

E-tevékenységek 2.

TOVÁBBKÉPZÉS A FAIPARI INNOVÁCIÓVEZÉRELT VÁLLALKOZÁSFEJLESZTÉSÉRT

I. Blokk, 5. Képzési modul INFORMATIKAI ALAPÚ IRÁNYÍTÁSI ÉS KOMMUNIKÁCIÓS RENDSZEREK

Soós Sándor
egyetemi adjunktus
soossandor@inf.nyme.hu

Nyugat-magyarországi Egyetem
Faipari Mérnöki Kar
Informatikai és Gazdasági Intézet

2008. január 25-26.

Az Internet használatának két szintje

- Az Internet passzív használata
 - a World Wide Web használata
 - elektronikus levelezés
 - stb.
 - *erről beszéltünk délelőtt*
- Az Internet aktív használata
 - megjelenés az Interneten, hogyan tudjuk mindezt mi magunk is megvalósítani?
 - *erről lesz szó a hátralévő időben*
 - estére elkészítünk egy igazi saját weblapot, otthon mindenki megnézheti és megmutathatja bárkinek

Megjelenés az Interneten

- Az Internet aktív használata:
 - saját domain név bejegyzése
 - webes tárhely biztosítása
 - saját honlap elkészítése és elhelyezése
 - az oldalunk bevezetése az Internetes "ismertségbe"
 - folyamatos karbantartás, frissítés
- Intranet
 - Az Interneten használt technikákat és módszereket használhatjuk a cég belső hálózatán is.
 - Belső, csak a cégen belülről elérhető honlap, a publikus honlaptól eltérő tartalmakkal és funkciókkal.

Saját domain név bejegyzése

- Név kiválasztása
- Milyen nevet válasszunk?
- Valamelyik regisztráló szervezetnél regisztrálnunk kell a nevet
- Ha a 14 napos várakozási idő alatt senki nem emel kifogást, akkor miénk a név.
- <http://www.domain.hu/>

Webes tárhely

- Saját szerveren
- Idegen szerveren
- Ingyen, vagy pénzért
- Erre a tárhelyre fog mutatni a domain név
- Az itt elhelyezett fájlok lesznek elérhetőek az Interneten keresztül
- Ehhez egy webszerver programra van szükség
- Például:
 - <http://www.extra.hu>
 - <http://soossandor.extra.hu>
 - <http://fata.extra.hu>

Bevezetés a webes „ismertségbe”

- Ha valahol elhelyezünk egy új weblapot, arról még senki sem fog tudni. Gondoskodnunk kell róla, hogy minél többen, de legalábbis a remélt ügyfeleink, vásárlóink tudomást szerezzenek róla.
- Lehetséges megoldások:
 - a hagyományos reklámozási módszerek (újság, TV, rádió, szórólap, stb.) itt is használhatók
 - gondoljuk meg, hogy a délelőtt tárgyalt keresési módszerekkel hogyan találunk rá a saját oldalunkra
 - regisztrálás a nagyobb keresőszervereken
 - <http://webmester.lap.hu/#b13737923>
 - keresőoptimalizálás (SEO), mit jelent ez?
 - *vannak csodák: Google: Java oktatás*

Bevezetés a webes „ismertségbe”, folytatás

- regisztrálás linkgyűjteményekbe
- linkek elhelyezése, ahol lehet
 - társoldalak
 - linkcsere, a keresők díjazzák, ha egy oldalra sok link mutat
- webes reklámok
 - banner
 - e-mail reklám!!! Óvatosan!!! SPAM
- hogyan tartsuk meg az egyszer hozzánk eljutott látogatót?
 - hasznos szolgáltatások
 - új ötletek
 - rendszeres frissítés, de a jó dolgokat tartsuk meg, minden valaha fent volt lényeges dolog maradjon meg az archívumban!

