

CSS

I. KÖNYV: ALAPOK

I. CSS – Bevezetés

Mi a CSS?

A CSS a **Cascading Style Sheets** (=lépcsőzetes stílus-oldalak) kifejezés rövidítése. A stílusok segítségével meghatározhatjuk a HTML-elemek megjelenését. A HTML 4.0-ás verziója óta alkalmazzák őket.

Külső stílus-oldal(ak) készítésével sok munkát spórolhatunk meg; egész weboldalakat formázhatunk meg egyetlen stílus-oldalra való hivatkozással (<link> elemmel a HTML dokumentum **head** részében). Emiatt a HTML/XHTML elsajátítása *után* érdemes megtanulnunk a CSS-fájlok készítésével is!

Egy gyors példa

Mint az alábbi példában látni fogjuk, egy adott weboldalt sokféle (akár végtelen) stílusban meg lehet jeleníteni.

Először is beállítjuk az alábbi keretcsoportot (a http://www.w3schools.com/css/demo_default.htm oldalként):

```
<html>
<head>
<title>CSS in Action</title>
</head>
<frameset cols="130,*" frameborder="1">
  <frame name="contents" src="demo_style_content.asp"
noresize="noresize">
  <frame name="main" src="demo_style_main.asp">
  <noframes>
  <body>
  <p>This page is using frames. Your browser does not support it.</p>
  </body>
  </noframes>
</frameset>
</html>
```

Majd megírjuk a **demo_style_content.asp** weboldal tartalmát:

```
<html>
<body>
<b>View the styles:</b>
<br />
<br />
<a href='demo_style_main.asp?style=style1' target='main'>
View Style1</a><br />
<a href='demo_style_main.asp?style=style2' target='main'>
View Style2</a><br />
<a href='demo_style_main.asp?style=style3' target='main'>
View Style3</a><br />
<br />
<a href="demo_style_main.asp" target="main">Without styles</a>
<br />
<br />
<b>View the stylesheets:</b>
<br />
<br />
```

```
<a href='demo_stylesheet_main.asp?style=style1'
target='main'>Style1</a><br />
<a href='demo_stylesheet_main.asp?style=style2'
target='main'>Style2</a><br />
<a href='demo_stylesheet_main.asp?style=style3'
target='main'>Style3</a><br />
</body>
</html>
```

Mint látjuk, a „**main**” keretben a **demo_style_main.asp** oldal jelenik meg, alapértelmezésben az 1. (**style1** szerinti) formázásban. Ezen oldal kódja:

```
<html>
<head>

<link rel="stylesheet" type="text/css" href="style1.css">

</head>

<body>
<h1>Heading 1</h1>
<p>This is some text in a paragraph.</p>
<p>This is another paragraph.</p>
<hr />

<h2>Heading 2</h2>
<table border="1" width="100%">
<tr>
    <th align="left">Name</th>
    <th align="left">E-mail</th>
    <th align="left">Phone</th>
</tr>
<tr>
    <td width="25%">Doe, John</td>
    <td width="25%">jdoe@example.com</td>
    <td width="25%">555-789-7222</td>
</tr>
<tr>
    <td width="25%">Smith, Eva</td>
    <td width="25%">esmith@example.com</td>
    <td width="25%">555-324-3693</td>
</tr>
</table>
<br />
<hr />

<h3>Heading 3</h3>
<p>Visit our
    <a href="http://www.w3schools.com/">
    Home Page</a>
or our
    <a href="http://www.w3schools.com/css/">
    CSS Tutorial</a>.
</p>
<p>What you should already know:</p>
<ol>
```

```

    <li>HTML</li>
    <li>XHTML</li>
</ol>

<p>Favorite drinks:</p>
<ul>
    <li>Smoothie</li>
    <li>Green tea</li>
    <li>Coffee</li>
</ul>

</body>
</html>

```

A formázatlan („without styles”) oldal **<head>** része üres. A 1. és 3. stílusnak megfelelő oldal **<head>** részében rendre **style2.css** és **style3.css** szerepel a **<link>** tagben.

Végül lássuk a CSS-eket, elsőként a **style1.css**-t:

```

body
{
font-size:75%;
font-family:verdana,arial,'sans serif';
background-color:#FFFFFF0;
color:#000080;
margin:10px;
}

h1 {font-size:200%;}
h2 {font-size:140%;}
h3 {font-size:110%;}

th {background-color:#ADD8E6;}

ul {list-style:circle;}
ol {list-style:upper-roman;}

a:link {color:#000080;}
a:hover {color:red;}

```

A style2.css:

```

body
{
font-size:75%;
font-family:"lucida calligraphy",arial,'sans serif';
background-color:#DCDCDC;
color:#8A2BE2;
margin:10px;
}

h1 {font-size:200%;}
h2 {font-size:140%;}
h3 {font-size:110%;}

th {background-color:#D3D3D3;}
td {background-color:#FFFAF0;}

```

```
a:link {color:#8A2BE2;text-decoration:none;}
a:hover {color:red;font-weight:bold;text-decoration:none;}
a:visited {text-decoration:none;}
```

A style3.css:

```
body
{
font-size:75%;
font-family:verdana,arial,'sans serif';
background-image:url('gradient.png');
background-repeat:repeat-x;
background-color:#FFFFFF0;
color:#000080;
margin:70px;
}
```

```
h1 {font-size:200%;}
h2 {font-size:140%;}
h3 {font-size:110%;}
```

```
th {background-color:#ADD8E6;}
```

```
ul {list-style:circle;}
ol {list-style:upper-roman;}
```

```
a:link {color:#000080;}
a:hover {color:red;}
```

Mint a fenti három CSS-ből látszik, a betűméret, betűtípus és -szín, a margók, a táblázat-címek és -cellák, lista-típusok, és nem látogatott (link) az egérrel érintett (hover) linkek színezése és aláhúzottsága (text-decoration) teljességgel szabályozható velük.

A CSS: megoldás egy komoly gondra

A HTML-dokumentumoknak elvileg nem kéne formázó tageket tartalmazniuk. A HTML-tageket az oldal *tartalmának* leírására (pl. `<p>Paragraph</p>` és `<h1>Heading</h1>`), és nem annak formázására találták ki.

Amikor a **** -hoz hasonló tageket és a szín-attribútumokat bekerültek a HTML 3.2 előírásai közé, a weboldal-fejlesztők munkája hirtelen rémálommá változott.

A nagy (sok oldalból álló) webhelyek fejlesztése és fenntartása, ahol betű-megjelenítési és -szín-adatokat kellett minden egyes HTML-elemhez csatolni, hosszú és drága dologgá vált.

A probléma megoldására a World Wide Web Consortium (W3C) megalkotta a CSS-T. A HTML 4.0-ás verziójától kezdve az összes formázási adatot a HTML-dokumentumoktól elkülönítve, CSS-fájlokban lehetett tárolni.

Ma már minden böngésző támogatja a CSS-ek használatát.

A CSS-sel sok munkát takaríthatunk meg!

A CSS-sel meghatározhatjuk a HTML-elemek *kinézetét* (nem a tartalmukat!).

A stílus-adatokat általában külső .css fájl(ok)ba mentik el. Egy ebben végzett egyszerű módosítással a webhely számos (vagy akár összes) oldalának beosztását, kinézetét megváltoztathatjuk.

II. CSS – mondattan

Két gyors példa

Az alábbi weboldal a <link> taggel hivatkozik az **ex1.css** stílus-oldalra:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="ex1.css" />
</head>

<body>
<h1>This header is 36 pt</h1>
<h2>This header is blue</h2>
<p>This paragraph has a left margin of 50 pixels</p>
</body>
</html>
```

Az **ex1.css** fájl tartalma:

```
body
{
background-color:yellow;
}
h1
{
font-size:36pt;
}
h2
{
color:blue;
}
P
{
margin-left:50px;
}
```

Ennek eredményeként a weboldal sárga háttérrel jelenik meg, melyben egy 36-os betűméretű **<h1>**, egy kék **<h2>** és egy 50 pixelnnyire jobbra húzott bekezdés látható.

A következő példában hasonló elrendezéssel találkozunk; az alábbi weboldal az **ex2css**-re hivatkozik:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="ex2.css" />
</head>

<body>
<h1>This is a header 1</h1>
<hr />
<p>You can see that the style sheet formats the text</p>
<p><a href="http://www.w3schools.com" target="_blank">
This is a link</a></p>
</body>
</html>
```

Az **ex2.css** tartalma:

```
body {background-color:tan;}
h1 {color:maroon;font-size:20pt;}
hr {color:navy;}
p {font-size:11pt;margin-left:15px;}
```

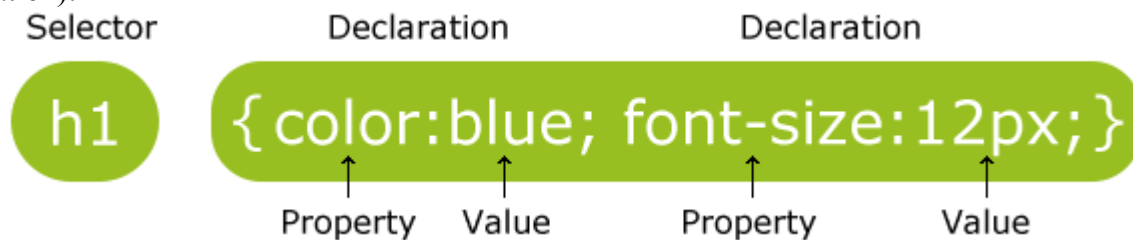
```
a:link {color:green;}
a:visited {color:yellow;}
a:hover {color:black;}
a:active {color:blue;}
```

Az eredmény egy világos-mogyoróbarna háttérű weboldal, 20-as betűméretű, bordó <h1>-gyel, kék <hr />-rel, 11-es betűméretű, 15 pixellel jobbra húzott <p>-vel, és egy sárga linkkel (mivel a www.w3schools.com oldalt már korábban megnyitottam), ami feketére vált, ha az egérrel föléje „kalandozunk” (=hover), és kékre vált, miután rákattitottunk.

Jegyezzük meg, hogy a px=pixel(=képpont) mértékegység nem azonos a pt=point[(=tipográfiai) pont] egységgel; az utóbbi nagyobb!

CSS-mondattan

A CSS-utasítások két fő részből állnak: kijelölésből (=selector) és egy vagy több meghatározásból (=declaration):



A kijelölés általában a formázandó HTML-elem neve.

A meghatározások két részből, jellemzőből és értékből állnak; azaz voltaképpen a beállítani kívánt stílus-attribútumból (pl. **color**) és a hozzá rendelt értékből (pl. green).

Egy újabb példa:

A CSS-meghatározások mindig pontosvesszőre végződnek, a kijelöléshez tartozó meghatározás-csoportokat pedig kapcsos zárójelek fogják közre:

```
p {color:red;text-align:center;}
```

A CSS olvashatóbb, ha az egyes meghatározásokat külön sorba írjuk:

```
p
{
color:red;
text-align:center;
}
```

Az alábbi oldal belső CSS-t tartalmaz, az első írásmódban:

```
<html>
<head>
<style type="text/css">
p{color:red;text-align:center;}
h1{color:green;font-size:20px;}
h2{color:blue;font-size:20pt;}
</style>
</head>

<body>
<h1>Header1</h1>
<h2>Header2</h2>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
```

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

CSS-megjegyzések

A CSS-kódhoz fűzött magyarázatok segítségével a későbbi javítások, átalakítások könnyebben elvégezhetők. A böngészők nem veszik figyelembe őket.

A CSS-megjegyzések „/*” jelekkel kezdődnek és „*/”-re végződnek, amint azt példánk is mutatja:

```
/*This is a comment*/
p
{
text-align:center;
/*This is another comment*/
color:black;
font-family:arial;
}
```

CSS-mondattan:

CSS-utasítás (általános alak): *kijelölés {jellemző:érték;jellemző:érték;}*

CSS-utasítás (konkrét példa): *h1 {color:blue;font-size:12px;}*

Külső stílus-oldal:

Hivatkozás külső stílus-oldalra (a HTML-dokumentum fejrészében):

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

Egy külső .css-fájl tartalma [csak stílus-utasításokat (kijelöléseket + meghatározásokat) tartalmaz]:

```
body {background-color:yellow;}
h1 {font-size:36pt;}
h2 {color:blue;}
p {margin-left:50px;}
```

Belső stílus-oldal:

Általános alak:

```
<style type="text/css">
kijelölés {jellemző:érték;jellemző:érték;}
kijelölés
{
jellemző:érték;
jellemző:érték;
jellemző:érték;
jellemző:érték;
}
</style>
```

Konkrét példa:

```
<style type="text/css">
p {color:red;text-align:center;}
h1 {color:green;font-size:20px;}
h2 {color:blue;font-size:20pt;}
</style>
```

CSS-megjegyzés [a .css fájlba ill. a belső stílus-oldalba (<style> elembe) írható]:

```
/*A megjegyzés szövege*/
```

III. CSS – id és class kijelölések

Az id és class kijelölések

Az egyes HTML-elemtípusok formázása mellett a CSS-ben lehetőség van egyéni kijelölésekre is, az **id** és **class** kijelölésekkel.

Az id kijelölés

Az id ijlölés egyetlen elem egyedi formázására szolgál. Az **id="név"** attribútummal ugyanis elnevezhetjük a HTML-dokumentum egyes elemeit, majd a style sheetben a szokott kettőskeresztes jelöléssel hivatkozhatuk rá, mint egyedi kijelölésre:

```
#para1
{
text-align:center;
color:red;
}
```

Ez a formázás tehát csak arra az elemre vonatkozik, melynek **id**-attribútuma: **id="para1"**; amint ez az alábbiakból látszik:

```
<html>
<head>
<style type="text/css">
#para1 {text-align:center;color:red;background-color:green;}
p {text-align:left;color:black;background-color:red;font-size:20pt;}
</style>
</head>

<body>
<p id="para1">Hello World!</p>
<p>This paragraph is not affected by the style.</p>
</body>
</html>
```

Példánkban az első bekezdés középre igazítva, piros betűszínnel, zöld háttérrel és 20-as betűmérettel fog megjelenni, míg a második balra igazítva, fekete betűszínnel, piros háttérrel és 20-as betűmérettel. Mint látszik, az általánosságban (<p>-re) megadott jellemzők a **#para1**-re is vonatkoznak, annak formázása csak abban tér el az általános alaktól, amit külön megadunk.

Fontos, hogy az **id**-attribútum értéke (az objektum neve) soha ne kezdődjön számmal, mert így a Mozilla Firefox nem tudja feldolgozni!

A class kijelölés

A **class** kijelöléssel egész elemcsoportokat formázhatunk. Az **id**-vel ellentétben tehát többnyire számos elem kijelölésére szolgál. Egy adott **class**-ba bármilyen típusú HTML-elem tartozhat; a böngésző az összeset azonosan fogja formázni.

A formázandó elemeket a tagjeikbe írt **class="név"** attribútummal jelöljük ki az adott class elemlévé, a stílus-lapon pedig a **.név** alakban hivatkozunk rá, amint az alább is látszik:

```
<html>
<head>
<style type="text/css">
.cent{text-align:center;}
</style>
</head>

<body>
<h1 class="cent">Center-aligned heading</h1>
<p class="cent">Center-aligned paragraph.</p>
<h2>Simple heading</h2>
```



```
<p>Simple paragraph.</p>
</body>
</html>
```

Itt az első címsor és bekezdés (melyek beletartoznak a **cent** class-ba) középre, a második kettő pedig balra van igazítva.

Fontos megjegyeznünk, hogy a **class**-nevek sem kezdődhetnek számmal, mert ezt csak az Internet Explorer támogatja, a többi böngésző nem!

Arra is van lehetőség, hogy a classból az azonos típusú elemeket külön formázzuk. Az alábbi példában az összes **cent** class-ba tartozó **<p>** jobbra van igazítva:

```
<html>
<head>
<style type="text/css">
.cent{text-align:center;}
p.cent{text-align:right;color:red;}
h2.cent{text-align:left;color:green;}
</style>
</head>

<body>
<h1 class="cent">This heading will not be affected</h1>
<h3 class="cent">Nor this.</h3>
<p class="cent">This paragraph will be right-aligned and red.</p>
<p class="cent">This too.</p>
<h2 class="cent">This heading will be left-aligned and green</h2>
<h2>This heading will be left-aligned (as default)</h2>
<p>This paragraph will be left-aligned (as default).</p>
</body>
</html>
```

A **<h2>**-elemek pedig balra. Ezek külön (piros ill. zöld) színt is kapnak. A csoport összes többi eleme (pl. a **<h1>** és **<h3>**) a **.cent** általános kijelölésben foglaltak szerint formázódik. (Azok a tagek, melyek nem tartalmazzák a **class** atribútumot, továbbra is alapértelmezett formátumban jelennek meg.)

Természetesen hogyha az egész csoportra vonatkozó **.cent** kijelölésben megváltoztatjuk pl. a betűméretet, akkor az a csoport mindazon tag-típusát megváltoztatja, melyben nincs speciális érték megadva erre. Pl. az alábbi esetben a **<h1>** és **<h2>** kék lesz, a **<p>** piros, a **<h3>** pedig alapértelmezett:

```
<html>
<head>
<style type="text/css">
.cent{text-align:center;color:blue;}
p.cent{text-align:right;color:red;}
h2.cent{text-align:left;}
</style>
</head>

<body>
<h1 class="cent">Header1</h1>
<h2 class="cent">Header2</h2>
<p class="cent">Paragraph.</p>
<h3>Header3</h3>
</body>
</html>
```

CSS id és class kijelölések:

Egyéni kijelölés (id):

Egyéni kijelölés egy elemben: `<p id="név"></p>`

Az elem kijelölése a CSS-ben: `#név`

FONTOS: A *név* nem kezdődhet számmal!

Csoportos kijelölés (class):

Egy csoportba sorolt elem: `<p class="csoportnév"></p>`

A csoport kijelölése a CSS-ben: `.név`

A csoportba tartozó, adott típusú elemek kijelölése a CSS-ben:

`típus.név` (pl.: `h1.class_1`)

FONTOS: A *név* nem kezdődhet számmal!

IV. CSS – módszertan

A CSS beolvasásával a böngésző annak megfelelően formázza a weboldalt.

A CSS három beillesztési módja

A CSS-t háromféleképpen kapcsolhatjuk a weboldalhoz:

- Külső stílus-oldalként
- Belső stílus-oldalként
- A szövegtestbe írt (=inline) stílus-attribútumként

Külső stílus-oldal

A külső stílus-oldal alkalmazása ideális akkor, ha annak tartalmát számos oldalra kívánjuk vonatkoztatni. A benne véghezvitt módosítással így egy egész webhely megjelenését átalakíthatjuk. Az összes formázni kívánt oldalnak hivatkoznia kell a stílus-oldalra, a **<link>** tag segítségével, melyet a **<head>** részükbe írunk:

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

A külső stílus-oldalt bármely szövegszerkesztőben megírhatjuk. Egyetlen HTML-taget sem szabad tartalmaznia! A dokumentumot **.css** kiterjesztéssel kell elmenteni. Az alábbi sorokban egy igen egyszerű stílus-oldal tartalmát látjuk:

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

Megjegyzés: a meghatározások érték-részébe írt számérték és mértékegység között ne hagyjunk szóközt! (Tehát pl. "margin-left:20 px" helyett mindig "margin-left:20px"-t írjunk!) Ezt ugyanis csak az Internet Explorer tolerálja, a Firefox és az Opera böngészők nem!

