htm-mé válik, ami kavargást okozhat. Ha barátunk lapját nem érjük el, esetleg próbáljuk meg az egyébként nem kötelező index.html-t index.htm-ként beírni a cím végére! (Vagy esetleg az index.html-t, ha a szolgáltató éppen a 8+3-at tekintette alapértelmezésnek.) Van amikor segít...

Más lapok elérésénél már semmit nem takaríthatunk meg, a teljes utat és az állomány nevét is meg kell adnunk. A példában Jakab cdlista.html állományát címeztük meg.

- **Kiegészítők**

A # jel után egy HTML lapon belül egy adott részre hivatkozunk, ezt a lap írójának előre ki kell jelölnie a megfelelő HTML jellel. Hosszú tematikus listáknál például jól jön. Jakab sokféle CD-t hallgat, de nekünk csak Koncz Zsuzsa kell. Tehát megy a végére az, hogy #koncz.

A ? után viszont paramétert ad át a böngésző. Ez már túlmutat azon, amivel mi foglalkozunk, de ha ilyet látunk valahol, érdemes figyelni és találgatni, hogyan is működik.

6.3 A LINK JELÖLÉSE

Linkre nekünk is szükségünk lesz. Háromféle "távolságra" vethetjük ki a lasszót: távolra, helybe, vagy az oldalon belülre. A linket az `` leíróval jelezzük és ezt zárni is kell a `` jelzővel.

- **Link távolra**

A teljes URL címet meg kell adnunk:

```
<a href="http://www.homepageschool.edu/~donald/cdlista.html">Donald CD-i</a>
```

Ezzel amerikai Donald barátunk listájára mutató linket helyeztünk el a lapunkban. A lapon a "Donald CD-i" felirat jelenik meg (olyan színben, amelyet a `link=` vagy a `vlink=` határoz meg -- alapesetben kéken). Az egeret odairányítva a nyílból kéz lesz, azaz kattinthatunk. Kattintás után érkezik meg Donald barátunk lapja.

- **Link helybe**

Jakab a saját `public_html` könyvtárán belül az állományok neveivel (és az esetleges alkönyvtárnevekkel kiegészítve) hivatkozhat a saját lapjaira.

```
<a href="cdlista.html">A saját CD-im</a>
```

Jakab a(z `index.html`) lapjában egy másik, `cdlista.html` nevű lapjára hivatkozott. Ezt írhatta volna így is:

```
<a href="http://www.honlapiskola.hu/~jakab/cdlista.html">A saját CD-im</a>
```

Ez nemcsak hosszabb, de rugalmatlanabb is. Ha véletlenül költözik a lapunk egy másik hostra, akkor az ilyen címeket egyenként mind át kell írni! Jobb a rövid (ún. relatív) cím és kényelmesebb is!

- **Link dokumentumon belüli ugrásra**

Egy hosszú dokumentumon belüli ugráláshoz elég csak a **címkét** megadni.

```
<a href="#koncz">Koncz Zsuzsa CD-k</a>
```

Ehhez azonban ezeket a címkéket el kell helyezni. Erre az `` leíró szolgál, ezt is `` zárja.

```
<a name="koncz"></a>
```

Itt nem, de a linkeknél muszáj valamit a nyitójel és a záró jel közé tenni, mert különben nem lesz mit kattintani!

Link egyébként nem csak HTML lapra, hanem bármi másra is mutathat. A böngésző vagy meg tudja jeleníteni (.txt, .gif, .jpg), vagy át tudja adni valamilyen csatolt segédprogramnak (.mpg-t videolejátszónak, .au-t hanglejátszónak), amennyiben ilyet el tud érni, végső esetben pedig felajánlja, hogy letölti nekünk a munkakönyvtárunkba, aztán azt csinálunk vele, amit akarunk.

6.4 HOVA KERÜLHET LINK A LAPON?

Linket bárhová rakhatunk. Egy szövegbe, egy szóhoz, egy kifejezéshez rendelve.

```
<a href="magamrol.html">Jakab</a> vagyok, most tanulom a honlapkészítés rejtelseit.
```

Jakab ebben a példában valószínűleg hosszabban értekezik önmagáról a `magamrol.html` lapon, de ezt csak akkor olvashatjuk el, ha a "Jakab" szóra kattintunk.

Akár egy egész bekezdést is elláthatunk linkkel, ha ennek van valami értelme (pl. egy cikkből idézünk röviden, de ezt inkább az idézet végére írt irodalmi hivatkozás pár szavára illesztjük).

Ha valamit listaszerűen oldunk meg, arra van némi plusz segítségünk. Jakab a CD-it így jelöli meg:

Jakab fantasztikus otthoni CD gyűjteménye:

```
<ul>
```

```
<li><a href="#cseh">Cseh Tamás</a>
```



```
<li><a href="#koncz">Koncz Zsuzsa</a>
<li><a href="#pici">Presser Gábor</a>
<li><a href="#zoran">Sztevanovity Zorán</a>
</ul>
```

A tételek egymás alatt lesznek, kicsit beljebb, mint a lap széle és egy kis tömör ponttal kezdődnek majd. A bekezdést az ****, a pontokat (és az új tétel miatt az új sort) a **** jelzi. Pontozott lista helyett számozottat is kérhetünk az **** leíróval.

Linket tehetünk képre is. Mondjuk feltettük a lapunkra ősi és veretes családi címerünket JPEG képként, s azt szeretnénk, hogy az eredeti kutyabőr teljes szövegét elolvashassa az, aki rákattint a címerre. Nézzük:

```
<a href="kutyabor.html"></a>
```

A képre kattintva máris jön a kutyabor.html lapunk.

