

Webprogramozás szakkör

Előadás 3. (2013.03.19)

Bevezető

HTML felelevenítés

HTML elemei

- Tag-ek, például: <html>, <h1>, ,
- Tulajdonságok, például: size, bgcolor
- Értékek, például: 4, black, #FFFFFF

Oldalkeret

```
<html>
  <head><title></title></head>
  <body>TARTALOM HELYE</body>
</html>
```

Táblázatok

```
<table>
  <tr>
    <td>1. sor, 1. cella</td>
    <td>1. sor, 2. cella</td>
    <td>1. sor, 3. cella</td>
  </tr>
  <tr>
    <td>2. sor, 1. cella</td>
    <td colspan="2">2 sor, 2-3. cellái összevonva</td>
  </tr>
</table>
```

HTML gyengeségei

Formázás: a weboldal elrendezése és a használt stílusok (színek, elválasztó vonalak, betűk... stb.) összessége

- HTML-ben is formázható az oldal, de nem elég hatékonyan
 - Minden helyre külön beírjuk, hogy például piros, félkövér szöveg legyen
 - Ez egy adott stílus, hogy piros és félkövér
- Így
 - Fölösleges plusz időtöltés a formázás
 - Sok fölösleges kód → átláthatóság
 - Sok fölösleges kód → nagyobb fájl méret
 - Összemosódik az oldal szerkezete és a formázás

CSS koncepció

CSS: egymásba ágyazott stíluslapok

- Írjuk meg egyszer, hogy milyen stílusokat szeretnénk használni
- Adjunk nevet a stílusoknak
- Ahol használni szeretnénk, ott csak a nevet kell megadni

Persze ez nem ilyen egyszerű...

CSS előnyök, célok

- Rendezettebb lesz a honlapod
- Egy sablont hozol létre, majd ha változtatni akarsz, akkor elég csak a css fájlt megváltoztatni
- Egy külön fájlba (*.css) tárolod az általad használt sablonokat
- Jegyzetömbben lehet megírni a fájlt, és pl.: "styles.css" néven kell elmenteni
- A html kódba csak hivatkoznod kell rá
- Leegyszerűsíti az oldalak forráskódjának írását
- Szétválik a tartalom a szerkezettől illetve a megjelenítéstől

CSS szabályai

A nyelv megtanulása talán könnyebb, mint a HTML, csak ennyit kell megjegyezni:

```
szelektor {  
    tulajdonság1: érték1;  
    tulajdonság2: érték2;  
}
```

Ez egy sablon, a fentebb látható elemei a következők:

A „szelektor” helyére valamilyen nevet írunk.

A „tulajdonság” helyén azt adjuk meg, hogy mit formázunk (pl. szín, betűméret).

Az „érték” esetében pedig megadjuk, hogy pl. a betűméret az mekkora is legyen pontosan.

Konkrét példával:

```
.piros_fk {  
    color: red;           // piros betűszín  
    font-weight: bold;   // vastag betűk (félkövér)  
}
```

CSS kód helye

Hova írhatunk ilyen sablonokat? 3 lehetőség van:

- Egy külön fájlba, aminek .css kiterjesztést adunk
- HTML kód HEAD részébe
- Tetszőleges HTML tag-ba

Külön fájlba írt CSS kód

Semmi mást nem kell tenni, mint az előző példánkat egy külön fájlba elhelyezni, a fájlt „css” kiterjesztéssel elmenteni (pl. style.css néven), majd importálni a fájlt a HTML kód HEAD részében (<link> tag):

```
<html>
<head>
  <title>Honlapom</title>
  <link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
</body>
</html>
```

Előnye, hogy több oldalba is importálhatjuk ugyanazt a fájlt, így minden sablon használható lesz az összes oldalon. Ide a sokat használt sablonokat helyezzük el.

