

015.503



*Nemzeti Információs
Infrastruktúra
Fejlesztési Program*

Információs Füzetek 11.10.

PERLAKI ATTILA

HOGYAN KÉSZÍT SÜNK
SAJÁT HONLAPOT A
WEBEN?

*Budapest
1996*

Perlaki Attila

143522

Hogyan készítsünk saját honlapot a Weben?

II. 10.

MTAK



N.I.F.

Budapest, 1996

Bevezetés / 7

Mi kell a honlaphoz? / 9

Hogyan készítsünk saját honlapot a Weben?

Extrák / 32

Még szebb szövegek / 32

Táblázatok / 32

Keretek / 33

Űrlapok / 33

Scriptek / 33

II. 10.

Merre tovább? / 36

Java / 36

VRML / 36

Honlap honlap / 37

Pár hasznos UNIX parancs DOS motoron

Függelék / 42

HTML írók / 42

Különböző betű írói / 45

Ekezetes betű írói / 46

Szövegszerkesztő UNIX alatt / 47

Színkezelés

Képműködés

GIF-működés

MTAK



Példalap / 52

Ajánlott irodalom / 55

N.I.I.F.

Budapest, 1996

gépelt 015503
128200
2578A
NIIF Információs Füzetek II. 10.

© Perlaki Attila (irtpa@gold.uni-miskolc.hu)

<<http://www.lib.uni-miskolc.hu/ali>>

Sorozatszerkesztők:

Drótos László (kondrot@gold.uni-miskolc.hu)

Kokas Károly (kokas@bibl.u-szeged.hu)

Lektor:

Drótos László (kondrot@gold.uni-miskolc.hu)

MAGYAR
AKADÉMIA
KÖNYVTÁRA

ISBN 963 02 9998 4

ISSN 1219-5472

Kiadja a Nemzeti Információs Infrastruktúra Fejlesztési Program Koordinációs Iroda

NIIFKI vezetője: Nagy Miklós

A kiadásban közreműködött: Kornétás Kiadó

Ügyvezető igazgató: Puzstay Sándor

Műszaki szerkesztő: Gáspár Imre

Nyomta: Komáromi Nyomda és Kiadó Kft., Komárom

Felelős vezető: Kovács Jánosné ügyvezető igazgató

M. TUD. AKADÉMIA KÖNYVTÁRA
Könyvleltár 10.14.../19 97. SZ

Tartalom

Bevezetés

Bevezetés / 7

Mi kell a honlaphoz? / 9

Az első lap / 13

Olvasnivalók / 15

Képek / 20

Linkek / 24

Szerkezet / 30

Extrák / 32

Még szebb szövegek / 32

Táblázatok / 32

Keretek / 33

Űrlapok / 33

Scriptek / 33

Merre tovább? / 36

Java / 36

VRML / 36

Honlap holnap / 37

Pár hasznos UNIX parancs DOS motorosoknak / 38

Függelék / 42

HTML leírók / 42

Különleges betűk leírói / 45

Ékezetes betűk leírói / 46

Szövegszerkesztők UNIX alatt / 47

Színkódok / 48

Képmanipulátorok / 48

GIF-manipulátorok / 50

Példalap / 52

Ajánlott irodalom / 55

Bevezetés

Kellene egy honlap, mert már az összes kollégának van. De mégis, mi az?

Töltsük be a számítógépünkre a Netscape vagy az Internet Explorer (vagy más, meglevő) **netsurfer** programunkat, amit kissé szabadon, de értelemnek megfelelően böngészőnek fogunk nevezni. Bökjük meg az **Open** gombot és írjuk be az ablakba ezt:

<http://www.lib.uni-miskolc.hu/ali>

Betöltődött? Nos, ez egy **homepage**, azaz **honlap**. Egy lap a több mint tízmillióból az Interneten. Ez a füzet azoknak készült, akik növelni szeretnék ezt a számot a saját honlapjukkal.

Honlapot intézmények és magánszemélyek egyaránt készítenek. Előbbiek prospektus és reklám mellett vagy helyett, illetve valamilyen szolgáltatást nyújtva (pl. keresőrendszerek, adattárak); utóbbiak önmaguk, vagy kedvenc dilijük bemutatása céljából. Van honlapja a Földnek, Magyarországnak, újságoknak, focicsapatoknak, a Télapónak és számos rajzfilmhősnek. A honlap a szigorú hivatalosságtól a korlátlan örületig mindent tartalmazhat - egyfajta önmegvalósítás. *Hé világ! Itt a honlapom, nézd!* És a világ megnézi!

