

SZAKDOLGOZAT

Szondi György

Debrecen

2010

Debreceni Egyetem
Informatika Kar

A CSS 2.0 BEMUTATÁSA PÉLDÁKON
KERESZTÜL

Témavezető:

Dr. Boda István

tanszékvezető egyetemi docens

Készítette:

Szondi György

informatikus könyvtáros

Debrecen

2010

Tartalomjegyzék

Bevezető	4
A CSS története	5
A CSS nyelvtana.....	5
Rangsor.....	7
Mértékegységek.....	9
Színek megadása.....	11
Háttér	13
Bekezdésformázás	16
Betűformázás	16
Szövegformázás.....	20
Pozicionálás	27
Float	27
Absolute és relative	30
Fixed.....	33
Menü készítése CSS segítségével.....	36
Összefoglalás	41
Irodalomjegyzék	42

Bevezető

Az internet széleskörű elterjedése a kilencvenes évek közepére, illetve az ezzel párhuzamosan zajló technikai fejlődés és társadalmi változások következtében megnőtt az igény a látványosabb, egyedibb webes tartalom megjelenítésére. A világháló infrastruktúrájának nagymértékű fejlődése lehetővé tette a bonyolultabb, jobban használható, nagyobb vizuális élményt nyújtó honlapok készítését. Mindezek velejárója volt egy egyszerű, csak a vizuális megjelenítésért felelős, a már meglévő HTML-t kiegészítő kódrendszer megszületése.

1996-ban a World Wide Web Consortium (W3C) szabványszerkesztő csoportja létrehozta a CSS-t. A CSS (Cascading Style Sheets) egymásba ágyazott vagy rangsorolt stíluslapokat jelent. Egy egyszerű nyelv, mellyel befolyásolhatjuk HTML oldalaink megjelenését, többek között a HTML elemek elhelyezkedését, színét, méretét, stb.

A CSS jelentősége abban rejlik, hogy a honlap készítők lehetőségeit nagymértékben kiterjeszti, míg a felhasználók a kisebb adatmennyiségnek köszönhetően szebb, gyorsabb, könnyebben átlátható oldalakon böngészhetnek.

Készítői oldalról a CSS egyik legnagyobb előnye, hogy akár egyetlen fájl megváltoztatásával egész weblaprendszerek megjelenítését lehet egyszerűen megváltoztatni. Ezzel időt és munkát lehet megtakarítani a karbantartás és frissítés terén.

Újdonság az is, hogy adott esetben egy olyan felhasználó, aki rendelkezik minimális HTML és CSS ismeretekkel egyedi megjelenésű weblapot hozhat létre.

A dolgozatom célja, hogy bemutassam a CSS-t azok számára, akik alapszinten tisztában vannak a HTML használatával, de szeretnének további ismereteket szerezni.

A munkámban egyszerű példákkal, HTML és CSS kódokkal és a jobb érthetőség kedvéért képekkel mutatom be a rangsorolt stíluslapok működését.

A CSS története

A HTML 4 és a CSS megjelenése előtt a megjelenítés a HTML kódba volt beépítve. Ennek több hátulütője is volt a CSS létrehozása előtt. A tervezőknek nem volt más, egyszerűbb lehetőségük a megjelenítésre csak az a több száz HTML elemhez tartozó meghatározás, ami miatt megnőtt a weblapok mérete és nagyobb sávszélesség kellett a letöltésükhöz.

A pontos pozícionálás szinte lehetetlen volt. Hogy ezt kiküszöböljék, az elemeket táblázat segítségével helyezték el a weblapon. Ez azonban felvetette azt a problémát, hogy a csökkentlátók által használt képernyő felolvasók nem tudták értelmezni a táblázatban elhelyezett szövegeket, különösen akkor, ha több táblázat volt egymásba ágyazva.

További lényeges hibája, hogy az elemeket nem csak arra használták a tervezők, amire eredetileg szánták őket. Ilyen például a szövegbehúzás problémája, amit listákkal vagy idézetblokkokkal tudtak megvalósítani.

Ezért 1996. december 17-én megjelent a CSS szabvány leírása a W3C honlapján (<http://www.w3.org>). A szabványt azóta már tovább fejlesztették és 1998-ban megjelent a CSS 2 és később a CSS 2.1 is, a CSS 3 kidolgozása pedig folyamatban van (1).

A CSS nyelvtana

Mint már említettem a CSS egy egyszerű nyelv, tehát a nyelvtana is viszonylag egyszerű, azonban eltér a HTML nyelvtanától.

A CSS-ben a stílusokat kijelölők és meghatározások segítségével lehet megadni.

A kijelölők többnyire olyan HTML tagok, amelyek megadják, hogy a stílus a HTML lap mely elemeire vonatkozzon. A legegyszerűbb kijelölő a *p* tag, ami kijelöli a weblapon található összes bekezdést.

A kijelölőhöz írunk kell egy meghatározást, amivel megadjuk, hogy a kijelölt elem hogyan jelenjen meg a weblapon.

A kijelölő és a meghatározás együtt alkotják a stílust.

Egy példa a kijelölőre és meghatározásra:

```
p {font-family: Arial, Georgia, sans-serif; font-size: medium; color: red;}
```

A példában a p a kijelölő és a két kapcsos zárójel közötti rész a meghatározás blokk.

A meghatározás összetevői: a tulajdonság (pl.: font-family), melyet kettőspont követ, majd egy szóköz, végül az érték (pl.: Arial), végül a meghatározást lezáró pontosvessző.

Gyakran egyértelműbben kell megadni egy elem stílusát. Erre két lehetőségünk van: az azonosító kijelölő illetve az osztálykijelölő.

Egy azonosító kijelölőt egy lapon csak egyszer lehet alkalmazni. Általában olyan elemeknél használjuk, amelyeket egy szerkezeti egységként szeretnénk formázni (pl.: fejléc, lábléc, menü). Az azonosító kijelölőket kettős kereszttel (#) jelöljük.

```
#lablec{ color: green;}
```

Az osztálykijelölőket egy lapon belül annyiszor használhatjuk, ahányszor csak akarjuk. Az osztálykijelölőket ponttal (.) jelöljük (2,3).

```
.tartalom {background-color: #f7fbfb;}
```

Rangsor

- Böngészőstílus
- Felhasználói stílus
- Külső stílus
- Begyázott vagy belső stílus
- Szövegekőzi stílus

A rangsor a gyakorlatban a stílusok hierarchiáját jelenti.

Böngészőstílus

A böngészőstílus az oldal alapértelmezett stílusát jelenti. Másképpen mondva, ha nincs semmilyen más stílus megadva az adott laphoz, akkor a böngésző stílusa határozza meg a lap megjelenését.

Felhasználói stílus

Minden böngészőben a felhasználó beállíthat saját stílust, melyben lehetősége van megváltoztatni például a betűstílust, méretet, betűtípust, színt, de akár még a betűkódolást is.

Külső stílus

A külső stíluslap egy .css kiterjesztésű szöveges dokumentum, amit hozzákapcsolunk a HTML oldalhoz. Ez a külső stíluslap tartalmazza a CSS kódokat. Pl.:

```
body{margin-top:10px;  
margin-bottom:10px;  
background-image:url(images/bbg.png);}
```

Külső stíluslap használata abban az esetben ajánlott, ha több lapot is ugyanazzal a stíluslappal szeretnénk megjeleníteni.

A hivatkozáselemet a HTML dokumentum fejlécében kell elhelyezni a title elem után.

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

Külső stíluslapot nem csak a link elemmel lehet hozzákapcsolni egy HTML laphoz, hanem az @import utasítással is. A legfőbb oka, hogy az @import utasítást még használják az, hogy bizonyos régi böngészők, mint például a Netscape Navigator 4.x még nem ismerik a link elemet, csak az @importot.

```
<head>
<title>CSS</title>
<style type="text/css">@import url(style.css);</style>
<link href="style.css" rel="stylesheet" type="text/css" />
</head>
```

Beágyazott vagy belső stílusok

Stílusokat a HTML dokumentum fejlécében is elhelyezhetünk. A belső stílusok csak arra a dokumentumra érvényesek, amelyben elhelyezték őket. Akkor érdemes belső stílusokat használni, ha különálló weblapot szeretnénk készíteni vagy ha a weblaphoz kapcsolódik külső stíluslap, de kismértékben megszeretnénk változtatni azt. Hasonlóan a külső stíluslaphoz a beágyazott stílust is a HTML dokumentum fejlécében kell elhelyezni a title elem után.

```
<head>
<title>CSS</title>
<style type="text/css">
p {font-size: small; color: #000;}
</style>
</head>
```

Szövegekői stílus

A szövegekői (a szövegben elhelyezett) stílusok egyszer használatos formázási utasítások. Technikailag ezek is beágyazott stílusok. A rangsor fogalmai szerint ez van a legközelebb a formázandó szöveghez. A szövegekői stílusok erőteljes használata megnöveli a letöltési időt és lehetetlenné teszi azt, hogy a dokumentumok megjelenítését egyetlen dokumentum segítségével szabályozzuk.

Amikor szövegekői stílust használunk, mindig az adott elembe kell beírunk. Pl.:

```
<p style="color: red">Ez egy bekezdés</p>
```

(4,5)

Mértékegységek

A méretek megadásakor beszélünk abszolút és relatív értékekről.

Pozitív és negatív értékeket is megadhatunk a tulajdonságoktól függően.

A mértékegységet közvetlenül a szám után kell írni, mindenfajta elválasztójel nélkül.