Weblap elkészítése

- Miből áll egy weblap?
 - Közöséges szövegfájlok
 - HTML nyelv
- Mikor születik meg az oldal?
 - Statikus oldal
 - Előre elkészítjük az oldalakat
 - A kész HTML nyelvű fájlokat tároljuk a webtárhelyen
 - Dinamikus weblapok
 - A böngésző kérésére készül el az oldal
 - A szerveren fut egy program, ami képes generálni egy HTML nyelvű fájlt, pl. PHP interpreter
 - Például egy adatbázis lekérdezést tudunk indítani a böngésző programból. Az eredményt HTML formátumban kapjuk.

A Web fejlődése, szkriptek

- Sok érdekes hatást lehet a segítségükkel elérni, a lehetőségeknek valószínűleg csak a képzelet szab határt
- Ennél (informatikus szemmel) fontosabbnak tűnik, hogy hasznos feladatok elvégzésére is felhasználhatók, pl. űrlapok ellenőrzése.
- A szkriptek terjedésével párhuzamosan egy új fogalommal is meg kellett ismerkednie a webes közösségnek: beköszöntött a böngészőháború.
- A szoftvergyártók különböző eszközökkel megpróbálják magukhoz csalogatni a felhasználókat

A Web fejlődése

- A háború mára gyakorlatilag eldőlt:
 - A Microsoft (az amerikai bíróság szerint) visszaélve erőfölényével, a vetélytársak fölé kerekedett.
 - A mai helyzet: az Internet Explorer kb. a Windowshoz hasonló dominanciával rendelkezik, szinte már csak a Unixos/Linuxos közösség „áll ellen”.
- Ugyanakkor a Web minél széleskörűbbé válásával a csillogó-villogó személyes honlapok helyét átvették a profi, gyakran haszonszerzési céllal készült oldalak.
- Egyre nagyobb szerepet kapott a felhasználók kezelése, visszajelzések fogadása, az interaktivitás.
- Ehhez a webkiszolgálón futóprogramok, adatbázisok szükségesek.

Újabb eszközök

- Ma a legismertebb kiszolgáló-(szerver-) oldali nyelvek:
 - PHP (PHP: Hypertext Preprocessor): ingyenes, (ma már) általános célú szkriptnyelv
 - ASP (Active Server Pages): a Microsoft megoldása (csak az IIS-sel működik)
 - ASP.NET: az előbbi átültetése .NET környezetbe, gyakorlatilag teljesen megújult
- Egyéb technikák:
 - Java kisalkalmazások (appletek)
 - Flash grafikák
 - stb.

Portálok

- A technikák változatos tárházát kihasználva, a szolgáltatások széles skáláját nyújtó portálok jelentek meg.
- Ezek mindenféle közérdeklődésre számot tartó funkcióval rendelkeznek:
 - Hírek
 - Keresés
 - Fórumok
 - Szavazógépek
 - Képeslapküldés
 - Csevegés (chat)
 - **REKLÁMOK**

Saját honlap

- Az, hogy a cégünknek honlapra van szüksége, ma már nem is kérdéses. Hogy ez milyen legyen, az már annál inkább!
- Általános recept nincs, de egy-két jó tanács megfogalmazható.
- Ki készíti:
 - Külső cég: szebb, jobb, gyorsabban kész lehet
 - Belső fejlesztés: ha gyakori frissítés kell, akkor érdemes lehet

Honlapkészítési tanácsok

- **Legyünk mindig naprakészek!**
 - Az elavult információt, „halott” hivatkozásokat mielőbb szedjük le
 - Tilos olyat kiírni, hogy „két nap múlva kész az oldal, látogass vissza” (úgysem fogjuk betartani)
 - Mindezek azt az érzést kelthetik, hogy nincs frissítve az oldal → a látogatók nem térnek vissza
- **Legyünk minél inkább platform-függetlenek!**
 - Ha csak adott technológiával, adott minőségű monitoron, adott programok telepítésével nézhető meg az oldalunk, potenciális ügyfeleket veszíthetünk.
 - Gondoljunk a lassabb kapcsolattal rendelkezőkre is (esetleg több változat).