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 egészként 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". A szabvány jelenleg a 3.2-es verziónál tart. Nem célom a "história" ismertetése, és nem is a kifejlesztés, hanem a hasznosság alapján választottam ki a fejezetek témáinak sorrendjét, ennek ellenére a 2.0-ás szabvány után bevezetett újításokat – kevés kivételtől eltekintve – csupán extrákként említem. (Az extrákkal nem árt vigyázni, mert nem minden böngészőprogram ismeri fel ezeket, ami miatt olvashatatlaná válhat egyes felhasználók számára a lapunk!)

Nem lesz szó mindenről. A cél az, hogy aki olvassa e sorokat, merjen belevágni önállóan is a honlapkészítésbe - ez nem ördögösség! Az alapok elsajátítása után pedig egyet tudok ajánlani: a **View/Source** gomb szorgos nyomogatását, ha érdekes ötlettel találkozunk barangolásaink során! A HTML lapokban talált megoldások ugyanis közkincsnek számítanak, szabadon felhasználhatók a bennük talált trükkök (a tartalom viszont nem biztos). El is menthetjük a **File/Save as/Source** kiválasztásával s ebből tanulhatunk tovább.

Vágjunk bele!

Mi kell a honlaphoz?

Először is egy

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

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 Net-en, 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

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 rejtelmek 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, hogy nekünk a Jakab név jutott. Jelentkezzünk be! Valami hasonlót látunk majd:

Welcome at www.honlapiskola.hu

login: jakab

password: *****

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

jakab@www.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

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

A public_html könyvtár

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

```
mkdir public_html
```

Ellenőrizni kell még az

Elérési jogok

néven ismert szörnyűséget is. Arról van szó, hogy a sokak által használt és a hálózatra kapcsolt gépeken ésszerűen korlátozva van, ki mit írhat és olvashat, nehogy vicces kedvű t-boyok 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 írnák, 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 ilyenkor.

Megjegyzem, azért bátorkodom 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"... ;-) (*Már megint ez a jel! Mi ez? Smiley. Lesz még...*)

Kész a "telek", kezdhethetjük építeni a "házat". Először is lépünk be a "telekre":

```
cd public_html
```

Ha bent vagyunk, akkor a ház bejárata

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

A lap címe

az Interneten:

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

Akár meg is nézhetjük.

Az első lap

elkészítéséhez tudnunk kell

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 máshoz, 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, a függelékben kettőt röviden ismertetek.

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

Egy HTML lap szerkezete

a HTML "nyelv" szabályai szerint így néz ki:

```
<html>
```

```
<head>
```

```
Fej
```

```
</head>
```

```
<body>
```

```
Törzs
```

```
</body>
```

```
</html>
```

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. :-)

Mi kerül a törzsbe?

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

Ezek után Jakob számára

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@www.honlapiskola.hu

```
</pre>
```

```
</body>
```

```
</html>
```

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

Mi kerüljön az első lapra?

Amikor az ember honlapot készít, nyilván igyekszik bemutatni magát. Ezt számtalan módon meg lehet tenni, ez itt 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 teleagatni 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 továbbben az Interneten bolyongó!

Olvasnivalók

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

Kész szövegek bevágása

a lapba. A legegyszerűbb, ha van egy "primitív" szövegszerkesztővel elkészített, általában .txt végződést 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 zárjunk `<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 őket 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 forrás-szöveget 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** megtúrt a web-lapokon (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

Konverterek

Annyiféle formátum van már, hogy az elképesztő! Reménytelen vállalkozás tehát univerzális átalakítót keresni vagy írni. Nem is érdemes. Léteznek eredményes kísérletek olyan állományformátumokra, amelyek az összes elképzelhető szerkesztési lehetőségre fel vannak készítve; 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 például ismeri.)

A már megszerkesztett szöveget tehát ebben a formátumban újra el kell mentenünk. Utána pedig keresni kell egy olyan programot, amelyik .rtf állományból .htm(1) 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

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 (zárnival 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 kezdetét (pl. párbeszédeknel) a
 jelet. Ez üres sort nem hagy.

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

Egy hosszabb szövegben valószínűleg fejezet és alfejezetcímek is előfordulnak, ezeket a <h1>, <h2>, ..., <h6> jelekkel emelhetjük ki. Figyelem, ezt a jelzést zárnival 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 címeket, ha ez nem tetszik, a <center> jellel kérhetjük a középreigazítást. Ezt a jelzést is zárnival kell! Összevonható, sőt illik is összevonni a címjelzővel, ekkor így alakul: <h1 align=center> és értelemszerűen elég csak így zárnival: </h1>.