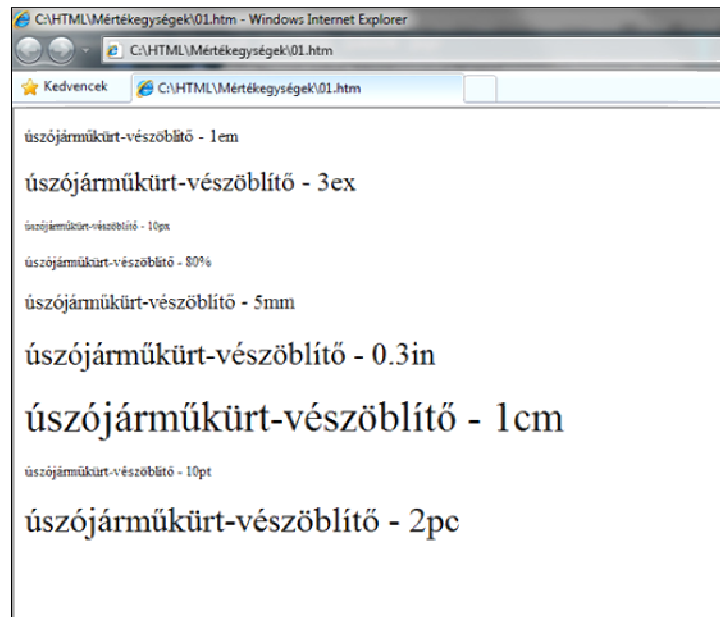
Amennyiben az érték 0, nem szükséges kiírni a mértékegységet, ugyanis ebben az esetben mindegy milyen mértékegységre gondolunk, mindegyikben ugyanazt jelenti.

Nem egész szám megadásakor a számítástechnikában megszokott tizedes pontot kell alkalmaznunk (nem tizedes vessző).

Relatív mértékegységek

Ide tartozik az em, ex, px és a százalékban megadott méret.

em: Relatív értéket jelöl, 1em az adott környezet betűméretét jelöli. A font-size aktuális értéke, illetve ha konkrét font-size definícióban belül használjuk, akkor a szülő elem betűmérete. Alapértelmezetten a dokumentum gyökerében 16 pixelt szokott jelenteni.



ex: Relatív érték, hasonlít az em-hez. Az adott betűkészlet kis "x" betű magasságát jelöli. Általában a böngészők az em felének veszik, tehát 8 pixel-t jelöl.

px: képpontban (pixelben) megadott méretet jelöl. Azért soroljuk a relatív mértékegységekhez, mert különböző megjelenítő eszközökön máshogy jelenhet meg (pl.: kisebb felbontásra állított monitor). Jele: px.

Százalékos megadás: mindig relatív értéket képez attól függően, hogy mihez viszonyítjuk. Minden tulajdonságnál, ahol százalékos megadást használhatunk, definiálva van, hogy mihez képest értendő.

Abszolút mértékegységek

Ide tartozik a mm, in, cm, pt és a pc.

Az abszolút mértékegységek neve onnan ered, hogy mind a kijelzőn, mind nyomtatásban a szemlélő számára ugyanakkora méretet jelentenek.

mm: a milliméter rövidítése, pl.: 10mm.

in: az inch rövidítése, nevezik még colnak vagy hüvelyknek is. 2,54 centimétert jelöl. Pl.: 2in.

cm: a centiméter rövidítése, pl.: 2cm.

pt: a pont rövidítése, tipográfiai mértékegység és $1/72$ hüvelyket jelöl. Pl.: 16pt.

pc: pica rövidítése, 12 pontot jelöl. Pl.: 3pc. (6,7)

Színek megadása

A weblapok megjelenítésére szolgáló eszközök három szín, a piros, a zöld, és a kék színek keverésével állítják elő a színeket. Ezeket az angol nevükből eredően (red, green, blue) RGB színeknek nevezzük.

Háromféleképp adhatjuk meg, hogy ezeket a színeket milyen arányban szeretnénk használni.

Az első módszer a hexadecimális számokkal történő megadás. Tizenhatos számrendszerbeli számokkal (azaz számokkal és betűkkel) adhatjuk meg a színeket. Például #0066FF egy kék árnyalatot jelent. Az első kettő karakter (00) a piros, a középső kettő (66) a zöld és az utolsó kettő (FF) a kék mennyiségét jelenti az adott színben. Amennyiben mindhárom érték ugyanaz akkor fehér (#FFFFFF), fekete (#000000) vagy a szürke (#C0C0C0) valamilyen árnyalata.

A színek hexadecimális megadásánál kötelező használnunk a számok előtt a kettős keresztet.

Ha a hexadecimálisan megadott RGB érték három azonos betűt vagy számot tartalmazó párból áll, elég megadnunk a karakterpár helyett egy karaktert. Például a #0066FF ugyanazt a színt adja, mint a #06F.

Webtűró színeknek nevezzük a 00, 33, 66, 99, CC és FF karakterpárok kombinációjából létrehozott bármilyen színt. Ilyen például a #009933 ami egy sötétzöld árnyalat. A számítógépek monitorai több millió színt képesek megjeleníteni, azonban az eszközök és operációsrendszerek különbözősége miatt csak 216 szín számít webtűrónek, ami nagyjából azonos árnyalatban jelenik meg minden számítógépen.

A második módszer az RGB színek százalékban történő megadása. Ebben az esetben nem színkóddal hanem százaléokban adjuk meg az egyes színek arányát a színkeverésnél. Például a fehér színt a következőképp adhatjuk meg: `rgb(100%,100%,100%)`, míg a kéket pedig: `rgb(0%,0%,100%)`.

A harmadik színmegadási mód a 0 és 255 közötti decimális (tízes számrendszerbeli) értékekkel való megadás. A fehér szín például a következőképpen adható meg: `rgb(255,255,255)`. (8,9)

A fehér színt tehát három módon is megadhatjuk a stíluslapon:







- color: #FFFFFF; vagy color: #FFF;
- color: rgb(100%,100%,100%);
- color: rgb(255,255,255);

Nevükkel is meg lehet adni a színeket. Ezeket nevezzük előre meghatározott színeknek. 16 ilyen szín van: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow.

	aqua	#00FFFF		navy	#000080
	black	#000000		olive	#808000
	blue	#0000FF		purple	#800080
	fuchsia	#FF00FF		red	#FF0000
	gray	#808080		silver	#C0C0C0
	green	#008000		teal	#008080
	lime	#00FF00		white	#FFFFFF
	maroon	#800000		yellow	#FFFF00

A fent említett 16 színen kívül létezik még 140 további névvel megadható szín, amelyek a böngészők többségében működik, azonban a W3C CSS referenciájában nem szerepel.

Ezeket a színeket érdemes a hexadecimális értékük szerint megadni, amennyiben szeretnénk érvényes (valid) HTML és CSS dokumentumot készíteni.(10,11) Pl.:

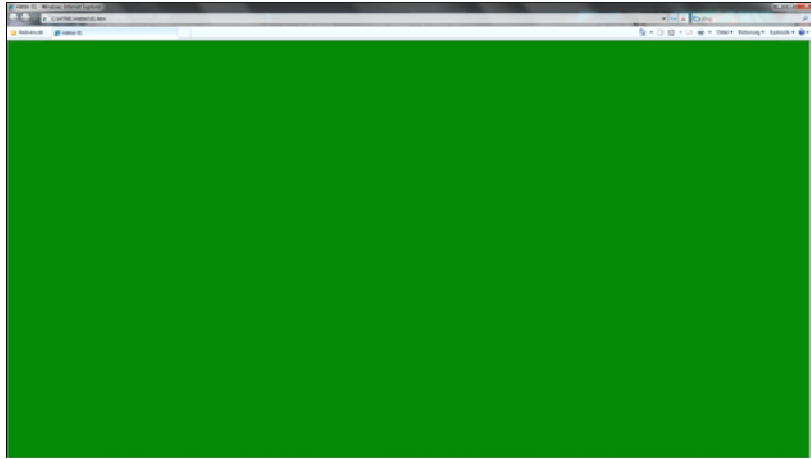
	BlueViolet	#8A2BE2		ForestGreen	#228B22
	CornflowerBlue	#6495ED		MistyRose	#FFE4E1
	FireBrick	#B22222		Tan	#D2B48C

Háttér

Az elemek háttérét többféleképpen lehet megadni. A CSS öt „background” tulajdonsággal rendelkezik.

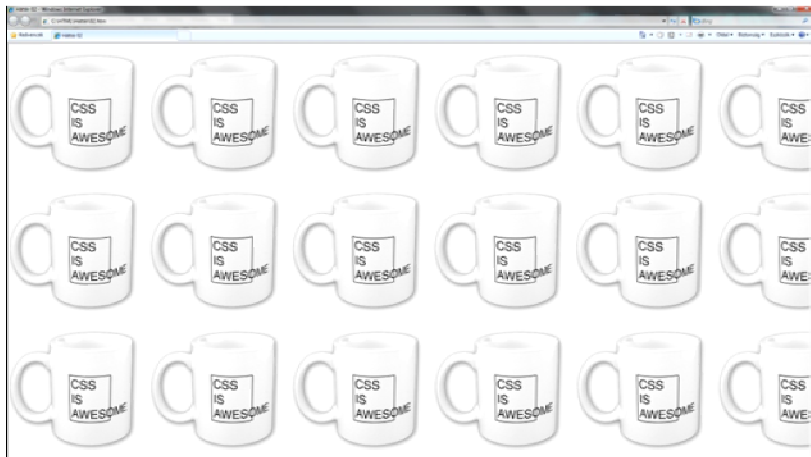
- `background-color`: az elem háttérszínét adja meg. A színeket meghatározhatjuk a „Színek megadása” című fejezetben leírtak szerint.

Pl.: `body {background-color: green;}`



- `background-image`: az elem háttérének egy képet állít be (12).