Belső stílus-oldal

Egyetlen (vagy kevés) oldal egyedi formázására alkalmazandó. A belső stílus-adatokat a HTML-dokumentum fejrészébe írjuk, a **<style>** tag használatával, pl. így:

```
<head>
<style type="text/css">
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>
```

Egy érdekes feltételezés: a **<link>** taggel ugyebár egy külső dokumentum tartalmát idézzük be. Ha van egy oldal (címe: **URL**), melynek **<head>** részébe egy belső stílus-oldalt írtunk, és ezt a részt az **** tagekkel könyvjelöltük, akkor a **<link rel="stylesheet" type="text/css" href="URL#név" />** taggel annak tartalmát egy másik oldalra is beidézhetjük(!?)

Ez azonban, mint a „html-gyakorlat” mappa „kulso-belso_CSS” fájljai mutatják, nem lehetséges, mivel a **<link>** tag a linkelt dokumentum teljes tartalmát beidézi; ami a fejrészben egy új, értelmezhetetlen weblap-tartalom megjelenését eredményezi. Tehát külső CSS csakis olyan fájl lehet, melyben csak CSS kijelölések és meghatározások szerepelnek, vagyis **<a>** tag sem! Így linkelni csak külső CSS-fájlt (vagy más, csak CSS-kódot tartalmazó dokumentumot) lehet!

Szövegtestbe írt (inline) stílus-attribútumok

A stílus-adatokat a dokumentumtestbe írva a stílus-oldalakkal járó sok előnyt nélkülöznünk kell. A formázás keveredése a tartalmi elemekkel nehezebben áttekinthető adatsort szül. Ezért csak a legszükségesebb esetben vegyük igénybe ezt a lehetőséget!

A szövegtesetn belüli formázást a megfelelő tagbe írt **style** attribútumra végezzük, mely bármely CSS-tulajdonságot tartalmazhatja. Az alábbi példában egy bekezdés betűszínét és bal margójának beállító stílus-attribútumot találunk:

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

Többszörös formázás (Multiple Style Sheets)

Ha az adott oldalra vonatkozó különböző (belső vagy külső) stílus-oldalak ugyanarra a kijelölésre [pl. a harmadrendű (<h3>) címsorokra] vonatkozólag többféle formázást is megadnak, akkor a böngésző a stílus-oldalak hierarchiája, és a formázások részletessége alapján, összetett módon formázza az adott elemet.

Például egy külső stílus oldal a **h3** kijelölést a következőképpen formázza:

```
h3
{
color:red;
text-align:left;
font-size:8pt;
}
```

A formázandó weblapba írt (belső) stílus oldal azonban a következőket tartalmazza:

```
h3
{
text-align:right;
font-size:20pt;
}
```

A szövegben található egyik <h3> címsor pedig így néz ki:
<h3 style="font-size:40pt">Heading_3</h3>

Ekkor az előbbi bekezdés a következő stílussal jelenik meg:

```
color:red;
text-align:right;
font-size:40pt;
```

Tudniillik a külső stílus-oldal szín tekintetében részletesebb a belsónél, így ezt a formázási adatot a belső stílus-oldal nem érinti. A belső stílus-oldal a külsőnél magasabb rendű lévén, felülírja annak szövegigazítási és -méretbeli utasításait. Így az oldal „közönséges” <h3> elemeit a böngésző a következőképpen formázza:

```
color:red;
text-align:right;
font-size:20pt;
```

Mint láttuk, a fenti <h3> taghez külön betűméretet adtunk meg, ami „inline” stílus-attribútumként még a belső stílus-oldalnál is magasabb rendű lévén felülírja az előbbi formázást ebben a tekintetben, ahogy az első formázási eredményben láttuk.

A többszörös formázás egyé olvad

Mint láttuk, a formázási jellemzőket megadhatjuk a HTML-elemben, melyre vonatkozik, illetve belső vagy külső stílus-oldalban. Sőt, egy dokumentumhoz több külső CSS-t is kapcsolhatunk (<link> taggel, a fejrészből).

Hogyha egy HTML-elemre többféle stílus-információt is megadtunk, akkor ezek egy új, virtuális stílussá összegződnek, a következő prioritási sorrendben:

1. Szövegtestbe írt stílus (**style** attribútum a HTML-elemben);
2. Belső stílus-oldal (a dokumentum **<head>** részében);
3. Külső stílus-oldal;
4. Böngésző-alapértelmezés.

Azaz a HTML-elembe írt „inline” stílusnak van a legmagasabb prioritása. Ez felülírja mind a fejrészben, mind a külön tárolt CSS-t, és a böngésző alapértelmezett stílusát is.

Hogyha pedig a külső stílus-oldal linkjét a belső stílus-leírás után írjuk a dokumentum **<head>** részébe, akkor a külső stílus-oldal tartalma felülírja a belsőét!

CSS-módszertan:

A CSS beillesztése:

1. **Külső CSS:** `<link rel="stylesheet" type="text/css" href="URL" />`

(A külső forrás egy különálló CSS-fájl, melyben csakis CSS kijelölések és meghatározások lehetnek, egyéb tartalom nem! A link tag az oldal teljes tartalmát bemásolja!)

HIBÁS: A külső forrás lehet egy különálló .css-fájl, vagy egy másik weboldal belső stílus-oldala. Utóbbit egy `<a>` taggel tehetjük linkelhetővé (pl. a belső CSS-t az `` taggel könyvjelzőzzük, és linkeljük: `<link rel="stylesheet" type="text/css" href="URL#név" />`).

2. **Belső CSS:** `<style type="text/css">`
kijelölés {jellemező:érték;}
`</style>`

3. **Szövegtestbe írt (inline) stílus-attribútumok, pl.:**

`<p style="color:green;margin-left:20px"></p>`

A CSS-típusok prioritási sorrendje:

1. Szövegtestbe írt stílus (**style** attribútum a HTML-elemben);
2. Belső stílus-oldal (a dokumentum **<head>** részében);
3. Külső stílus-oldal;
4. Böngésző-alapértelmezés.

A sorban előbb állók felülírják az utóbbiakat. A csak egy helyen meghatározott (tehát lejjebb felül nem írt) jellemzők az adott kijelölés szintjén ill. az alatt érvényesek.

CSS

II. KÖNYV: FORMÁZÁS

I. CSS – hátterek

A CSS háttér-jellemzők a HTML-elemek háttér-effektusainak beállítására valók.

Az egyes háttér-hatások elérésére szolgáló CSS-jellemzők:

- **background-color**
- **background-image**
- **background-repeat**
- **background-attachment**
- **background-position.**

Background-color (háttérszín)

A **background-color** jellemző az elemek háttérszínének beállítására szolgál.

Az egész HTML-oldal háttérének beállítására a **body** kijelölést használjuk:

```
<html>
<head>
<style type="text/css">
body {background-color:#b0c4de;}
</style>
</head>
```

```
<body>
<h1>My CSS web page!</h1>
<p>Hello world! This is a W3Schools.com example.</p>
</body>
</html>
```

A háttérszín névvel (pl.: "red") vagy RGB- ill. HEX-értékkel jellemezhetjük (pl.: "rgb(255,0,0)" ill. "#ff0000").

Az alábbi példában a **<h1>**, **<p>** és **<div>** elemeknek eltérő háttérszínük van:

```
<html>
<head>
<style type="text/css">
h1 {background-color:#6495ed;}
p {background-color:#e0ffff;}
div {background-color:#b0c4de;}
</style>
</head>

<body>
<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has it's own background color.</p>
We are still in the div element.
</div>
</body>
```

```
</html>
```

A **<div>** elem egy tatalmi részt (=division) jelöl ki a HTML-dokumentumon belül. Gyakran tagek egységes formázáshoz való öszegyűjtésére ill. csoportosítására szolgál.

Background-image (háttér-kép)

A **background-image** jellemző az elem háttérül szolgáló kép beállítására szolgál. Alapértelmezett esetben a böngésző a képet többször ismételve töltheti ki az elem háttérét.

Egy egész oldal háttér-kép – beállítását mutatja a következő példa:

```
<head>
<style type="text/css">
body {background-image:url('paper.gif');}
</style>
</head>
```

```
<body>
<h1>Hello World!</h1>
</body>
</html>
```

Ebben az esetben tehát a viszonylag kicsi papír-textúra – képet a böngésző sok példányban helyezi a **Hello World!** felirat mögé. Amint látjuk, a hivatkozott dokumentum címét egyszeres idézőjelbe tettük. Ez azonban nem feltétlenül szükséges.

Hasonló példát találunk alább:

```
<html>
<head>
<style type="text/css">
body {background-image:url('bgdesert.jpg');}
</style>
</head>
```

```
<body>
<h1>Hello World!</h1>
<p>This text is not easy to read on this background image.</p>
</body>
</html>
```

Itt a rosszul megválaszott (sok szabálytalan sötét vonalat tartalmazó) **bgdesert.jpg** háttér-kép miatt a bekezdés szövege szinte olvashatatlan.

Background-repeat, azaz a háttérkép vízszintes ill. függőleges ismétlése

Alapértelmezésben a **background-image** jellemző a képet vízszintesen és függőlegesen egyaránt ismételteti a kitöltéshez. Egyes képeket azonban csak vízszintesen vagy függőlegesen célszerű ismételni, különben furcsán néznek ki, pl.:

```
<html>
<head>
<style type="text/css">
body {background-image:url('gradient2.png');}
</style>
</head>
```

```
<body>
<h1>Hello World!</h1>
</body>
```

```
</html>
```

Itt felülről lefelé halványodó sávokat látunk egymás felett, holott elegendő volna az oldal tetejére egyetlen sáv.

Ezért használjuk a **background-repeat** jellemzőt, melynek értéke itt **repeat-x** legyen:

```
<html>
<head>
<style type="text/css">
body {background-image:url('gradient2.png');background-repeat:repeat-
x;}
</style>
</head>
```

```
<body>
<h1>Hello World!</h1>
</body>
</html>
```

Így a kép csak egyetlen csíkban ismétlődik az oldal tetején, ami kedvezőbb látványt nyújt.

No-repeat és background-position

Mint fentebb láttuk, lyán háttér-képre van szükség, mely nem zavarja a szöveg olvashatóságát.

A háttérkép egyszeri megjelenítéséhez a **background-repeat** jellemző értékét **no-repeat** -nek vegyük:

```
<html>
<head>
<style type="text/css">
body {background-image:url('img_tree.png');background-repeat:no-
repeat;}
</style>
</head>
```

```
<body>
<h1>Hello World!</h1>
<p>W3Schools background image example.</p>
<p>The background image is only showing once, but it is disturbing the
reader!</p>
</body>
</html>
```

Így az összes felirat mögött egyetlen világos-rózsaszín lombú fa áll, barna törzssel és az ágán ugyanolyan bagollyal. A bagoly és a törzs színe nehezíti a fekete betűk elolvasását.

Hogyha most a **no-repeat** értéket **repeat-x** -re ill. **repeat-y** -ra cseréljük, a fás kép egy vízszintes ill. függőleges sávban lesz látható; hogyha pedig az egész **background-repeat** jellemzőt kitöröljük, akkor mindkét irányban többször ismétlődve, az egész ablakot kitölti.

A fenti zavaró hatás kiküszöbölhető, ha a képet áthelyezzük. Ezt a **background-position** jellemzővel érhetjük el:

```
<html>
<head>
<style type="text/css">
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
background-position:right top;
```



```
margin-right:215px;
}
</style>
</head>

<body>
<h1>Hello World!</h1>
<p>W3Schools background no-repeat, set position example.</p>
<p>Now the background image is only show once, and positioned away from
the text.</p>
<p>In this example we have also added a margin on the right side, so
the background image will never disturb the text.</p>
</body>
</html>
```

Itt a kép csak egyszer, az ablak jobb-felső sarkában szerepel, a szöveget pedig jobbról egy 215 pixeles margó tartja távol tőle.

Háttér-jellemzők – gyorsírással!

Amint látjuk, a háttér beállításakor sokféle jellemzővel van dolgunk. A kód lerövidítésére a meghatározásokat (azaz jellemző+érték párokat) összevonhatjuk, egyetlen hosszú meghatározássá. Ezt a jellemzők gyorsírásának (=shorthand property) nevezett módszerrel végezzük.

A **background** jellemző gyorsírásban szintén **background**, ahogy az az előző példa gyorsírást változatában, alább látszik:

```
<html>
<head>
<style type="text/css">
body
{
background:#ffffff url('img_tree.png') no-repeat right top;
margin-right:215px;
}
</style>

</head>
<body>
<h1>Hello World!</h1>
<p>Now the background image is only show once, and positioned away from
the text.</p>
<p>In this example we have also added a margin on the right side, so
the background image will never disturb the text.</p>
</body>
</html>
```

Az előzővel megegyező eredményt értünk el: a háttér-kép a szövegtől jobbra, elkülönülve áll. A **background** jellemzőhöz tartozó értékek szóközzel elválasztva sorakoznak. Sorrendjük fontos:

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

Nem baj, ha valamelyik értéket kihagyjuk (nem specifikáljuk); csak az a fontos, hogy az egyes értékek ebben a sorrendben kövessék egymást.

A következő példában egy komolyabb belső stílus-oldalt látunk, az előbbi rövidítési elvek alkalmazásával:

```

<html>
<head>
<style type="text/css">
body {margin-left:200px;
background:#5d9ab2 url('img_tree.png') no-repeat top left;}
.container {text-align:center;}
.center_div {border:1px solid gray;
margin-left:auto;
margin-right:auto;
width:90%;
background-color:#d0f0f6;
text-align:left;
padding:8px;}
</style>
</head>

<body>
<div class="container">
<div class="center_div">
<h1>Hello World!</h1>
<p>This example contains some advanced CSS methods you may not have
learned yet. But, we will explain these methods in a later chapter in
the tutorial.</p>
</div>
</div>
</body>
</html>

```

A **container** class-ba tartozó **div**-et a képtől jobbra rendelkezésére álló hely közepre igazítottuk.

Egy további példa

Itt a böngésző a kicsi háttér-képet egyetlenszer használja, amikor az alapértelmezett bal-felső sarokba kirajzolja. A **background-attachment:fixed** meghatározás értelmében az oldal legördítésekor a háttérkép nem mozdul el, hanem mindig azonos helyen látszik:

```

<html>
<head>
<style type="text/css">
body
{
background-image:url('smiley.gif');
background-repeat:no-repeat;
background-attachment:fixed;
}
</style>
</head>

<body>
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
(...Ugyanez sokszor, mint hosszú oldal-tartalom...)
<p>The background-image is fixed. Try to scroll down the page.</p>
<p>The background-image is fixed. Try to scroll down the page.</p>
</body>
</html>

```

CSS background jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
<u>background</u>	Az összes háttér-érték egyszeri beállítására szolgál.	<i>background-color</i> <i>background-image</i> <i>background-repeat</i> <i>background-attachment</i> <i>background-position</i> inherit	1
<u>background-attachment</u>	Megadja, hogy a háttérkép rögzített, vagy mindig látható (az oldal legördítésére is).	scroll fixed inherit	1
<u>background-color</u>	Az elem(ek)háttér-színét határozza meg.	<i>color-rgb</i> <i>color-hex</i> <i>color-name</i> transparent inherit	1
<u>background-image</u>	Az elem(ek)háttér-képét határozza meg.	url(URL) none inherit	1
<u>background-position</u>	A háttér-kép (kezdő)helyzetét határozza meg.	left top left center left bottom right top right center right bottom center top center center center bottom <i>x% y%</i> <i>xpos ypos</i> inherit	1
<u>background-repeat</u>	A háttér-képpel való térkitöltés módját adja meg.	repeat repeat-x repeat-y no-repeat inherit	1

CSS – hátterek:

Háttér-szín: {background-color:#ffffff;} (HEX)
 {background-color:rgb(255,255,255);} (RGB)
 {background-color:green;} (triviális színnév)

HTML-dokumentumrész: <div></div>

Háttér-kép:

Alapértelmezett (a kitöltéshez végtelenig ismételt) háttérkép-beállítás:

body {background-image:url('URL');}

Kitöltési hatás beállítása:

body {background-repeat:repeat/repeat-x/repeat-y/no-repeat/inherit;}

Háttérkép pozicionálása (no-repeat mellett):

body {background-position:left top/left center/left bottom/right top/right center/ right bottom/center top/center center/center bottom/100% 100%/100px 100px/inherit;}

Háttérkép rögzítése (no-repeat mellett):

body {background-attachment:scroll/fixed/inherit;}

Háttér-jellemzők gyorsírása:

{background:színnév url('URL') kitöltés-típus pozíció;}

Egy konkrét példa:

{background:#ffffff url('image.jpg') no-repeat right top;}

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

1. background-color

2. **background-image**
3. **background-repeat**
4. **background-attachment**
5. **background-position**

II. CSS – szöveg

Egy gyors példa:

```
<html>
<head>
<style type="text/css">
h1
{
text-align: center;
text-transform: uppercase;
color: #A7C942;
}
p
{
text-indent: 50px;
text-align: justify;
letter-spacing: 3px;
}
a
{
text-decoration: none;
}
</style>
</head>

<body>
<h1>text formatting</h1>
<p>This text is styled with some of the text formatting properties. The
heading uses the text-align, text-transform, and color properties.
The paragraph is indented, aligned, and the space between characters is
specified. The underline is removed from the
<a target="_blank" href="tryit.asp?filename=trycss_text">
"Try it yourself"</a> link.</p>
</body>
</html>
```

Mint látszik, a **<h1>** nagybetűkkel, zöld színben, középre igazítva fog megjelenni. A **<p>** elején egy 50 pixeles tabulátor van; a szöveg sorkizárt, 3 pixeles betűközzel. A link az alapértelmezéssel szemben nincs aláhúzva, csak a betűszíne kék.

Betűszín

A **color** jellemző a szöveg betűszínére vonatkozik. Megadható névvel, RGB- és HEX-értékkel, pl.:

- "red"
- "rgb(255,0,0)"
- "#ff0000"

Az szövegtest alapértelmezett betűszínét a **body** kijelölés mellett állíthatuk be, pl.:

```
<html>
<head>
<style type="text/css">
body {color:red;}
h1 {color:#00ff00;}
p.ex {color:rgb(0,0,255);}
</style>
</head>

<body>
```

```
<h1>This is a green Heading 1</h1>
<h2>This is a Heading2</h2>
<p>This is an ordinary paragraph. Notice that this text is red. The
default text-color for a page is defined in the body selector.</p>
<p class="ex">This is a paragraph with class="ex". This text is
blue.</p>
</body>
</html>
```

Itt a tulajdonképpeni szövegtest csupán a `<p>` bekezdésből és a `<h2>` címsorból áll, melyek betűszíne piros. A `<p class="ex">` bekezdés, a `ex` „csoport” részeként kék betűszínű. A `<h1>` címsorokra is külön, zöld betűszínű formázás vonatkozik.

Figyelem! A W3C előírásai szerint a `color` jellemzővel együtt a `background-color` jellemzőt is be kell állítani!

Szövegigazítás

A `text-align` jellemzőt a szöveg vízszintes pozicionálására használjuk. A szöveget balra (=left), középre (=center) vagy jobbra (=right) igazíthatjuk, ill. sorkizárttá tehetjük (=justify).