A szövegben belül **vastag** és *dőlt* betűs kiemeléssel élhetünk, erre a és az <i> jelek szolgálnak és ezeket is zárnival 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 *idézhetünk* is a <cite> jelzővel. Levélcímünket pedig illik <address> jelzők közé zárnival - 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 rejtjelmeit.
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@www.honlapiskola.hu<br>
</address>
</body>
</html>

```

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 az ô és û használt, ezek azok a "kalapos" betűk, amelyeket kis jóindulattal elfogadhatunk (a képernyő és a nyomtató karakterkészleteinek magyarra cserélésével a kalapok helyett a rendes hosszú ékezetek is megjeleníthetők). A lélek nyugalma itt sem teljes, ugyanis a kalapos õ helyett a hullámvonalas is elterjedt - miért pont ebben lenne egység? (Halkan merem csak megjegyezni, hogy ha már szegény û kalapos, logikusabbnak tűnik az õ-re is kalapot húzni...)

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

árvíztűrő tükörfúrógép
 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
ű	û	Ű	Û

Ékezetes szöveget így írni hosszú távon fárasztó, ezért érdemes valami konvertálót beszerezni (pl. az **rtftohtm** 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).

A kész képet el kell helyezni a lapon valahogy és valahova. Természetesen csak a törzsben. Figyelem! A kép nem mutat a szövegterületnek, nem belénkuk, csak jelezzük, hogy ott lesz. Az elhelyezés a böngésző feladata. Van ennek egy kellemetlen következménye: ha el akartuk "helyezni" egy oldalban, az nem történik, hiszen ezt szabad, ha a gazdája nem írja, akkor annak képe is bárhol is meg lehet szedgetni... A Netscape-ben ezt a jobb egérgombbal a képre kattintva tehetjük meg. Léteznek egyes segédprogramok, amelyek már képeket máshova elment együtt elmenteni. Mások anyagok azonban nem telepíthetők el, azaz egy nekünk tetsző lapot ne tegyünk ki a szöveggel, mert nem illik.

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

Pedig lehetne! Tehetjük a képet balra (alapszámolás) vagy jobbra, igazán jóluk a szöveget a felső széléhez, az aljához, vagy középre. Ez könnyes, ugyanis a kép egy szövegsor magas a böngésző logikája szerint. A kép mellé több sor írnunk lehet első nekifutásra nem lehet! (Azért van rá megoldás.)

Az elhelyezésre vonatkozó választást az `img` után elhelyezett `align` tagra írjuk, mint az angolok: jobbra elhelyezés, balra left, fölül top, le bottom, középre middle. Jobbra képe így válik a lap részévé.

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

A méret és az elhelyezés

Könyörögve kérek mindenkit: böhöm nagy képet **NE** tegyen fel huszasával a lapjára, mert nincs az a birka, aki türelemmel ki bírná várni! Azaz: mértékletesség!

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, aki meg Windows alatt nézi, amit néz, az még ennél is kevesebbet lát.

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. Ezt számos programmal el lehet végezni, a függelékben az xv (UNIX), a **PhotoStyler** (Windows) és a **Graphic Workshop** (DOS) programokat ismertetem röviden.

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 kellemtelen 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=** beszúráásával. A keret méretét pixelben adjuk meg, a 10 már szép széles.

Vannak, akik az információs szupersztrádán gyalogriksán közlekednek, grafikus felület nélkül. Ők a legszebb képünkéből 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=** beszúráásával. 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

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 kontrasztdús. 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 iméltlődő 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.) Háttérrel 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

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.

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

java

Ez egy új fejlesztés, majd megemlítem, de nem ennek az iskolának az anyaga.

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).

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. "Társbérlek" 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 állománynevet 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 állománynevet 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 állományneveket. 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.

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/cdlist.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">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">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 nyitó jel 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 illetékes tud élni, végső esetben pedig felajánlja, hogy letölti nekünk a munkakönyvtárunkba, aztán azt csinálunk vele, amit akarunk.

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 rejtjelmeit.
```

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-t 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="horgaszt.html">
```

```
<area shape="rect" coords="190, 20,210, 40" href="kocsm.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 felejtjük 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:

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)!

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.

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 rendszerezni 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 ajánlott 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 betűk és írásjelek nem, vagy csak igen körmönfont (s ezért felesleges) módon vehetők 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-hívő úgyszem 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 is ilyen!), az letöltéskor csonka, egymást felülíró állományokba futhat, ha nem vagyunk előzékenyek.

Extrák

Még szebb szövegek

Ha valamit **nagyon** szeretnénk kiemelni, akkor két lehetőségünk is van: megnövelhetjük (vagy csökkenthetjük) a betűk nagyságát, illetve átszínezhetjük 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>
```

Táblázatok

Ha sok adatot akartunk táblázatokban közölni, akkor azt régebben előre tördelt szöveggként vágtuk 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>Szentkanalas SC<td align=middle>0 : 0
<tr><td>Macskási Vasasok  <td>Akasztódombi SE<td align=middle>1 : 3
</table>
```

Nem erre találták ki, de a 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 másikon a szöveg.