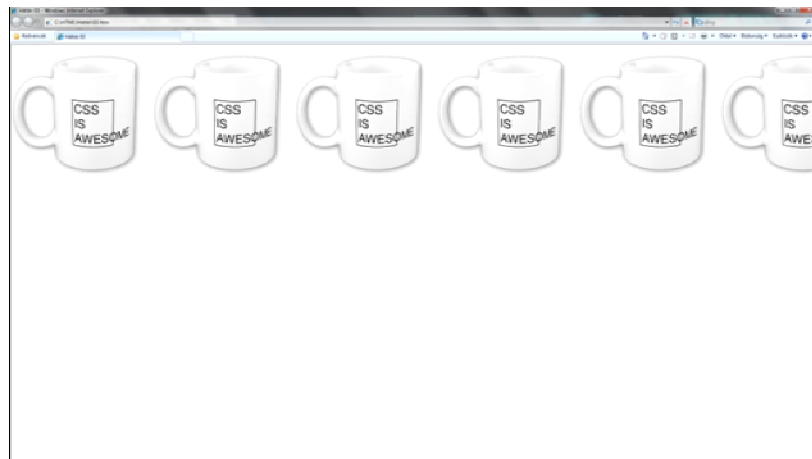
Pl.: `body {background-image: url(hatter.jpg);}`



- background-repeat: az elem háttéréként beállított kép ismétlődését állíthatjuk be. Négy értéke van:

- repeat (vízszintes és függőleges ismétlődés)
- no-repeat (ne ismétlődjön)
- repeat-x (vízszintes ismétlődés)
- repeat-y (függőleges ismétlődés)

Pl.: `body {background-image: url(hatter.jpg); background-repeat: repeat-x;}`



- background-position: a háttérkép helyzetét adja meg.
 - top (felülre)
 - center (középre)
 - bottom (alulra)
 - left (balra)
 - right (jobbra)
 - százalékérték
 - hossz

Pl.: `body {background-image: url(hatter.jpg); background-repeat: repeat-x; background-position: center;}`



- background-attachment: a héttérkép rögzítettségét határozza meg.
 - fixed: nem mozdul a háttérgörgetéskor
 - scroll: görgetéssel együtt mozog a háttér

A háttér meghatározások kombinálhatóak, tehát elég megadnunk annyit, hogy:
background: green url(hatter.jpg) repeat-y center;.

A background tulajdonságnak, ha rövidítünk, a következő értékei lehetnek az alábbi sorrendben (13):

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

Bekezdésformázás

Betűformázás

- `font-family`: az adott elem betűcsaládját határozza meg. Érdeemes több betűcsaládot megadni, mivel nem ugyanazzal a betűtípuskészlettel van ellátva minden számítógép. Elsőbbségi sorrendben kell megadni a betűcsaládokat. Célszerű a sor végére egy általános betűcsaládot írni, amelyet a rendszer alapértelmezett betűcsaládként tud alkalmazni. Ilyen betűcsaládok a serif, sans-serif, cursive, fantasy és a monospace.

Pl.:

```
<html>
<head>
<style type="text/css">
  p {font-family: Arial, Helvetica, serif;}
</style>
</head>
<body>
<p>úszójárműkürt-vészöblítő</p>
</body>
</html>
```

- `font-style`: a betűtípus „dőltségét” adja meg. Lehetséges értékei:
 - normal
 - italic (dőlt)
 - oblique (ferde)

Az oblique érték csak az oblique-kal jelölt betűtípusoknál működik megfelelően, egyébként az italic-kal megegyező képet ad.

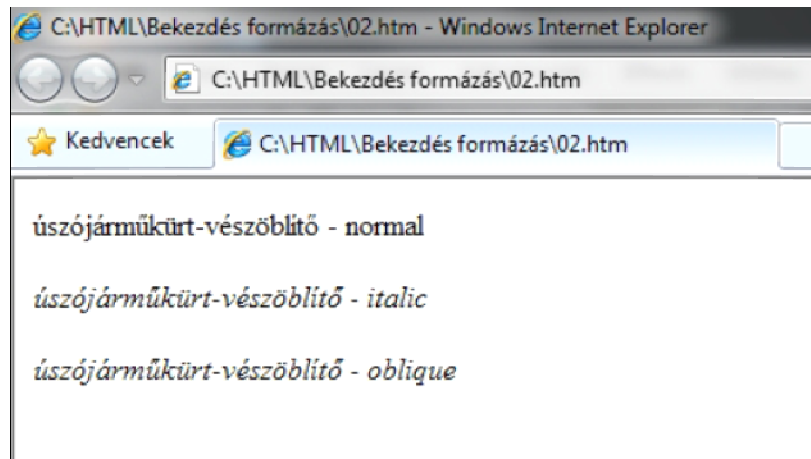
Pl.:

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

```

<body>
<p style="font-style: normal;">úszójárműkürt-vészöblítő - normal</p>
<p style="font-style: italic;">úszójárműkürt-vészöblítő - italic</p>
<p style="font-style: oblique;">úszójárműkürt-vészöblítő - oblique</p>
</body>
</html>

```



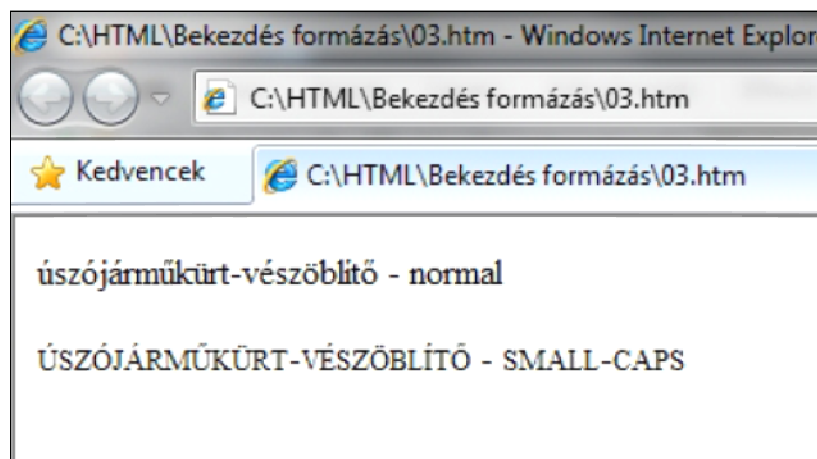
- font-variant: a font változat megadása. Lehet normál és kiskapitális: normal, small-caps

Pl.:

```

<html>
<head></head>
<body>
<p style="font-variant: normal;">úszójárműkürt-vészöblítő - normal</p>
<p style="font-variant: small-caps;">úszójárműkürt-vészöblítő - small-caps</p>
</body>
</html>

```



- font-size: a betűtípus méretét adja meg. A méreteket meghatározhatjuk pixelben, pontban, százalékban, szövegesen (xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger).

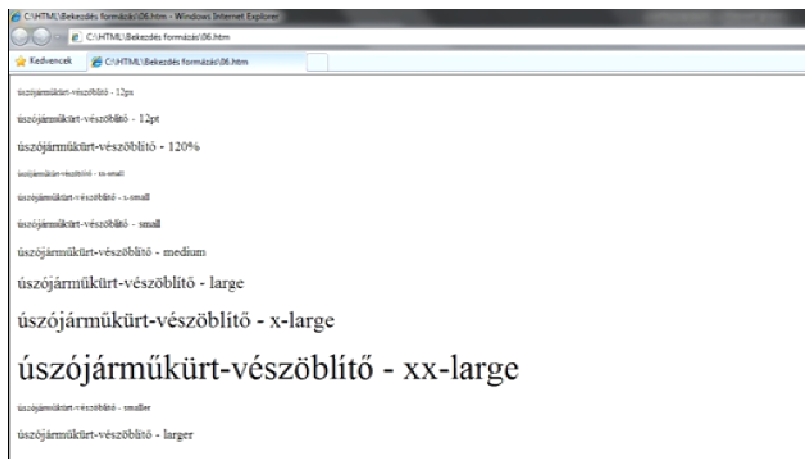
Pl.:

```
<html>
<head></head>
<body>
<p style="font-size: 12px;">űszójárműkűrt-vészöblítő - 12px</p>
<p style="font-size: 12pt;">űszójárműkűrt-vészöblítő - 12pt</p>
<p style="font-size: 120%;">űszójárműkűrt-vészöblítő - 120%</p>
<p style="font-size: xx-small;">űszójárműkűrt-vészöblítő - xx-small</p>
<p style="font-size: x-small;">űszójárműkűrt-vészöblítő - x-small</p>
<p style="font-size: small;">űszójárműkűrt-vészöblítő - small</p>
<p style="font-size: medium;">űszójárműkűrt-vészöblítő - medium</p>
<p style="font-size: large;">űszójárműkűrt-vészöblítő - large</p>
<p style="font-size: x-large;">űszójárműkűrt-vészöblítő - x-large</p>
<p style="font-size: xx-large;">űszójárműkűrt-vészöblítő - xx-large</p>
<p style="font-size: smaller;">űszójárműkűrt-vészöblítő - smaller</p>
<p style="font-size: larger;">űszójárműkűrt-vészöblítő - larger</p>
</body>
</html>
```

Csoportos megadásra is lehetőségünk van a font tulajdonsággal (14,15).

Pl.:

```
<html>
<head></head>
<body>
  <p style="font: italic bold 14px Arial, Helvetica, sans-serif; ;">úszójárműkürt-
vészöblítő - italic bold 14px Arial, Helvetica, sans-serif</p>
</body>
</html>
```

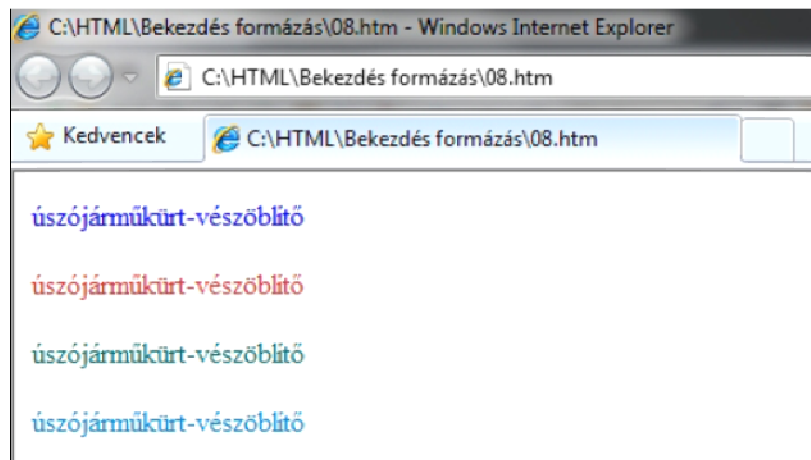


Szövegformázás

- color: a betűk színét határozza meg a „Színek megadása” című fejezetben foglaltak szerint.