Honlapkészítési tanácsok

- **Legyen markáns arculatunk!**
 - Olyan megjelenést, arculatot válasszunk, ami egyedi, és könnyen kezelhető.
 - Ehhez kapcsolódik, hogy a tartalom is olyan legyen, amiért megéri megnézni az oldalt (információk, érdekes webes szolgáltatások stb.)
- **Ügyeljünk arra, hogy megtaláljanak minket!**
 - Reklám, könnyen megjegyezhető név
 - A címünk regisztrálása, terjesztése
 - Regisztrálás a keresőszervereken
- **Egy honlap sosincs „kész”:** rendszeresen frissíteni, ellenőrizni kell, és néhány évenként tanácsos a teljes átalakítás is (új arculat, új technológiák, új szolgáltatások stb.)

Az Internet biztonsága

- Ahogyan bármely más informatikai rendszert, az Internetet is állandó támadások érik
- A nyíltság különösen kedvez a támadóknak
- Fokozott figyelmet kell szentelni az Internetre kapcsolt rendszerek védelmének

Támadási formák

- A fenyegetések legfontosabb fajtái:
 - Vírusok, férgek
 - DoS támadások
 - Betörők (hackerek, crackerek)
- Vírusok, férgek:
 - Különbség: a féreg önálló életre képes, azaz magától terjed, nem kell hozzá hordozó program
 - Megoldás: tűzfal, állandóan futó vírusvédelmi programok
- Denial of Service támadás, DoS (szolgáltatásbénítés):
 - Már megszerzett („zombivátett”) gépekkel egyszerre olyan mennyiségű kérést juttatnak a gépünkhöz, amit az nem tud kiszolgálni
 - Védekezés: nehéz...

Támadási formák

- Ma az informatikus szlengben különbséget szoktak tenni két csoport között:
 - Hacker: nem ártó szándékú (feltöri a gépet, de nem okoz kárt, figyelmezteti a rendszergazdát)
 - Cracker: rossz szándékú
- Akármelyik „látogat meg” bennünket, nem öröm.
- Védekezés:
 - Tűzfal
 - Biztonsági javítások folyamatos telepítése
 - Új programok használata
 - Körültekintő konfigurálás, csak a szükséges programok fussanak, csak a szükséges dolgokat éri el a felhasználók
 - A rendszer állandó figyelése, naplózás

Tűzfal

- Tűzfal (firewall)
 - Eredetileg: olyan házfal, amin nincs ablak
 - Az informatikában: olyan (biztonságos) gép, vagy program, amin a védett gép vagy hálózat forgalma áthalad, és csak azok a csomagok mehetnek rajta keresztül, amelyeket kimondottan megengedünk.
 - Nem hagyja tárva-nyitva az ablakokat a külvilág felé.
 - A gyakorlatban ez azt jelenti, hogy portszintű szűrést lehet végezni, a forráscím (és a célcím) ill. további feltételek vizsgálata alapján.
- Ma már kötelező!
- Belső hálózathoz: Network Address Translation (NAT)

Védekezés

- További fontos szabály: lehetőleg ne használjunk titkosítatlan adatátviteli formákat!
- Az eddig bemutatott szolgáltatásoknak mára már mind megvannak a titkosított változataik:
 - Secure SHell(SSH): Rlogin és RSH
 - Secure CoPy(SCP): RCP
 - SecureFTP (SFTP): FTP
 - HTTP overSSL (HTTPS): HTTP (a Secure Sockets Layer(SSL) olyan kommunikációs réteg, ami a titkosítást végzi, alkalmazástól függetlenül felhasználható)
 - IMAP over SSL (IMAPS): IMAP
 - stb.

Weblap készítés alapjai

- Ennyi előkészítés után lássunk neki a weblap elkészítésének!
- Egy teljesen működő igazi weblap elkészítéséhez elegendő megismerkedni néhány alapeszközzel
 - A HTML nyelv alapjai
 - Egy HTML szerkesztő programmal
 - Az eddig tanult alapfogalmakkal

Miért érdemes megismerkedni a HTML nyelvvel?