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 szerver és a külvilág közé, hogy a scriptre vonatkozó és a szerver 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 "fejléc-információval" 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".

Számláló (példa)

Teendőink sorrendben:

Készítsük el counter néven az alábbi UNIX scriptet tetszésünk szerinti szöveg-szerkesztővel a public_html könyvtárba:

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

Nagyon ügyeljünk minden egyes jelre, még a szóközőkre 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 sejtheti, 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 szervereken nem működik!

A szerver rendszerváltozói

Több érdekes információt tartalmazó szabványos rendszerváltozót használnak a szerverek, 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 kiírását is elintézhetjük 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.)

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 nyűstölhetünk kérdéseinkkel. Messze kerültünk a kezdetektől és alaposan meghaladtuk a svájci atomtudósok igényeit.

A fejlődés egyik útja a

Java

A Java a HTML lapban a mozgás, az élet - szó szerint. Lapjaink ezek után számológépek, elektronikus jőpofáskodógépek, dadaista versgenerátorok, kártyajátékok, köszönőrobotok és egyéb agyatlanságok tűnhetnek 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 rutingyűtemény, 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 sorról 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

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.

Pár hasznos UNIX parancs DOS motorosoknak

Ezt csak azok olvassák el, akik a DOS-táborból kerültek hirtelen "ellenséges területre".

Ami ugyanaz

A `cd`, az `mkdir` és az `rmdir` nagyjából ugyanúgy működik, mint DOS-ban, bár a `cd` önmagában paraméter nélkül nem az aktuális könyvtár útvonalát adja vissza (erre a `pwd` parancs szolgál), hanem a bejelentkezési (bázis) könyvtárba visz minket. A `cd` (és minden más parancs mögé is) kötelező szóközt tenni, azaz például visszalépésnél nem a `cd..`, hanem a `cd ..` a helyes.

Állománylista

ls

A DOS alatt ezt `DIR` (pontosabban `DIR /W`) parancsnak ismerjük. Kilistázza a könyvtárban található állományokat. A rejtett állományokat nem. UNIX alatt ezek a ponttal kezdődő állományok. Ezekhez a

ls -a

parancs kiadása szükséges.

ls -l

Részletes listát ad, többet, mint a DOS `DIR` parancsa. Közli az állomány elérési engedélyeit, a tulajdonosát és csoportját, a létrehozás dátumát és a méretet is.

```
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/
```

Az első oszlop az állomány típusa (a "d" könyvtár, a "-" hétköznapi állomány). Utána kilenc oszlopban 3x3-as rendben a tulajdonos, a tulajdonos csoportja és a kívüljáról elérési jogai láthatók kódolva. Az "r" olvasási, a "w" írási, az "x" pedig végrehajtási jog. (Ez utóbbit könyvtárra is ki kell adni.)

A tulajdonos neve jó, ha egyezik a mi bejelentkezési nevünkkel, különben aligha tudunk felhőtlenül és zavartalanul dolgozni. A csoport neve most lényegtelen.

Ügyeljünk arra, hogy minden könyvtárunk elérési kódja a fenti példában szereplő `public_html` kódjának feleljen meg, míg a többi állományé ugyanez "x" nélkül (maradhat rajta, de okozhat zavart: `.txt` szövegállományok pl. eltérően viselkednek, "x" nélkül preformattáltak veszi őket a Netscape, míg "x" jelenlétében nem, azaz összefolytatja a szöveget...).

Az elérési kód beállítására a

`chmod`

parancs szolgál. Könyvtár szokásos beállítása:

```
chmod 755 public_html
```

Állomány szokásos beállítása:

```
chmod 644 index.html
```

Az `ls` nem mond semmit arról, mekkora hely áll még a rendelkezésünkre. Erre a

`quota -v`

parancs szolgál. Vegyük komolyan az "addig nyújtózkodj, amíg a takaród ér!" mondást! A `quota` túllépése figyelmeztetést, sőt `account` letiltást is eredményezhet.

Ha véletlenül azt üzeni vissza a gép, hogy nincs `quota` megszorítás a gépen, akkor elvileg az egész a miénk lehet, de ezt ne próbáljuk ki. A fizikai kapacitás korlátjáról is érdeklődhetünk, ezt a

`df`

parancs által szolgáltatott statisztikából olvashatjuk ki. Keressük a `/home` bejegyzést, illetve ha ez nincs, akkor a `/` bejegyzés sorát.