Pl.:

```
<html>
<head></head>
<body>
<p style="color: blue;">úszójárműkürt-vészöblítő</p>
<p style="color: #CC3333;">úszójárműkürt-vészöblítő</p>
<p style="color: rgb(000,102,102);">úszójárműkürt-vészöblítő</p>
<p style="color: rgb(0%,50%,80%);">úszójárműkürt-vészöblítő</p>
</body>
</html>
```



- letter-spacing: a betűk közti távolságot csökkenti vagy növeli.

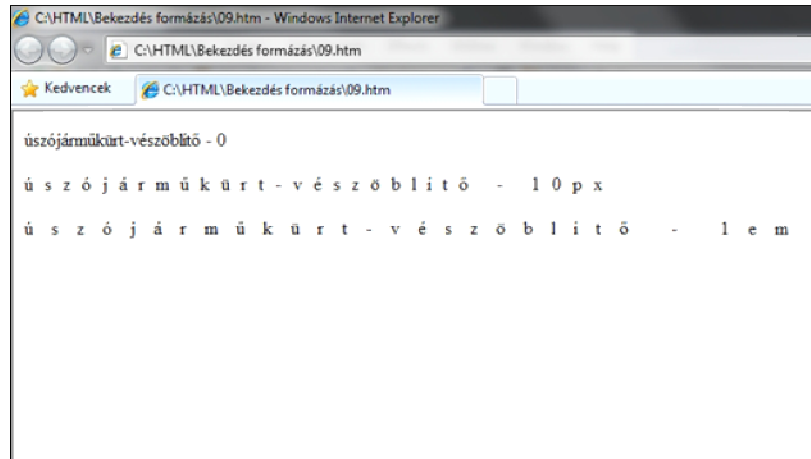
Pl.:

```
<html>
<head></head>
<body>
<p style="letter-spacing: 0;">úszójárműkürt-vészöblítő - 0</p>
<p style="letter-spacing: 10px;">úszójárműkürt-vészöblítő - 10px</p>
```

```

<p style="letter-spacing: 1em;">úszójárműkürt-vészöblítő - 1em</p>
</body>
</html>

```



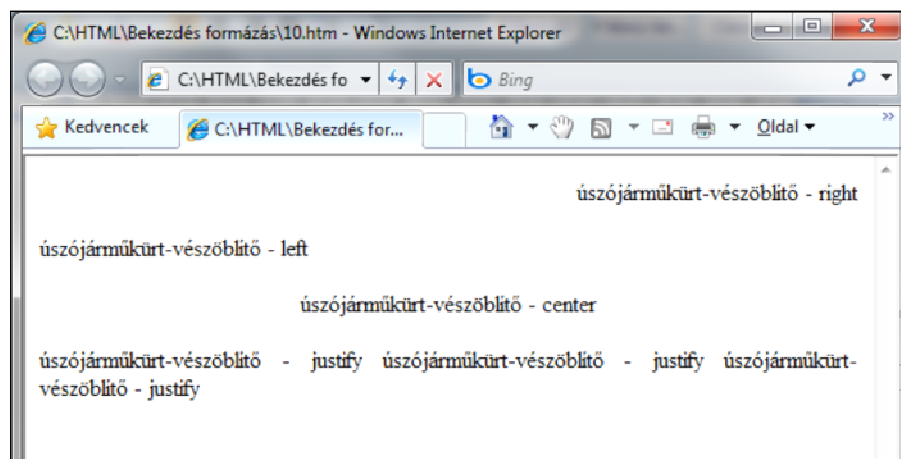
- text-align: a szöveg igazítását adja meg, lehet jobbra (right), balra (left), középre (center) és sorkizárt (justify).

Pl.:

```

<html>
<head>
</head>
<body>
<p style="text-align: right;">úszójárműkürt-vészöblítő - right</p>
<p style="text-align: left;">úszójárműkürt-vészöblítő - left</p>
<p style="text-align: center;">úszójárműkürt-vészöblítő - center</p>
<p style="text-align: justify;">úszójárműkürt-vészöblítő - justify úszójárműkürt-
vészöblítő - justify úszójárműkürt-vészöblítő - justify</p>
</body>
</html>

```



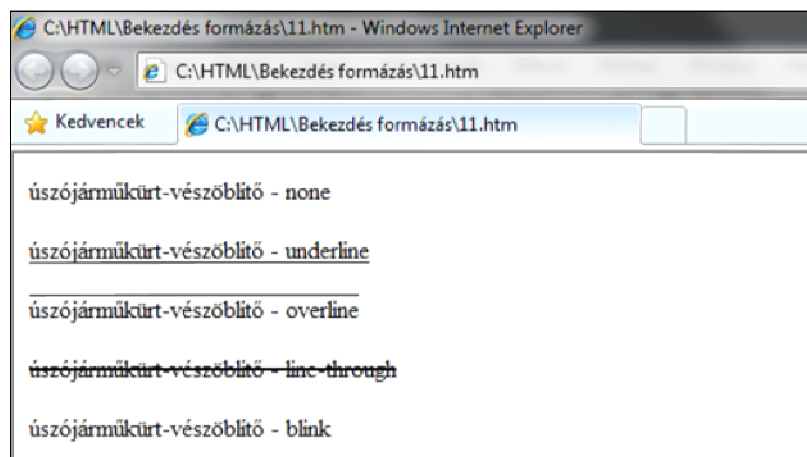
- text-decoration: a szöveg dekorációját adja meg. Lehetséges értékei:
 - none (dísztelen)
 - underline (aláhúzott)
 - overline (felülhúzott)
 - line-through (áthúzott)
 - blink (villogó – Internet Explorer 8 alatt nem működik)

Pl.:

```

<html>
<head></head>
<body>
<p style="text-decoration: none;">úszójárműkürt-vészöblítő - none</p>
<p style="text-decoration: underline;">úszójárműkürt-vészöblítő -
underline</p>
<p style="text-decoration: overline;">úszójárműkürt-vészöblítő - overline</p>
<p style="text-decoration: line-through;">úszójárműkürt-vészöblítő - line-
through</p>
<p style="text-decoration: blink;">úszójárműkürt-vészöblítő - blink</p>
</body>
</html>

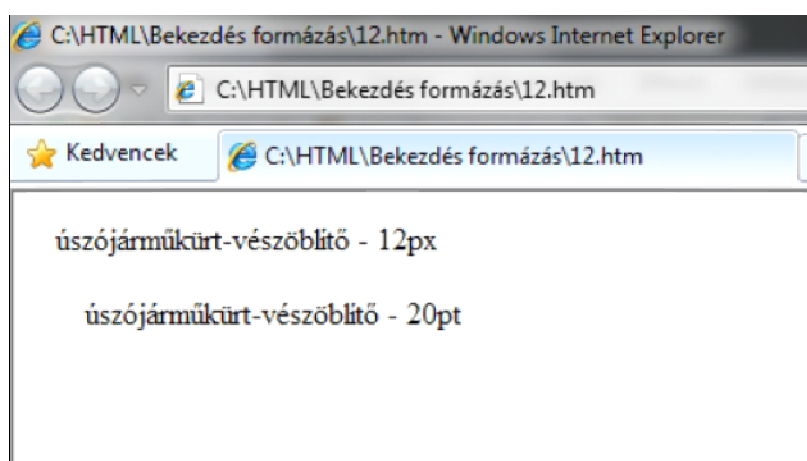
```



- text-indent: a bekezdések első sorának behúzását állítja be a „Mértékegységek” fejezetben foglaltak szerint.

Pl.:

```
<html>
<head></head>
<body>
<p style="text-indent: 12px;">úszójárműkürt-vészöblítő - 12px</p>
<p style="text-indent: 20pt;">úszójárműkürt-vészöblítő - 20pt</p>
</body>
</html>
```

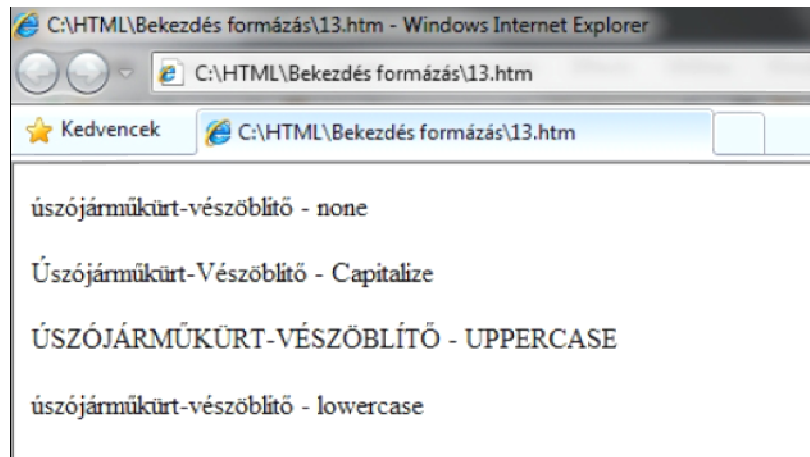


- text-transform: szöveg átalakítás kis vagy nagy betűssé.
 - none (módosítatlan)
 - capitalize (első betű nagy)

- uppercase (összes betű nagy)
- lowercase (összes betű kicsi)

Pl.:

```
<html>
<head></head>
<body>
<p style="text-transform: none;">úszójárműkürt-vészöblítő - none</p>
<p style="text-transform: capitalize;">úszójárműkürt-vészöblítő - capitalize</p>
<p style="text-transform: uppercase;">úszójárműkürt-vészöblítő -
uppercase</p>
<p style="text-transform: lowercase;">úszójárműkürt-vészöblítő -
lowercase</p>
</body>
</html>
```



- text-shadow: árnyékot rajzol a szöveg köré. Négy értéket kell meghatározunk az árnyék megadásánál: a lefelé való távolságot, a jobbra való távolságot, az elmosódottság sugarát és a színét. A színt megadása kötelező, a pozíció nem. Negatív értékeket is felvehetnek a pozíciónál meghatározott távolságok.