Érdekes és viszonylag új lehetőség képen belül bizonyos területek kijelölése. Azt szeretnénk például, hogy kedvenc üdülőfalunk, Baltásasszonyfa térképét feltéve a nevezetesebb intézményekről (horgásztanya, kocsmá) rövid leírást adhatnánk. Kicsit bonyolult, de nem veszélyes.

Először is kell egy baltásasszonyfai térkép mondjuk GIF-ben, ennek mérete pl. 320x200 pixel (nem nagy falu, tömegben nem lehet pihenni!).

```

```

Ebben az új a **usemap=** jelzés volt. Ez is lehet más lapra, sőt távoli lapra való hivatkozás is, de célszerűbb rögtön helyben elintézni a feldolgozást:

```
<map name="helyek">
```

```
<area shape="rect" coords="110, 90,130,110" href="horgasztanya.html">
```

```
<area shape="rect" coords="190, 20,210, 40" href="kocsmá.html">
```

```
<area shape="rect" coords=" 0, 0,319,199" nohref">
```

```
</map>
```

A **shape=** megjelöli, hogy milyen alakzatot használunk, de egyelőre ez csak a téglalap (**rect**) lehet, ennek bal felső és jobb alsó koordinátáit adjuk meg pixelben. Ezek után a képen ide kattintva a **href=** után megadott lap hívódik meg, akár egy linkről. A **nohref** sor elhagyható.

Okos dolog, de ne felejtsük el, hogy aki kép nélkül hívja le ezt az oldalt, az nem tudja használni ezt a lehetőséget! Nyakra-főre ne éljünk vele, ha pedig mégis kell, akkor találjunk ki valamit szegény karakteres felhasználóknak is – mondjuk egy sima kis listát a kép alá, ugyanezekkel a lapokkal.

A végére egy kis fekete leves:

6.5 GONDOK A LINKEKKEL

Sok link, sok gond. Az Internet állandóan átalakul, a szolgáltatások költöznek és megszűnnek, vagy csak átmenetileg bezárnak, s újabbak nyílnak. Aki tehát nagy nekibuzdulással ezer linket szedett össze az oldalára, az nagy munkát szakasztott a saját nyakába: ha nem akar leégni a vakvilágba mutató elavult linkjeivel, azt folyamatosan ellenőriznie és frissítenie kell! Ez pedig sok időt igényel. Szóval mértékletesség! Tényleg csak oda tegyünk távoli linket, ahova feltétlenül szükséges és időnként ellenőrizzük! Az első "not found" még nem jelenti azt, hogy nincs többé a címzett lap, dugulások előfordulnak. Próbálkozzunk máskor! Egy hete elérhetetlen lap azonban már gyanús, kezdjük el keresni az új címét (ha van még egyáltalán)!

7 SZERKEZET

Előbb-utóbb szükségessé válik egy kis rendrakás. Lapjaink elszaporodnak, képeinkről pedig egyre nehezebb lesz kideríteni, melyik laphoz is tartoznak tulajdonképpen. De ezen lehet segíteni.

7.1 ÁLLOMÁNYNEVEK, KÖNYVTÁRAK

Hogyan is válik őserdővé egy könyvtár? Eleinte, amikor még csak az index.html lapunk üldögél, esetleg egy-két képpel együtt a public_html könyvtárunkban, még nincs baj.

Illetve van. Ugyanis ahhoz, hogy a lapunkat a világból olvasni lehessen, meg kell rá adnunk az engedélyt. Igen, erről már volt szó, egy következményéről viszont nem. A UNIX logikájából következik, hogy a saját alapkönyvtárunkat, ahová a bejelentkezéskor jutunk, szintén olvasásra elérhetővé kell tennünk. Azaz, elvileg ezzel lehetőséget adunk rá, hogy ebbe is beleolvassanak idegenek. Akit ez zavar, az alapkönyvtárában ne tartson zűrös anyagokat. (Sőt inkább sehol se.) Ha valamit el akarunk zárni a külvilág elől, azt külön könyvtárba tegyük. (Persze állományonként is adhatunk vagy visszavonhatunk jogokat, de ez macerás...)

Egy idő után azonban annyira belejövünk (remélhetőleg) a lapok gyártásába, hogy feltétlenül megszerezni leszünk kénytelenek. Erre adok néhány tippet.

- **Témák szerint csoportosítsunk!**

Tegyük fel, hogy zenéről, irodalomról és természetjárásról is több lapot követtünk már el. Logikus, hogy ezeket témánként külön könyvtárba tegyük. Használjuk fel azt, hogy az index.html-t külön (általában) nem kell kiírni, így a téma vezető lapját (de ha csak egy lap az egész, akkor is) index.html néven helyezzük el a megfelelő könyvtárban. A témához tartozó képek és egyéb állományok is ide kerülhetnek.

Másik megoldás, hogy a téma vezető lapja a téma könyvtárán kívül, de nevében megegyezően kerül tárolásra (pl. irodalom.html és irodalom/ könyvtár). Ez akkor tűnik megfelelő megoldásnak, ha a témavezető lapok megjelenése egységes, pl. ugyanazt a háttérképet használják.

Ha valamit minden lapon használunk (pl. ikonokat, logókat), azokat szintén érdemes külön könyvtárba gyűjteni. Itt azonban felmerül az a probléma, hogy visszafelé nem lehet hivatkozni a belső könyvtárból. Ekkor sajnos hosszú, azaz teljes URL-t kell használnunk.

- **Hosszú nevek, rövid nevek**

Ma már több operációs rendszer is engedi a hosszú, beszédes állománynevek használatát, bár szóköz, ékezetes betű és írásjelek nem, vagy csak igen körmönfont (s ezért felesleges) módon vethetők be. A hosszú neveket ettől függetlenül érdemes alkalmazni.