- Nem csak weboldalak készítőinek hasznos.
- Egyike a kevés platformfüggetlen dokumentum formátumnak, ezért ha ebben a formátumban készítünk el egy dokumentumot, akkor az bármilyen operációs rendszer alatt olvasható lesz.
- Kényelmesen lehet kapcsolatokat létrehozni a különböző dokumentumok között.
- Az így elkészült dokumentum csomagokat nagyon egyszerű közzétenni az Interneten.
- Példa: közös munka pesti munkatársakkal

A HTML fogalma

- Az Interneten található információ tömeg alapvető alkotóelemei a HTML dokumentumok.
- Mi az a HTML?
 - HyperText Markup Language
 - Hipertext jelölő nyelv
- Mi az a hipertext?
 - Olyan dokumentumok hálózata, amit kereszthivatkozások szőnek át.

A keresztivatkozás

- Mi az a keresztivatkozás?
 - Egy kapcsolat egy vagy két dokumentum két pontja között.
 - Rákattintva a hivatkozásra (link) elugrunk a kapcsolat másik végére.
- Hova mutathat egy keresztivatkozás?
 - egy másik dokumentumra (az elejére)
 - bármely dokumentum egy kiválasztott pontjára

A dokumentum

- Mit értünk dokumentum alatt?
 - tetszőleges nyelvű formázott, vagy formázatlan szöveg
 - kép
 - hanganyag
 - mozgókép

 - vagy ezek tetszőleges kombinációja!
- Hogyan tudunk ilyen dokumentumot létrehozni?
 - egy néhány órás tanfolyam alatt megtanulható, most kicsit belekóstolunk

A HTML nyelv

- A HTML egy jelölő nyelv, amivel hipertext dokumentumokat tudunk készíteni.
- Nem programozási nyelv!
- Használatához nem kell tudni programozni.
- Aki meg tud írni egy egyszerű levelet a számítógépen, az HTML dokumentumokat is tud készíteni.
- Egy HTML dokumentum egy egyszerű szövegfile amibe bizonyos szabályok szerint parancsokat illesztünk.

A HTML nyelv

- Az első néhány szabály megtanulása után máris képesek vagyunk teljes értékű HTML dokumentumokat készíteni.
- Ahogy haladunk előre a HTML nyelv megismerésében, egyre érdekesebb, összetettebb, "profibb" dokumentumokat tudunk majd elkészíteni.
- Ennek megfelelően már az első óra után elkészítheti mindenki a saját honlapját, amit apránként tud fejleszteni a megtanult új lehetőségekkel.

A HTML nyelv parancsai

- A HTML parancsok formája:
 - **<parancs paraméterek> . . . </parancs>**
 - Minden parancs hatása a parancsot követő szövegrészre vonatkozik a parancs lezárásáig (**</parancs>**).
 - A parancsokat szabadon írhatjuk kis és nagybetűkkel is.
- A 3 legfontosabb HTML parancs:
 - **<HTML> . . . </HTML>**: a dokumentum eleje és vége
 - **<HEAD> . . . </HEAD>**: fejrész eleje és vége, ide tesszük a dokumentum egészére vonatkozó információkat, utasításokat.
 - **<BODY> . . . </BODY>**: a törzs eleje és vége, ide tesszük a dokumentum tartalmát.

Az első HTML dokumentumunk

```
<HTML>
```

```
Ez az első HTML dokumentumom.
```

```
</HTML>
```

- Ha ezt a 3 sort beírjuk egy szövegfile-ba, pl. **elso.html** néven, akkor azt máris megnézhetjük egy böngésző programban.
- (Egyébként ez még a `<HTML>...</HTML>` nélkül is működik! Ugyanis az egyszerű szövegfile-okat is meg tudják jeleníteni a böngészők.)