Az Internet Explorer 8 még nem képes megjeleníteni a text-shadow-t.

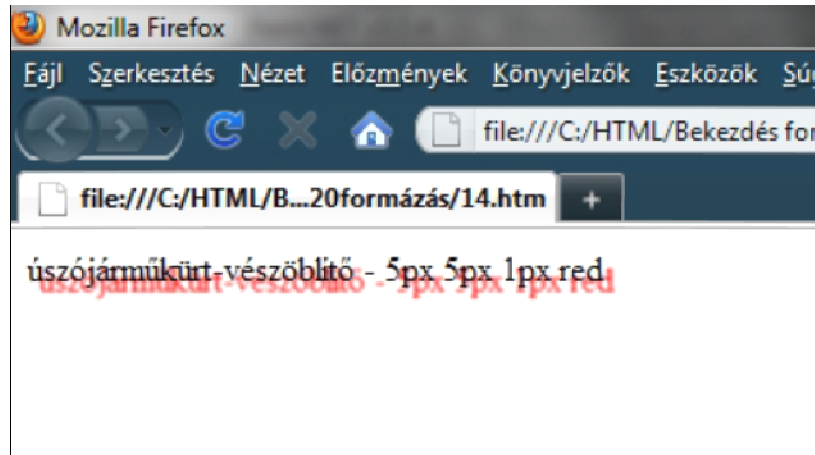
Pl.:

```
<html>
<head></head>
<body>
```

```

<p style="text-shadow: 5px 5px 1px red;">úszójárműkürt-vészöblítő - 5px 5px
1px red</p>
</body>
</html>

```



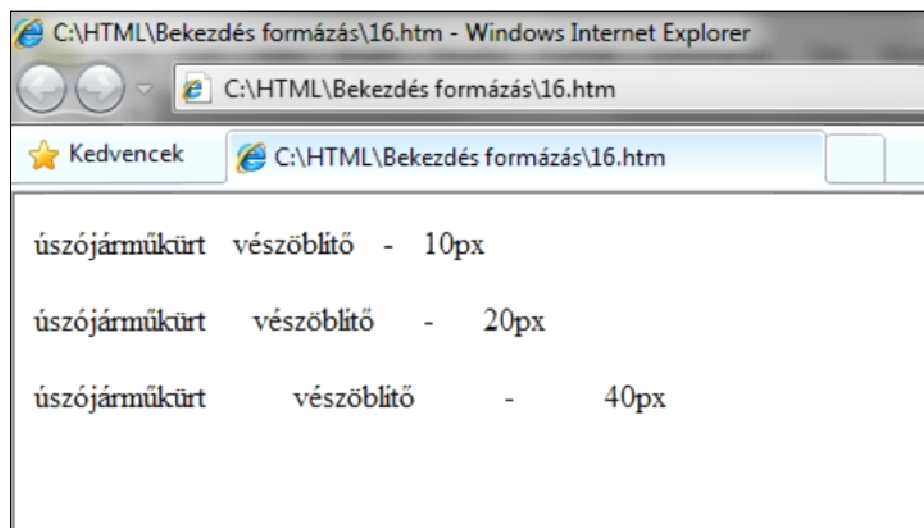
- word-spacing: szavak közti távolságot határozza meg. Felvehet negatív értéket is.

Pl.:

```

<html>
<head></head>
<body>
<p style="word-spacing: 10px;">úszójárműkürt vészöblítő - 10px</p>
<p style="word-spacing: 20px;">úszójárműkürt vészöblítő - 20px</p>
<p style="word-spacing: 40px;">úszójárműkürt vészöblítő - 40px</p>
</body>
</html>

```



- line-height: a sormagasságot állítja be. Érdekes a betűméretnél nagyobb sormagasságot beállítani, mert különben összecúsznak a sorok (16,17).

Pl.:

```
<html>
```

```
  <head></head>
```

```
  <body>
```

```
    <p style="line-height: 0px;">úszójárműkürt vészöblítő - 0px</p>
```

```
    <p style="line-height: 10px;">úszójárműkürt vészöblítő - 10px</p>
```

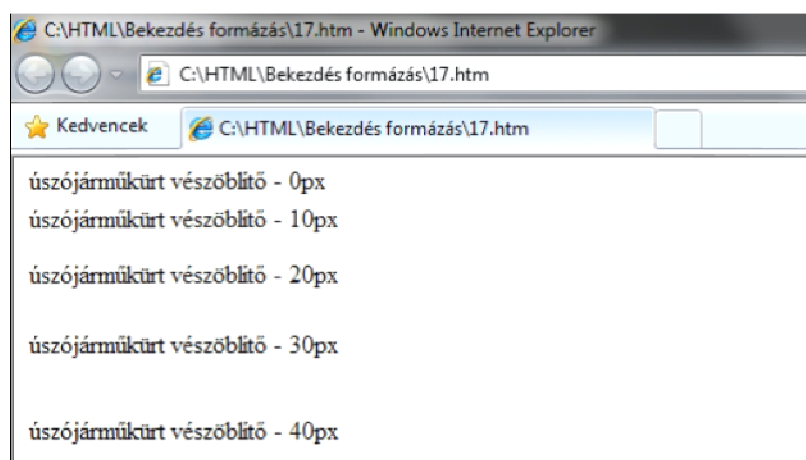
```
    <p style="line-height: 20px;">úszójárműkürt vészöblítő - 20px</p>
```

```
    <p style="line-height: 30px;">úszójárműkürt vészöblítő - 30px</p>
```

```
    <p style="line-height: 40px;">úszójárműkürt vészöblítő - 40px</p>
```

```
  </body>
```

```
</html>
```



Pozicionálás

A CSS-ben a pozicionálásnak 4 típusa van:

- float:
 - left;
 - right;
 - none;
- position: absolute;
- position: relative;
- position: fixed;

Float

Ha a float jellemzővel egy adott elemet úsztatunk (float=úsztatás) akkor a tartalom többi része körbefolyja. Ez a stílus hasonló a Microsoft Wordben alkalmazott körbefuttatás parancshoz. Legtöbbször a float-ot akkor használjuk, amikor egy képet szeretnénk körbefuttatni szöveggel (18,19).

Nézzünk ehhez egy egyszerű példát:

```
<html>
```

```
<head>
```

```
<title>Float</title>
```

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

```
</head>
```

```
<body>
```

```
<div id="kulso">
```

```
<h1>1. Hol terem a magyar vitéz</h1>
```

<p>Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a magyar foglyokat. Gábor pap megkegyelmez Jumurdzsáknak, csak a talizmánját veszi el. Dobó Gáspárra bízta a

kisfiút, hogy vigye el Török Bálinthoz, Szigetvárba. Örömmel fogadja a várkapitány, s két fiával együtt neveli.

`</div>`

`</body>`

`</html>`

Egy egyszerű HTML kóddal kezdünk, amiben létrehozunk egy kulso divet amiben elhelyezzük a képet, a címsort és a szöveget. A példában azért van a címsorban (`<h1>`tagok között) a kép, hogy később a kép teteje és a cím egy vonalba kerüljön.

`#kulso{`

`width: 500px;`

`margin: 10px;`

`padding: 5px;`

`background-color: #FFC;`

`border: solid 1px black;}`

A kulso div szélességét beállítjuk 500 px-re, mert így kevesebb szöveggel lehet szemléltetni a float működését. Megadunk egy belső és egy külső margóval értéket. A háttérszín és a keret is a jobb szemléltethetőség miatt kell.

A kép úsztatása nélkül a kép alsó vonala és a címsor van egyvonalban.

`img{`

`float:left;`

`padding: 5px;}`

A példában a képet balra szeretnénk helyezni ezért a `float: left;` stílust állítjuk be. Ezzel a jellemzővel a helyére is került a képünk a szövegben.





A floatnak természetesen nem csak a left (balra) értéket lehet beállítani. A float: right;-tal jobbra úsztathatjuk az elemet. A none érték az alapértelmezett, aminek hatására sem jobbra, sem balra nem úszik az elem. Ez az érték egyenértékű azzal, ha nem állítunk be float tulajdonságot egy elemhez.

Absolute és relative

Az absolute és relative pozicionálást legkönnyebben egy háromszlopos egyszerű weboldallal lehet bemutatni.

```
<html>
<head>
<title>Absolute</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
<div id="kulso">
  <div id="első"> Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török
dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré
nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a
magyar foglyokat.</div>
  <div id="masodik"> Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török
dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré
nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a
magyar foglyokat.</div>
  <div id="harmadik"> Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török
dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré
nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a
magyar foglyokat.</div>
</div>
</body>
</html>
```

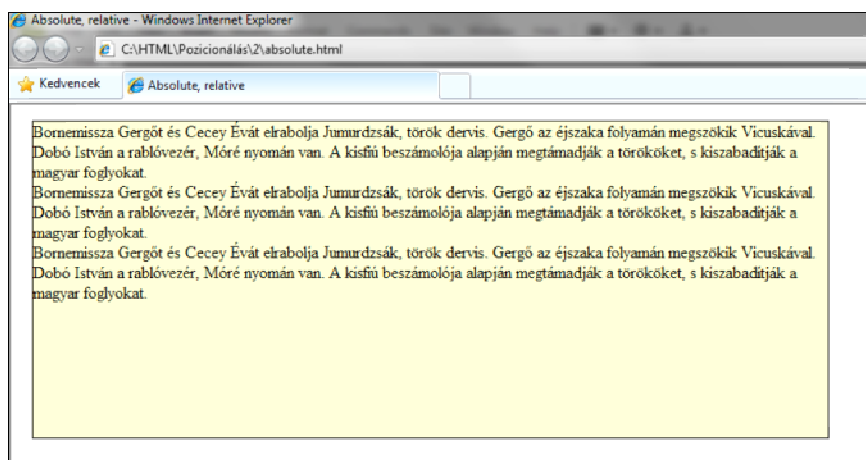
Ebben a példában is külső CSS fájlt fogunk használni, ezért el kell készítenünk a hivatkozást a fájlra.