Állománymozgatás

`cp`

A DOS alatt ezt `COPY` parancsnak ismerjük. Fontos eltérés, hogy mindig ki kell tennünk a célt jelölő paramétert is, tehát az aktuális könyvtárba másoláskor is (ezt egyébként egyszerűen egy ponttal jelöljük).

```
cp /tmp/test.jpg .
```

A DOS alatt ismert

```
COPY *.DOC *.TXT
```

trükköt pedig felejtsük el, nem megy. Viszont ha egy sereg `.txt` állományt másolunk mondjuk egy könyvtárba, UNIX alatt nem kell a pontot beütni:

cp *txt stories/

Ha egy másik gépről szeretnénk másolni, azt az

ftp

tudja, de ez már inkább program. Paraméterként a másik gép nevét vagy IP számát írjuk be. Kérdésre login-nevet és password-öt kell megadnunk, ezek után pedig egy nagyon korlátozott parancsfelületbe jutunk. Listázni itt is az **ls** tud, állományt a **get** hoz le, a **put** meg feltesz. Ha nem szöveget mozgatunk, adjuk ki előtte a **bin** utasítást! Kilépni a **bye** paranccsal lehet. Az **ftp** ennél többet is tud, erről a ? parancs mesél - persze angolul. (Az FTP részletes leírását a legtöbb Internet könyvben megtaláljuk - már magyarul is.)

mv

A DOS alatt ezt **REN** parancsnak ismerjük. Sajnos ez sem ismeri a

REN *.DOC *.TXT

trükköt. Könyvtárba viszont ezzel is lehet egyszerre több állományt mozgatni.

rm

A DOS alatt ezt **DEL** parancsnak ismerjük.

Nagyon fontos eltérés, hogy UNIX alatt nincs mód a letörölt állományok visszahozatalára!

UNIX alatt egyes programok nagy szemetet hagynak maguk után hiba esetén. Ezek szinte mindig **core** névről vagy végződésre "hallgatnak". Sokra nem megyünk velük, érdemes eltakarítani őket, ha felbukkannak:

rm *core

Állományok megtekintése

Itt most csakis egyszerű szövegállományokról lesz szó. Egy szöveget a

cat

parancs ír a képernyőre. Ezt a DOS alatt **TYPE** parancsnak ismerjük. A hibája is ugyanaz: ha hosszú a szöveg, leszalad a képernyőről. Használjuk helyette a

more

parancsot! Ez lapokra bontja a kiírást és csak space-re lép tovább. Ha meguntuk, 'q'-val befejezhetjük.

A UNIX-ról bővebben...

A UNIX kezeléséről remek könyvek jelentek meg, így azt tanácsolom, hogy idővel ezeket is érdemes elolvasni azoknak, akik használják ezt a rendszert. Barátságosnak tűnik, de nemcsak megszokható, meg is kedvelhető.

```
... <tbody> ... </tbody> </table>
<table border="1" style="width: 100%; border-collapse: collapse;">
| Címek |
| --- |
| A társaságok            ... </table> |


... </table> ... </div>
<div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">
  ... </div>
<div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">
  ... </div>
<div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">
  ... </div>
<div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">
  ... </div>
<div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">
  ... </div>

```

Függelék

HTML leírók

A lapszerkezet

```
<html> ... </html> A lap HTML része. (Általában az egész lap.)  
<head> ... </head> Fejlécinformációk.  
<body> ... </body> A lap törzse.  
background=(file|url) Háttérkép.  
bgcolor=XXXXXXX Háttérszín.  
text=XXXXXXX Szövegszín.  
link=XXXXXXX Meg nem nézett link színe.  
vlink=XXXXXXX Megnézett link színe.  
alink=XXXXXXX 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|auto).  
noresize Az ablak nem méretezhető át.  
<noframes> ... </noframes> Kereteket nem kezelő böngészőknek szól.
```

Fejlécelemek

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

Címsorok

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

`<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 oszlopnyi 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=` Cimke.

`` 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.

Levelezési cím

`<address> ... </address>` Levelezési cím.