HTML kód HEAD részébe írt CSS kód

Elhelyezhetünk CSS kódot az alábbi módon is, a HTML <style> tagba:

```
<html>
<head>
  <title>Honlapom</title>
  <style type="text/css">
    .piros_fk {
      color: red;
      font-weight: bold;
    }
  </style>
</head>
<body>
</body>
</html>
```

Ennek hátránya, hogy csak azon az oldalon lesz érvényes, ahol megadtuk, de érdemes lehet itt megadni sablonokat, ha azt más oldalon nem használjuk.

Tetszőleges HTML tag-ba írt CSS kód

```
<html>
<head>
  <title>Honlapom</title>
</head>
<body>
  <p style="color:red;font-weight:bold;">Szöveg</p>
</body>
</html>
```

Ekkor a HTML tag-ek „style” nevű tulajdonságában adhatjuk meg azt, amit a fenti példákban a kapcsos zárójelek közé írtunk.

CSS kiválasztók (szelektorok)

A szelektor helyén egy nevet kell megadunk, mely 3 féle lehet.

HTML tag neve

Ha egy HTML tag nevét adjuk meg, akkor az oldalunkon minden ilyen nevű HTML tag-ra érvényesek lesznek a megadott stílusok. Például az összes <h1> címsor legyen kék:

```
h1 {
    color: blue;
}
```

Azonosító

A HTML oldalunkban a tag-eknek adhatunk egy azonosítót (id):

```
<p id="alma">Szöveg</p>
```

Itt a <p> HTML tag azonosítója az „alma”. Az azonosítóknak egyedinek kell lenniük, tehát egy oldalon csak egyszer szerepelhet minden azonosító név. Ha azt szeretnénk, hogy az alma azonosítón belül lévő részre érvényes legyen egy CSS sablon, akkor a kiválasztót egy kettős kereszttel kell kezdenünk:

```
#alma {
    color: blue;
}
```

Ez tehát egy oldalon belül csak egyetlen HTML tag-ra lehet érvényes.

Tetszőleges elnevezésű sablon

A leggyakoribb, hogy egy tetszőlegesen választott nevet adunk a sablonunknak, majd ezt bármennyi alkalommal használjuk. Ekkor a CSS kiválasztót egy PONT-tal kell kezdeni:

```
.barack {
    color: blue;
}
```

Ebben az esetben is valamilyen módon meg kell adnunk, hogy hol legyen érvényes, ezt a HTML tag-ek „class” tulajdonságában tehetjük meg:

```
<p class="barack">Szöveg</p>
```

Összetett kiválasztók

Példa 1: „csak azokra a h1 elemekre legyen érvényes, ahol alma sablont is megadtunk”.

```
CSS:
h1.alma {
    color: blue;
}

HTML:
<h1>Erre nem lesz érvényes, nem lesz kék a szöveg</h1>
<h1 class="alma">Erre viszont igen</h1>
```

Példa 2: „csak azokra a h1 elemekre legyen érvényes, ami egy alma sablonon belül van”.

CSS:

```
.alma p {
    color: blue;
}
```

HTML:

```
<p>Erre nem lesz érvényes</p>
<p class="alma">Erre sem</p>
<h1 class="alma">Erre nem <p>de erre már igen</p> itt újra nem</h1>
```

Kiválasztók csoportosítása

Ha ugyanazokat a stílusokat szeretnénk több kiválasztóval alkalmazni, akkor vesszővel elválasztva adjuk meg a kiválasztókat. Például a h1 és h2-es címsorok legyenek aláhúzottak:

```
h1, h2 {
    text-decoration: underline;
}
```

További megjegyzések

Fontos megjegyezni szabályként, hogy minden HTML kódrészletre az a sablon lesz érvényes, ami a „legközelebb van hozzá”. Tehát ha egy külső fájlban megadnánk, hogy piros legyen a szöveg, az importálás után pedig megadjuk, hogy kék legyen, akkor kék lesz, hiszen azt adtuk meg később, és felülírja a később megadott a korábbi.