Először létrehozunk egy div-et amiben később elhelyezzük a másik három div-et (a három oszlopot), ennek a div-nek legyen az azonosítója kulso. A kulso div-be pedig tegyük bele az első, második és harmadik azonosítójú div-eket. Ezzel el is készült a HTML kódunk.

```
#kulso{
    position: relative;
    width: 750px;
    height: 300px;
    margin: 10px;
    background-color: #FFC;
    border: solid 1px black;}
```

Elsőként a három hasábot tartalmazó div tulajdonságait állítjuk be. Legelső ilyen tulajdonság a `position: relative;`. Ennek szerepe, hogy így a három hasáb igazodik a környezetéhez, tehát ahhoz a div-hez amiben benne van. Amennyiben `position: relative;` sort kihagynánk, úgy az első, második és harmadik div-ek a böngészőablakhoz viszonyítva helyezkednének el és nem pedig a kulso div-hez képest. Ennek az lenne a hátránya, hogy sokkal nehezebb lenne a három hasábot pozícionálni a megfelelő helyre.

A kulso div-nek a következő lépésben megadunk egy szélességet és magasságot. A jobb láthatóság érdekében háttérszínt és keretet is beállítunk.



```
#első{
    position: absolute;
    top: 0px;
    left: 0px;
    width: 250px;
    background-color: #3CF;}
```

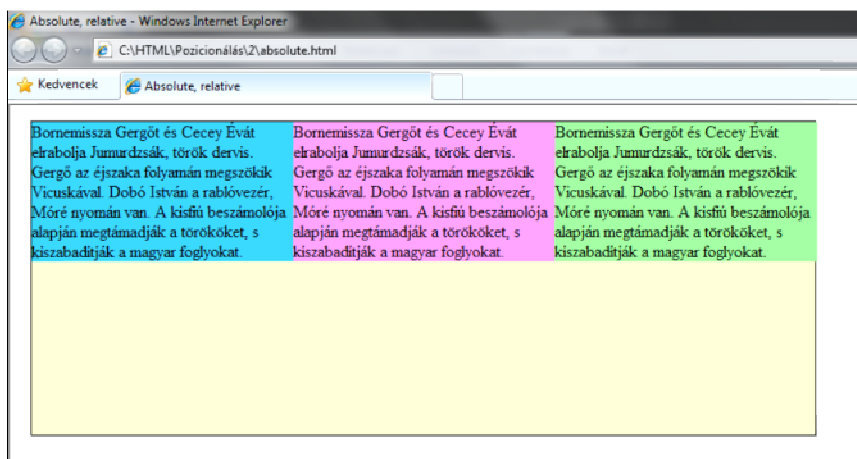
Mivel az előző részben a kulso div-et beállítottuk a relative-ra, ezért az első hasábot a kulso div alapján helyezhetjük el. A kulso div bal felső sarkába helyezzük az első hasábot: top: 0px; left: 0px; parancsokkal (20,21).

```
#masodik{  
    position:absolute;  
    top:0px;  
    left:250px;  
    width:250px;  
    background-color:#F9F9;}
```

A második hasáb helyét az abszolút pozicionálás miatt pontosan kell meghatároznunk. Az első hasáb szélessége 250px volt ezért a második hasábot 250px-lel el kell tolni jobbra: left: 250px;.

```
#harmadik{  
    position:absolute;  
    top:0px;  
    left:500px;  
    width:250px;  
    background-color:#9F9;}
```

A harmadik hasábnál ugyanúgy járunk el mint a másodiknál, azzal a különbséggel, hogy az első két hasáb szélességével kell eltolnunk a harmadik hasábot: left: 500px;.



Fixed

A position: fixed; nagyon hasonlít a position: absolute; beállításához. A különbség, hogy fixed értéknél az elemek a helyükre rögzítve maradnak, tehát akkor is ugyanabban a pozícióban látjuk, amikor lejjebb görgetjük a weboldalt.

Az előző fejezetben használt példát fogom felhasználni a fixed bemutatására.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Fixed</title>
```

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

```
</head>
```

```
<body>
```

```
<div id="kulso">
```

```
<div id="első"> Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a magyar foglyokat.
```

```
 Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a magyar foglyokat.</div>
```

```
<div id="második">Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a magyar foglyokat.
```

```
 Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák, török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér, Móré nyomán van. A kisleány beszámolója alapján megtámadják a törököket, s kiszabadítják a magyar foglyokat.</div>
```

```
<div id="harmadik">Bornemissza Gergőt és Cecey Évát elrabolja Jumurdzsák,
török dervis. Gergő az éjszaka folyamán megszökik Vicuskával. Dobó István a rablóvezér,
Móré nyomán van. A kisfiú beszámolója alapján megtámadják a törököket, s kiszabadítják a
magyar foglyokat.</div>
```

```

```

```
</div>
```

```
</body>
```

```
</html>
```

Az első dolog, ami feltűnhet a `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">` sor. A Mozilla Firefox igen, az Internet Explorer 8 azonban nem képes kezelni a `position: fixed;` beállítást e nélkül a sor nélkül. Az újabb HTML szerkesztők új dokumentum létrehozásánál automatikusan beírják a dokumentumtípus-meghatározást a HTML kódba.

```
#kulso{width: 750px;
margin: 10px;
background-color: #FFC;
border: solid 1px black;}
```

```
#elso{width:250px;
background-color:#C1E6D2;}
```

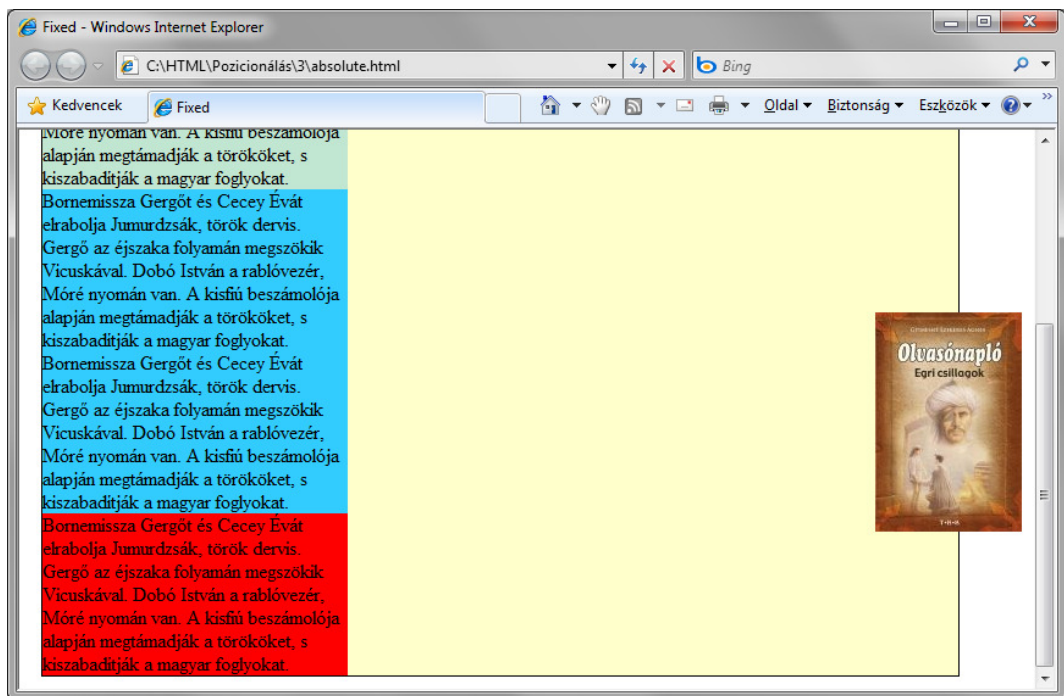
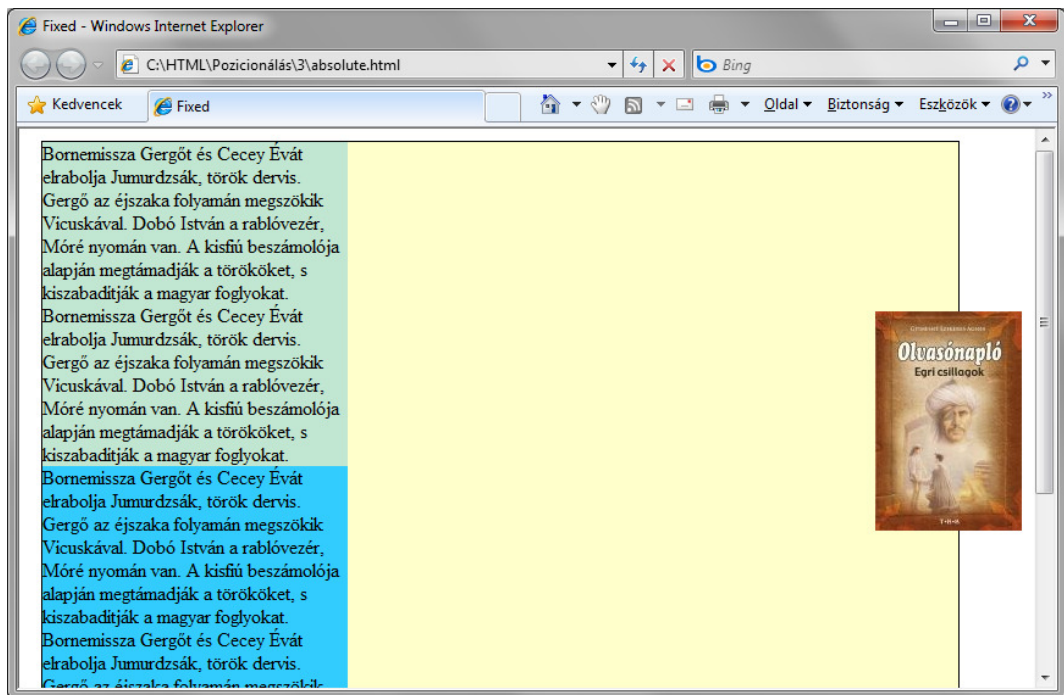
```
#masodik{width:250px;
background-color:#3CF;}
```