Van kivétel. Ha a lapunk, vagy az abból nyíló szöveg, kép, stb. letöltésére komolyan számítunk és szeretnénk is, ez esetben úgy kell tennünk, mint a menekülő elefántcsapatnak. Minden csapatban van egy leglassúbb elefánt, s az egész csapat az ő sebességéhez alkalmazkodik. Esetünkben a leglassúbb elefánt a DOS a maga 3+8 karakteres állományneveivel. Szöveget, képet ezért érdemes még UNIX alatt is a fenti megkötés szerint elnevezni. Ha viszont valami kemény UNIX-os anyagunk van (egy jó C program, egy ügyes script), akkor használjuk csak ki a névadás szabadságát -- ezzel egy DOS-hívó úgysem tudna mit kezdeni. Vigyázzunk arra is, hogy DOS alatt a kisbetű ugyanaz, mint a nagy, a UNIX alatt viszont nem! Aki DOS alapú környezetben dolgozik (a sokak által használt Windows 3.x ilyen!), az letöltéskor csonka, egymást felülíró állományokba futhat, ha nem vagyunk előzékenyek.

8 EXTRÁK

8.1 MÉG SZEBB SZÖVEGEK

Ha valamit **nagyon** szeretnénk kiemelni, akkor két lehetőségünk is van: megnövelhetjük (vagy csökkenthetjük) a betűk nagyságát, illetve átszínezhajjuk őket. A kulcsszó a **font**.

Átméretezés:

```
<font size=+3>Ez nagy.</font><font size=-3>Ez kicsi.</font>
```

Színezés:

```
<font color=#FF0000>Ez piros.</font>
```

8.2 TÁBLÁZATOK

Ha sok adatot akarunk táblázatokban közölni, akkor azt régebben előre tördelt szöveggént vágjuk be, most viszont van saját, igaz kicsit bonyolult megadási módja:

A táblázat a **<table>** jellel kezdődik és a **</table>** zárja. Ezen belül a sorokat a **<tr>**, az oszlopokat a **<td>** határolja el egymástól. (Ha a táblázat oszlopneveit ki akarjuk emelni, akkor **<td>** helyett **<th>** jelet használunk.) A táblázat szélességét a **width=** módosítóval megváltoztathatjuk, **border=** megadásával keretszélességet definiálunk, a táblázat celláin belül pedig az **align=** segítségével jobbra **right**, balra **left**, középre **middle**, föl **top** és le **bottom** igazíthatjuk az adatot. Lássuk a helyi focieredményeket!

```
<table border=2>
<tr><th>Hazai csapat <th>Vendégcsapat <th>Eredmény
<tr><td>Baltásasszonyfa FC<td>Redvestáka TC <td align=middle>2 : 1
<tr><td>Rongyosi Falábúak <td>Szentkanalasi SC<td align=middle>0 : 0
<tr><td>Macskási Vasasok <td>Akasztódombi SE <td align=middle>1 : 3
</table>
```

Nem erre találták ki, de két (vagy több) hasábos kiírást is táblázattal oldunk meg. A szöveget **<table>** és **<td>** jelekkel vezetjük be, majd a hasáb végén újabb **<td>** jelet szúrunk be, végül **</table>** jellel zárjuk. Az **align=** legyen **top**.

Kép mellé ugyanígy írhatunk, csak akkor az egyik "cellában" a kép van, a másikban a szöveg.

8.3 KERETEK

A böngésző ablakát több részre oszthatjuk **frame**, azaz keret definiálásával. Ekkor tulajdonképpen több különböző HTML lapunk szorong egy osztott ablakban. A megosztáshoz egy vezetőlapot kell készítenünk, amin a felosztás szerepel, valamint itt helyezhetünk el üzenetet azok számára, akik nem rendelkeznek kereteket kezelő böngészővel. Egy példa:

```
<head>
<title>Jakab ablakai</title>
</head>

<frameset rows="*,120">
<frame name=fent src=alap.html>
<frame name=lent src=menu.html>
</frameset>
```

```
<noframe>
```

```
<body>
```

Te sajnos nem látsz be az ablakaimon. Keress egy nagyobb verziójú "létrát"!

```
</body>
```

```
</noframe>
```

Azt, hogy egy kattintás után melyik ablakban jelenjen meg az új lap, a **target=** határozza meg. Ide kell beírni az ablak nevét, amit a **name=** után írtunk.
``

8.4 ŰRLAPOK

Egy HTML lapot nem csak passzívan megtekinteni lehet, hanem megfelelő kiegészítésekkel szabályos kérdőívet, szavazólapot, vendégkönyvet, stb. kaphatunk, amelybe aztán be is írhatunk. Egy ilyen lap nyomógombokat, adatbeviteli mezőket, menüket tartalmazhat. A bevitt adatokat azonban a HTML keretei közt **nem lehet feldolgozni**, ahhoz kiegészítőprogram szükséges: vagy egy szerver oldali program, vagy egy Java alkalmazás.

8.5 SCRIPTEK

Bár a scriptek sem tartoznak szorosan véve a HTML-hez, egy rövid és igen gyakran használt példa erejéig mégis kitérek rá. Számlálót ugyanis nagyon sokan szeretnének a lapjukra tenni.

Mi is egy script? Röviden és DOS akcentussal – a script a UNIX ".bat"-ja. Csakhogy sokkal többet tud. Gyakorlatilag programnak minősül. Lapunkról ilyen kis (vagy akár nagy) programok indítására van lehetőség!

A világ nem egyszerű, a scriptek meghívására is kétféle megoldást dolgoztak ki: egy elsőt és egy jót.