A `text-align` értékét `justify`-nak választva minden sor egyenlő szélességben tölti ki a bal és jobb (függőleges) margó közti helyet, az újságok cikk-oszlopaihoz hasonlóan.

Egy szövegigazítási példa:

```
<html>
<head>
<style type="text/css">
h1 {text-align:center;}
p {text-align:left;}
p.date {text-align:right;}
p.main {text-align:justify;}
</style>
</head>

<body>
<h1>CSS text-align Example</h1>
<p class="date">May, 2009</p>
<p class="main">In my younger and more vulnerable years my father gave
me some advice that I've been turning over in my mind ever since.
'Whenever you feel like criticizing anyone,' he told me, just remember
that all the people in this world haven't had the advantages that
you've had.'</p>
<p><b>Note:</b> Resize the browser <br />window to see how the <br
/>value "justify" works.</p>
</body>
</html>
```

Itt a címsor és a három paragrafus mindegyikénél más szövegigazítási módot alkalmaztunk.

Text-decoration

A `text-decoration` jellemző különféle (köztük alapértelmezett) szöveghatások hozzáadására ill. eltávolítására szolgál.

Leggyakrabban a linkek aláhúzásának kikapcsolására használjuk (azaz esztétikai céllal):

```
<html>
<head>
<style type="text/css">
a {text-decoration:none;}

```

```
</style>
</head>

<body>
<p>Link to: <a href="http://www.w3schools.com">W3Schools.com</a></p>
</body>
</html>
```

A fenti példában a „W3schools.com” felirat aláhúzás nélkül jelenik meg.

A következő példában különböző szöveghatásokat állítunk be:

```
<html>
<head>
<style type="text/css">
h1 {text-decoration:overline;}
h2 {text-decoration:line-through;}
h3 {text-decoration:underline;}
h4 {text-decoration:blink;}
</style>
</head>

<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<h3>This is heading 3</h3>
<h4>This is heading 4</h4>
<p><b>Note:</b> The "blink" value is not supported in IE, Chrome, or
Safari.</p>
</body>
</html>
```

Az **overline**, **line-through**, **underline** és **blink** értékek beállításával rendre felülhúzottá, áthúzottá, aláhúzottá illetve villogóvá tehetjük a szöveget.

Megjegyezzük, hogy nem célszerű a linkeken kívül más feliratot aláhúzni, mert ez megzavarhatja a felhasználót!

Text-transform

A **text-transform** jellemzővel a szöveg is- ill. nagybetűs szedését szabályozhatjuk:

```
<html>
<head>
<style type="text/css">
p.uppercase {text-transform:uppercase;}
p.lowercase {text-transform:lowercase;}
p.capitalize {text-transform:capitalize;}
</style>
</head>

<body>
<p class="uppercase">This is some text.</p>
<p class="lowercase">This is some text.</p>
<p class="capitalize">This is some text.</p>
</body>
</html>
```

Mint látjuk, az **uppercase** értékkel csupa nagybetűs, a **lowercase**-szel csupa kisbetűs szöveget érhetünk el, annak eredeti aakjától függetlenül. A **capitalize** érték a szöveg minden szavának kezdőbetűjét nagygyá teszi.

Szöveg-behúzás (text-indentation)

A **text-indentation** jellemzőt a szöveg (bekezdés) első sorának tabulálására használjuk:

```
<html>
<head>
<style type="text/css">
p {text-indent:50px;}
</style>
</head>

<body>
<p>In my younger and more vulnerable years my father gave me some
advice that I've been turning over in my mind ever since. 'Whenever you
feel like criticizing anyone,' he told me, just remember that all the
people in this world haven't had the advantages that you've had.'</p>
</body>
</html>
```

A fenti példában a **<p>** kezdete 50 pixelre lesz a bal margótól.

További példák

Első példánkban a betűköz-beállítást változtatjuk:

```
<html>
<head>
<style type="text/css">
h2.egy {letter-spacing:3px;}
h2.ketto {letter-spacing:-3px;}
</style>
</head>

<body>
<h2>This is heading 2</h2>
<h2 class="egy">This is heading 2</h2>
<h2 class="ketto">This is heading 2</h2>
</body>
</html>
```

Az első címsorban normális, a másodikban annál nagyobb, a harmadikban pedig annál kisebb a betűköz. A **letter-spacing=0** érték a szokványos szövegalaknak felel meg.

A következő példában a sorköz-beállítás módját láthatjuk:

```
<html>
<head>
<style type="text/css">
#small {line-height:90%;}
#big {line-height:200%;}
</style>
</head>

<body>
<p>
This is a paragraph with a standard line-height.
The default line height in most browsers is about 110% to 120%.
This is a paragraph with a standard line-height.
</p>
<p id="small">
This is a paragraph with a smaller line-height.
This is a paragraph with a smaller line-height.
```



```
This is a paragraph with a smaller line-height.
</p>
<p id="big">
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
This is a paragraph with a bigger line-height.
</p>
</body>
</html>
```

Az első bekezdés normál (**line-height=100%**-nak megfelelő) sorközzel bír, a második, egyedleg kijelölt bekezdés annál kisebbel, a harmadik pedig nagyobbbal.

A következő példában a szövegirány módosítását tanulmányozhatjuk:

```
<html>
<head>
<style type="text/css">
div.ex1 {direction:rtl;}
p {direction:rtl;}
</style>
</head>
<body>

<div>Some text. Default writing direction.</div>
<div class="ex1">Some text. Right-to-left direction.</div>
</body>
</html>
```

Itt az **ex1** osztályba tartozó részbe írt szöveget a böngésző jobbra igazítva jeleníti meg, úgy, hogy az írásjel a mondat elejére kerül.

Az alábbi példában a szóköz-beállítást láthatjuk:

```
<html>
<head>
<style type="text/css">
p {word-spacing:30px;}
</style>
</head>

<body>
<p>This is some text. This is some text.</p>
</body>
</html>
```

A böngésző így minden egyes szó közt 30 pixel helyet hagy.

A következő példában szerepő hosszú bekezdést a böngésző nem tördeli az ablak méretének megfelelően, hanem az ablakméretet igazítja ahhoz (egy alsó csúszka segítségével):

```
<html>
<head>
<style type="text/css">
p {white-space:nowrap;}
</style>
</head>

<body>
<p>
This is some text. This is some text. This is some text.
```

```

This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
</body>
</html>

```

A formázott szöveg „kilóg” látómezőből, egy csúszkával lehet megnézni az éppen nem látszó részét. E beállítás mellett a többszörös szóközök – mint rendszeren – egygyé dolvadnak; ám a szöveg csak a **
** tageknél szenved sortörést, egyébként egyenesen fut a végtelenségig.

Ha a **white-space** jellemző értéke **pre**, akkor a böngésző a szöveget szerkezetileg formázatlanul jeleníti meg; azaz megőrzi a halmozott szóközöket és csak a **
** tagek jelenléte esetén végez sortörést.

Alábbi példánkban a képek szöveghez képesti pozícionálását figyelhetjük meg:

```

<html>
<head>
<style type="text/css">
img.top {vertical-align:text-top;}
img.bottom {vertical-align:text-bottom;}
</style>
</head>

<body>
<p>An  image with a default alignment.</p>
<p>An  image with a text-top alignment.</p>
<p>An  image with a text-bottom alignment.</p>
</body>
</html>

```

Az első kép az alapértelmezés szerinti **bottom** helyzetben látszik, azaz az alja egybeesik a szöveg aljával. A második kép teteje pedig egybeesik a szöveg tetejével, így **top** helyzetű.

CSS szöveg-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
color	Betűszín	<i>color</i>	1
direction	Szövegirány	ltr rtl	2
line-height	Sorköz	normal <i>number</i> <i>length</i> %	1
letter-spacing	Betűköz	normal <i>length</i>	1
text-align	Szövegigazítás	left right center justify	1
text-decoration	Szöveghatás	none underline overline line-through blink	1
text-indent	Kezdő tabulátor	<i>length</i> %	1

text-shadow		none color length	
<u>text-transform</u>	Nagy/kisbetűs vagy kapitális kezdetű szedés	none capitalize uppercase lowercase	1
unicode-bidi		normal embed bidi-override	2
<u>vertical-align</u>	A HTML-elem függőleges elhelyezkedését szabja meg.	baseline sub super top text-top middle bottom text-bottom length %	1
<u>white-space</u>	A szövegtördelést befolyásolja.	normal pre nowrap	1
<u>word-spacing</u>	Szóköz	normal length	1

CSS – szöveg:

Betűszín (itt az egész szövegtestre): `body {color:green/rgb(255,255,255)/#ffffff;}`

FONTOS: A szín megadásához a background-color-t is be kell állítani!

Szöveghatások (text-decoration; text-transform):

Nincs/alapértelmezett: `{text-decoration:none;}`

Felülhúzott: `{text-decoration:overline;}`

Áthúzott: `{text-decoration:line-through;}`

Aláhúzott: `{text-decoration:underline;}`

Villogó: `{text-decoration:blink;}`

Nagybetűs: `{text-transform:uppercase;}`

Kisbetűs: `{text-decoration:lowercase;}`

Nagy kezdőbetűk: `{text-decoration:capitalize;}`

Nincs/alapértelmezett: `{text-decoration:none;}`

Szövegirány: `{direction:ltr/rtl;}`

Térközök a szövegben:

Betűköz: `{letter-spacing:±100px/normal;}` **(Alapértelmezett értéke: 0px.)**

Szóköz: `{word-spacing:100px/normal;}`

Sorköz: `{line-height:/normal/100/100px/100%;}` **(Alapértelmezett értéke: 100%.)**

Tabulálás, szöveg-behúzás (text-indent):

`{text-indent:100px/100%;}`

Szövegigazítás (text-align):

Balra zárt: `{text-align:left;}`

Középre: `{text-align:center;}`

Jobbra zárt: `{text-align:right;}`

Sorkizárt: `{text-align:justify;}`

Beágyazott elem (függőleges) pozicionálása a szöveg-sorhoz képest (vertical-align):

`img {vertical-align:baseline/sub/super/top/text-top/middle/bottom/text-bottom/100px/100%;}`

Automatikus szövegtördelés:

Engedélyezve (alapértelmezett; a szóközök egybeolvadnak, és a szöveg megtörhet rajtuk):

`{white-space:normal;}`

Tiltva (a szóközök egybeolvadnak, de a szöveg nem törhet meg rajtuk):

```
{white-space:nowrap;}
```

Kikapcsolva (a szóközök megmaradnak és a szöveg nem törhet meg rajtuk):

```
{white-space:pre;}
```

[A szöveg-formázásnak nincs gyorsírási jellemzője (pl.: *text*).]

III. CSS – betűk

A CSS **font** jellemző a betűtípus és -méret kiválasztására ill. a különféle betűhatások (pl. félkövér) és stílusok hozzáadását szolgálja.

A serif és sans-serif betűtípusok közti különbség
a következő ábrán látszik:



A serif „talpakkal” rendelkező, a sans-serif pedig azt nélkülöző betűképet jelent. Utóbbiak a számítógépes felhasználás szempontjából kedvezőbbek, mivel könnyebben olvashatók a képernyőn.

CSS – betűcsaládok és betűtípusok

A CSS-ben kétféle betűtípus-fogalommal találkozunk:

generic family (=betűcsalád) név alatt az azonos tulajdonságú (pl. „Serif” illetve „Monospace”) betűtípusok összességét értjük.

font family (=betűtípus) egyedi tipográfiájú betűtípus. Különféle változatai többféle betűcsaládba tartozhatnak.

A következő táblázatban a főbb betűcsaládokat és az azokba tartozó legjellegzetesebb betűtípusokat látjuk:

Betűcsalád	Betűtípus	Leírás
Serif	Times New Roman Georgia	Egyes serif betűk végén kis kacsok találhatóak.
Sans-serif	Arial Verdana	A sans-serif (=„kacs-nélküli”) betűk kacs-nélküliek.
Monospace	Courier New Lucida Console	A monospace betűk mindegyike azonos szélességű.

A betűcsalád/betűtípus beállítása

a **font-family** jellemzővel történik. Ide számos betűtípust beírhatunk, arra az esetre, hogyha a böngésző a sorban legelöl állótnem támogatná. Ekkor a következővel próbálja megjeleníteni az oldalt, s ha azt sem ismeri, akkor a következővel, stb.. Célszerű tehát a listát a kiválasztott betűtípussal kezdeni, és egy betűcsalád nevével befejezni, hogy a böngésző végső esetben ezen családból ismert betűkészletéből válasszon betűtípust.

Hogyha a betűtípus neve több szóból áll, akkor idézőjelbe kell tenni, pl. „Times New Roman”. A lista tagjai közé egyszerű vesszőt (,) írunk, pl.:

```
p{font-family:"Times New Roman", Times, serif;}
```

Az alábbi példa két bekezdését eltérő betűtípusokkal formáztuk:

```
<html>
<head>
<style type="text/css">
p.serif{font-family:"Times New Roman",Times,serif;}
p.sansserif{font-family:Arial,Helvetica,sans-serif;}
</style>
</head>

<body>
```

```

<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>
</body>
</html>

```

A leggyakrabban használt betűnév-kombinációkról az alábbi táblázat segítségével tájékozódhatunk:

Serif Fonts

font-family	Example text
Georgia, serif	This is a heading This is a paragraph
"Palatino Linotype", "Book Antiqua", Palatino, serif	This is a heading This is a paragraph
"Times New Roman", Times, serif	This is a heading This is a paragraph

Sans-Serif Fonts

font-family	Example text
Arial, Helvetica, sans-serif	This is a heading This is a paragraph
Arial Black, Gadget, sans-serif	This is a heading This is a paragraph
"Comic Sans MS", cursive, sans-serif	This is a heading This is a paragraph
Impact, Charcoal, sans-serif	This is a heading This is a paragraph
"Lucida Sans Unicode", "Lucida Grande", sans-serif	This is a heading This is a paragraph
Tahoma, Geneva, sans-serif	This is a heading This is a paragraph
"Trebuchet MS", Helvetica, sans-serif	This is a heading This is a paragraph
Verdana, Geneva, sans-serif	This is a heading This is a paragraph

Monospace Fonts

font-family	Example text
"Courier New", Courier, monospace	This is a heading This is a paragraph
"Lucida Console", Monaco, monospace	This is a heading This is a paragraph

Font-style

A **font-style** jellemzőt leginkább a dőlt betűhatás előidézésére használjuk. Háromféle értéke lehet:

- **normal** (a szöveg szokványosan jelenik meg)
- **italic** (dőltén jelenik meg)
- **oblique** [„ferdén” jelenik meg (az **italic**-hoz igen hasonló, de a böngésző által kevésbé elterjedten támogatott hatás)]

A következő bekezdések mindháromra adnak példát:

```
<html>
<head>
<style type="text/css">
p.normal {font-style:normal;}
p.italic {font-style:italic;}
p.oblique {font-style:oblique;}
</style>
</head>

<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>
</body>
</html>
```

Az első normálisan, a második kettő pedig dőlt alakban jelenik meg.

Betűméret

A **font-size** jellemzővel beállíthatjuk a szöveg betűméretét. Ez a webes formázásban igen fontos művelet, de nem adokumentum struktúrálására való (tehát pl. a bekezdéseinket ne próbáljuk címsornak ill. a címsorainkat bekezdéseknek feltüntetni vele). Ahhoz mindig a megfelelő (<h> és <p>) tageket használjuk!

A **font-size** értéke lehet abszolút ill. relatív.

Az **abszolút** betűmérettel egy megadott nagyságot állíthatunk be. Egyes böngészőkben azonban ez nem teszi lehetővé az olvashatóság-segítő betű-átméretezést; azaz felhasználói szempontból gondot okozhat. Akkor előnyös alkalmazni, hogyha a felhasználói képernyő fizikai méreteit ismerjük.

A **relatív** betűméret a többi szöveges elemhez viszonyítva értendő. Így a dokumentum betűméretekkel jelzett hierarchiája minden megjelenítés mellett megmarad.

Hogyha a betűméretet külön nem adjuk meg, úgy annak autmatikus értéke 16px = 1em.

A betűméret megadása pixelekből

A betűméret pixelekből való (abszolút) megadása a szövegméret teljeskörű „kézivezérlését” jelenti, pl.:

```
<html>
<head>
<style>
```

```

h1 {font-size:40px;}
h2 {font-size:30px;}
p {font-size:14px;}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in px allows Firefox, Chrome, and Safari to
resize the text, but not Internet Explorer.</p>
</body>
</html>

```

Példánkban a bekezdések szövege a normálisnál (16 pixelnél) kisebb betűmérettel kerül megjelenítésre. Ahogy a példa szövege is tartalmazza, a Firefox, Chrome és Safari a pixelekben megadott abszolút betűméretezés esetén is lehetővé teszi a szöveg olvashatóság érdekében való felnagyítását; az Internet Explorer azonban nem támogatja ezt. Az átméretezést a böngészők „Zoom” parancsával végezhetjük; ez azonban nemcsak a szöveget, hanem az egész oldalt felnagyítja.

A betűméret megadása em-ben

Az Internet Explorerrel való átméretezhetetlenség elkerülésére sok webes fejlesztő az **em** egységet használja a pixel helyett. Ezt a módszert támogatja a W3C is.

1 em megfelel az alapértelmezett – azaz jelenleg 16 pixeles – betűtípusnak. Tehát az em alapértelmezett értéke jelenleg 16px.

A betűméret em értéke tehát egyszerűen kiszámítható a pixelekben megadott méretből: $\text{pixel}/16 = \text{em}$.

Következő példánk megegyezik az előzővel, csak hogy itt a speciális (abszolút) betűméreteket em-ben adjuk meg, ahogy az a kódba írt megjegyzésekből is kiolvasható:

```

<html>
<head>
<style>
h1 {font-size:2.5em;} /* 40px/16=2.5em */
h2 {font-size:1.875em;} /* 30px/16=1.875em */
p {font-size:0.875em;} /* 14px/16=0.875em */
</style>
</head>
<body>

```

```

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in em allows Internet Explorer, Firefox,
Chrome, and Safari to resize the text.
Unfortunately, there is still a problem with IE. When resizing the
text, it becomes larger/smaller than it should.</p>
</body>
</html>

```

Ahogy említettük, az em betűméret-egységgel formázott szöveget minden böngésző át tudja méretezni, az Internet Explorer is. Utóbbinál azonban fennmarad egy probléma, ti. az átméretezés ugyan működik, de nem méretarányos (azaz nem felel meg a nagyítási %-nak, hanem annál mindig nagyobb ill. kisebb, nagyítás és kicsinyítés esetén).

Használjuk a százalék és az em keverékét!

Minden böngészőben működő megoldás a fenti problémára, hogyha az egész szövegtörzsnek egy, %-os betűméretet adunk:

```
<html>
<head>
<style>
body {font-size:100%;}
h1 {font-size:2.5em;}
h2 {font-size:1.875em;}
p {font-size:0.875em;}
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in percent and em displays the same size in
all browsers, and allows all browsers to resize the text!</p>
</body>
</html>
```

A százalékos értékkel kijelöltük a szövegtörzs abszolút betűméretét, a böngésző-alapértelmezés valahány %-ában. Az em értékekkel ehhez az abszolútértékhez viszonyítva adhatunk meg kisebb vagy nagyobb méreteket. Mivel minden betűméret alapja ugyanaz az egység, a betűk egymáshoz képest minden böngészőben és minden nagyításban ugyanakkorák lesznek.