Fejécselemek

Címsorok

Hangsúlyos: `Hangsúlyos`

Idézet: `<CITE>Idézet</CITE>`

Változó neve: `<VAR>Változó neve</VAR>`

Erősen hangsúlyos: `Erősen hangsúlyos`

Felső index: `^{Felső index}`

Alsó index: `_{Alsó index}`

Forráskód: `<CODE>Forráskód</CODE>`

Példa: `<SAMP>Példa</SAMP>`

BEGÉPÉLT 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>`

Különleges betűk leírói

<code>&quot;</code>	"	idézőjel	<code>&macr;</code>	—	felső vonal
<code>&amp;</code>	&	kereskedelmi "és"	<code>&deg;</code>	°	fok
<code>&lt;</code>	<	kisebb, mint	<code>&plusmin;</code>	±	plusz-mínusz jel
<code>&gt;</code>	>	nagyobb, mint	<code>&sup2;</code>	²	négyzet
<code>&nbsp;</code>		nem vágható szóköz	<code>&sup3;</code>	³	köb
<code>&iexcl;</code>	¡	fordított felkiáltójel	<code>&acute;</code>	'	felső vessző
<code>&cent;</code>	¢	amerikai cent	<code>&micro;</code>	μ	mikro
<code>&pound;</code>	£	angol font	<code>&para;</code>	¶	bekezdéshatároló
<code>&curren;</code>	¤	általános pénzjel	<code>&middot;</code>	·	kis középső pont
<code>&yen;</code>	¥	japán yen	<code>&cedil;</code>	˘	farkinca
<code>&brvbar;</code>	¦	függőleges vonal	<code>&sup1;</code>	¹	első hatvány
<code>&sect;</code>	§	paragrafusjel	<code>&ordm;</code>	♠	hímnem
<code>&uml;</code>	¨	kétpont (umlaut)	<code>&raquo;</code>	»	belső idézőjel (vég)
<code>&copy;</code>	©	copyright	<code>&frac14;</code>	¼	egyenegyed
<code>&ordf;</code>	♀	nőnem	<code>&frac12;</code>	½	fél
<code>&laquo;</code>	«	belső idézőjel (kezdő)	<code>&frac34;</code>	¾	háromnegyed
<code>&not;</code>	¬	logikai tagadás	<code>&iquest;</code>	¿	fordított kérdőjel
<code>&shy;</code>	-	kötőjel	<code>&times;</code>	×	szorzásjel
<code>&reg;</code>	®	kereskedelmi védjél	<code>&div;</code>	÷	osztásjel

Ékezetes betűk leírói

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

Megjegyzés:

Ha ASCII 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.

Pico

Teljes képernyős szövegszerkesztő, a kurzornyilak és a backspace (megfelelő terminálbeállítás esetén) használhatók. Egyéb szerkesztőbillentyűk:

^X	Kilépés mentéssel.
^C	Kilépés mentés nélkül.
^R	Másik állomány beemelése.
^W	Keresés.
^Y	Lapozás felfelé.
^V	Lapozás lefelé.
^K	Sortörlés.
^U	Utoljára törölt sor visszahozása.
^O	Mentés kilépés nélkül.
^G	Blokk kijelölés

(utána ^K és ^U blokkáthelyezésre és duplikálásra használható)

Joe

Teljes képernyős szövegszerkesztő, a kurzornyilak és a backspace (megfelelő terminálbeállítás esetén) használhatók. Egyéb szerkesztőbillentyűk:

^KX	Kilépés mentéssel.
^C	Kilépés mentés nélkül.
^KR	Másik állomány beemelése.
^KF	Keresés.
^U	Lapozás felfelé.
^V	Lapozás lefelé.
^Y	Sortörlés.
^-	Utoljára törölt sor visszahozása.
^KD	Mentés kilépés nélkül.
^KB ... ^KK	Blokk kijelölés.
^KH	Help.

Megjegyzés:

Számos egyéb szövegszerkesztő létezik még.

Színkódok

Szín	RGB kód	Szín	RGB kód
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 bitre alakítják át a kapott értékeket. Ügyeljünk arra, hogy egy ilyen konverzió nem tegye olvashatatlanná lapunkat! Nézzük meg egy 256 színű (VGA), vagy szürkeárnyalatos (mono VGA) monitoron is!

Képmanipulátorok

XV

UNIX/X alatt futó program, eredetileg képnézegetőnek készült. Indítása:

```
xv &
```

FIGYELEM! Karakteres üzemmódban nem fut!

Kép betöltése

A **Load** nyomógombot választva a megjelenő ablakban kell kiválasztanunk, vagy beírni a betöltendő kép nevét.

Formátumkonverzió

Az XV a főbb képfarmátumokat ismeri. A betöltött képet más formátumban a **Save** nyomógombot megnyomva lehet elmenteni. A megjelenő ablakban a GIF, JPEG,... (változó feliratú) nyomógomb mögött található listával módosíthatjuk a mentés formátumát. A program elvileg a kiterjesztést automatikusan a helyesre állítja, ez azonban egy hiba miatt nincs mindig így, ellenőrizni, esetleg javítani kell!