Az első ilyen megoldás a

- **CGI**

azaz a **Common Gate Interface**. Ez egy "patkolás", amely beül a buta CERN httpd server és a külvilág közé, hogy a scriptre vonatkozó és a server által nem ismert utasításokat feldolgozza. Ez néhány kellemetlen jelenséggel jár. Az egyik, hogy a meghívott scriptek nem paraméterezhetők. A másik, hogy a scripteknek külön **cgi-bin** könyvtárban kell gyülekezniük. A scripteket el kell látni "fejlécinformációval" is. Mindezek ellenére a CGI nagy karriert futott be, így a fejlettebb szerverekbe eleve beépítettek egy ehhez hasonló, de kellemesebb lehetőséget, ez a

- **SSI**

azaz Server Side Include. Itt már a script hívása egyszerűbb és kényelmesebb, s együtt maradhat az őt hívó oldallal. Ez azért jó, mert pl. a cgi-bin könyvtárra vagy kapunk írási jogot, vagy sem, de még ha kapunk is, ott mások is "nyomulnak", közös lónak meg ugye... A saját könyvtárunkban viszont nyugodtan fejleszthetünk. A példát ez utóbbi szerverhez írtam meg, mert mentes a CGI "sallangjaitól".

8.6 SZÁMLÁLÓ (PÉLDA)

Teendőink sorrendben:

Készítsük el **counter** néven az alábbi UNIX scriptet tetszésünk szerinti szövegszerkesztővel a public_html könyvtárba:

```
N=`cat ./counter.$1`
```

```
expr $N + 1 > ./counter.$1
```

Nagyon ügyeljünk minden egyes jelre, még a szóközökre is, mert a UNIX haragos öregúrként viselkedik, ha pontatlanok vagyunk!

A kész állományra adjunk végrehajtási jogot!

```
chmod 755 counter
```

Készítsük el az index.html "adatlapját"!

```
echo 0 > counter.index
```

Erre adjunk általános írási jogot, hiszen a világon mindenki belátogathat a lapunkra, növelve a számláló állását!

```
chmod 666 counter.index
```

Végül helyezzük el az index.html-ben a számláló hívását (célszerű a legelső sorba a <head> elé):

```
<!--#exec cmd="./counter index"-->
```

Ezzel a számláló kész és működik is, csak épp nem látjuk a lapon az eredményét. Ehhez egy újabb sort csempésszünk a lap törzsébe:

Ezt a Jakab-féle lapot

```
<!--#exec cmd="cat ./counter.index"-->
```

boldogtalan elektronikus csavargó látta eddig.

A szám valóban megjelenik a szövegben, mi több a forrásban is, így aki lapunk "kézirátát" olvasgatja, az sem sejtetheti, milyen csodaprogrammal értük el ezt az eredményt. Más lapokon is bevethető ugyanez a script, mivel eleve úgy íródott, hogy csak az index cserélendő le más névre a lapban a hívásnál. Persze az "adatlapokat" külön-külön el kell készíteni! És ez még csak a kezdet!...

FIGYELEM! Ez a megoldás korszerűtlenebb servereken nem működik!

8.7 A SERVER RENDSZERVÁLTOZÓI

Több érdekes információt tartalmazó szabványos rendszerváltozót használnak a serverek, ezek közül emelek ki néhányat:

- `$LAST_MODIFIED`

A lap utolsó módosítási dátumát tartalmazza. Ahol lényeges, hogy kövessük, mikor frissült utoljára a lap, az állandó dátumátírás helyett ennek a változónak a kiíratásával is elintézhethjük a kérdést:

```
<!--#exec cmd="echo $LAST_MODIFIED"-->
```

- `$REMOTE_HOST`

Ebben a változóban rejlik a lapunkat éppen meghívó kíváncsi "ügyfél" gépének neve (maga a felhasználó anonim marad). Sokan szívesen veszik, ha a lap "felismeri" őket, így ezt is kiírathatjuk a fenti módon.

- `$DATE_LOCAL`

A mai dátum a helyi idővel együtt. Ha valaki nagyon lassúnak találja a vonalat, ezt az adatot ismerve rádöbbenhet, hogy nem munkaidőben kellene próbálkoznia... ;-) (Nem árt, ha olykor eszünkbe jut, hogy az Interneten mindig máshol van éjszaka.)

9 MERRE TOVÁBB?

A HTML fejlődése ellentmondásos. Népszerűsége óriási, alapjai biztosak, jól átgondoltak, de azóta mindenféle torzulások keletkeztek benne. A baj az, hogy eredetileg egy nagyon egyszerű dokumentumleíró nyelvet definiáltak, most pedig arra tart a HTML, hogy **GUI**, azaz grafikus felhasználói felület lesz belőle (plusz multimédia alkalmazás és reklámhordozó, plusz DTP kiadványszerkesztő és -terjesztő eszköz, plusz belső vállalati információs rendszer és üzleti tranzakciók közvetítője, stb.). Ma már rendelhetünk és fizethetünk is a WWW-ben bolyongva, laborokba kukkanthatunk be kandi kamerával, rádiót hallgathatunk egyenes adásban és mindenféle adatbázis-lekérdezőket nyüstölhetünk kérdéseinkkel. Messze kerültünk a kezdetektől és alaposan meghaladtuk a svájci atomtudósok igényeit.

A fejlődés egyik útja a

9.1 JAVA

A Java a HTML lapban a mozgás, az élet – szó szerint. Lapjainkon ezek után számológépek, elektronikus jőpofáskodógépek, dadaista versgenerátorok, kártyajátékok, köszönőrobotok és egyéb agyatlanságok tűnnek fel. A Java valódi programnyelv, amely kommunikációs felületként használja a HTML lapot.