További példák

Első példánkban a kövér betűhatásra találunk példákat:

```
<html>
<head>
<style type="text/css">
p.normal {font-weight:normal;}
p.light {font-weight:lighter;}
p.thick {font-weight:bold;}
p.thicker {font-weight:900;}
</style>
</head>

<body>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>
</html>
```

Az elő két bekezdés között nincs különbség; a második kettő pedig egyaránt félkövérnek látszik. Hogyha a **font-weight** (100-tól 900-ig százasaival terjedhető értékét 500-nál nagyobbobnak vesszük, akkor a böngésző a betűket félkövérként jeleníti meg (egybként pedig normálként), legalábbis a Firefoxban. Az Internet Explorerben a 600-as betű-súly átmenet a normál és a félkövér között:

```
<html>
<head>
<style>
p {font-size:300%;}
</style>
</head>
```

```
<body>
<p style="font-weight:100">This is a paragraph.</p>
<p style="font-weight:200">This is a paragraph.</p>
<p style="font-weight:300">This is a paragraph.</p>
<p style="font-weight:400">This is a paragraph.</p>
<p style="font-weight:500">This is a paragraph.</p>
<p style="font-weight:600">This is a paragraph.</p>
<p style="font-weight:700">This is a paragraph.</p>
<p style="font-weight:800">This is a paragraph.</p>
<p style="font-weight:900">This is a paragraph.</p>
</body>
</html>
```

A 900 feletti betű-súlyt a böngésző normálnak tekinti.

Következő példánkban két újabb (a **text-transform** jellemző értékeihez hasonló) betű-variáns beállításával találkozhatunk, a **font-variant** jellemző révén:

```
<html>
<head>
<style type="text/css">
p.normal {font-variant:normal;}
p.small {font-variant:small-caps;}
</style>
</head>
```

```
<body>
<p class="normal">My name is Hege Refsnes.</p>
<p class="small">My name is Hege Refsnes.</p>
</body>
</html>
```

Az első bekezdés szedése normális, a másodiké kiskapitális.

A következő példában, a jellemzők gyorsírása révén egyetlen hosszú meghatározással élünk a szövegformázás tekintetében:

```
<html>
<head>
<style type="text/css">
p.ex1 {font:15px arial,sans-serif;}
p.ex2 {font:italic bold 12px/30px Georgia,serif;}
</style>
</head>
```

```
<body>
<p class="ex1">This is a paragraph. This is a paragraph. This is a
paragraph. This is a paragraph. This is a paragraph. This is a
paragraph. This is a paragraph. This is a paragraph.</p>
<p class="ex2">This is a paragraph. This is a paragraph. This is a
paragraph. This is a paragraph. This is a paragraph. This is a
paragraph. This is a paragraph. This is a paragraph.</p>
</body>
</html>
```

Az **ex1** osztályba tartozó bekezdés 15 pixeles betűmérettel, Arial betűtípussal fog megjelenni, míg a **<p class="ex2">** dőlt, félkövér, 12 pixeles betűméretű, 30 pixeles sorközü Georgia betűtípussal.

A gyorsírást **font** jellemző tehát az összes betű-jellemzőt egy meghatározásba sűríti. A beállítható jellemzők sorrendben: **font-style**, **font-variant**, **font-weight**, **font-size/line-height**, **font-family**. A **font-**

size és **font-family** jellemzők kifejtése kötelező, különben a szöveg alapértelmezésben jelenik meg. A **font-family** jellemző értékeit vesszővel elválasztva, szóközök nélkül soroljuk fel; a több szóból álló neveket idézőjelbe tesszük. A sorrend itt is fontos!

CSS betű-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
<u>font</u>	Az összes betű-jellemző egyszeri beállítására szolgál.	<i>font-style</i> <i>font-variant</i> <i>font-weight</i> <i>font-size/line-height</i> <i>font-family</i> caption icon menu message-box small-caption status-bar inherit	1
<u>font-family</u>	Specifies the font family for text	<i>family-name</i> <i>generic-family</i> inherit	1
<u>font-size</u>	Specifies the font size of text	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> % inherit	1
<u>font-style</u>	Specifies the font style for text	normal italic oblique inherit	1
<u>font-variant</u>	Specifies whether or not a text should be displayed in a small-caps font	normal small-caps inherit	1
<u>font-weight</u>	Specifies the weight of a font	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit	1

CSS – betűk:

Betűcsaládok:

Serif: talpakkal rendelkező betűk (pl.: Times New Roman).

Sans-serif: talpakkal nem rendelkező betűk (pl.: Arial).

Monospace: egyenlő szélességű betűk (pl.: Courier).

Betűtípus/betűcsalád beállítása:

Általános alak: {font-family:"speciális betűtípus","általános betűtípus",betűcsalád;}

A több tagból álló betűtípus- ill. -család-neveket idézőjelbe írjuk.

Konkrét példa: {font-family:"Times New Roman",Times,serif;}

Betűhatások (font-style):

Normális (alapértelmezett): {font-style:normal;}

Dőlt: {font-style:italic;}

Ferde (kb. a dőlttel megegyezik): {font-style:oblique;}

Kövérség (font-weight): {font-weight:normal/lighter/bold/bolder/900/inherit;}

Kiskapitális (font-variant): {font-variant:normal/small-caps/inherit;}

Betűméret (font-size):

Abszolút betűméret névvel: {font-size:small/medium/large;}

Abszolút betűméret mérőszámmal: {font-size:100px/100pt/cm;}

Relatív betűméret: {font-size:100%/10.085em;}

Univerzális, nagyítható betűméretezés;

A formázandó szöveget tartalmazó elem (<body>) betűméretét az böngésző-alapértelmezés %-ában megadjuk, majd az egyes elemeket em-értékekben, relatívan méretezzük, pl.:

body {font-size:100%;}

h1 {font-size:3em;}

h2 {font-size:2em;}

p {font-size:1.05em;}

Betű-jellemzők gyorsírása (font):

{font:betűhatás kiskapitális kövérség betűméret/sorköz betűcsalád;}

Egy konkrét példa:

{font:italic small-caps 700 5.05em/5cm "Courier New";}

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

1. font-style
2. font-variant
3. font-weight
4. font-size/line-height
5. font-family

FONTOS: gyorsírásnál a font-size és font-family jellemzők kiföltése kötelező!

IV. CSS – linkek

A linkek sokféle stílussal rendelkezhetnek.

Link-stílusok

A linkeket bármely CSS-jellemzővel formázhatjuk (pl. **color**, **font-family**, **background** stb.). Formázásuk különlegessége, hogy állapotfüggő. A linkek ugyanis négyféle állapotban lehetnek, melyhez négyféle kijelölés tartozik:

- **a:link** (normális, meg nem nyitott link)
- **a:visited** (már látogatott oldal linkje)
- **a:hover** (ha az egér mutatója a link fölé megy)
- **a:active** (azon pillanat, mikor a linkre éppen rákattintunk).

Lássunk egy gyors példát a link-formázásra!

```
<html>
<head>
<style type="text/css">
a:link {color:#FF0000;} /* unvisited link */
a:visited {color:#00FF00;} /* visited link */
a:hover {color:#0000FF;} /* mouse over link */
a:active {color:#000000;} /* selected link */
</style>
</head>

<body>
<p><b><a href="http://muszeroldal.hu" target="_blank">This is a
link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition
in order
to be effective.</p>
</body>
</html>
```

Itt a meg nem nyitott link színe piros; a már látogatotté zöld; az egérgomb közelségére kék, és aktív állapotban (míg az egérgombot lenyomva tartjuk) fekete.

Amint a példa-szövegben is láttuk, itt is fontos a sorrendiség, tehát az egyes jellemzők csakis a fenti sorrendben követhetik egymást; akkor is, ha egyiket-másikat kihagyjuk.

További link-stíluselemek

A fenti példában a link állapot szerinti színeváltozását figyeltük meg. Következzenek most a további link-stíluselemek:

Text-decoration

A **text-decoration** jellemzőt leginkább a linkek alapértelmezés szerinti aláhúzásának eltávolítására alkalmazzuk:

```
<html>
<head>
<style type="text/css">
a:link {text-decoration:none;} /* unvisited link */
a:visited {text-decoration:none;} /* visited link */
a:hover {text-decoration:underline;} /* mouse over link */
a:active {text-decoration:underline;} /* selected link */
</style>
</head>
```

```

<body>
<p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition
in order
to be effective.</p>
</body>
</html>

```

Példánkban a link-szöveg passzív állapotban (azaz **a:link** és **a:visited** kijelölés mellett) nincs aláhúzva, aktív állapotban (azaz **a:hover** és **a:active** kijelölés mellett) azonban igen.

Background-color

A **background-color** jellemzőt a linkek háttérszín-beállítására alkalmazzuk:

```

<html>
<head>
<style type="text/css">
a:link {background-color:#B2FF99;} /* unvisited link */
a:visited {background-color:#FFFF85;} /* visited link */
a:hover {background-color:#FF704D;} /* mouse over link */
a:active {background-color:#FF704D;} /* selected link */
</style>
</head>

<body>
<p><b><a href="http://freemail.hu" target="_blank">This is a
link</a></b></p>
<p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS
definition in order to be effective.</p>
<p><b>Note:</b> a:active MUST come after a:hover in the CSS definition
in order
to be effective.</p>
</body>
</html>

```

Példánkban a még nem látogatott (**a:link**) link betűszíne kék, háttere zöld.

A látogatott (**a:visited**) betűje lila, háttere sárga.

Az egér fölévitelkor (**a:hover**) a link színe változatlan (kék vagy lila), a háttér piros.

Kattintáskor (**a:active**) a link betűszíne változatlan, a háttér piros. (Mint látjuk, a betűszín a példában alapértelmezett.)

A jellemzők megadási sorrendje itt is fontos.

További példák

Első példánkban további link-stíluselemeket próbálunk ki:

```

<html>
<head>
<style type="text/css">
a.one:link {color:#ff0000;}
a.one:visited {color:#0000ff;}
a.one:hover {color:#ffcc00;}

a.two:link {color:#ff0000;}
a.two:visited {color:#0000ff;}
a.two:hover {font-size:150%;}

```

```

a.three:link {color:#ff0000;}
a.three:visited {color:#0000ff;}
a.three:hover {background:#66ff66;}

a.four:link {color:#ff0000;}
a.four:visited {color:#0000ff;}
a.four:hover {font-family:monospace;}

a.five:link {color:#ff0000;text-decoration:none;}
a.five:visited {color:#0000ff;text-decoration:none;}
a.five:hover {text-decoration:underline;}
</style>
</head>

```

```

<body>
<p>Mouse over the links to see them change layout.</p>

```

```

<p><b><a class="one" href="default.asp" target="_blank">This link
changes color</a></b></p>
<p><b><a class="two" href="default.asp" target="_blank">This link
changes font-size</a></b></p>
<p><b><a class="three" href="default.asp" target="_blank">This link
changes background-color</a></b></p>
<p><b><a class="four" href="default.asp" target="_blank">This link
changes font-family</a></b></p>
<p><b><a class="five" href="default.asp" target="_blank">This link
changes text-decoration</a></b></p>
</body>
</html>

```

Mint látjuk, a linkek (**a:link** kijelöléshez tartozó) alapszíne piros, látogatott (azaz **a:visited** kijelöléshez tartozó) színük pedig kék. Hogyha az első link fölé visszük az egeret, annak betűszíne sárgára vált. A második link betűmérete az egerre másfélszeresre nő; a harmadiknak zöld háttere lesz, a negyediknek a betűtípusa vélt át (Monospace-re), az ötödik alatt pedig megjelenik az amúgy kikapcsolt **text-decoration**, azaz aláhúzás.

A következő példában egy sokkal összetettebb link-formázást láthatunk:

```

<html>
<head>
<style type="text/css">
a:link,a:visited
{
display:block;
font-weight:bold;
color:#FFFFFF;
background-color:#98bf21;
width:120px;
text-align:center;
padding:4px;
text-decoration:none;
}
a:hover,a:active
{
background-color:#7A991A;
}
</style>

```

```
</head>
```

```
<body>
```

```
<a href="default.asp" target="_blank">This is a link</a>
```

```
</body>
```

```
</html>
```

A félkövén szedett, fehér link-szöveg ekkor egy világoszöld téglalap közepén jelenik meg, aláhúzás nélkül. Az egérrel érintett négyzet színe sötétebbre vált.

Mint látjuk, az egyszerre meghatározható kijelöléseket vesszővel elválasztva írjuk a meghatározásokat tartalmazó kapcsos zárójel elé. A zárójelen belül az elválasztás továbbra is pontosvesszőkkel történik.

CSS – linkek:

Link-típusok elnevezése:

Normális, nem látogatott: link

Már meglátogatott: visited

Egérrel érintett: hover

Működő (éppen kattintott): active

FONTOS: Az egyes kijelölések mindig ebben a sorrendben követik egymást!

Link-kijelölések:

A dokumentum összes linkje: *a:link/visited/hover/active*

Egy csoportba tartozó linkek: *a.csoportnév:link/visited/hover/active*

Egy id alá tartozó link(ek): *a#név:link/visited/hover/active*

Többszörös kijelölésnél az egyes elnevezéseket vessző választja el (pl.: *a:link, a:visited*).

Link-színek:

a:link {color:#FF0000;}

a:visited {color:#00FF00;}

a:hover {color:#0000FF;}

a:active {color:#000000;}

Link-szövegátalakítások (text-decoration; lásd feljebb a CSS-szöveg részt!):

Nincs: *{text-decoration:none;}*

Aláhúzott (alapértelmezett): *{text-decoration:underline;}*

Link-betűátalakítások (font; lásd feljebb a CSS-betűk részt!):

Betűméret: *{font-size:100px/100em;}*

Betűcsalád: *{font-family:serif/sans-serif/monospace;}*

Kövérség: *{font-weight:normal/lighter/bold/bolder/900/inherit;}*

A betűátalakításokat gyorsírással, a *font* jellezővel is meghatározhatjuk!

Link-szövegigazítás (text-align, lásd feljebb a CSS-szöveg részt!):

{text-align:left/center/right/justify;}

Link-háttérszín (background-color; lásd feljebb a CSS-hátterek részt!):

{background-color:green/rgb(255,255,255)/#ffffff;}

A háttérrel gyorsírással, a *background* jellemzővel is meghatározhatjuk!

Link-alak:

Téglalap: *{display:block;}*

Link-méret:

Szélesség: *{width:100px/100cm/100%;}*

Térköz: *{padding:100px/100cm/100%;}*

V. CSS – listák

A CSS lista-jellemzőkkel különféle jeleket illetve egyedi képeket állíthatunk be a rendezett vagy rendezetlen listák elemeinek jelölésére.

Listák – ismétlés

Kétféle HTML-lista van:

- rendezetlen lista (=unordered list); melynek elemeit kis ábráskákkal jelzik, és
- rendezett lista (=ordered list), melynek elemei ABC vagy számsorrendben állnak.

A CSS-ben az egyszerű fekete-fehér ábráskák mellett további formázásra és egyedi jel-képek beállítására is lehetőség van.

Listaelem-jelek

A listaelem-jel típusát a **list-style-type** jellemzővel állíthatjuk be. A jellemző bizonyos értékei a rendezett, mások pedig a rendezetlen listákra vonatkoznak:

```
<html>
<head>
<style type="text/css">
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
ol.c {list-style-type:upper-roman;}
ol.d {list-style-type:lower-alpha;}
</style>
</head>
```

```
<body>
<p>Example of unordered lists:</p>
```

```
<ul class="a">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
<ul class="b">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ul>
```

```
<p>Example of ordered lists:</p>
```

```
<ol class="c">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

```
<ol class="d">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Coca Cola</li>
</ol>
```

```
</body>
```

</html>

A példa első (rendezetlen) listájának elemei előtt fekete körvonalak, a második előtt fekete négyzetek, a harmadik (rendezett) lista előtt pedig nagy római számok, végül a negyedik előtt kis betűk vannak.

List-style-type – értékek rendezetlen listákhoz

Érték	Leírás
none	Nincs jelölés.
disc	Korong (alapértelmezett).
circle	Körvonal.
square	Négyzetlap.

List-style-type – értékek rendezett listákhoz

Érték	Leírás
armenian	Tradicionális örmény számokkal jelölt lista (az Internet Explorer nem támogatja).
decimal	Arab számos lista (1, 2, 3, stb.).
decimal-leading-zero	Arab számos lista, kezdő nullákkal (01, 02, 03, stb.).
georgian	Tradicionális grúz számokkal jelölt lista (az Internet Explorer nem támogatja).
lower-alpha	Kis alfabetikus jelölés (a, b, c, d, e, stb.).
lower-greek	Kis görög betűs lista.
lower-latin	Kis latinbetűs jelölés (a, b, c, d, e, stb.), gyakorlatilag megegyezik a kis alfabetikussal.
lower-roman	Kis római számozás (i, ii, iii, iv, v, stb.).
upper-alpha	Nagy alfabetikus jelölés (A, B, C, D, E, etc.).
upper-latin	Nagy latinbetűs jelölés (A, B, C, D, E, etc.), gyakorlatilag megegyezik a nagy alfabetikussal.
upper-roman	Nagy római számozás (I, II, III, IV, V, etc.).

Az Internet Explorer egyetlen verziója (beleértve az IE8-at is) sem támogatja a következő értékeket (hacsak nincs megfelelő DOCTYPE beállítva): **decimal-leading-zero**, **lower-greek**, **lower-latin**, **upper-latin**, **armenian**, **georgian**.

Kép, mint listaelem-jel

Képek listajellé tételére a **list-style-image** jellemzőt használhatjuk:

```
<html>
<head>
<style type="text/css">
ul {list-style-image:url('sqpurple.gif');}
</style>
</head>

<body>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

Példánkban a lista-tételek előtt kis, térhatású szegéllyel rendelkező, lila négyzetlaposkák állnak.

A jelöléseket a különféle böngészők eltérően jelenítik meg; az Internet Explorer és Opera egy kicsit magasabbra teszi (a betűk tetejéhez igazítja) őket, mint a Firefox, Chrome vagy Safari (melyek a betűk aljához igazítják őket).

Arra az esetre, hogyha a jelöléseket minden képpen azonos helyen kell megjeleníteni, egy böngészők közt egyaránt alkalmazható megoldást mutatunk be (=crossbrowser solution).

Univerzális megjelenésű lista-képjelek

A következő példa lista-jelei minden böngészőben azonosan jelennek meg:

```
<html>
<head>
<style type="text/css">
ul
{
list-style-type:none;
padding:0px;
margin:0px;
}
li
{
background-image:url('sqpurple.gif');
background-repeat:no-repeat;
background-position:0px 5px;
padding-left:14px;
}
</style>
</head>

<body>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

Mint látjuk, ez esetben valójában kikapcsoljuk a nem-hierarchikus lista-jelölést, és helyette háttérként állítunk be egy-egy képet az elemekhez, olyan elhelyezésben, mintha lista-jel volna.

Az **ul** kijelöléshez tartozó meghatározások értelmében tehát eltávolítjuk a listajeleket (**list-style-type:none**) és a térkitöltési jellemzőket nullának vesszük (**padding:0px;margin:0px;**), hogy a listát minden böngésző ugyanúgy helyezze el (a többi elemhez képest).

Ezután a **li** kijelölésben hivatkozunk az alkalmazandó képre (**background-image:url('sqpurple.gif')**), a kitöltést egyszeresnek állítjuk be (**background-repeat:no-repeat**), majd elhelyezzük a lista-elemhez rendelt térrész bal felső sarkához képest (**background-position: 0px 5px**), végül pedig a szöveg helyét is meghatározzuk e térrészen belül, a **padding-left:14px**.

Lista-formázás – gyorsírással!

A listák is formázhatók gyorsírással, a **list-style** jellemző használatával:

```
<html>
<head>
<style type="text/css">
ul
{
```

```
list-style:square url("sqpurple.gif");
}
</style>
</head>
```

```
<body>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

Az értékek sorrendje a gyorsírásakor:

1. **list-style-type**
2. **list-style-position** (lásd az alábbi magyarázatot ill. táblázatot)
3. **list-style-image**.