Méretezés

Az **Image size** nyomógomb mögötti listában dupla, fél, stb. előre definiált méretet, vagy saját beállítását adhatunk meg. A képet ezek után még menteni kell!

Vágás

Keretes képeket automatikusan körbeszab az **autocrop**. Más esetben a képen a bal egérgomb egyszeri lenyomásával kezdhetünk téglalap alakú terület kijelölésébe. A gombot felengedve rögzül a terület, ezt azonban utólag a szélek elhúzásával még módosíthatjuk. A **Crop** megnyomásával a vágás megtörténik, ezt az **Uncrop** még vissza tudja állítani.

Photostyler

Windows 3.x alatt futó program, kimondottan képek manipulálására. Indítása a megfelelő ikonra való kattintással történik.

Kép betöltése

A **File** menü **Open** opcióját válasszuk ki, majd a megjelenő listából válasszuk ki a minket érdeklő képet.

Formátumkonverzió

A betöltött képet a **File** menü **Save as** opciójával és a kívánt formátum kiválasztásával menthetjük el új formátumban.

Méretezés

A betöltött képet a **Transform** menü **Resize** opciójával méretezhetjük át. A képet ezek után még menteni kell!

Vágás

A lebegő ikontáblából válasszuk ki az ollót, majd a kivágandó képrészlet sarkától a bal egérgombot lenyomva folyamatosan húzzuk a kívánt terület kijelöléséig az egeret. A kijelölt területre duplán rákattintva a vágás megtörténik. Ezt a **Edit** menü **Undo** opciója tudja visszavonni.

Graphic Workshop (DOS)

DOS alatt futó program, kimondottan képek manipulálására. Indítása:

GWS . EXE

Kép betöltése

A képernyőn látható listából a kurzornyalakkal keresünk rá a kívánt képre, majd ENTER.

Formátumkonverzió

A kiválasztott képre **F2**-t ütve a megjelenő ablakban választjuk ki új formátumot és ENTER.

Méretezés

A kiválasztott képre **F8**-at ütve írhatjuk be a nagyítás illetve kicsinyítés mértékét százalékban, majd **F10**-re ezt véglegesítjük. A program ezután az általunk választott formátumban menti a képet.

Vágás

A kiválasztott képre **F5**-öt ütve áttekintő (fekete-fehér) kép jelenik meg. A kurzorjilakkal mozgathatjuk a vágás határait, a bal shifttel a bal oldali és felső, a jobb shifttel a jobb oldali és alsó határolóvonalat. **F10**-re ezt véglegesítjük. A program ezután az általunk választott formátumban menti a képet.

Megjegyzés

Mindhárom program számos egyéb lehetőséggel rendelkezik, illetve más hasonló funkciójú programok is léteznek.

GIF-manipulátorok

Giftool

A giftool váltottsoros letöltődésű és áttetsző képeket állít elő. Használata:

```
giftool [kapcsolók] [kép]
```

```
giftool (-p|-c|-B) [kapcsolók] [képek]
```

- B Felülírja a bemeneti képet.
- i Váltottsoros képet állít elő.
- +i Váltottsoros képből normál képet állít elő.
- p Információt ad a kép(ek)ről.
- c Kiírja a képhez mellékelte megjegyzést.
- +c Megjegyzést mellékel a képhez.
- C Törli a megjegyzést.
- o A kimeneti képállomány neve.
- rgb name Az áttetszővé teendő szín neve.
- rgb ##,##,## Az áttetszővé teendő szín RGB kódja.
- ### Az áttetszővé teendő szín sorszáma.

Példa: A könyvtár összes GIF képét váltottsorossá tesszük:

```
giftool -B -i *.gif
```

Whirlgif

A whirlgif animált képeket állít elő. Használata:

```
whirlgif [-o kimenetkép] [-loop [számszor]] [-time #várakozás] [!<br>[-i listaállomány] kép1 [-time #várakozás] kép2 ...</pre></div><div data-bbox="92 151 757 227" data-label="List-Group"><ul style="list-style-type: none; padding-left: 0;"><li>-o Az animált (kimeneti) kép neve.</li><li>-loop Az ismétlések száma (ha elmarad, az végtelen ciklus).</li><li>-time Várakozási idő az egyes képek közt.</li><li>-i Az animáció fázisait tartalmazó lista neve.</li></ul></div><div data-bbox="440 950 470 966" data-label="Page-Footer"><p>51</p></div>
```

Példalap

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

<!-- Itt 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 készítettünk 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 kerétszélességet nem adtunk meg, az nulla, -- >
<!-- azaz láthatatlan. -->

<td width=110>
<!-- Az első cella szélessége 110 pixel, de egyébként üres. -->

<td valign=top>
<!-- A második cella elfoglalja a kép további részét. -->
<!-- A cellában minden a felső vonalhoz igazodik -->