Háromféle Java létezik. Az igazi Java egy komoly objektumorientált nyelv. Ebben teljes alkalmazásokat írhatunk meg. Van egy kistestvére, amely csak annyiban különbözik bátyjától, hogy nem önálló program, hanem inkább egyfajta rutinyűjtemény, melynek különböző részeit a lapról hívjuk meg. Végül egy elfajzott rokon is van, a Javascript, ez nem önálló, hanem a HTML lapba beleírható nyelv. Míg az "igazi" Java-t le kell fordítani, addig ezt egy interpreter értelmezi sorrol sorra, mint az otthoni C64 a Basic programokat. Képességei nagyon korlátozottak, igazából csak hasonlít a Java-ra, de nem az (valamikor Mocha névre hallgatott). Előnye, hogy ott van a lapban, de ez egyben a hátránya is: bárki betekinthez programozói munkásságunkba. Ráadásul lassú is. De van, létezik, használható. A Java alkalmazások csak a friss kiadású böngészőprogramokkal használhatók, s van olyan is, amelyik csak a Javascriptet érti.

Teljesen más utakon halad a

9.2 VRML

azaz a **Virtual Reality Markup Language**. Ez egy háromdimenziós világot ír le, amelyben mászkálhatunk, foroghatunk, közelre és távolra tekinthetünk, környezetünket pedig alakíthatjuk. Aki játszott már DOOM-szerű játékkal, az tudja, milyen érzés. Ez már nem lap, nem kiadvány! Ez világ! Saját személyes, egyéni világ építhető fel.

A VRML jelenleg viszonylag egyszerű térbeli alapelemekkel dolgozik, de árnyékolás és perspektíva kezdettől van benne. A generált világok még elég DOOM-stílusúak, s még így is nyög tőle a vonal és a gép. Egy VRML "lap" megjelenítéséhez jóval nagyobb teljesítmény szükséges, mint egy HTML laphoz.

A WWW világához úgy kapcsolódik a VRML, hogy ennek böngészői felimerik a HTML lapokat és vagy maguk dolgozzák fel azt, vagy átadják egy HTML olvasónak. A VRML világában egy HTML lap csak "falragasz".

9.3 HONLAP HOLNAP

Végetért a honlapiskola. Remélem megérte a szöveg elolvasása és a példák megfejtése. Ha az Internet világa újabb oldalakkal gazdagodik, az mindenkinek jó.

Honlapunk címét elektronikus leveleinkben (a .signature állományban) és akár névjegykártyánkon is feltüntethetjük. Bejegyeztethetjük a Hungary Network Ki-Kicsoda

lapjára <http://www.hungary.com/kiki/>, vagy más szolgáltatókhoz. Más keresőszolgáltatások viszont hamarosan maguktól is rátalálnak a lapunkra. A látogatók száma eleinte kevés lesz, de ha jó az anyagunk, hamarosan egymás sarkát tapossák majd.

A jó lap nem avul el, azaz illik rendszeresen a karbantartással foglalkozni. Az elavult anyagokat és linkeket törölni kell, az újdonságokat és a még fejlesztés alatt álló oldalakat pedig külön érdemes jelezni, hogy a visszajáró látogatók könnyebben megtalálják. Ha véletlenül költözünk, akkor a régi címen (ha lehetséges) hagyjunk egy lapot. Ezen jó nagy betűkkel jelentsük be, hogy elköltöztünk, az új címünk pedig ez és ez, amit persze kattinthatóan építünk be.

Van egy-két dolog, amit nem szerencsés az oldalakra feltenni. A kimondottan reklámra szakosodott szolgáltatókon kívül a reklám általában tiltott tevékenység. Nem illik másokat megsérteni, bosszantani, és politizálni sem közkedvelt dolog. Ne tegyünk ki olyan anyagokat sem, amit a törvény tilt – vagy az etikett. Ezeket egy kirakatba sem tennénk ki – és a honlap nem más, mint egy elektronikus kirakat. A rendszergazdától érdeklődjünk, mit tekint ő nem kívánt anyagnak.

Mindezek nem lehetnek túl szigorú megkötések a végtelen fantázia számára. S lám, a megvalósítás nem is oly bonyolult. Nincs mese, bele kell vágni a lapalapításba!

10 FÜGGELÉK

10.1 HTML LEÍRÓK

A lapszerkezet

- `<html> ... </html>` A lap HTML része. (Általában az első sortól az utolsóig.)
- `<head> ... </head>` Fejlécinformációk.
- `<body> ... </body>` A lap törzse.
 - `background=(file|url)` Háttérkép.
 - `bgcolor=XXXXXX` Háttérszín.
 - `text=XXXXXX` Szövegszín.
 - `link=XXXXXX` Meg nem nézett link színe.
 - `vlink=XXXXXX` Megnézett link színe.
 - `alink=XXXXXX` Épp most kiválasztott link színe.
- `<!-- ... -->` Megjegyzés.

Keretek

- `<frameset> ... </frameset>` Keretleírás.
 - `rows="p1,p2,...,*"` Osztás vízszintes vonallal a megadott pixel(ek)nél.
 - `cols="p1,p2,...,*"` Osztás függőleges vonallal a megadott pixel(ek)nél.
- `<frame>` Ablak a kereten belül.
 - `src=(file|url)` A lap forrása.
 - `name=(file|url)` Az ablak neve.
 - `marginheight=[pixel]` A margó magassága.
 - `marginwidth=[pixel]` A margó szélessége.
 - `scrolling=(yes|no|auto)` Az ablak görgethető (mindig|soha|ha nagy).
 - `noresize` Az ablak nem méretezhető át.
- `<noframes> ... </noframes>` Kereteket nem kezelő böngészőknek szóló rész.

Fejlécelemek

- `<title> ... </title>` A fejlécben megjelenő szöveg.