```
#harmadik{width:250px;
background-color:#F00;}
```

```
img{position:fixed;
top:150px;
right:10px;}
```

A css is hasonló az előző részben használtakhoz, annyi különbséggel, hogy a pozícionálást kivettem a divekből és hozzáírtam az `img` taghoz tartozó meghatározásokat.

Ebben az esetben egy képet helyezek el a jobb oldaltól 10 pixelre, az oldal tetejétől 150 pixel távolságra és beállítom a position: fixed;-et (22,23).



Menü készítése CSS segítségével

A menü elkészítését egy példán keresztül fogom bemutatni.

Elsőként szükségünk van a HTML kódra.

Egy külső css fájlba fogjuk írni a css kódokat ezért hivatkoznunk kell erre a fájlra. A fájl neve legyen style.css.

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

A következő lépésben létrehozuk a menüt tartalmazó div-et. A css fájlban erre a divre kell majd hivatkoznunk, ezért adnunk kell neki egy azonosítót.

```
<div id="menu"></div>
```

A menüpontokhoz listákat használunk, melyet az imént elkészített div-be írunk be. Elkészítjük a menüpontokhoz tartozó hivatkozásokat is.

```
<div id="menu">
  <ul>
    <li><a href="#"><span class="left"></span><span class="text">menü
1</span><span class="right"></span></a></li>
    <li><a href="#"><span class="left"></span><span class="text">menü
2</span><span class="right"></span></a></li>
    <li><a href="#"><span class="left"></span><span class="text">menü
3</span><span class="right"></span></a></li>
    <li><a href="#"><span class="left"></span><span class="text">menü
4</span><span class="right"></span></a></li>
  </ul>
  <div class="clear"></div>
</div>
```

Egyelőre nem szükséges az egyes menüpontokhoz tartozó pontos hivatkozásokat beírunk, elég ha #-el jelöljük a címet.

Azért fontos minden listaelemhez beírni a <a> tagot, mert később a css-ben meg fogjuk változtatni a tulajdonságait.

Elkészült tehát a menühöz szükséges HTML fájl, amit elmentünk index.html néven.

A css kódot a menüpontok száma nem befolyásolja, tehát a css fájl elkészülte után is kiegészíthetjük a menüt további menüpontokkal.

Térjünk át a css fájlra. Elsőként az oldal alaptulajdonságait állítsuk be.

```
body {  
font-size: 12px;  
font-family: Tahoma, Verdana, Arial;  
color: #000000;  
background-color: #999999;}  
.clear, span.clear {clear: both;}
```

A body-nak állítsunk be egy betűméretet, a betűcsaládot, egy betűszínt és a háttérszínt.

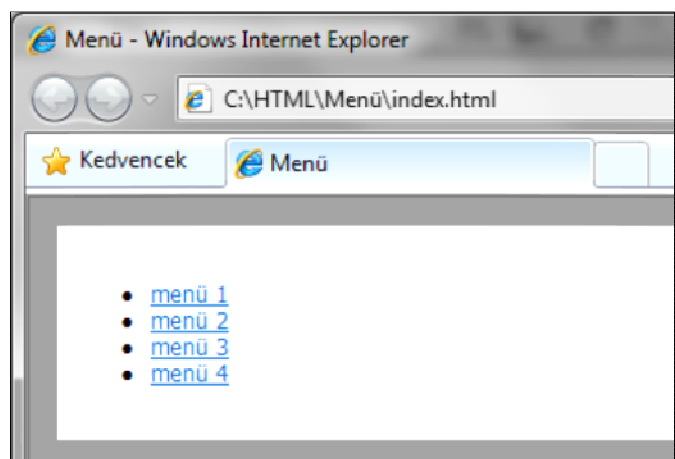
Beállítjuk a float-olt elemek után használatos clear tulajdonságait.

Háttérszínt azért érdemes beállítani, mert később jobban elkülönül majd a menüt tartalmazó div a háttértől.

```
#menu {  
margin: 5px;  
padding: 10px;  
background-color: #ffffff;}
```

A menü div margóját 5px-re és belső margóját 10 px-re állítjuk és beállítunk egy háttérszínt, amivel kiemeljük a bodyből.

A következő lépésekben a listaelemek tulajdonságait állítjuk be.

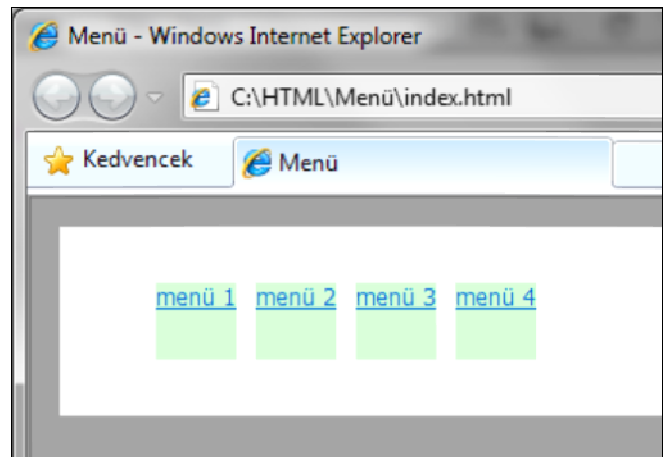


```
#menu ul li {
    display: block;
    float: left;
    height: 40px;
    margin: 0px 10px 0px 0px;
    background-color: #cfc; }
```

A `display: block;` tulajdonság azért kell, hogy később méretezhetőek legyenek a listaelemek.

A menüpontokat balra float-oljuk hogy egymás mellé kerüljenek. Így alakítjuk ki a vízszintes menüt.

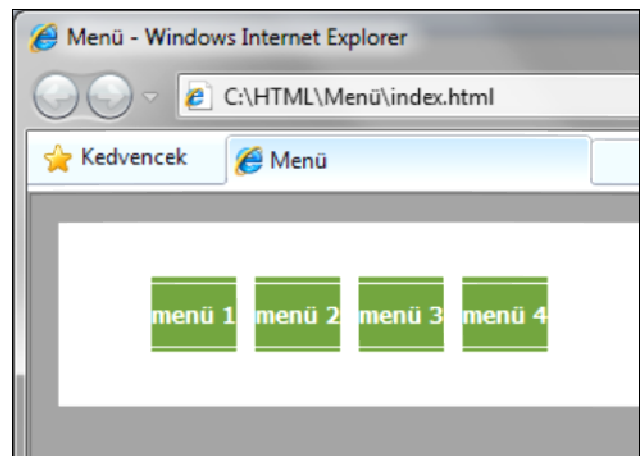
Beállítjuk a menü magasságát és a margóval a menüpontok közötti távolságot, valamint az alapértelmezett háttérszínt.



A következő rész a menüelemekben lévő linkek tulajdonságait határozza meg.

```
#menu ul li a {
    display: block;
    float: left;
    height: 40px;
    line-height: 40px;
    vertical-align: middle;
    background: url(images/bg-center.gif);
    font-weight: bold;
    text-decoration: none;
    color: #fff;
    cursor: pointer; }
```

A hivatkozások magassága és a sor magassága egyezzen meg a listaelemek magasságával, mert így nem csak a szöveg lesz hivatkozás, hanem annak háttere is. A



vertical-align: middle; tulajdonság a szöveget függőlegesen középre helyezi. Minden gomb 3 képből áll, ebben a szakaszban a link háttérképét választjuk ki, ez lesz a középső kép. Az utolsó paranccsal pedig a kurzort változtatjuk kézre, amikor a link fölé visszük az egeret.

```
#menu ul li a span.left, #menu ul li a span.right {  
    display: block;  
    float: left;  
    width: 13px;  
    height: 40px;}
```

A következő lépésben a linkek jobb és bal oldalára kerülő képek helyét készítjük el.

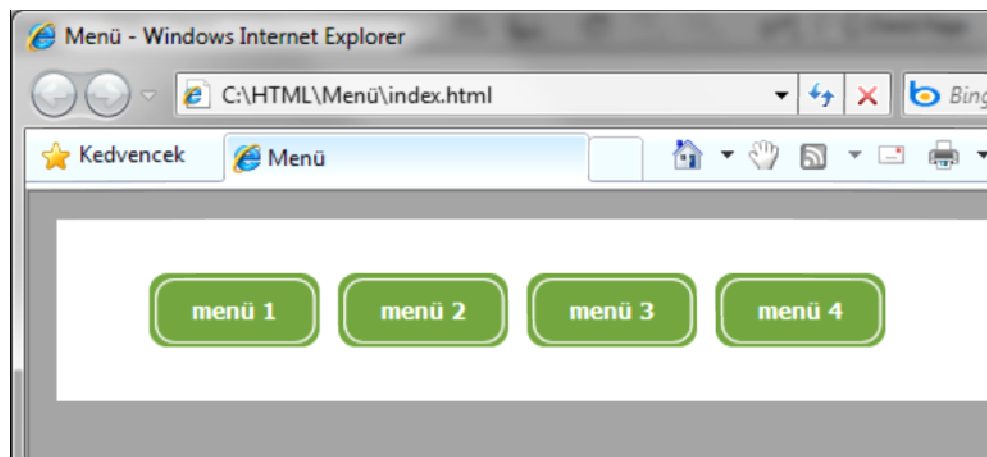
A display: block;-ot és a balra úsztatást itt is be kell állítanunk. A szélességet 13 pixelre állítjuk, mert a jobbra és balra kerülő képek is 13px szélességűek. A magasságot pedig az eddigiek szerint állítjuk be.