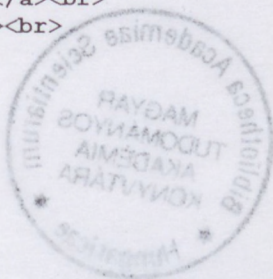


<!-- Itt a kép, amihez a tartománylista tartozik. -->
<!-- Nincs kerete és aki grafika nélkül nézi, -->
<!-- semmit nem lát belőle, még egy [IMAGE] felíratot sem, -->
<!-- mert az idézőjelek közt semmi sincs. -->
```

```

<h5>It is high time I developed a homepage... :-(</h5>
<!-- Ez egy kisbetűs kiírás, azaz inkább morgolóadás... ;-)<!-- -->
<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 íratjuk ki a számláló értékét, -->
<!-- a hozzá tartozó szöveggel együtt. -->
<p>
<!-- Ez az elhatárolás azért kell, hogy ne csússzon 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 "leírásával", -->
<!-- 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 méretu betűkkel írja ki a szöveget. -->
<!-- Minden sor egy link, hasonlóan az előző 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>,
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 rendelkezik. Nem véletlen, hiszen --->
<!-- megjelenésében illeszkednie kell ahhoz. -->
<!-- A benne hívott script az utolsó módosítás dátumát írja ki. -->
</body>

<!-- Itt a vége, fuss el véle! ;- ) -->

```



Ajánlott irodalom

Füstös János: **World Wide Web Internet bevezetés a hálózati információszolgáltató rendszer tervezésébe és használatába.** - Szak Kiadó Kft. Bicske, 1995

Hargittai Péter: **Internet haladóknak.** - LSI Budapest, 1996

Magyar nyelvű leírások. WWW-vel kapcsolatos dokumentumok. -

<http://majsola.iit.uni-miskolc.hu/~gonda1/leirasok.html>

HTML doksi magyarul. - <http://goliat.eik.bme.hu/~danya/html.html>

HTML alapismeretek. - http://www.vpsz.k.bme.hu/vpsz.k/HTML_leiras/start.html

ISO 8859/2 (Latin2) tesztoldal. - <http://scosrv.ibis.kfki.hu/test/test.html>

Infopen Library (HTML) - <http://www.eunet.hu/infopen/library.htm#html>

Martin C. Martin: **Quick Reference to HTML Tags.** -

<http://www.frc.ri.cmu.edu/~mcm/tags.html>

Csáky István - Mörk Péter: **Orlando - UNIX iskola** -

<gopher://gopher.mek.iif.hu/11/porta/szint/muszaki/szamtech/os/orlando/>

A NIIF Információs Füzetek sorozatban az alábbi köteteket tervezzük. A címek melletti csillag (*) jelzi, hogy mely füzet készült el.

I. sorozat

1. Rajta vagy már a hálózaton? (*)
2. Kalandozás a Gopherrel
3. Böngészés a WWW-vel
4. Keresgélés a WAIS-szel
5. Gyűjtögetés az FTP-vel
6. Kapcsolattartás e-mail útján az X.25-ön
7. Kapcsolattartás e-mail útján az Interneten (*)
8. Vitatkozás a USENET newsgroupokban
9. Kutatás a hálózati könyvtári katalógusokban (*)
10. Információszerzés kereskedelmi szolgáltatók adatbázisaiból
11. Beilleszkedés a hálózat virtuális világába (*)
- 12.1 A hálózat használata a molekuláris biológia területén (*)
- 12.2 A hálózat használata a környezetvédelem területén (*)
- 12.3 A hálózat használata a számítógépes grafika területén (*)
- 12.4 A hálózat használata a csillagászat és az űrkutatás területén (*)
13. A hálózat használata a könyvtárakban
14. A hálózat használata az iskolákban (*)
15. A hálózat használata elektronikus publikáláshoz
16. A hálózat használata Windowsból (*)
17. Szórakozás és játék hálózati szoftverekkel

II. sorozat

1. Hogyan csináljunk saját Gophert? (*)
2. Hogyan csináljunk saját WWW-t?
3. Hogyan csináljunk saját FTP archívumot?
4. Hogyan indítsunk saját BITNET/INTERNET levelezőcsoportot?
5. Hogyan indítsunk saját USENET newsgroupot?
6. Hogyan csináljunk saját OPAC-ot?
7. Hogyan integráljuk hálózati információs rendszereinket?
8. Hogyan védjük hálózatra kötött számítógépes rendszereinket? (*)
9. Hálózati kasszótár
10. Hogyan készítsünk saját honlapot a Weben? (*)