Címsorok

- `<h1> ... </h1>` Cím. További címjelek: h2, h3, h4, h5, h6.
 - `align=(left|center|right)` Elhelyezés.

Elhatárolók

- `<p> ... </p>` Bekezdés.
 - `align=(left|center|right)` Elhelyezés.
- `
` Sordobás.
- `<nobr> ... </nobr>` Nem engedi megtörni a sort a lap szélén.
- `<wbr> ... </wbr>` A szöveg újsor karakterei hatásosak.
- `<hr>` Határolóvonal.
 - `size=[pixel]` Vastagság.
 - `align=(left|right|center)` Elhelyezés.
 - `width=([pixel] | [%])` Hossz.

Táblázatok

- `<table> ... </table>` A táblázat.
 - `border=[pixel]` Keretszélesség.
 - `width=([pixel] | [%])` Táblázatszélesség.
- `<caption> ... </caption>` Táblajegyzet.
 - `align=(left|right|center)` Elhelyezés.
- `<tr> ... </tr>` Sorváltás a táblázatban.
 - `align=(left|right|center)` Elhelyezés.
 - `valign=(top|middle|bottom|baseline)` Elhelyezés cellán belül.
- `<th> ... </th>` Oszlopváltás a táblázatban kiemelt szöveggel.
 - `align=(left|right|center)` Elhelyezés.
 - `valign=(top|middle|bottom|baseline)` Elhelyezés cellán belül.
 - `nowrap` Nem engedi a tördelést.
 - `colspan=[columns]` Több oszlopnyi széles bejegyzés.
 - `rowspan=[rows]` Több sornyi hosszú bejegyzés.
 - `width=([pixel] | [%])` Cellaszélesség.
- `<td> ... </td>` Oszlopváltás a táblázatban.
 - `align=(left|right|center)` Elhelyezés.
 - `valign=(top|middle|bottom|baseline)` Elhelyezés cellán belül.
 - `nowrap` Nem engedi a tördelést.

- **colspan=[columns]** Több oszloponyi széles bejegyzés.
- **rowspan=[rows]** Több sornyi hosszú bejegyzés.
- **width=(**[pixel]**||**[%]**)** Cellaszélesség.

Listák

- ** ... ** Számozott lista.
- ** ... ** Számozatlan lista.
- **<menu> ... </menu>** Menülista.
- **<dir> ... </dir>** Könyvtárlista alak.
- **** Listaelem.
 - **type=[A|a|I|i|1]** Elemjelölő alakja (számozott listáknál).
 - **type=[disc|circle|square]** Elemjelölő alakja (számozatlan listáknál).
- **<dl> ... </dl>** Definíció.
 - **<dt>** Definíció fejrésze.
 - **<dd>** Definíció törzsrésze.

Tördelés

- **<pre> ... </pre>** Előre tördelt szöveg.
 - **width=[chars]** Szöveg szélessége betűben.

Kapcsolatok

- **<a> ... ** Kapcsolatjelzés.
 - **href=(file|url)** Link.
 - **name=** Címke.
- **** Beépített kép jelzése.
 - **src=(file|url)** A kép forrása.
 - **href=(file|url)** Link.
 - **align=(left|right|top|texttop|middle|absmiddle|baseline|bottom|absbottom)** Elhelyezés.
 - **width=(**[pixel]**||**[%]**)** Képszélesség.
 - **height=(**[pixel]**||**[%]**)** Képmagasság.
 - **vspace=(**[pixel]**||**[%]**)** Kép környezetének szélessége.
 - **hspace=(**[pixel]**||**[%]**)** Kép környezetének magassága.
 - **border=[pixel]** Keretszélesség.

- **usemap=(file|url)** Linktérkép kapcsolat.
 - **<map> ... </map>** Linktérkép.
 - **<area>** Tartomány.
 - **shape=rect** Téglalap alakú tartomány.
 - **coords="x1,y1,x2,y2"** Koordináták.
 - **href=(file|url)** Link.
 - **nohref** Inaktív terület.

Levezézési cím

- **<address> ... </address>** Levezézési cím.

Stílusok

- *Hangsúlyos*: **Hangsúlyos**
- *Idézet*: **<CITE>Idézet</CITE>**
- *Változó neve*: **<VAR>Változó neve</VAR>**
- Erősen hangsúlyos: **Erősen hangsúlyos**
- Alsó index: **_{Alsó index}**
- Felső index: **^{Felső index}**
- Forráskód: **<CODE>Forráskód</CODE>**
- Példa: **<SAMP>Példa</SAMP>**
- Begépelte szöveg: **<KBD>Begépelte szöveg</KBD>**
- Nagyított kiírás: **<BIG>Nagyított kiírás</BIG>**
- Kicsinyített kiírás: **<SMALL>Kicsinyített kiírás</SMALL>**

- **Hét** egyre kisebb különböző méretű betűkészlet választható:
** ... **.

- Alapértelmezett betűméret: **<BASEFONT size=-2>méret</BASEFONT>**

10.2 KÜLÖNLEGES BETŰK LEÍRÓI

"	" idézőjel	¯	¯ felső vonal
&	& kereskedelmi "és"	°	° fok
<	< kisebb, mint	±	± plusz-minusz jel
>	> nagyobb, mint	²	² négyzet
 	nem vágható szóköz	³	³ köb
¡	! fordított felkiáltójel	´	´ felsővessző
¢	¢ amerikai cent	µ	μ mikro
£	£ angol font	¶	¶ bekezdéshatároló