Az előbbi példát átalakíthatjuk hierarchikussá is:

```
<html>
<head>
<style type="text/css">
#ol1 {list-style:decimal inside;}
#ol2 {list-style:decimal outside;}
#ol3 {list-style:decimal inherit;}
</style>
</head>
```

```
<body>
<ol>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol id="ol1">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol id="ol2">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
```

```
</ol id="ol3">
<ol id="ol1">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
</body>
</html>
```

Itt az első lista alapértelmezésben, azaz **outside** alakban jelenik meg; tehát a lap (bal „külső”) szélén. A második bekezdés az alapértelmezéstől eltérő, laphoz közelebbi helyen látszik (**inside** értékkel). A harmadik lista képe megegyezik az alapértelmezéssel, mivel a meghatározásában **outside** pozíció szerepel. A negyedik bekezdés megörökli a második **inside** értékét (az **inherit** érték révén); egyébként ha az **#ol1** formázást elhagyjuk, vagy pozíció-értékét **outside**-nak vesszük, az összes lista egyaránt az oldal (bal) szélén lesz.

Egy további példa

Példánkban megtaláljuk szinte az összes lista-jelölést. Természetesen ezek csak megfelelő böngészővel vagy **DOCTYPE**-beállítással jelennek meg:

```
<html>
<head>
<style type="text/css">
ul.a {list-style-type:disc;}
ul.b {list-style-type:circle;}
ul.c {list-style-type:square;}
ul.d {list-style-type:none;}
ol.e {list-style-type:decimal;}
ol.f {list-style-type:decimal-leading-zero;}
ol.g {list-style-type:lower-roman;}
ol.h {list-style-type:upper-roman;}
ol.i {list-style-type:lower-alpha;}
ol.j {list-style-type:upper-alpha;}
ol.k {list-style-type:lower-greek;}
ol.l {list-style-type:lower-latin;}
ol.m {list-style-type:upper-latin;}
ol.n {list-style-type:armenian;}
ol.o {list-style-type:georgian;}
</style>
</head>

<body>
<ul class="a">
<li>Disc type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>

<ul class="b">
<li>Circle type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>

<ul class="c">
<li>Square type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>

<ul class="d">
<li>The "none" type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
```

```
<ol class="e">
<li>Decimal type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="f">
<li>Decimal-leading-zero type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="g">
<li>Lower-roman type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="h">
<li>Upper-roman type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="i">
<li>Lower-alpha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="j">
<li>Upper-alpha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="k">
<li>Lower-greek type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="l">
<li>Lower-latin type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="m">
<li>Upper-latin type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="n">
<li>Armenian type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="o">
<li>Georgian type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

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

CSS lista-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
list-style	A lista egyszeri megformázására szolgál.	<i>list-style-type</i> <i>list-style-position</i> <i>list-style-image</i> inherit	1
list-style-image	Kép listaelem-jelként való beállítására való.	URL none inherit	1
list-style-position	Meghatározza, hogy a listaelem-jelek a szövegtörzs vonalán belül vagy kívül helyezkedjenek-e el.	inside outside inherit	1
list-style-type	A listajel típusát adja meg.	none disc circle square decimal decimal-leading-zero armenian georgian lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman inherit	1

CSS – listák:

A formázandó elemek kijelölése ugyanolyan, mint a korábbiakban (pl.: *ul* vagy *ul.név* vagy *ul#név*).

Rendezetlen-lista jeltípusok:

Nincs jel: {list-style-type:none;}
Korong: {list-style-type:disc;}
Körvonal: {list-style-type:circle;}
Négyzet: {list-style-type:square;}

Rendezett-lista jeltípusok:

Arab szám (1, 2, 3): {list-style-type:decimal;}
Arab szám kezdő nullával (01, 02, 03): {list-style-type:decimal-leading-zero;}
Nagy rómaisám (I, II, III): {list-style-type:upper-roman;}
Kis rómaisám (i, ii, iii): {list-style-type:lower-roman;}

Nagy ABC (A, B, C): {list-style-type:upper-alpha;}

Kis ABC (a, b, c): {list-style-type:lower-alpha;}

Kis görög betűk (α , β , γ): {list-style-type:lower-greek;}

Egyes rendezett-lista jeltípusok használatához a DOCTYPE beállítandó!

Rendezetlen lista jelölése egyedi képpel:

```
{list-style-image:url('URL');}
```

Rendezetlen lista jelölése háttérképpel:

```
ul {list-style-type:none;padding:0px;margin:0px;}
```

```
li {background-image:url('URL');background-repeat:no-repeat;
```

```
background-position:0px 5px;padding-left:14px;}
```

Listaelem-jel – pozíció:

A szövegtörzs vonalán belül: {list-style-position:inside;}

A szövegtörzs vonalán kívül: {list-style-position:outside;}

Lista-jellemzők gyorsírása (list):

```
{list:lista-típus listaelem-jel-pozíció listajelölő-kép-URL;
```

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

1. **list-style-type**
2. **list-style-positon**
3. **list-style-image.**

VI. CSS – táblázatok

A HTML-táblázatok kinézetét a CSS nagyban javíthatja:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK
Paris spécialités	Marie Bertrand	France
The Big Cheese	Liz Nixon	USA
Vaffeljernet	Palle Ibsen	Denmark

Táblázat-szegélyek

A táblázat-szegélyek beállítására a **border** jellemző szolgál a CSS-ben.

Az alábbiakban fekete keretet határozunk meg a táblázat egésze, illetve a **<th>** és **<td>** elemek számára:

```
<html>
<head>
<style type="text/css">
table,th,td
{
border:1px solid black;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
</tr>
</table>
</body>
</html>
```

Mint látjuk, a példában többszörös (vesszővel elválasztott elemekből álló) kijelölés, és gyorsírással jellemzés található. A megadott táblázat így néz ki:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Lois	Griffin
------	---------

Mint látjuk, kettős kerettel rendelkezik; ennek oka, hogy mind a táblázat egészének, mind a benne lévő elemeknek van külön kerete. A táblázat egyszeres keretezéséhez a **border-collapse** jellemzőt használhatjuk.

A border-collapse jellemző

A border-collapse jellemzővel beállíthatjuk, hogy a táblázat-elemek szegélyei egyetlen, közös keretté olvadjanak-e össze, vagy különállóan megmaradjanak:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
table {border-collapse:collapse;}
table, td, th {border:1px solid black;}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
</tr>
</table>
<p><b>Note:</b> If a !DOCTYPE is not specified, the border-collapse
property can produce unexpected results.</p>
</body>
</html>

```

Amint látjuk, a **DOCTYPE** a váratlan hatások elkerülése érdekében meghatározandó; bár az adott (egyszerű) példára a hiánya nincs hatással. Hogyha a **border-collapse** jellemzőhöz tartozó kijelölésbe **table** helyett **th**, **td**-t írunk, nem történik hatérvonal-egyesülés.

Táblázat-szélesség és -magasság

A táblázat szélességét és magasságát a **width** és **height** jellemzőkkel állíthatjuk be.

A következő példában a táblázat szélessége 100%, a **<th>** elemek magassága pedig 50 pixel:

```

<html>
<head>
<style type="text/css">
table,td,th {border:1px solid black;}
table {width:100%;}
th {height:50px;}
</style>
</head>

```

```

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>

```

Itt az egyes táblázat-elemek, és annak egésze is, külön-külön (1 pixeles, fekete) határvonallal rendelkezik; a táblázat a rendelkezésre álló oldal- ill. keret-szélességet 100%-ban kitölti, a táblázat-címek magassága pedig 50px:

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

Hogyha a belső stílus-oldalba elvégezzük az alábbi módosításokat:

```

<head>
<style type="text/css">
table,td,th {border:1px solid black;}
table {width:80%;border-collapse:collapse;height:800px}
th {height:50px;}
</style>
</head>

```

akkor a táblázat minden része csak egyszeres szegéllyel fog rendelkezni; szélessége csak 80%-a lesz a rendelkezésre állónak s bár a **<th>** elemek magassága továbbra is 50px, ez eltörpül a táblázat 800px-es magasságához képest, amiből a cellákra egyenként $(800-50)/4 = 187.5\text{px}$ esik (azaz méretük a címsornak több mint háromszorosa).

Táblázat – szövegigazítás

A táblázatba írt szöveget a **text-align** és **vertical-align** jellemzőkkel pozícionálhatjuk.

A **text-align** jellemző a vízszintes szövegigazítást adja meg (pl. **left**, **right**, **center**):

```
<html>
<head>
<style type="text/css">
table {width:90%;}
table,td,th {border:1px solid black;}
td {text-align:right;}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

Itt a meglehetősen széles táblázatban jól látszik, hogy az alapértelmezetten középre igazított címsor alatt a táblázat-értékek (az alapértelmezett ballal szemben) jobbra zártak. Természetesen a táblázat-címsor számára is bármiféle szövegigazítást előírhatunk, a **th** kijelölés segítségével.

A **vertical-align** jellemzővel a függőleges szövegigazítást (pl.: **top**, **middle**, **bottom**) adhatjuk meg:

```
<html>
<head>
<style type="text/css">
table {width:80%;}
table, td, th {border:1px solid black;}
td {height:50px;vertical-align:bottom;}
</style>
```

```

</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>

```

E példánkban a vízszintes szövegigazítás alapértelmezés szerinti, azaz a címsorok közepén, a közöséges cella-adatok (<td>) pedig baloldalt vannak. A beállítás lényege, hogy az 50px magas <td> elemkbe írt szövegek a cellák bal-**alsó (bottom)** sarkában látszanak.

Táblázat – térköz

A táblázat szegélye és tartalma közti térköz beállítását a **padding** jellemzővel végezhetjük (pl. **td** és **th** kijelölés mellett):

```

<html>
<head>
<style type="text/css">
table {width:80%;}
table, td, th {border:1px solid black;}
td {padding:15px;}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>

```

```

<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>

```

Így az alapértelmezetten balra igazított cella-szövegek körül minden irányban legalább 15px üres hely van (jobbról nyilván több).

Táblázat – színek

A következő példában a **<th>** elemek szegély-, szöveg- és háttér-színét állítjuk be (valamint a többi elem szegély-színét is, egy kijelölésben):

```

<html>
<head>
<style type="text/css">
table, td, th {border:1px solid green;}
th {background-color:green;color:white;}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>

```

```

<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>

```

Így az egész táblázat zöld szegéllyel jelenik meg; a címsornak pedig emellett zöld háttérszíne és fehér betűszíne van. A színt a **border** jellemzőhöz a **solid** előtag segítségével adhatjuk meg, pl.:

```

{border: 2px solid yellow;}
{border: 2px solid #15ffe7;}
{border: 2px solid rgb(254,255,21);}

```

További példák:

Első példánkban az ezen fejezet elején közölt táblázat HTML-kódját láthatjuk:

```

<html>
<head>
<style type="text/css">
#customers
{
font-family:"Trebuchet MS", Arial, Helvetica, sans-serif;
width:100%;
border-collapse:collapse;
}
#customers td, #customers th
{
font-size:1em;
border:1px solid #98bf21;
padding:3px 7px 2px 7px;
}
#customers th
{
font-size:1.1em;
text-align:left;
padding-top:5px;
padding-bottom:4px;
background-color:#A7C942;
color:#ffffff;
}
#customers tr.alt td
{
color:#000000;
background-color:#EAF2D3;
}
</style>
</head>

<body>
<table id="customers">
<tr>

```

<th>Company</th>	<th>Contact</th>	<th>Country</th>
<td>Alfreds Futterkiste</td>	<td>Maria Anders</td>	<td>Germany</td>
<td>Berglunds snabbköp</td>	<td>Christina Berglund</td>	<td>Sweden</td>
<td>Centro comercial Moctezuma</td>	<td>Francisco Chang</td>	<td>Mexico</td>
<td>Ernst Handel</td>	<td>Roland Mendel</td>	<td>Austria</td>
<td>Island Trading</td>	<td>Helen Bennett</td>	<td>UK</td>
<td>Königlich Essen</td>	<td>Philip Cramer</td>	<td>Germany</td>
<td>Laughing Bacchus Winecellars</td>	<td>Yoshi Tannamuri</td>	<td>Canada</td>
<td>Magazzini Alimentari Riuniti</td>	<td>Giovanni Rovelli</td>	<td>Italy</td>
<td>North/South</td>	<td>Simon Crowther</td>	<td>UK</td>
<td>Paris spécialités</td>	<td>Marie Bertrand</td>	<td>France</td>


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

Az első (**#customers**) kijelölés az egész (**id="customers"** megjelölésű) táblázatra vonatkozik; a meghatározások szerint az Trebuchet MS betűtípussal, az oldalszélesség 100%-án, egyszeres kerettel kerül megjelenítésre.

A második kijelölés (**#customers td**, **#customers tr**) a táblázat összes **<th>** és **<tr>** elemére vonatkozik; 1em betűméretről, 1 pixeles, sötétzöld keretről és a négy térköz-értékről szól; a térköz rendre: felül 3px, jobboldalt 7px, alul 2px, balról 7px.

A harmadik kijelölés (**#customers th**) szerint, a második kijelölésben meghatározottaktól eltérően, a betűméret 1.1em, a felső térköz 5, az alsó pedig 4px; és emellett (az alapértelmezett középtől eltérő) baloldali szövegigazítás, világoszöld háttérszín és fehér betűszín kerül meghatározásra.

A negyedik, és egyben utolsó (**#customers tr.alt td**) kijelölésben a **customers id** alá tartozó **<tr>** elemek közül a **class="alt"**-ba tartozókat formázzuk; e soroknak világoszöld háttérszínt adunk. Mint látjuk, itt egyúttal a fekete betűszín is beállításra kerül, holott ez elvileg alapértelmezett. A korábban tanultak szerint azonban a háttérszín szabályszerű meghatározása magában foglalja a betűszín megadását is; nehogy a szöveg egy esetleges kedvezőtlen alapértelmezett betűszín miatt olvashatatlan legyen.

Így tehát biztosak lehetünk benne, hogy az **alt** csoportba tartozó elemekmindig fekete betűszínnel és halványzöld háttérrel jelennek majd meg; hasonlóképpen a **<th>** elemek is mindig azonos módon; míg a közönséges **<tr>** elemek az alapértelmezéstől függő betűszínnel és háttérrel jelentkeznek. Tehát elképzelhető, hogy egy böngészőben fekete háttérrel és fehér betűkkel jelenik meg; ami változtat ugyan az egyes sorok színmélységen alapuló rangsorán, de azok elkülönített volta észlelhető marad. Mindenesetre a legbiztosabb megoldás az volna, ha a egy ötödik kijelöléssel a fehéren maradó sorok színét is meghatároznánk:

```
#customers tr
{
color:#000000;
background-color:#ffffff;
}
```

Következő példánkban bemutatjuk a táblázatszócím pozícionálására szolgáló **caption-side** jellemzőt:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
caption {caption-side:bottom;}
</style>
</head>

<body>

<table border="1">
<caption>Table 1.1 Customers</caption>
<tr>
<th>Company</th>
<th>Contact</th>
<th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
```

```

</tr>
<tr>
<td>Berglunds snabbköp</td>
<td>Christina Berglund</td>
<td>Sweden</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
<tr>
<td>Ernst Handel</td>
<td>Roland Mendel</td>
<td>Austria</td>
</tr>
<tr>
<td>Island Trading</td>
<td>Helen Bennett</td>
<td>UK</td>
</tr>
<tr>
<td>Magazzini Alimentari Riuniti</td>
<td>Giovanni Rovelli</td>
<td>Italy</td>
</tr>
<tr>
<td>North/South</td>
<td>Simon Crowther</td>
<td>UK</td>
</tr>
</table>

```

<p>Note: Internet Explorer 8 (and higher) supports the caption-side property if a !DOCTYPE is specified.</p>
</body>
</html>

Ebben az esetben a következőt látjuk:

Table 1.1 Customers		
Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK

Mozilla Firefoxban a cím helyesen, azaz a táblázat alatt kerül megjelenítésre. Az Internet Explorer 8-as és afölötti verziói (sőt, a 7-es is!) ezt csak megfelelő **DOCTYPE** beállítása mellett végzi el, egyébként mindig az alapértelmezett, táblázat-teteji megjelenítést alkalmazza.

A **caption-side** jellemző lehetséges értékei:

- **top** (alapértelmezett); a cím a táblázat fölött, középre igazítva jelenik meg
- **bottom**; a cím a táblázat alatt, középre igazítva jelenik meg
- **inherit**; a böngésző az adott elem megjelenítésében a magasabb szintű elemek beállításaihoz igazodik.

Hogyha a belső stílus-oldalt kissé átíjtjuk:

```
<style type="text/css">
caption {caption-side:bottom;}
table, th, td{border: 3px solid green;}
table, th, td{border-collapse:collapse;}
</style>
```

akkor pedig ezt látjuk (az Internet Explorerben):

Table 1.1 Customers

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK

CSS-táblázatok:

Táblázat: `<table></table>`

Tábláztcím: `<caption></caption>` [A `<table>` elembe írandó!]

Tábláztcím-pozíció: `caption {caption-side:top/bottom/inherit,}`

Szegélyek/szegélyszín:

Táblázat többszörös szegéllyel: `table,th,td {border:1px solid black;}`

Táblázat egyszeres szegéllyel: `table {border-collapse:collapse;}`
`table,td,th {border:1px solid black;}`

A *border-collapse* jellemző használatához DOCTYPE kellhet!

[A szegély-típusokról bővebben lásd alább a CSS-szegélyek részt!]

Betűszín: `{color:green/rgb(255,255,255)/#ffffff;}`

Háttérszín: `{background-color:green/rgb(255,255,255)/#ffffff;}`

Betűtípus: `{font-family:"speciális betűtípus", "általános betűtípus", "betűcsalád";}`

Betűméret: `{font-size:100px/100em/100%;}`

Szélesség/magasság:

Szélesség: `{width:100px/100%;}`

Magasság: `{height:100px/100%;}`

Szövegigazítás:

Vízszintesen: `{text-align:left/center/right;}`

Függőlegesen: `{vertical-align:top/middle/bottom;}`

Térköz: `{padding:100px/5cm/100%;}`

[Részletesebben lásd alább, a CSS-térköz részben!]

CSS

III. KÖNYV: DOBOZ-MODELL

I. CSS doboz-modell

A CSS doboz-modell

A HTML-elemek dobozok gyanánt is felfoghatók. A CSS-ben a „doboz-modell” (=box model) kifejezést a tervezésnél, szerkezeti beosztásnál használjuk.