Egy igazi HTML dokumentum

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Ez a dokumentum címe</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Ez a dokumentum törzse!
```

```
</BODY>
```

```
</HTML>
```

Néhány egyszerű HTML parancs

- Néhány parancs ahhoz, hogy feldobhassuk a dokumentumot:
 - **<center>...</center>**: középre igazítás
 - **...**: félkövér szöveg
 - **<i>...</i>**: dőlt szöveg
 - **<u>...</u>**: aláhúzott szöveg
 - **<strike>...</strike>**: áthúzott szöveg
 - **_{...}**: alsóindex
 - **^{...}**: felsőindex
 - **
**: új bekezdés, enélkül minden egy bekezdés lenne, majd a böngésző program tördeli sorokra.

Szövegformázó parancsok

- Bekezdések igazítása:

`<div align="left">...</div>`

balra

`<div align="center">...</div>`

középre

`<div align="right">...</div>`

jobbra

`<div align="justify">...</div>`

sorkizárt

- Elválasztó vonal, illetve téglalap:

`<hr>`

egyszerű

`<hr width="50%" size="2" align="center" >`

formázott

abszolút (pixel), vagy
relatív (%) szélesség

Címsorok

- Különböző szintű címeket határozhatunk meg. Ezek végleges formája a böngészőtől függ, csak az biztos, hogy fokozatosan "csökkennek" a címek.

<H1>...</H1>

<H2>...</H2>

<H3>...</H3>

<H4>...</H4>

<H5>...</H5>

<H6>...</H6>

Képek előkészítése

- A kép előkészítése
 - Használható képformátumok:
 - JPEG, GIF, PNG
 - Méret és felbontás beállítása
 - Döntsük el, hogy milyen célra szánjuk a képet, mert ettől függ, hogy milyen képet érdemes feltenni a weblapra!
 - Törekedjünk a minél kisebb fileméretre a letöltés gyorsítása érdekében.
 - Külön beállíthatjuk a kép tényleges és látható méretét.
- Az Irfanview használata

Képek elnevezése

- Filenév:
 - Figyeljünk a kis és nagybetűkre, mert a Linux megkülönbözteti őket. Ez minden más file-ra is igaz!
 - Használjunk egységes formátumot és ügyeljünk erre a szerverre való másolásakor is!
- Relatív útvonal a HTML file-hoz képest
 - A HTML file-hoz képest hova helyezzük a képeket
 - ugyanabba a könyvtárba
 - egy alkönyvtárba, pl. images
 - Használhatunk abszolút útvonalat is, de a másolásakor ez problémát okoz. Helyben látszólag minden rendben, felmásolás után nem jelenik meg a kép!

Képek elhelyezése a weblapon

```

```

parancs kezdete
a képfile neve útvonallal
megjelenítési szélesség
megjelenítési magasság
alternatív szöveg
parancs vége

- Tetszés szerint tagolhatjuk sorokra a parancsot.
- A paramétereket szóközzel választjuk el egymástól.

Kereszthivatkozás elhelyezése

- Ugrás egy adott webcímre:

```
<a  
href="webcim"  
>  
a link szövege  
</a>
```

parancs kezdete
a link amire hivatkozunk,
parancs vége
ez fog megjelenni a lapon
a kereszthivatkozás vége

- Például:

```
<a href="http://inf.nyme.hu" >inf.nyme.hu</a>
```

Kereszthivatkozás elhelyezése 2

- Nem csak egy weboldal elejére ugorhatunk.
- A **href** paraméterben megadhatunk mást is:
 - **href=" ../het01/elso.html"** helyi file
 - **href="#top"** aktuális oldal tetejére
 - ezt nem minden böngésző támogatja automatikusan
 - **href="#könyvjelző"** saját könyvjelző
 - és ezek kombinációi, például:

**
Ugrás a példák aljára**

Saját könyvjelző létrehozása

- Ha egy dokumentumra ugrunk az `<a ...>` paranccsal, akkor a dokumentum elejére kerülünk.
- Amikor mi hozunk létre egy dokumentumot, akkor megadhatunk olyan pontokat, amelyekre később ugorhatunk. Ezt nevezzük könyvjelzőnek.
- Könyvjelző létrehozása:
 - ``**
 - Könyvjelzőből tetszőleges számút létrehozhatunk szabadon választott névvel (az angol ábécé betűiből), csak annyi a megkötés, hogy egy dokumentumban nem lehet két azonos nevű könyvjelző!

Könyvjelzőre ugrás

- Ugrás dokumentumon belül:
Ugrás a konyvjelzo-re
- Ugrás másik dokumentumból:
Ugrás a masik.html filebeli konyvjelzo2-re
- Ugrás másik weblap egy könyvjelzőjére:
Ugrás a www.honlap.hu -beli konyvjelzo3-ra
– Ehhez tudnom kell, hogy a www.honlap.hu készítője milyen nevű könyvjelzőt készített!

A link szövege

- Mit írhatunk a kereszthivatkozásban a `<a>` és a `` közé?
 - **Bármit**, amit egyébként beírhatunk a HTML file-ba!
 - tetszőleges formázott szöveget
 - képet
 - stb.
- Például hogyan helyezünk el takarékos módon nyomtatni való képeket a honlapunkra?
 - Készítsük elő a képeket két méretben
 - kis méretben (thumbnail-hüvelykujjköröm)
 - nagy méretben
 - Tegyük fel az oldalra a thumbnail-t, amit beállítunk kereszthivatkozásnak, ami a nagy file-ra mutat!

Fényképalbum

- Az előző dián lévő példa:
``
``
``
- Ilyen linkekből egymás mellé tehetünk tetszőleges számút. Ezzel elkészíthetünk egy fényképalbumot.

Felsorolások, listák

- Feladat: Szeretnék betenni a dokumentumba egy felsorolást, listát.
- HTML-ben háromféle listát használhatunk:

Sorszámozott:
(Ordered list)

Például:

- 1. Első listaelem**
- 2. Második listaelem**

Nem sorszámozott:
(Unordered list)

Például:

- **Első listaelem**
- **Második listaelem**

Definíciók listája,
mint a szótárban
(Definition list)

Például:

Fogalom
Definíció, leírás

Felsorolások, listák

- Hogyan tudunk ilyen listákat létrehozni?
 - Egy HTML paranccsal meghatározzuk a lista kezdetét és végét.
 - Ezzel egyben meghatározzuk a lista típusát is.
 - A listán belül egymás után megadjuk a pontokat szintén egy HTML paranccsal.
 - Listák létrehozása:
 - ``: sorszámozott lista (ordered list)
 - ``: nem sorszámozott lista (unordered list)
 - (`<dl></dl>`: definíció lista (definition list) – ezzel most nem foglalkozunk)
 - Listaelem létrehozása:
 - ``: egy lista elem ``, vagy `` típusú listához (list item)

Példák listákra

- Sorszámozott lista:

Gyümölcsök számozva:

```
<ol>
```

```
<li>alma</li>
```

```
<li>körte</li>
```

```
<li>szilva</li>
```

```
</ol>
```

- Nem sorszámozott lista:

Gyümölcsök listája:

```
<ul>
```

```
<li>alma</li>
```

```
<li>körte</li>
```

```
<li>szilva</li>
```

```
</ul>
```

- A listaelemen belül a **** és **** között használhatunk minden formázó parancsot, újabb listát is.

Táblázatok

- Ezzel az eszközzel nem csak hagyományos táblázatokat készíthetünk, hanem ez a fő eszköze az elemek elrendezésének is.
- Minden táblázat a következőképpen épül fel:
 - Táblázat eleje (**table**)
 - 1. sor kezdete (**table row - tr**)
 - 1. cella kezdete (**table data cell - td**)
 - 1. cella vége
 - 2. cella kezdete
 - 2. cella vége
 - ...
 - 1. sor vége
 - ...
 - Táblázat vége

Táblázatok, folyt.

- Ugyanez HTML kódban:

```
<table>
```

```
<tr>
```

```
    <td>Az 1. sor 1. cellája</td>
```

```
    <td>Az 1. sor 2. cellája</td>
```

```
</tr>
```

```
<tr>
```

```
    <td>A 2. sor 1. cellája</td>
```

```
    <td>A 2. sor 2. cellája</td>
```

```
</tr>
```

```
</table>
```

Táblázatok, folyt.

- Egy cellában a `<td>` és a `</td>` között minden lehet, ami egy HTML fájlban leírható, akár újabb táblázat is.
- Így kialakíthatunk egy tetszőleges oldalelrendezést néhány egymásba ágyazott táblázat segítségével.
- Természetesen mindegyik parancsnak vannak különböző paraméterei, amelyekkel befolyásolhatjuk a működését.
- A következőkben megismerkedünk a legfontosabbakkal.

A `<table>` parancs legfontosabb paramétere

- **width** (pl. `width="100%"`)
 - a táblázat szélessége abszolút (pixel), vagy relatív (%)
- **border** (pl. `border="2"`)
 - a külső keret vastagsága, ha 0, akkor nem látszik
- **align** (pl. `align="center"`, vagy `"left"`, vagy `"right"`)
 - a táblázat igazítása az őt tartalmazó területen belül
- **bgcolor** (pl. `bgcolor="red"`)
 - a háttér színe, a színek angol neveit, vagy hexakódját használhatjuk.
- **bordercolor** (pl. `bordercolor="blue"`)
 - a keretvonalak színe

A `<tr>` (table row) parancs legfontosabb paramétere

- **height** (pl. `height="100"`)
 - a sor magassága abszolút pixelben
- **align** (pl. `align="center"`, `"left"`, `"right"`, vagy `"justified"`)
 - a sor minden cellájának igazítása
- **valign** (pl. `align="top"`, `"bottom"`, vagy `"middle"`)
 - a sor minden cellájának igazítása függőlegesen
- **bgcolor** (pl. `bgcolor="red"`)
 - a sor háttér színe, a színek angol neveit, vagy hexakódját használhatjuk.

A `<td>` (table datacell) parancs legfontosabb paramétere

- **height** (pl. `height="100"`)
 - a sor magassága abszolút pixelben, egy soron belül nem lehet eltérő
- **width** (pl. `width="100%"`)
 - a cella szélessége abszolút (pixel), vagy relatív (%) a táblázat szélességéhez képest, egy oszlopban nem lehet eltérő
- **rowspan** (pl. `rowspan="2"`)
 - az adott cella két sort fogjon át
- **colspan** (pl. `colspan="2"`)
 - az adott cella két oszlopot fogjon át
- **align**, **valign**, **bgcolor** mint a `<tr>` esetében, de csak a cellára vonatkozik

Egyéb kiegészítő parancsok a táblázatokhoz

- `<caption>`: a táblázat címe
Pl. `<caption>A táblázat címe</caption>`
- `<th>`: az oszlopok felső címkéje
Pl.
`<th>Az 1. oszlop fejléce</th>`
`<th>A 2. oszlop fejléce</th>`

Jó tanácsok a táblázatok sikeres használatához

- Mindig páronként írjuk be a HTML parancsokat! Amikor leírtuk a `<table>` parancsot, akkor néhány sorral lejjebb írjuk be a `</table>` -t is, majd `<tr> .. </tr>`, `<td> .. </td>`!
- Ez általában is hasznos a HTML szerkesztéskor, de a táblázatoknál elengedhetetlen!
- Beljebb kezdéssel tagoljuk a sorokat és a cellákat!
- A `border="2"` paraméterrel rajzoljuk ki a határolóvonalakat, hogy jól lássuk a táblázatot! Amikor már minden rendben van, akkor kivehetjük a vonalakat.
- Munka közben gyakran ellenőrizzük az eredményt egy böngészőben.

Hogyan készítsünk weblapot?

- Tetszőleges texteditor (szövegszerkesztő)
 - ahogyan eddig dolgoztunk
- Grafikus webszerkesztő programok
 - Ugyanúgy dolgozunk, mint egy megszokott szövegszerkesztőben, vagy grafikus programban
 - A program állítja elő a HTML kódot
- Grafikus szövegszerkesztő program
 - A szövegszerkesztőben elkészítjük a dokumentumot, majd exportáljuk HTML formátumban
- HTML szerkesztő programok
 - HTML kódot szerkesztünk, de segítséget kapunk ehhez.

HTML szerkesztő programok