¤t;	⌘ általános pénzjel	·	· kis középső pont
¥	¥ japán yen	¸	˘ "farkinca"
¦	¡ függőleges vonal	¹	¹ első hatvány
§	§ paragrafusjel	º	º hímnem
¨	¨ kétpont (umlaut)	»	» belső idézőjel (vég)
©	© copyright	¼	¼ egynegyed
ª	ª nőnem jele	½	½ fél
«	« belső idézőjel (kezdő)	¾	¾ háromnegyed
¬	¬ logikai nem	¿	¿ fordított kérdőjel
­	– kötőjel	×	× szorzásjel
®	® kereskedelmi védjél	÷	÷ osztásjel

10.3 ÉKEZETES BETŰK LEÍRÓI

Æ	Æ	æ	æ	Ï	ï	ï	ï
Á	Á	á	á	Ñ	Ñ	ñ	ñ
Â	Â	â	â	Ó	Ó	ó	ó
À	À	à	à	Ô	Ô	ô	ô
Å	Å	å	å	Ò	Ò	ò	ò
Ã	Ã	ã	ã	Ø	Ø	ø	ø
Ä	Ä	ä	ä	Õ	Õ	õ	õ
Ç	Ç	ç	ç	Ö	Ö	ö	ö
Ð	Ð	ð	ð	ß	ß		
É	É	é	é	Þ	Þ	þ	þ
Ê	Ê	ê	ê	Ú	Ú	ú	ú
È	È	è	è	Û	Û	û	û
Ë	Ë	ë	ë	Ù	Ù	ù	ù
Í	Í	í	í	Ü	Ü	ü	ü
Î	Î	î	î	Ý	Ý	ý	ý
Ì	Ì	ì	ì	ÿ	ÿ		

MEGJEGYZÉS

Ha kód szerint akarunk valamit beírni, úgy az & jel után egy # jelet kell írunk, majd a kódot decimálisan és ugyanúgy a ; jellel kell zárunk, mint más speciális jeleket.

10.4 SZÍNKÓDOK

	R	G	B		R	G	B
Fekete	#000000			Sötétszürke	#545454		
Kék	#0000A8			Halványkék	#5454FF		
Zöld	#00A800			Halványzöld	#54FF54		
Ciánkék	#00A8A8			Halványcián	#54FFFF		
Vörös	#A80000			Halványvörös	#FF5454		
Lila	#A800A8			Halványlila	#FF54FF		
Barna	#A85400			Sárga	#FFFF54		
Világosszürke	#A8A8A8			Fehér	#FFFFFF		

MEGJEGYZÉS

A további színeket innen kiindulva könnyű kikeverni. Bizonyos megjelenítők azonban nem képesek a 24 bites színbontásra, ezek 16, 8 vagy 4 bite alakítják át a kapott értékeket. Ügyeljünk arra, hogy egy ilyen konverzió nem tege olvashatatlanná lapunkat! (Nézzük meg egy 256 színű (VGA), vagy szürkeáryalatos (monoVGA) monitoron!)

11 PÉLDALAP

```
<!--#exec cmd="./.ct index"-->
<!--#exec cmd="./.lg index"-->
```

```
<!-- A lap elején két script indul, az egyik a számlálót növeli, -->
<!-- a másik pedig a látogatók gépeinek neveit gyűjti (hiúságból). ;- ) -->
```

```
<head>
<title>Ali's Homepage</title>
</head>
```

```
<!-- Ez itt a fej, benne a fejlécszöveg. -->
```

```
<body background="bg.gif" text="#ffffff" link="#ffff00" vlink="#00ffff">
```

```
<!-- A lapnak kékes árnyalatú tapétája lesz, de míg az be nem töltődik -->
<!-- a háttér fekete. Az írás színe fehér, a linkek meglátogatásuk -->
<!-- előtt sárgák, utána világoskékek lesznek -->
```

```
<map name="facemap">
```

```
<area shape="rect" coords="195,120,254,159" href="monitor.html">
<area shape="rect" coords="100, 20,180, 80" href="friend/il.jpg">
<area shape="rect" coords=" 0, 0,319,199" nohref>
```

```
</map>
```

```
<!-- Egy "facemap" nevű tartománylistát készítünk egy képhez. -->
<!-- Az egyik esetben egy helyi lapot tölt be, a másik esetben -->
<!-- egy helyi könyvtárban lévő képet. A kép többi része érzéketlen. -->
```

```
<table>
```

```
<!-- Táblázat segítségével felosztjuk a lap további részét. -->
<!-- Miután keretszélességet nem adtunk meg, az nulla, -->
<!-- azaz láthatatlan. -->
```

```
<td width=110>
```

```
<!-- Az első cella szélessége 110 pixel, de egyébként üresen marad. -->
```

```
<td valign=top>
```

```
<!-- A második cella elfoglalja a kép további részét. -->
<!-- A cellában minden a felső vonalhoz igazodik -->
```

```