A CSS értelmezésében a doboz-modell „doboz” egy, a HTML-elemet körülvevő geometriai alakzat, mely négy részből áll: margóból (=margin), határvonalból (=border), térközből (=padding) és az elem tartalmából (=content), amint az az alábbi kép ábrázolja:



A doboz-modell tehát lehetővé teszi az elemek körülhatárolását, és egymáshoz képest való elhelyezését. Alkotórészeinek részletesebb magyarázata:

- **margin:** a határvonal körüli szabad rész méretét adja meg. Háttérszíne nincs, azaz átlátszó (=transparent).
- **border:** az elem tartalmát és az akörüli térközt körülvevő szegély. Színe a doboz háttérszínéhez igazodjon!
- **padding:** az elem tartalma és a szegély köztzi térköz. Színe megegyezik a doboz háttérszínével.
- **content:** a doboz tartalma, ahol az elembe tartozó szöveg, kép vagy egyéb adat megjelenik. Háttérszíne adja az egész dobozét is.

Hogy az egyes elemek szélességét és magasságát minden böngészőre nézve megfelelően beállíthassuk, ismernünk kell a doboz-modell használatát.

Az elemek szélessége és magassága

Lényeges tudnivaló, hogy amikor a CSS segítségével beállítjuk egy elem **width** és **height** jellemzőjét, akkor ez csak a tartalmi (content) részre vonatkozik. A teljes elem méretezéséhez meg kell adnunk a térköz, szegély és margó méretét is.

Az alábbi stílussal rendelkező elem szélessége 300 pixel lesz:

```
{  
width:250px;  
padding:10px;
```

```
border:5px solid gray;
margin:10px;
}
```

Tudniillik az elem-tartalom szélessége **250px**, melyet körülvesz $2*10 = 20\text{px}$ bal- és jobboldali térköz, $2*5 = 10\text{px}$ bal- és jobboldali, szürke határvonal és $2*10 = 20\text{px}$ bal- és jobboldali margó.

Ezek összege: $250+20+10+20 = 300\text{px}$.

Hogyha csak 250px hely áll rendelkezésünkre, akkor nyilván egy ugyanekkora szélességű elemet tudunk csak elhelyezni:

```
<html>
<head>
<style type="text/css">
div.ex
{
width:220px;
height:70px;
padding:10px;
border:5px solid gray;
margin:0px;
}
</style>
</head>
```

```
<body>
<br /><br />
```

```
<div class="ex">The line above is 250px wide.<br />
The total width of this element is also 250px.</div>
```

```
<p><b>Important:</b> This example will not display correctly in IE!<br />
```

```
However, we will solve that problem in the next example.</p>
```

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

Példánk, mint az a tartalmából is kiderül, Internet Explorerrel nem, Firefox-szal azonban megfelelően működik. Az oldal tetején egy $250*100$ pixeles fekete téglalap látszik; alatta pedig egy ugyanekkora, 5px vastag, szürke keret, melyben az **ex** csoportba sorolt dokumentumrész szövege van. A szöveg rendelkezésére álló hely tekintetében figyelembe vettük a térköz és határvonal által elfoglalt 30px helyet; így a 100px-es össz-magasság folytán erre csak 70px hely marad.

Az elemek össz-szélessége tehát így számítható ki:

Össz-szélesség = elem-szélesség + bal térköz + jobb térköz + bal szegély + jobb szegély + bal margó + jobb margó.

Ugyanígy az **össz-magasság** = elem-magasság + felső térköz + alsó térköz + felső szegély + alsó szegély + felső margó + alsó margó.

Böngésző-kompatibilitási megoldás

Az előző példa Internet Explorerrel történt kipróbálásakor az objektum/doboz szélessége nem volt pontosan 250 pixel. Az Internet Explorer ugyanis a térközt és a szegélyt beleszámítja az elem-szélesség értékebe (emiatt ott nem 220, hanem 250px szélességet kellett volna beállítani); kivéve, hogyha megfelelő **DOCTYPE**-ot állítunk be.

A **DOCTYPE** hozzáadásával tehát minden böngészőre nézve megoldódik a problémánk:

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
div.ex
{
width:220px;
height:70px;
padding:10px;
border:5px solid gray;
margin:0px;
}
</style>
</head>

<body>
<br /><br />

<div class="ex">The line above is 250px wide.<br />
Now the total width of this element is also 250px.</div>

<p><b>Note:</b> In this example we have added a DOCTYPE declaration
(above the html element), so it displays correctly in all browsers.</p>

</body>
</html>

```

Valóban, ez minden böngészőben ugyanúgy jelenik meg.

Hogyha megfigyeljük az ebben a példában szereplő belső stílus-oldalt:

```

{
width:220px;
height:70px;
padding:10px;
border:5px solid gray;
margin:0px;
}

```

akkor látjuk, hogy ez a **DOCTYPE**-meghatározással együtt mind az Internet-Explorerben, mind a Firefoxban működik; utóbbiban pedig **DOCTYPE** nélkül is azonos eredményt kapunk. Hogyha azonban az IE-ben akarunk **DOCTYPE** nélkül azonos eredményt kapni, a 220 ill. 70 pixeles értékekhez hozzá kell adnunk 30-at:

```

{
width:250px;
height:100px;
padding:10px;
border:5px solid gray;
margin:0px;
}

```

mert a **padding** és **border** mérete az IE szerint az elem-méretbe is beleszámít.

CSS doboz-modell:

A HTML-elem, mint doboz, részei:

- **content:** a doboz tartalma, ahol az elembe tartozó szöveg, kép vagy egyéb adat megjelenik. Háttérszíne adja az egész dobozét is.

- **padding:** az elem tartalma és a szegély köztüi térköz. Színe megegyezik a doboz háttérszínével.
- **border:** az elem tartalmát és az akörüli térközt körülvevő szegély. Színe a doboz háttérszínéhez igazodjon!
- **margin:** a határvonal körüli szabad rész méretét adja meg. Háttérszíne nincs, azaz átlátszó (=transparent).

Doboz-méretezés:

```
{width:100px;
height:100px;
padding:100px;
border:100px solid green;
margin:100px;}
```

A doboz össz-szélessége ill. -magassága a Firefox-hoz hasonló böngészőkben

(valamint IE-ben, megfelelő DOCTYPE mellett) =

elem-méret + bal térköz + jobb térköz + bal szegély + jobb szegély + bal margó + jobb margó.

Ehhez megfelelő DOCTYPE-declaration (melyet a dokumentum fejrésze elé írunk):

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

A doboz össz-szélessége ill. -magassága az Internet Explorerhez hasonló böngészőkben

(más DOCTYPE mellett) =

elem-méret (beleértve a térközt és szegélyt is) + bal margó + jobb margó.

II. CSS – szegélyek

CSS szegély-jellemzők

A CSS szegély-jellemzők lehetővé teszik az elemek szegély-színének és -kialakításának beállítását.

Border-syle

A **border-syle** jellemzőt a szegély típusának megadására használjuk. A további szegély-jellemzők is csak a **border-style** megadása után, azzal együtt lépnek érvénybe!

Border-style – értékek:

- **none**: nincs keret
- **dotted**: pontozott keret
- **dashed**: kötőjelekből álló keret
- **solid**: egyetlen vonalból álló keret
- **double**: Két vonalból álló keret, amelyek külső és belső éltávoldága megegyezik a keretvastagsággal.
- **groove**: „vajt” térhatású keret; kinézete függ a **border-color** jellemző értékétől is.
- **ridge**: „domború” térhatású keret; kinézete függ a **border-color** jellemző értékétől is.
- **inset**: „besüllyesztett” térhatású keret; kinézete függ a **border-color** jellemző értékétől is.
- **outset**: „kiemelt” térhatású keret; kinézete függ a **border-color** jellemző értékétől is.
- **hidden**: rejtett, nem látszó keret

A következő példában az egyes keret-típusok beállítására láthatjuk:

```
<html>
<head>
<style type="text/css">
p.none {border-style:none;}
p.dotted {border-style:dotted;}
p.dashed {border-style:dashed;}
p.solid {border-style:solid;}
p.double {border-style:double;}
p.groove {border-style:groove;}
p.ridge {border-style:ridge;}
p.inset {border-style:inset;}
p.outset {border-style:outset;}
p.hidden {border-style:hidden;}
</style>
</head>

<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>
</html>
```

A következőben pedig látjuk a fentebb, a **double** szegélytípusnál leírtak igazolását:

```
<html>
```



```

<head>
<style type="text/css">
p.solid {border:25px solid black;}
p.double {border: 25px double black;}
</style>
</head>

<body>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
</body>
</html>

```

A két keret pontosan ugyanakkora helyet foglal el. A kettős keret voltaképpen három (fekete+fehér+fekete), egyenlő vastagságú vonalból áll, melyek összvastagsága 25px.

Border-width

A **border-width** jellemző a szegély vonalvastagságának beállítására szolgál. Értékeit pixelben, vagy három alapértelmezett egységben (**thin**, **medium** és **thick**) adhatjuk meg. A **border-width** jellemző is csak a **border-style** megadása után alkalmazható, amint az az alábbi példában látszik:

```

<html>
<head>
<style type="text/css">
p.one
{
border-style:solid;
border-width:5px;
}
p.two
{
border-style:solid;
border-width:medium;
}
p.three
{
border-style:solid;
border-width:1px;
}
</style>
</head>

```

```

<body>
<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p><b>Note:</b> The "border-width" property does not work if it is used
alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>

```

A **thin** vastagság a Firefoxban 1px, az IE-ben 2px; a **medium** is rendre 3px ill. 4px, a **thick** pedig ugyanígy 5px és 6px. Tehát a megjelenítés a böngésző alapértelmezésétől függ.

Border-color

A **border-color** jellemző a szegélyszín beállítására való. Ahogy eddig, a színt itt is háromféleképpen, azaz névvel (pl.: **"black"**), RGB- (pl.: **"rgb(0,0,0)"**) ill. HEX-értékkel (pl.: **"#000000"**) adhatjuk meg.

A szegélyszín **transparent** is lehet

A **border-color** jellemző sem működik önmagában, csak a **border-style**-lal együtt.

Lássunk egy példát:

```
<html>
<head>
<style type="text/css">
p.one
{
border-style:solid;
border-color:red;
}
p.two
{
border-style:solid;
border-color:rgb(152,208,33);
}
p.three{
border-style:solid;
border-color:#98bf21;
}
p.four
{
border-style:solid;
border-color:transparent;
}
</style>
</head>

<body>
<p class="one">A solid red border</p>
<p class="two">A solid green border</p>
<p class="three">A solid green border</p>
<p class="four">A transparent border</p>
<p><b>Note:</b> The "border-color" property does not work if it is used
alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>
```

Az első bekezdés szegélye piros, a következő kettőé pedig egy megegyező zöld-árnyalat. A negyedik bekezdés körül nem látszi szegély, de hogyha az összes kijelöléshez beírjuk a

```
border-width:20px;
```

parancsot, akkor az egymás fölötti bekezdés-szövegek pontosan ugyanoda kerülnek, tehát a **transparent** szegély is jelen van, a többive megegyező vatagságban, csak épp nem látszik.

Hogyha azonban a negyedik bekezdés meghatározásait átírjuk:

```
p.four
{
border-style:hidden;border-width:20px;
border-color:black;
}
```

a keret ténylegesen eltűnik, és a szöveg, a többivel ellentétben, az oldal bal margójára kerül.

Ugyanez a helyzet a

```
p.four {border-style:none;}
beállítás mellett is.
```

Oldalspecifikus szegélyezés

A CSS segítségével a keret különböző oldalait eltérően formázhatjuk, pl.:

```
<html>
<head>
<style type="text/css">
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
</style>
</head>

<body>
<p>2 different border styles.</p>
</body>
</html>
```

Itt a bekezdés teteje és alja pontozott, oldalai pedig egyszeres szegéllyel rendelkeznek.

Ugyanezt a hatást elérhetjük egy rövidített meghatározással is:

```
<html>
<head>
<style type="text/css">
p {border-style:dotted solid;}
</style>
</head>

<body>
<p>2 different border styles.</p>
</body>
</html>
```

Tehát ennél a gyorsírásnál a vízszintes határvonalak típusát előre, a függőlegeseket pedig azután írjuk.

A **border-style** jellemző értékét ugyanis négyféleképpen adhatjuk meg:

1. **border-style:dotted;**
a keret mind a négy oldala pontozott.
2. **border-style:dotted solid;**
a keret vízszintes oldalai pontozottak, a függőlegesek szimpla-vonalasak.
3. **border-style:dotted solid double;**
a keret teteje pontozott, oldalai szimpla-, alja pedig dupla-vonalas.
4. **border-style:dotted solid double dashed;**
a keret teteje pontozott, jobboldala szimpla-, alja dupla-vonalas, baloldala pedig kötőjeles.

FONTOS: Habár a fenti felsorolásban a **border-style** jellemzőt vettük alapul, a leírtak a **border-width** és **border-color** jellemzőkre is érvényesek!

Például:

```
<html>
<head>
<style type="text/css">
p
{
```

```
border-style:dotted double dashed solid;
border-color:red green blue;
border-width:30px medium;
}
</style>
</head>
```

```
<body>
<p>
This is some text in a paragraph.<br />
This is some text in a paragraph.<br />
This is some text in a paragraph.<br />
This is some text in a paragraph.<br />
</p>
</body>
</html>
```

a fentiek szerint a bekezdés felső szegélye pontozott, piros és 30px-es; a két oldalsó zöld és közepes méretű, a jobboldali kettős, a baloldali pedig egyszeres; az alsó szegély pedig kötőjeles, kék és 30px-es.

A szegély-jellemzők gyorsírása

Amint a fenti példákban is látszik, a szegélyek beállításakor számos jellemzővel foglalkozunk. A kód lerövidítése érdekében az összes szegély-jellemzőt egyben, azaz gyorsírással is megadhatjuk.

A szegély-jellemzők összevont megnevezése **”border”**:

```
<html>
<head>
<style type="text/css">
p {border:5px solid red;}
</style>
</head>
```

```
<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```

A fenti szegély 5px vastag, egyszeres és piros.

A **border** gyorsíró-jellemző használatakor, az értékek hagyományos jellemzők szerinti sorrendje a következő:

1. **border-width**
2. **border-style**
3. **border-color**

Az utolsó előtti példánk, azaz a

```
p
{
border-style:dotted double dashed solid;
border-color:red green blue;
border-width:30px medium;
}
```

kód gyorsírással változata tehát elvileg ez volna:

```
p {border:30px medium dotted double dashed solid red green blue;}
```

csak hogy ezt sem az IE, sem a Firefox nem fogadja el; ti. a bekezdést **medium** vastagságú, **solid** és **blue** kerettel látja el, azaz a legutolsó jellemzőket veszi csak figyelembe.

További példák

Első példánkban bemutatjuk, hogyan lehet a felső határvonalra vonatkozó összes beállítást egyszerre megadni:

```
<html>
<head>
<style type="text/css">
p
{
border-style:solid;
border-top:thick double #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```

Az eredmény: a bekezdés keretének alsó és oldalsó élei egyszeres, fekete vonalak (alapértelmezés szerint), a felső szegély azonban vastag, kettős és piros.

Ez alapján már érthetjük, miért bizonyult helytelennek a fenti gyorsírásunk: az egyes oldalakra vonatkozó meghatározásokat külön-külön kellett volna csoportosítanunk; azaz a

```
p
{
border-style:dotted double dashed solid;
border-color:red green blue;
border-width:30px medium;
}
```

gyorsírásos változata helyesen ez:

```
p
{
border-top:dotted red 30px;
border-right:double green medium;
border-bottom:dashed blue 30px;
border-left:solid green medium;
}
```

csak hogy ez hosszabb az előzőnél.

A következő példánkban szereplő bekezdések felső és oldalsó határa az alapértelmezett **solid black**, az alsó típusát azonban mindegyiknél átállították:

```
<html>
<head>
<style type="text/css">
p {border-style:solid;}
p.none {border-bottom-style:none;}
p.dotted {border-bottom-style:dotted;}
p.dashed {border-bottom-style:dashed;}
p.solid {border-bottom-style:solid;}
p.double {border-bottom-style:double;}
p.groove {border-bottom-style:groove;}
p.ridge {border-bottom-style:ridge;}
p.inset {border-bottom-style:inset;}
p.outset {border-bottom-style:outset;}
</style>
```

```
</head>
```

```
<body>
```

```
<p class="none">No bottom border.</p>
```

```
<p class="dotted">A dotted bottom border.</p>
```

```
<p class="dashed">A dashed bottom border.</p>
```

```
<p class="solid">A solid bottom border.</p>
```

```
<p class="double">A double bottom border.</p>
```

```
<p class="groove">A groove bottom border.</p>
```

```
<p class="ridge">A ridge bottom border.</p>
```

```
<p class="inset">An inset bottom border.</p>
```

```
<p class="outset">An outset bottom border.</p>
```

```
</body>
```

```
</html>
```

A keret oldalai 45 fokos szögben kapcsolódnak egymáshoz, így a formázások e határvonaltól érvényesek/változnak meg.

A következő példában a keret baloldalát vastagítjuk meg:

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
p
```

```
{
```

```
border-style:solid;
```

```
border-left-width:15px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p><b>Note:</b> The "border-left-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
```

```
</body>
```

```
</html>
```

Mint látjuk, a **border-left-width** jellemző használatához a **border-style**-t is be kell állítani.