- Az eddig megtanult módon szerkeszthetjük a HTML kódot, de különböző szintű segítséget kaphatunk a programtól.
- Megnézzük a gyakorlatban.
- Sok freeware (teljesen ingyen használható) program létezik ebből a típusból. Például
 - HTML Builder XP
 - <http://www.code-builders.com/hbxp.shtml>
 - jelenleg a 6.0 beta verzió a legfrissebb, ezt fogjuk megnézni
 - MAX's HTML Beauty++ 2004
 - <http://www.htmlbeauty.com/>
 - jelenleg a 2.0-s verzió a legfrissebb

Internet a HTML nyelven túl

- Az előző diákon megismerkedtünk a HTML nyelv alapjaival.
- Ezekkel az eszközökkel csak statikus oldalakat tudunk létrehozni.
- De HTML nyelv képes befogadni számos kiegészítést, amelyek dinamikussá tehetik az oldalainkat.
- Egy igazi honlapnak használnia kell ezeket a kiegészítéseket.

Szerver oldali és kliens oldali megoldások

- A HTML nyelv kiegészítései két típusba sorolhatók
 - Szerver oldali megoldások
 - A webservert nem elégzik meg azzal, hogy az eltárolt statikus HTML oldalt elküldi a böngészőnek
 - Ő maga készíti el az oldalt tetszőleges eszközökkel, különböző programok futhatnak, dolgozhatnak együtt ehhez
 - A végeredmény egy közös HTML oldal, ezt küldjük el a böngészőnek.
 - Kliens oldali megoldások
 - A böngésző programba kerülnek bele kiegészítések, amelyek az alap HTML nyelvnél több funkcióval rendelkeznek.
- Mindkét megoldásnak vannak előnyei és hátrányai.

Szerver oldali megoldások

- Előnyök:
 - a szerver teljesen a mi kezünkben van, azt teszünk rá, amit akarunk, illetve ami szükséges
 - A klientsől semmilyen extra tudást nem várunk el
 - A program nem utazik a hálózaton, csak az eredmény
- Hátrányok:
 - csak olyan dolgokat tudunk megcsinálni, amit a legegyszerűbb kliens is képes értelmezni
- Főként a tartalom elkészítésére használjuk
- Például: PHP interpreter
 - egy teljes értékű programozási nyelv, mindenre képes, amire ma egy programozási nyelv képes
 - A program eredménye mindig egy HTML fájl.

Kliens oldali megoldások

- Előnyök:
 - A kliensek minden extra tudását felhasználhatjuk
 - Látványos megoldások, játékok, trükkök
- Hátrányok:
 - Fel kell készülnünk a sok különböző kliens korrekt kezelésére.
 - Csak olyan dolgokat tudunk megcsinálni, ami a kliens hatókörében elvégezhető
- Főként a felület elkészítésére használjuk
- Például: JavaScript, VBScript
 - apróbb-nagyobb programokat, scripteket ágyazunk bele a HTML fájlba. A böngésző értelmezi és végrehajtja ezeket. Ha tudja!

CSS - Cascading Style Sheet

- Egy nagyobb weblap sok-sok weboldalból áll, ezeket jó lenne egységes kinézetűvé tenni.
- Az alap HTML nem segítette ezt.
- A CSS bevezetésével már van lehetőség arra, hogy ne egyedileg formázzunk meg minden elemet a weblapon, hanem stílusokat hozunk létre, és azokat használjuk fel az elemek formázására.
- Ezek a stílusok később is módosíthatók, a következő letöltéskor már az új stílus szerint fog megjelenni minden weblap.
- A CSS olyan formázási lehetőségeket is kínál, amelyeket az alap HTML nem.
- Némileg böngészőfüggő a CSS használhatósága, de a mai böngésző verziók már mind támogatják.

Ingyenes webszolgáltatások

- <http://ingyenweb.lap.hu/>
 - tárhely
 - rövid URL
 - virtuális merevlemez
 - számláló
 - vendégkönyv
- Lásd még: társoldalak!

Ingyenes webszolgáltatások 2

- <http://webmester.lap.hu>
 - regisztrálás keresőkbe
 - bannercsere
 - fórum
 - levelezőlista
- Lásd még: társoldalak!

Egy igazi weblap elkészítése

- fata.extra.hu
- index.html rajta:
 - bemutatkozás
 - a két előadás anyaga
 - névsor
 - csoportkép
 - stb.