```

```
<!-- Itt a kép, amihez a tartománylista tartozik. -->
<!-- Nincs kerete és aki grafika nélkül nézi, semmit nem lát belőle, -->
<!-- meg egy [IMAGE] felíratot sem, mert az idézőjelek közt semmi sincs. -->
```

```
<h5>It is high time I developed a homepage... :-(</h5>
```

```
<!-- Ez egy kisbetűs kiírás, azaz inkább morgolódás... ;-)>
```

```
<h1>Ali's Homepage</h1>
```

```
<!-- Ez viszont nagybetűs, mivel a lap főcíme. -->
```

```
<!--#exec cmd="echo 'This page has been accessed <b>`cat .ct.index`'</b>
times since 11/07/96.'"-->
```

```
<!-- Itt írjuk ki a számláló értékét, a hozzá tartozó szöveggel együtt. -->
```

```
<p>
```

```
<!-- Ez az elhatárolás azért kell, hogy ne csusszon rá -->
<!-- a következő blokkra a kiírás. -->
```

```
<tr>
```

```
<!-- Vége a cellának és ennek a cellasornak is, új kezdődik. -->
```

```
<td valign=top width=110>
```

```
<!-- Ez a cella ismét 110 pixel széles és -->
<!-- ami benne lesz, az a felső vonalhoz igazodik. -->
```

```
<a href="cv.txt" ></a>
<a href="intro.html" ></a>
<a href="story.html" ></a>
<a href="friend.html" ></a>
<a href="work.html" ></a>
<a href="yellow.html" ></a>
<a href="miskolc.html"></a>
<a href="hungary.html"></a>
<a href="gb/"> </a>
```

```
<!-- Ezek apró kattintható ikonok. A sor első fele a link, -->
<!-- utána maga az ikon jön képként és egybetűs karakteres megfelelővel -->
```

```
<!-- majd lezárjuk a linket. A linkek helyi állományokra mutatnak -->
<!-- illetve az egyik egy könyvtárra, ami mögött egy index.html -->
<!-- található, így azt nem kellett kiírni (kedvező esetben). -->
```

```
<td valign=top>
```

```
<!-- Itt az újabb cella, az előbbi mellett, ez is a felső sorhoz igazít. -->
```

```
<h2>
```

```
<a href="cv.txt">Curriculum Vitae / Önéletrajz</a><br>
<a href="intro.html">Introduction / Bemutakozás</a><br>
<a href="story.html">Stories / Történetek</a><br>
<a href="work.html">Works / Munkák</a><br>
<a href="ref.txt">References / Referenciák</a><br>
<a href="friend.html">Friends / Barátaim</a><br>
<a href="yellow.html">Links / Címjegyzék</a><br>
<a href="gb/">Guestbook / Vendégkönyv</a><br>
</h2>
```

```
<!-- Kisebb, 2-es méretű betűvel írja ki a szöveget. -->
<!-- Minden sor egy link, hasonlóan az előző ikonos megoldáshoz, -->
<!-- csak itt a szövegre lehet kattintani. -->
<!-- A sorokat, hogy össze ne keveredjenek, br választja el -->
```

```
<strong>
```

```
<address>
```

```
Perlaki Attila<br>
```

```
3527 <a href="miskolc.html">MISKOLC</a> Selyemrét 26<br>
```

```
<a href="hungary.html">HUNGARY</a>, <a href="europe.html">EUROPE</a><br>
```

```
Internet:
```

```
<a href="mailto:irtpa@gold.uni-miskolc.hu">irtpa@gold.uni-miskolc.hu</a>,<br>
```

```
h7476per@ella.hu<br>
```

```
FidoNet: 2:370/23<br>
```

```
</address>
```

```
</strong>
```

```
<!-- A címet address jelzők között illik megadni, a strong pedig -->
<!-- még jobban kiemeli ezt. A címben néhány szóhoz link -->
<!-- kapcsolódik, illetve a "kötelező" mailto link is -->
<!-- szerepel az internet címnél. A sorokat itt is br tördeli. -->
```

```
<p>
```

```
</table>
```

```
<!-- Ezt a táblázatot befejeztük. -->
```

```
<a href="http://www.lib.uni-miskolc.hu/imho"></a><br>
```

<!-- Itt egy nagyobb kép szerepel távoli linkkel és karakteres leíróval. -->

```
<table>
<td width=110>
<td>
<h6>
<!--#exec cmd="echo 'LAST MODIFIED AT ' $LAST_MODIFIED"-->
</h6>
</table>
```

<!-- Itt egy újabb táblázat szerepel, ami a fentihez hasonló -->
<!-- paraméterekkel rendelkeznek. Nem véletlen, hiszen -->
<!-- megjelenésében illeszkednie kell ahhoz. -->
<!-- A benne rejlő scripthívás az utolsó módosítás dátumát írja ki. -->

</body>

<!-- Itt a vége, fuss el véle! ;-)-->

12 TARTALOMJEGYZÉK

1	Bevezetés	1
2	Mi kell a honlaphoz?	2
2.1	Gép	2
2.2	Internet kapcsolat	2
2.3	Account	2
2.4	httpd	2
2.5	A public_html könyvtár	3
2.6	Elérési jogok	3
2.7	Az index.html	3
2.8	A lap címe	4
3	Az első lap	5
3.1	Mivel készítsük?	5
3.2	Egy HTML lap szerkezete	5
3.3	Mi kerül a fejbe?	5
3.4	Mi kerül a törzsbe?	5
3.5	Az első kész lap	5
3.6	Mi kerüljön az első lapra?	6
4	Olvasnivalók	7
4.1	Kész szövegek bevágása	7
4.2	Konverterek	7
4.3	Kézzel szerkesztve	8
4.4	Ékezetek	9
5	Képek	11
5.1	Képformátumok	11
5.2	A méret és az elhelyezés	11
5.3	A háttérkép (background)	12
5.4	Az átviteli sebességről...	13
6	Linkek	14
6.1	Az URL felépítése	14
6.2	Protokol	14
6.3	A link jelölése	16
6.4	Hova kerülhet link a lapon?	16
6.5	Gondok a linkekkel	17

7	Szerkezet	18
7.1	Állománynevek, könyvtárak	18
8	Extrák	19
8.1	Még szebb szövegek	19
8.2	Táblázatok	19
8.3	Keretek	19
8.4	Űrlapok	20
8.5	Scriptek	20
8.6	Számláló (példa)	20
8.7	A server rendszerváltozói	21
9	Merre tovább?	22
9.1	Java	22
9.2	VRML	22
9.3	Honlap holnap	22
10	Függelék	24
10.1	HTML leírók	24
10.2	Különleges betűk leírói	27
10.3	Ékezetes betűk leírói	28
10.4	Színkódok	28
11	Példalap	28
12	Tartalomjegyzék	33