A következő példában a szegélyszín leírását tanulmányozhatjuk; itt is elengedhetetlen a **border-style** használata:

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
p.one
```

```
{
```

```
border-style:solid;
```

```
border-color:#0000ff;
```

```
}
```

```
p.one_b
```

```
{
```

```
border-style:solid;
```

```
border-color:#0000ff;
```

```
border-left-color:#ff0000;
```

```
}
```

```
p.two
```

```
{
```

```

border-style:solid;
border-color:#ff0000 #0000ff;
}
p.three
{
border-style:solid;
border-color:#ff0000 #00ff00 #0000ff;
}
p.four
{
border-style:solid;
border-color:#ff0000 #00ff00 #0000ff rgb(250,0,255);
}
</style>
</head>

```

```

<body>
<p class="one">One-colored border!</p>
<p class="one_b">Two-colored border!</p>
<p class="two">Two-colored border!</p>
<p class="three">Three-colored border!</p>
<p class="four">Four-colored border!</p>
<p><b>Note:</b> The "border-color" property does not work if it is used
alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>

```

Mint látjuk, a gyorsírásnál az oldalak sorrendje itt is felül-jobbra-alul-balra.

Az **one** bekezdés szegélye kék (és **solid**).

Az **one_b** bekezdés szegélye kék, baloldala azonban piros; leírásához az előző példában megismert, jellemző-névvvel való kijelölést alkalmaztuk.

A **two** bekezdés teteje és alja piros, oldalai kékek.

A **three** bekezdés teteje piros, oldalai zöldek, alja kék.

A **four** bekezdés teteje piros, jobboldala zöld, alja kék, baloldala lila.

Az utolsó, **"four"** osztályra vonatkozó formázást a

```

p.four
{
border-style:solid;
border-color:#ff0000 #00ff00 #0000ff rgb(250,0,255);
}

```

kód helyet másképp, azaz külön kijelölésekkel is írhatjuk:

```

p.four
{
border-style:solid;
border-top-color:#ff0000;
border-right-color:#00ff00;
border-bottom-color:#0000ff;
border-left-color:rgb(250,0,255);
}

```

Ezúttal tehát az utóbbi gyorsírás-módszer bizonyul hosszabbnak.

CSS szegély-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
----------	--------	--------------------	-----

<u>border</u>	Az egész szegély összes jellemzőjének egyszeri meghatározására való.	<i>border-width border-style border-color</i>	1
<u>border-bottom</u>	Az alsó szegély egyszeri meghatározására való.	<i>border-bottom-width border-bottom-style border-bottom-color</i>	1
<u>border-bottom-color</u>	Az alsó szegély színét állítja be.	<i>border-color</i>	2
<u>border-bottom-style</u>	Az alsó szegély típusát állítja be.	<i>border-style</i>	2
<u>border-bottom-width</u>	Az alsó szegély vastagságát állítja be.	<i>border-width</i>	1
<u>border-color</u>	Az egész szegély színének egyszeri meghatározására való.	<i>color_name hex_number rgb_number transparent inherit</i>	1
<u>border-left</u>	A bal szegély egyszeri meghatározására való.	<i>border-left-width border-left-style border-left-color</i>	1
<u>border-left-color</u>	A bal szegély színét állítja be.	<i>border-color</i>	2
<u>border-left-style</u>	A bal szegély típusát állítja be.	<i>border-style</i>	2
<u>border-left-width</u>	A bal szegély vastagságát állítja be.	<i>border-width</i>	1
<u>border-right</u>	A jobb szegély egyszeri meghatározására való.	<i>border-right-width border-right-style border-right-color</i>	1
<u>border-right-color</u>	A jobb szegély színét állítja be.	<i>border-color</i>	2
<u>border-right-style</u>	A jobb szegély típusát állítja be.	<i>border-style</i>	2
<u>border-right-width</u>	A jobb szegély vastagságát állítja be.	<i>border-width</i>	1
<u>border-style</u>	Az egész szegély típusának egyszeri meghatározására való.	<i>none hidden dotted dashed solid double groove ridge inset outset inherit</i>	1
<u>border-top</u>	A felső szegély egyszeri meghatározására való.	<i>border-top-width border-top-style border-top-color</i>	1
<u>border-top-color</u>	A felső szegély színét állítja be.	<i>border-color</i>	2
<u>border-top-style</u>	A felső szegély típusát állítja be.	<i>border-style</i>	2
<u>border-top-width</u>	A felső szegély vastagságát állítja be.	<i>border-width</i>	1
<u>border-width</u>	Az egész szegély vastagságának egyszeri meghatározására való.	<i>thin medium thick length inherit</i>	1

CSS szegélyek:

Szegélytípus (border-style):

- Nincs keret:** {border-style:none;}
- Pontozott:** {border-style:dotted;}
- Kötőjeles:** {border-style:dashed;}
- Vonalas:** {border-style:solid;}
- Kettős vonal:** {border-style:double;}
- Vájt:** {border-style:groove;}

Domború: {border-style:ridge;}
Süllyesztett: {border-style:inset;}
Kiemelt: {border-style:outset;}
Elrejtett (nem látszó) keret: {border-style:hidden;}

Oldalspecifikus szegélytípus:

Felső szegély: {border-top-style:típusnév;}
Jobb szegély: {border-right-style:típusnév;}
Alsó szegély: {border-bottom-style:típusnév;}
Bal szegély: {border-left-style:típusnév;}

Szegélytípus-gyorsírás:

Minden oldal: {border-style:típusnév;}
Vízszintes/függőleges oldalak: {border-style:típusnév típusnév;}
Teteje/olalai/alja: {border-style:típusnév típusnév típusnév;}
Teteje/jobbjja/alja/balja: {border-style:típusnév típusnév típusnév típusnév;}

Vastagság: {border-width:thin/medium/thick/100px;}
Megadásához mindig állítsuk be a **border-style-t** is!

Oldalspecifikus szegélyvastagság:

Felső szegély: {border-top-width:méret;}
Jobb szegély: {border-right-width:méret;}
Alsó szegély: {border-bottom-width:méret;}
Bal szegély: {border-left-width:méret;}

Szegélyvastagság-gyorsírás:

Minden oldal: {border-width:méret;}
Vízszintes/függőleges oldalak: {border-width:méret méret;}
Teteje/olalai/alja: {border-width:méret méret méret;}
Teteje/jobbjja/alja/balja: {border-width:méret méret méret méret;}

Szín: {border-color:black/rgb(255,255,255)/#ffffff/transparent;}
Megadásához mindig állítsuk be a **border-style-t** is!

Oldalspecifikus szegélyszín:

Felső szegély: {border-top-color:szín;}
Jobb szegély: {border-right-color:szín;}
Alsó szegély: {border-bottom-color:szín;}
Bal szegély: {border-left-color:szín;}

Szegélyszín-gyorsírás:

Minden oldal: {border-color:szín;}
Vízszintes/függőleges oldalak: {border-color:szín szín;}
Teteje/olalai/alja: {border-color:szín szín szín;}
Teteje/jobbjja/alja/balja: {border-color:szín szín szín szín;}

Szegély-jellemzők gyorsírása (border):

Minden oldal: {border:vastagság/típus/szín}
Teteje: {border-top:vastagság/típus/szín}
Jobbjja: {border-right:vastagság/típus/szín}
Alja: {border-bottom:vastagság/típus/szín}
Balja: {border-left:vastagság/típus/szín}

III. CSS – kontúrozás

A kontúrvonalak (=outlines) az elemek köré rajzolt, szegélyen kívüli vonalak, melyek a kiemelés/figyelemfelkeltést ill. díszítési célt szolgálnak.

A kontúr jellemzői, a szegélyhez hasonlóan, a **style**, **color** és **width**.

Példák

A következőkben néhány példa segítségével fogunk megismerkedni a CSS-kontúrozással.

Az első példában egy egyszerű, szegélyt körülkerítő kontúrvonal készítését láthatjuk:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
p
{
border:1px solid red;
outline:green dotted thick;
}
</style>
</head>

<body>
<p><b>Note:</b> Internet Explorer 8 (and higher) supports the outline
property
if a !DOCTYPE is specified.</p>
</body>
</html>
```

Ez a példa, habár a **DOCTYPE** megadásra került, az IE-ben nem működött, csak a Firefoxban. Utóbbi esetben az 1px-es, piros körvonal körül vastag zöld pontozás látszott.

Következő példánk megint csak a Firefoxban működött:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
p {border:1px solid red;}
p.dotted {outline-style:dotted;}
p.dashed {outline-style:dashed;}
p.solid {outline-style:solid;}
p.double {outline-style:double;}
p.groove {outline-style:groove;}
p.ridge {outline-style:ridge;}
p.inset {outline-style:inset;}
p.outset {outline-style:outset;}
</style>
</head>
<body>

<p class="dotted">A dotted outline</p>
<p class="dashed">A dashed outline</p>
<p class="solid">A solid outline</p>
<p class="double">A double outline</p>
```

```
<p class="groove">A groove outline</p>
<p class="ridge">A ridge outline</p>
<p class="inset">An inset outline</p>
<p class="outset">An outset outline</p>
<b>Note:</b> Internet Explorer 8 (and higher) supports the outline
properties if a !DOCTYPE is specified.
</body>
</html>
```

A létrehozott hatások ugyanolyanok, mint a szegélyek esetében. Voltaképpen komplett szegélyeket rajzoltatunk a szegélyek köré.

Következő példánkban, mely szintén csak a Firefoxban használható, a bekezdés körül zöld és pontozott kontúr kerül beállításra:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
p
{
border:1px solid red;
outline-style:dotted;
outline-color:#00ff00;
}
</style>
</head>

<body>
<p><b>Note:</b> Internet Explorer 8 (and higher) supports the outline
properties if a !DOCTYPE is specified.</p>
</body>
</html>
```

Utolsó példánkban a kontúr vastagság-beállítását láthatjuk (szintén csak a Firefoxban):

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<style type="text/css">
p.one
{
border:1px solid red;
outline-style:solid;
outline-width:thin;
}
p.two
{
border:1px solid red;
outline-style:dotted;
outline-width:3px;
}
</style>
</head>

<body>
```

```

<p class="one">This is some text in a paragraph.</p>
<p class="two">This is some text in a paragraph.</p>
<p><b>Note:</b> Internet Explorer 8 (and higher) supports the outline
properties if a !DOCTYPE is specified.</p>
</body>
</html>

```

Az első bekezdés vékony-piros kerete körül ugyancsak vékony, fekete kontúr látható (a Firefox alapértelmezésében, mint láttuk, a **thin** értéke 1px; ami itt megegyezik a szegélyvastagsággal). A második bekezdés körül 3px vastag, pontozott kontúr van.

CSS kontúr-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
outline	Az összes kontúr-jellemző egyszeri beállítására szolgál.	<i>outline-color</i> <i>outline-style</i> <i>outline-width</i> inherit	2
outline-color	A kontúr színének beállítására szolgál.	<i>color_name</i> <i>hex_number</i> <i>rgb_number</i> invert inherit	2
outline-style	A kontúr-típus beállítására szolgál.	none dotted dashed solid double groove ridge inset outset inherit	2
outline-width	A kontúr vastagságának beállítására szolgál.	thin medium thick <i>length</i> inherit	2

CSS-kontúrozás:

Kontúr-típus:

Nincs kontúr: {outline-style:none;}
Pontozott: {outline-style:dotted;}
Kötőjeles: {outline-style:dashed;}
Vonalas: {outline-style:solid;}
Kettős vonal: {outline-style:double;}
Vájt: {outline-style:groove;}
Domború: {outline-style:ridge;}
Süllyesztett: {outline-style:inset;}
Kiemelt: {outline-style:outset;}
Elrejtett (nem látszó) kontúr: {outline-style:hidden;}

Kontúr-szín: {outline-color:*green/rgb(255,255,255)/#ffffff*;}
 Megadásához mindig állítsuk be az **outline-style-t** is!

Kontúr-vastagság: {outline-width:*thin/medium/thick/100px*;}
 Megadásához mindig állítsuk be az **outline-style-t** is!

Kontúr-jellemzők gyorsírása(outline):

`{outline:szín típus méret;`

A CSS kontúr-jellemzők csak DOCTYPE megadása mellett működnek!

Példa egy DOCTYPE-ra:

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

IV. CSS – margók

A CSS margó-jellemző a HTML-elemek közti térelosztás beállítására való.

Margók

A margó az elem (szegélye) körüli terület méretének és jellegének szabályozására szolgál. Háttérszíne nincs, azaz teljesen transzparens (=transparent).

A felső, alsó, bal és jobb margókat, a megfelelő jellemzőkkel, külön is beállíthatjuk. A gyorsírási **margin** jellemző mindegyikre egyszerre vonatkozik.

Lehetséges értékek

A margó méretét (illetve általában az objektum-méreteket) háromféleképpen adhatjuk meg, az alábbi táblázat értelmében:

Érték	Magyarázat
auto	A margó méretezését a böngészőre bizzuk. Az eredmény az eltérő program-alapértelmezések függvényében változó lehet.
<i>length</i>	Megadhatjuk a margó-méret abszolútértékét (pl. px, pt, em, cm).
%	Relatív méretezés: a margó-méretét az őt tartalmazó elem méretének %-aként adjuk meg.

A margó méretét negatívnak is választhatjuk, ha egyes tartalmakat fedésbe kívánunk hozni egymással.

Oldalspecifikus margó-beállítás

Ahogy a szegélyeket és kontúrokat, úgy a margókat is lehetséges külön-külön méretezni:

```
<html>
<head>
<style type="text/css">
p
{
background-color:yellow;
}
p.margin
{
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
margin-left:50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified margins.</p>
<p class="margin">This is a paragraph with specified margins.</p>
</body>
</html>
```

Itt az első bekezdés körül nincs (ill. alapértelmezett méretű) a) margó, míg a **margin** csoportba tartozó második fölött és alatt 100px, a két oldalán pedig 50-50px.

Margó-jellemzők gyorsírása

A kód rövidítéséhez a margóra vonatkozó meghatározásokat egyetlen **margin** kijelölés alá is összevonhatjuk:

```
<html>
<head>
<style type="text/css">
p
```

```

{
background-color:yellow;
}
p.margin
{
margin:100px 50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified margins.</p>
<p class="margin">This is a paragraph with specified margins.</p>
</body>
</html>

```

Mivel a margó-jellemzők gyorsírására a korábbi elemekével megegyező sorrendbeli szabályok vonatkoznak, egyértelmű, hogy az utóbi példa eredménye megegyezik az azt megelőzővel. Ti. a **margin** jellemző első értéke a felső és alsó oldalakra, míg a második a szélsőkre vonatkozik.

Az előzőekhez hasonlóan itt is felírhatunk az egyes esetekre vonatkozó példákat, a **margin** kijelölés értékeinek négyféle megadási módja szerint:

1. **margin:25px;**
mind a négy margó 25px-es.
2. **margin:25px 50px;**
a felső és alsó margó 25px, az oldalsók pedig 50px-esek.
3. **margin:25px 50px 75px;**
a felső margó 25px, az oldalsók 50px-esek, az alsó pedig 75px-es.
4. **margin:25px 50px 75px 100px;**
a felső margó 25px, a jobboldali 50px, az alsó 75px, a baloldali pedig 100px.

További példák

Első példánkban egy szövegrész felső margójának cm-ekben való megadását láthatjuk:

```

<html>
<head>
<style type="text/css">
p.ex1 {margin-top:2cm;}
</style>
</head>

<body>
<p>A paragraph with no margins specified.</p>
<p class="ex1">A paragraph with a 2cm top margin.</p>
<p>A paragraph with no margins specified.</p>
</body>
</html>

```

Az első és második bekezdés által elfoglalt két sor közt hozzávetőleg 2cm köz látható. Az **ex1** bekezdés szemmel láthatólag csak a felső margó meglétében tér el a többitől (azaz csak felső margója van).

Második példánkban alsó margót adunk meg, %-os értékkel:

```

<html>
<head>
<style type="text/css">
p.bottommargin {margin-bottom:25%;}
</style>
</head>

```

```

<body>
<p>This is a paragraph with no margin specified.</p>
<p class="bottommargin">This is a paragraph with a specified bottom
margin.</p>
<p>This is a paragraph with no margin specified.</p>
</body>
</html>

```

Itt a második bekezdés alatt látható nagyobb térköz, ami az egész böngésző ablak függőleges kiterjedésének kb. 25%-a (ha leszámítjuk a felső banner-sávot).

CSS margó-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
<u>margin</u>	Az összes margó-beállítás egyszeri (egyetlen, összetett meghatározással való) elvégzésére szolgál.	<i>margin-top</i> <i>margin-right</i> <i>margin-bottom</i> <i>margin-left</i>	1
<u>margin-bottom</u>	Az alsó margó beállítására való.	auto <i>length</i> %	1
<u>margin-left</u>	A bal margó beállítására való.	auto <i>length</i> %	1
<u>margin-right</u>	A jobb margó beállítására való.	auto <i>length</i> %	1
<u>margin-top</u>	A felső margó beállítására való.	auto <i>length</i> %	1

CSS-margók:

Egyedi (ill. „oldalspecifikus”) margó-beállítás:

Teteje: {margin-top:auto/100px/100pt/100em/100cm/100%;}
Jobbja: {margin-right:auto/100px/100pt/100em/100cm/100%;}
Alja: {margin-bottom:auto/100px/100pt/100em/100cm/100%;}
Balja: {margin-left:auto/100px/100pt/100em/100cm/100%;}

Csoportos margó-beállítás (a margó-jellemzők gyorsírása):

Minden oldal: {margin:méret;}
Vízszintes/függőleges oldalak: {margin:méret méret;}
Teteje/olalai/alja: {margin:méret méret méret;}
Teteje/jobbja/alja/balja: {margin:méret méret méret méret;}

V. CSS – térköz

A CSS térköz-jellemzők az elemek tartalma és szegélye közti térközt szabályozzák.

Térköz

A térköz (=padding) a szegélyen belül elhelyezkedő elem-tartalom körüli határmezsgyét jelenti. Színe megegyezik az elem háttérszínével.

A felső, jobboldali, alsó és baloldali térközt megfelelő jellemzőkkel külön-külön is meghatározhatjuk. A gyorsíráshoz **padding** jellemzővel az összes térközt egyszerre, egyféleképpen határozhatjuk meg.

Térköz – értékek

A térköz-jellemzők értékét kétféle módon adhatjuk meg:

Érték	Magyarázat
<i>length</i>	Megadhatjuk a térköz-méret abszolútértékét (pl. px, pt, em, cm).
%	Relatív méretezés: a térköz-méretét az őt tartalmazó elem méretének %-aként adjuk meg.

Oldalspecifikus térköz-beállítás

A CSS segítségével az elem egyes oldalainak térközét külön-külön is beállíthatjuk:

```
<html>
<head>
<style type="text/css">
p
{
background-color:yellow;
}
p.padding
{
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
padding-left:50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>
</html>
```

Példánk első bekezdése körül nincs térköz, míg a másodikat felülről és alulról 25px, balról 50px, jobbról pedig legalább 50px hely veszi körbe; melynek színe a betűk hátterével egyezően, sárga. Hogyha a bal margót 900, a jobbot pedig 500px-nek vesszük, akkor az oldal alján csúszka jelenik meg, és a szöveg egy keskeny, függőleges sávba tömörül; tőle balra nagyobb, jobbra kisebb hely látszik. Azaz a margó a szövegigazítástól függően adott (jobb vagy bal) irányban lehet a megadott értéknél nagyobb, de kisebb sohasem. Jól látszik ez, hogyha a **padding-right** értékét 1 vagy 0px-nek választjuk; az eredmény ugyanaz, mint 50px esetén, mivel a szöveg végétől jobbra rendelkezésre álló hely jóval nagyobb, mint 50px.

Térköz-beállítás gyorsírással

A kód lerövidítése érdekében a térközt egyetlen, gyorsíráshoz meghatározással is megadhatjuk, mely azonban minden oldalra érvényes. A térköz gyorsíráshoz jellemzője a **padding**:

```
<html>
<head>
<style type="text/css">
```

```

p
{
background-color:yellow;
}
p.padding
{
padding:25px 50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>
</html>

```

Példánk, mint a korábbi doboz-alkotórészek gyorsírásos jellemzéséből sejtethetjük, eredményében megegyezik a megelőzővel. A korábbiakhoz hasonlóan ugyanis itt is azonos sorrendbeli szabályok érvényesek az egyes oldalakhoz tartozó értékek megadására.

Itt is négyféle gyorsírásos értékmegadási lehetőség van, melyek eredményei rendre a következők:

1. **padding:25px;**
mind a négy térköz 25px-es.
2. **padding:25px 50px;**
a felső és alsó térköz 25px, az oldalsók pedig 50px-esek.
3. **padding:25px 50px 75px;**
a felső térköz 25px, az oldalsók 50px-esek, az alsó pedig 75px-es.
4. **padding:25px 50px 75px 100px;**
a felső térköz margó 25px, a jobboldali 50px, az alsó 75px, a baloldali pedig 100px.

További példák

Első példánkban az utóbbi **border** gyorsírásos jellemzőt használjuk két bekezdés térközeinek beállítására:

```

<html>
<head>
<style type="text/css">
p.ex1 {padding:2cm;}
p.ex2 {padding:0.5cm 3cm;}
</style>
</head>