```
#menu ul li a span.left { background: url(images/bg-left.gif) }  
#menu ul li a span.right { background: url(images/bg-right.gif); }
```

Háttérképként illesztjük be a jobb és bal oldali képeket.

```
#menu ul li a span.text {  
    display: block;  
    float: left;  
    padding: 0px 10px;  
    height: 40px;}
```

A belső margóval meghatározzuk a szöveg távolságát a gombok szélétől.

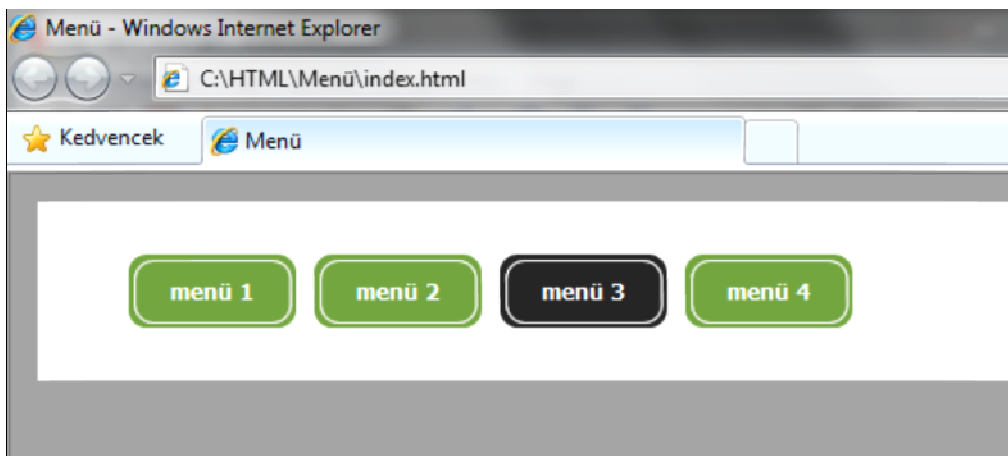


```
#menu ul li a: hover {  
    text-decoration: none;  
    color: #ffffff;  
    background: url(images/bg-center-alt.gif); }  
#menu ul li a: hover span.left { background: url(images/bg-left-alt.gif) }  
#menu ul li a: hover span.right { background: url(images/bg-right-alt.gif); }
```

Az utolsó lépésben a hover állapotú linkeket formázzuk (amikor az egeret a linkre visszük).

Alapértelmezetten a linkek aláhúzottak, de ebben az esetben semmi szükség az aláhúzásra ezért a text-decoration-nel ezt kikapcsoljuk.

Beállítjuk fehér színűre a szöveget. Új háttérképet állítunk be a linknek valamint jobb és bal oldalra is új képek kerülnek (17).



Összefoglalás

Dolgozatom célkitűzése a CSS bemutatása volt. A téma összetettsége miatt, már az elején tisztában voltam azzal, hogy a CSS-nek jelen keretek között csak az alapvető jelentőséggel bíró területeit tudom bemutatni.

Szót ejtettem a CSS nyelvtanáról, mely elengedhetetlen a téma megértéséhez, bemutattam a stílusok típusait hierarchiájuk szerint. Foglalkoztam a CSS-ben használható mértékegységekkel, valamint színmegadási módokkal, a háttér beállításának lehetőségeivel, a bekezdés formázás módjával, azon belül a betű és szövegformázással. Áttekintettem a CSS-ben használható pozicionálási lehetőségeket. Érdekességképpen bemutattam egy CSS-sel készült menüt, amely gyors működése miatt kedvelt eleme a honlapoknak.

Munkám során megállapítottam, hogy a CSS rugalmasságából és egyszerűségéből rengeteg előny fakad. Lehetővé teszi a weboldalak könnyű megváltoztatását, nagymértékben leegyszerűsítve ezzel a weblapkészítők munkáját, ugyanakkor a felhasználó gyors és praktikus oldalakat böngészhet.

Remélem a dolgozatomat olvasva mások számára is világossá válnak a CSS előnyei és útmutatóul szolgálhat a CSS-el ismerkedni kívánók számára.

A későbbiekben szándékomban áll jobban elmélyülni a témában, megvizsgálni többek között annak lehetőségét, hogy hogyan lehet a CSS-t együtt használni más weblapkészítési módszerekkel.

Irodalomjegyzék

1. HTML és CSS : webszerkesztés stílusosan : bevezetés / DeBolt, Virginia. - Kiskapu, Budapest, 2005. xvii-xviii.p.
2. HTML és CSS : webszerkesztés stílusosan : hogyan írjunk XHTML és CSS kódot? / DeBolt, Virginia. - Kiskapu, Budapest, 2005. 8-14.p.
3. w3schools.com : CSS Syntax , http://www.w3schools.com/css/css_syntax.asp letöltve: 2009, december 21.
4. HTML és CSS : webszerkesztés stílusosan : mindent a maga helyére, azaz: hová tegyük a stílusokat? / DeBolt, Virginia. - Kiskapu, Budapest, 2005. 20-29.p.
5. Cascading Style Sheets, level 2 CSS2 Specification : assigning property values, Cascading, and Inheritance / Bos, Bert et. al. - 1998. május 12. <http://www.w3.org/TR/2008/REC-CSS2-20080411/cascade.html> letöltve: 2010. január 24.
6. Weblabor : CSS alapjai VI. - 2004.június 1. <http://weblabor.hu/cikkek/cssalapjai6> letöltve: 2010. január 26.
7. Cascading Style Sheets, level 2 CSS2 Specification : CSS2 syntax and basic data types/ Bos, Bert et. al. - 1998. május 12. <http://www.w3.org/TR/2008/REC-CSS2-20080411/syndata.html#length-units> letöltve: 2010. január 26.
8. Cascading Style Sheets, level 2 CSS2 Specification : CSS2 syntax and basic data types/ Bos, Bert et. al. - 1998. május 12. <http://www.w3.org/TR/2008/REC-CSS2-20080411/syndata.html#color-units> letöltve: 2010. február 14.
9. HTML és CSS : webszerkesztés stílusosan : a weblapok alapjai: DOCTYPE, fejléc, törzs és a törzs stílusai/ DeBolt, Virginia. - Kiskapu, Budapest, 2005 . 48-49.p.
10. w3schools.com : CSS Colors. - http://www.w3schools.com/css/css_colors.asp letöltve: 2010. február 20.
11. w3schools.com : CSS Color Names. - http://www.w3schools.com/css/css_colornames.asp letöltve: 2010. február 20.
12. <http://blogs.atlassian.com/developer/jens/css-is-awesome.jpg> letöltve: 2010. február 26.

13. Weblabor : CSS alapjai II. - 2004. április 6. <http://weblabor.hu/cikkek/cssalapjai2>
letöltve: 2010. február 26.
14. Cascading Style Sheets, level 2 CSS2 Specification : fonts/ Bos, Bert et. al. - 1998.
május 12. <http://www.w3.org/TR/2008/REC-CSS2-20080411/fonts.html> letöltve:
2010. március 5.
15. Weblabor : CSS alapjai II. - 2004. április 6. <http://weblabor.hu/cikkek/cssalapjai2>
letöltve: 2010. március 5.
16. Cascading Style Sheets, level 2 CSS2 Specification : text/ Bos, Bert et. al. - 1998.
május 12. <http://www.w3.org/TR/2008/REC-CSS2-20080411/text.html> letöltve:
2010. március 12.
17. tutorial.hu : resizable tabbed navigation with CSS, avagy méretezhető füles menü/
Harder. - 2007. május 15. [http://www.tutorial.hu/resizable-tabbed-navigation-with-
css-avagy-meretezheto-fules-menu/](http://www.tutorial.hu/resizable-tabbed-navigation-with-css-avagy-meretezheto-fules-menu/) letöltve: 2010. március 20.
18. hotdog.hu : olvasónapló – Gárdonyi Géza: Egri csillagok. - 2008. január. 25.
[http://www.hotdog.hu/magazin/magazin_article.hot?m_id=24689&a_id=345338&h_i
d=74385](http://www.hotdog.hu/magazin/magazin_article.hot?m_id=24689&a_id=345338&h_i
d=74385) letöltve: 2010. március 26.
19. http://bookline.hu/product_images/921/B390191.JPG letöltve: 2010. március 26.
20. standardsmode.hu : méretezés, pozícionálás. - [http://www.standardsmode.hu/html-
css/pozicionalas/](http://www.standardsmode.hu/html-
css/pozicionalas/) letöltve: 2010. április 3.
21. Segédletek.hu : css pozícionálás divekkel/ requiem. - 2006. június 5.
http://www.segedletek.hu/segedletek/css/css_pozicionalas_divekkel.html letöltve:
2010. április 3.
22. HTML és CSS : webszerkesztés stílusosan : a weblap szakaszai: div címke a
weblapszerkezet és az elrendezés kialakításához/ DeBolt, Virginia. - Kiskapu,
Budapest, 2005 . 93-94.p.
23. Cascading Style Sheets, level 2 CSS2 Specification : visual formatting model/ Bos,
Bert et. al. - 1998. május 12. [http://www.w3.org/TR/2008/REC-CSS2-
20080411/visuren.html#fixed-positioning](http://www.w3.org/TR/2008/REC-CSS2-
20080411/visuren.html#fixed-positioning) letöltve: 2010. április 9.