Előfordulhat olyan eset, hogy nem tudjuk milyen HTML tag-ot lehetne használni, hogy megadhassuk a „class” tulajdonságot. Ekkor a nevű, semleges, vagyis semmit sem módosító HTML tag használata célszerű, például:

```
<p class="barack">Szöveg <span class="szilva">Ez más stílusú</span> ez újra brack stílusú</p>
```

HTML vs CSS

Hogyan állítottuk be egy szövegre HTML-ben, hogy piros színű és félkövér legyen?

```
<b><font color="red">Szöveg</font></b>
```

Egyszer leírva még kényelmes így is, de ha egy oldalon akár 50-100x meg kell ismételni, akkor már nem annyira, főleg, ha nem ilyen egyszerű formázást szeretnénk.

Hogyan adhatjuk meg CSS-ben?

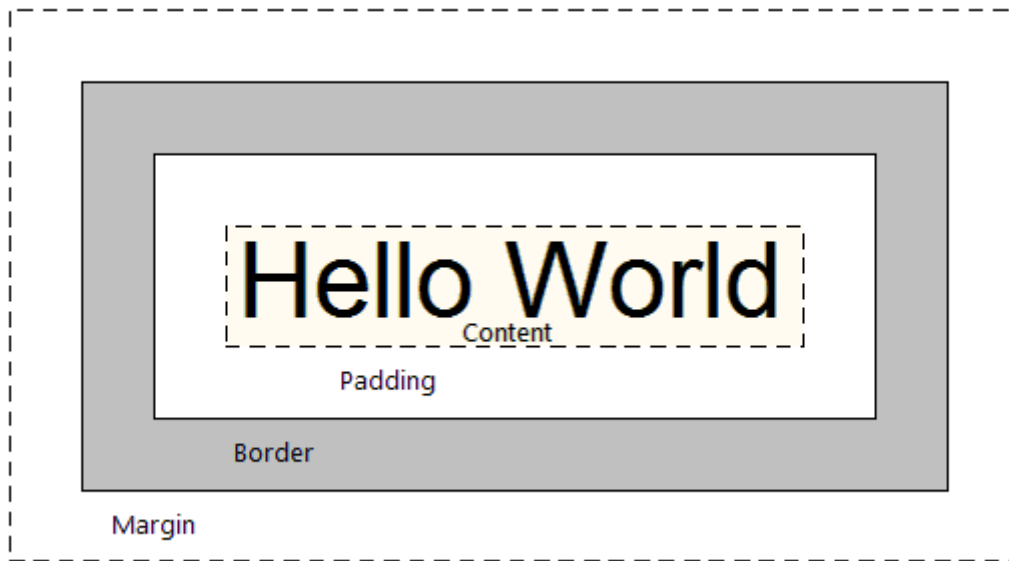
CSS:

```
.piros_fk {
    color: red;
    font-weight: bold;
}
```

HTML:

```
<span class="piros_fk">Szöveg</span>
```

Dobozmodell



Egy adott szöveg gyakorlatilag egy szabályos doboznyi helyet foglal el. A szöveget körülveszi a kitöltés (padding), annak a határán található a körvonal (border), majd a körvonal és az őt befoglaló doboz (szülő HTML elem) közti távolság a margó (margin). Mind a négy oldal (fent, jobbra, lent, balra) külön szabályozható.

Például egy `<p>` tag, nem egy egyszerű szöveget, hanem bekezdést jelöl, így egy teljes sort foglal el az általa alkotott doboz. Az ilyen elemeket blokk szintű HTML elemeknek hívjuk. Ha csak egy sima szövegről van szó, nem bekezdésről, az önmagában úgynevezett sorközi (inline) elem. Ekkor az őt körülvevő doboz csak addig tart, amíg a tartalom is, nem foglalja el a teljes sort.

A körvonalat sokszor használjuk a weboldal egyes részeinek tagolására vagy díszítőelemként, míg a kitöltés és a margó a szövegek tagolását, jobb áttekinthetőséget, térközök beállítását segíti.

Órai feladat

1. Apple (lásd a honlapon)
 - a. Betűtípus: a teljes dokumentumban Verdana.
 - b. Betű igazítása sorkizárt a teljes dokumentumban (text-align: justify);
 - c. Betű színek: a törzsben (body): #333333, egyes címsor: #666666, kettes címsor: #999999, hármas címsor: #CC6600.
 - d. Betűméretek: egyes címsor: 32pt, kettes címsor: 16pt, hármas címsor: 16pt.
 - e. Margók (margin): a dokumentum törzsnél 0, a fejlécnél 5 pixel.
 - f. Bélések (padding): a képnek (csak felül) 15 pixel és minden táblázatcellának (<td>) 15px minden oldalon.
 - g. A fejléc beállításai: alul 1 pixel széles folytonos (solid) keret (border) #666666 színnel. Balra igazított szöveg (text-align: left).

Otthoni gyakorló példa

1. A weboldalra feltöltött gyakorló feladatot (feladat 2) töltsük le, majd a „start” mappában lévő index.html fájlt nyissuk meg. Tetszőlegesen formázzuk, legyünk kreatívak és próbáljunk meg egy egyszerű, de szép oldalt készíteni. A végeredményre egy példát a „final” mappában találhatunk.
2. A hp könyvtár „final” mappájában található végeredménynek nézzük meg a CSS fájlját és válaszoljuk meg a következő kérdéseket:
 - a. Adjuk meg azoknak a sablonoknak a nevét, ahol azonosító segítségével választjuk ki, hogy hol legyen érvényes a sablon. Miért pont itt használtuk?
 - b. Adjuk meg azoknak a sablonoknak a nevét, ahol tetszőleges nevet választottunk a sablonunknak.
 - c. Milyen HTML tag-ekre készítettünk sablont?
 - d. Keressük meg, hogy a végeredményben látható piros vonalakat hogyan készítettük el.
 - e. Adjuk meg a szaggatott vonalat előállító kódrészt.
 - f. Szavakkal fogalmazzuk meg, hogy a láblécre (footer) milyen formátumot készítettünk (például: a betűmérete 25 pixel... stb.).
 - g. Kis módosítással érjük el, hogy a fejlécben lévő kép körül legyen egy körvonal, 2px távolságra a kép széleitől.

Haladó++ feladat

Táblázat helyett <div>-ekkel készítsünk két hasábos elrendezést.

Help: <http://info.berzsenyi.hu/halozatok/weblapok-formazasa-css-segitsegevel>

CSS kivonat

HTML <head> részbe importálás:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

CSS sablon általános leírása:

```
szelektor {  
    tulajdonság1: érték1;  
    tulajdonság2: érték2;  
}
```

Adott HTML elemre érvényes sablon

```
h1 {  
    color: red;  
    font-size: 14px;  
}
```

Egyedi névvel ellátott sablon

Fontosabb formázási utasítások

Háttérszín:	background-color: blue;	
Betűszín:	color: #000000;	(#000000 – fekete szín)
Betűméret:	font-size: 14px;	
Betűtípus:	font-family: Arial, Helvetica, sans-serif;	
Aláhúzás:	text-decoration: underline;	
Félkövér betűk:	font-weight: bold;	
Dőlt betűk:	font-style: italic;	
Szöveg igazítás (középre):	text-align: center;	
Körvonal:	border: 1px solid red;	(méret, típus, szín)
Kitöltés:	padding: 5px 2px 5px 2px; padding: 2px; padding-top: 5px;	(fent-jobb-lent-bal sorrend) (minden oldalt ugyanannyi) (kitöltés csak fent)
Margó:	margin: 5px 2px 5px 2px;	(hasonlóan a kitöltéshez)

```
.kicsi_piros {  
    color: red;  
    font-size: 14px;  
}  
HIVATKOZÁS:  
<p class="kicsi_piros">Ez lesz a  
kicsi piros szöveg</p>
```

Azonosító alapú kiválasztás

```
#kiemelt {  
    color: red;  
    font-size: 14px;  
}  
HIVATKOZÁS:  
<p id="kiemelt">Ez lesz a kiemelt  
szöveg</p>
```