<body>
<p class="ex1">This text has equal padding on each side. The padding on
each side is 2cm.</p>
<p class="ex2">This text has a top and bottom padding of 0.5cm and a
left and right padding of 3cm.</p>
</body>
</html>

```

Könnyebben áttekinthető lesz az eredmény, hogyha kereteket is beállítunk a bekezdéseknek:

```

p.ex1 {border-style:solid;border-width:5px;padding:2cm;}
p.ex2 {border-style:solid;border-width:5px;padding:0.5cm 3cm;}

```

Ekkor a felső bekezdés körül, a szegélyen belül, tényleg mindenhol 2cm-es hely van; és jól megfigyelhető, hogy a második fölött és alatt ennél kevesebb, két oldalán pedig több a térköz.

Következő példánkban csak baloldali térközöket állítunk be:

```

<html>

```

```
<head>
<style type="text/css">
p.padding {padding-left:2cm;}
p.padding2 {padding-left:50%;}
</style>
</head>
```

```
<body>
<p>This is a text with no left padding.</p>
<p class="padding">This text has a left padding of 2 cm.</p>
<p class="padding2">This text has a left padding of 50%.</p>
</body>
</html>
```

A bekezdéseknek nincs felső, alsó és jobb térközük (amint arról egy keret beállításával megbizonyosodhatunk; pl. a **border-style:solid;** paranccsal). Az első bekezdés az oldal jobb szélén van, a második 2cm-rel beljebb, a harmadik pedig kb. az oldal (ill. frame) közepén kezdődik.

Harmadik példánkban jobboldali térközöket állítunk be:

```
<html>
<head>
<style type="text/css">
p.padding {padding-right:2cm;border-style:solid;}
p.padding2 {padding-right:50%;}
</style>
</head>
```

```
<body>
<p>This is a text with no right padding. This is a text with no right
padding. This is a text with no right padding.</p>
<p class="padding">This text has a right padding of 2 cm. This text has
a right padding of 2 cm. This text has a right padding of 2 cm.</p>
<p class="padding2">This text has a right padding of 50%. This text has
a right padding of 50%. This text has a right padding of 50%.</p>
</body>
</html>
```

Mint látszik, a felső bekezdésnek nincs térköze; a másodiknak, ahol keretet is beállítottunk, 2cm-es jobb térköze van. A harmadik bekezdés jobb széle az oldal közepén van. (A bekezdéseknek itt is legfeljebb csak jobb térközük van.)

Negyedik példánkban a felső térköz egyedüli beállítását láthatjuk:

```
<html>
<head>
<style type="text/css">
p {border-style:solid;}
p.padding {padding-top:2cm;}
p.padding2 {padding-top:50%;}
</style>
</head>
```

```
<body>
<p>This is a text with no top padding. This is a text with no top
padding. This is a text with no top padding.</p>
<p class="padding">This text has a top padding of 2 cm. This text has a
top padding of 2 cm. This text has a top padding of 2 cm.</p>
```

```
<p class="padding2">This text has a top padding of 50%. This text has a top padding of 50%. This text has a top padding of 50%.</p>
```

```
</body>
```

```
</html>
```

Mint látjuk, a jobb kiértékelhetőség kedvéért minden `<p>` elemre vonatkozólag keretet állítottunk be. Az első bekezdésnek nincs térköze, a második fölött 2cm-nyi található; a harmadik fölött pedig annyi, amennyi a frame-ben lévő, csúszkával ellátott oldal magassága 50%-ának felel meg.

Utolsó, ötödik példánk:

```
<html>
```

```
<head>
```

```
<style type="text/css">
```

```
p {border:solid;}
```

```
p.padding {padding-bottom:2cm;}
```

```
p.padding2 {padding-bottom:50%;}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p>This is a text with no bottom padding. This is a text with no bottom padding. This is a text with no bottom padding.</p>
```

```
<p class="padding">This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm.</p>
```

```
<p class="padding2">This text has a bottom padding of 50%. This text has a bottom padding of 50%.</p>
```

```
</body>
```

```
</html>
```

Itt ugyanaz a helyzet, mint az előbb, csak a keretet egyrészt rövidebb alakban adtuk hozzá a bekezdésekhez, másrészt a térközök a második ill. harmadik bekezdések tartalma alatt vannak.

CSS térköz-jellemzők és értékeik

Az alábbi táblázatban összefoglaltuk a címben szereplő jellemzőket és azok lehetséges értékeit. A „CSS” oszlopba írt szám azt mutatja, hogy az adott jellemzőt a CSS mely verziója tartalmazta először:

Jellemző	Leírás	Lehetséges értékek	CSS
padding	Az összes oldali térköz egyszeri, egyféle méretezésére szolgál.	<i>padding-top</i> <i>padding-right</i> <i>padding-bottom</i> <i>padding-left</i>	1
padding-bottom	Az alsó térköz-méret beállítására való.	<i>length</i> <i>%</i>	1
padding-left	A baloldali térköz-méret beállítására való.	<i>length</i> <i>%</i>	1
padding-right	A jobboldali térköz-méret beállítására való.	<i>length</i> <i>%</i>	1
padding-top	A felső térköz-méret beállítására való.	<i>length</i> <i>%</i>	1

CSS-térköz:

Egyedi (ill. „oldalspecifikus”) térköz-beállítás:

Teteje: {padding-top:100px/100pt/100em/100cm/100%;}

Jobbja: {padding-right:100px/100pt/100em/100cm/100%;}

Alja: {padding-bottom:100px/100pt/100em/100cm/100%;}

Balja: {padding-left:100px/100pt/100em/100cm/100%;}

Csoportos térköz-beállítás (a térköz-jellemzők gyorsírása):

Minden oldal:	{padding:méret;}
Vízszintes/függőleges oldalak:	{padding:méret méret;}
Teteje/olalai/alja:	{padding:méret méret méret;}
Teteje/jobbjá/alja/balja:	{padding:méret méret méret méret;}

CSS-ÖSSZEFOGLALÁS

I. KÖNYV: ALAPOK

I. Bevezetés

II. CSS – mondattan

CSS-utasítás (általános alak): *kijelölés {jellemző:érték;jellemző:érték;}*

CSS-utasítás (konkrét példa): `h1 {color:blue;font-size:12px;}`

Külső stílus-oldal:

Hivatkozás külső stílus-oldalra (a HTML-dokumentum fejrészében):

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

Egy külső .css-fájl tartalma [csak stílus-utasításokat (kijelöléseket + meghatározásokat) tartalmaz]:

```
body {background-color:yellow;}
```

```
h1 {font-size:36pt;}
```

```
h2 {color:blue;}
```

```
p {margin-left:50px;}
```

Belső stílus-oldal:

Általános alak:

```
<style type="text/css">
```

```
kijelölés {jellemző:érték;jellemző:érték;}
```

```
kijelölés
```

```
{
```

```
jellemző:érték;
```

```
jellemző:érték;
```

```
jellemző:érték;
```

```
jellemző:érték;
```

```
}
```

```
</style>
```

Konkrét példa:

```
<style type="text/css">
```

```
p {color:red;text-align:center;}
```

```
h1 {color:green;font-size:20px;}
```

```
h2 {color:blue;font-size:20pt;}
```

```
</style>
```

CSS-megjegyzés [a .css fájlba ill. a belső stílus-oldalba (<style> elembe) írható]:

*/*A megjegyzés szövege*/*

III. CSS – id és class kijelölések

Egyéni kijelölés (id):

Egyéni kijelölés egy elemben: `<p id="név"></p>`

Az elem kijelölése a CSS-ben: `#név`

FONTOS: A név nem kezdődhet számmal!

Csoportos kijelölés (class):

Egy csoportba sorolt elem: `<p class="csoportnév"></p>`

A csoport kijelölése a CSS-ben: `.név`

A csoportba tartozó, adott típusú elemek kijelölése a CSS-ben:

`típus.név` (pl.: `h1.class_1`)

FONTOS: A név nem kezdődhet számmal!

IV. CSS – módszertan

A CSS beillesztése:

1. **Külső CSS:** `<link rel="stylesheet" type="text/css" href="URL" />`

(A külső forrás egy különálló CSS-fájl, melyben csakis CSS kijelölések és meghatározások lehetnek, egyéb tartalom nem! A link tag az oldal teljes tartalmát bemásolja!)

HIBÁS: A külső forrás lehet egy különálló .css-fájl, vagy egy másik weboldal belső stílus-oldala. Utóbbit egy `<a>` taggel tehetjük linkelhetővé (pl. a belső CSS-t az `` taggel könyvjelzőzzük, és linkeljük: `<link rel="stylesheet" type="text/css" href="URL#név" />`).

2. **Belső CSS:** `<style type="text/css">`
`kijelölés {jellemző:érték;}`
`</style>`

3. **Szövegtestbe írt (inline) stílus-attribútumok, pl.:**
`<p style="color:green;margin-left:20px"></p>`

A CSS-típusok prioritási sorrendje:

1. Szövegtestbe írt stílus (`style` attribútum a HTML-elemben);
2. Belső stílus-oldal (a dokumentum `<head>` részében);
3. Külső stílus-oldal;
4. Böngésző-alapértelmezés.

A sorban előbb állók felülírják az utóbbiakat. A csak egy helyen meghatározott (tehát lejjebb felül nem írt) jellemzők az adott kijelölés szintjén ill. az alatt érvényesek.

II. KÖNYV: FORMÁZÁS

I. CSS – hátterek

Háttér-szín: `{background-color:#ffffff;}` (HEX)
`{background-color:rgb(255,255,255);}` (RGB)
`{background-color:green;}` (triviális színnév)

HTML-dokumentumrész: `<div></div>`

Háttér-kép:

Alapértelmezett (a kitöltéshez végtelenig ismételt) háttérkép-beállítás:

`body {background-image:url('URL');}`

Kitöltési hatás beállítása:

`body {background-repeat:repeat/repeat-x/repeat-y/no-repeat/inherit;}`

Háttérkép pozicionálása (no-repeat mellett):

`body {background-position:left top/left center/left bottom/right top/right center/right bottom/center top/center center/center bottom/100% 100%/100px 100px/inherit;}`

Háttérkép rögzítése (no-repeat mellett):

`body {background-attachment:scroll/scroll/scroll/fixed/inherit;}`

Háttér-jellemzők gyorsírása:

`{background:színnév url('URL') kitöltés-típus pozíció;}`

Egy konkrét példa:

`{background:#ffffff url('image.jpg') no-repeat right top;}`

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

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

II. CSS – szöveg

Betűszín (itt az egész szövegtestre): `body {color:green/rgb(255,255,255)/#ffffff;}`

FONTOS: A szín megadásához a background-color-t is be kell állítani!

Szöveghatások (text-decoration; text-transform):

Nincs/alapértelmezett: `{text-decoration:none;}`

Felülhúzott: `{text-decoration:overline;}`

Áthúzott: {text-decoration:line-through;}
Aláhúzott: {text-decoration:underline;}
Villogó: {text-decoration:blink;}
Nagybetűs: {text-transform:uppercase;}
Kisbetűs: {text-decoration:lowercase;}
Nagy kezdőbetűk: {text-decoration:capitalize;}
Nincs/alapértelmezett: {text-decoration:none;}
Szövegirány: {direction:ltr/rtl;}

Térközök a szövegben:

Betűköz: {letter-spacing:±100px/normal;} (Alapértelmezett értéke: 0px.)
Szóköz: {word-spacing:100px/normal;}
Sorköz: {line-height:/normal/100/100px/100%;} (Alapértelmezett értéke: 100%.)

Tabulálás, szöveg-behúzás (text-indent):

{text-indent:100px/100%;}

Szövegigazítás (text-align):

Balra zárt: {text-align:left;}
Középre: {text-align:center;}
Jobbra zárt: {text-align:right;}
Sorkizárt: {text-align:justify;}

Beágyazott elem (függőleges) pozícionálása a szöveg-sorhoz képest (vertical-align):

img {vertical-align:baseline/sub/super/top/text-top/middle/bottom/text-bottom/100px/100%;}

Automatikus szövegtördelés:

Engedélyezve (alapértelmezett; a szóközők egybeolvadnak, és a szöveg megtörhet rajtuk):
{white-space:normal;}
Tiltva (a szóközők egybeolvadnak, de a szöveg nem törhet meg rajtuk):
{white-space:nowrap;}
Kikapcsolva (a szóközők megmaradnak és a szöveg nem törhet meg rajtuk):
{white-space:pre;}

[A szöveg-formázásnak nincs gyorsírási jellemzője (pl.: *text*).]

III. CSS – betűk

Betűcsaládok:

Serif: talpakkal rendelkező betűk (pl.: Times New Roman).
Sans-serif: talpakkal nem rendelkező betűk (pl.: Arial).
Monospace: egyenlő szélességű betűk (pl.: Courier).

Betűtípus/betűcsalád beállítása:

Általános alak: {font-family:"speciális betűtípus","általános betűtípus",betűcsalád;}
A több tagból álló betűtípus- ill. -család-neveket idézőjelbe írjuk.
Konkrét példa: {font-family:"Times New Roman",Times,serif;}

Betűhatások (font-style):

Normális (alapértelmezett): {font-style:normal;}
Dólt: {font-style:italic;}
Ferde (kb. a dőltenel megegyezik): {font-style:oblique;}

Kövérség (font-weight): {font-weight:normal/lighter/bold/bolder/900/inherit;}

Kiskapitális (font-variant): {font-variant:normal/small-caps/inherit;}

Betűméret (font-size):

Abszolút betűméret névvel: {font-size:small/medium/large;}
Abszolút betűméret mérőszámmal: {font-size:100px/100pt/cm;}
Relatív betűméret: {font-size:100%/10.085em;}

Univerzális, nagyítható betűméretezés:

A formázandó szöveget tartalmazó elem (<body>) betűméretét az böngésző-alapértelmezés %-ában megadjuk, majd az egyes elemeket em-értékekben, relatívan méretezzük, pl.:
body {font-size:100%;}


```
h1 {font-size:3em;}
h2 {font-size:2em;}
p {font-size:1.05em;}
```

Betű-jellemzők gyorsírása (font):

{font:betűhatás kiskapitális kövérség betűméret/sorköz betűcsalád;}

Egy konkrét példa:

{font:italic small-caps 700 5.05em/5cm "Courier New";}

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

1. font-style
2. font-variant
3. font-weight
4. font-size/line-height
5. font-family

FONTOS: gyorsírásnál a font-size és font-family jellemzők kiföltése kötelező!

IV. CSS – linkek

Link-típusok elnevezése:

Normális, nem látogatott: link
Már meglátogatott: visited
Egérrel érintett: hover
Működő (éppen kattintott): active

FONTOS: Az egyes kijelölések mindig ebben a sorrendben követik egymást!

Link-kijelölések:

A dokumentum összes linkje: a:link/visited/hover/active
Egy csoportba tartozó linkek: a.csoportnév:link/visited/hover/active
Egy id alá tartozó link(ek): a#név:link/visited/hover/active
Többszörös kijelölésnél az egyes elnevezéseket vessző választja el (pl.: a:link, a:visited).

Link-színek:

```
a:link {color:#FF0000;}
a:visited {color:#00FF00;}
a:hover {color:#0000FF;}
a:active {color:#000000;}
```

Link-szövegátalakítások (text-decoration; lásd feljebb a CSS-szöveg részt!):

Nincs: {text-decoration:none;}
Aláhúzott (alapértelmezett): {text-decoration:underline;}

Link-betűhatások (font; lásd feljebb a CSS-betűk részt!):

Betűméret: {font-size:100px/100em;}
Betűcsalád: {font-family:serif/sans-serif/monospace;}
Kövérség: {font-weight:normal/lighter/bold/bolder/900/inherit;}
A betűhatásokat gyorsírásosan, a font jellezővel is meghatározhatjuk!

Link-szövegigazítás (text-align, lásd feljebb a CSS-szöveg részt!):

{text-align:left/center/right/justify;}

Link-háttérszín (background-color; lásd feljebb a CSS-hátterek részt!):

{background-color:green/rgb(255,255,255)/#ffffff;}
A háttérrel gyorsírásosan, a background jellezővel is meghatározhatjuk!

Link-alak:

Téglalap: {display:block;}

Link-méret:

Szélesség: {width:100px/100cm/100%;}
Térköz: {padding:100px/100cm/100%;}

V. CSS – listák

A formázandó elemek kijelölése ugyanolyan, mint a korábbiakban (pl.: ul vagy ul.név vagy ul#név).

Rendezetlen-lista jeltípusok:

Nincs jel: {list-style-type:none;}
Korong: {list-style-type:disc;}
Körvonal: {list-style-type:circle;}
Négyzet: {list-style-type:square;}

Rendezett-lista jeltípusok:

Arab szám (1, 2, 3): {list-style-type:decimal;}
Arab szám kezdő nullával (01, 02, 03): {list-style-type:decimal-leading-zero;}
Nagy rómais szám (I, II, III): {list-style-type:upper-roman;}
Kis rómais szám (i, ii, iii): {list-style-type:lower-roman;}
Nagy ABC (A, B, C): {list-style-type:upper-alpha;}
Kis ABC (a, b, c): {list-style-type:lower-alpha;}
Kis görög betűk (α, β, γ): {list-style-type:lower-greek;}
Egyes rendezett-lista jeltípusok használatához a DOCTYPE beállítandó!

Rendezetlen lista jelölése egyedi képpel:

{list-style-image:url('URL');}

Rendezetlen lista jelölése háttérképpel:

ul {list-style-type:none;padding:0px;margin:0px;}
li {background-image:url('URL');background-repeat:no-repeat;
background-position:0px 5px;padding-left:14px;}

Listaelem-jel – pozíció:

A szövegtörzs vonalán belül: {list-style-position:inside;}
A szövegtörzs vonalán kívül: {list-style-position:outside;}

Lista-jellemzők gyorsírása (list):

{list:listatípus listaelem-jel-pozíció listajelölő-kép-URL;}

Az értékek jellemzők szerinti sorrendje a gyorsírásnál:

1. list-style-type
2. list-style-position
3. list-style-image.

VI. CSS – táblázatok

Táblázat: <table></table>

Táblázattípus: <caption></caption> [A <table> elembe írható!]

Táblázattípus-pozíció: caption {caption-side:top/bottom/inherit,}

Szegélyek/szegélyszín:

Táblázat többszörös szegéllyel: table,th,td {border:1px solid black;}

Táblázat egyszeres szegéllyel: table {border-collapse:collapse;}
table,td,th {border:1px solid black;}

A border-collapse jellemző használatához DOCTYPE kellhet!

[A szegély-típusokról bővebben lásd alább a CSS-szegélyek részt!]

Betűszín: {color:green/rgb(255,255,255)/#ffffff;}

Háttérszín: {background-color:green/rgb(255,255,255)/#ffffff;}

Betűtípus: {font-family:"speciális betűtípus", "általános betűtípus", "betűcsalád";}

Betűméret: {font-size:100px/100em/100%;}

Szélesség/magasság:

Szélesség: {width:100px/100%;}

Magasság: {height:100px/100%;}

Szövegigazítás:

Vízszintesen: {text-align:left/center/right;}

Függőlegesen: {vertical-align:top/middle/bottom;}

Térköz: {padding:100px/5cm/100%;}

[Részletesebben lásd alább, a CSS-térköz részben!]

III. KÖNYV: DOBOZ-MODELL

I. CSS doboz-modell

A HTML-elem, mint doboz, részei:

- **content:** a doboz tartalma, ahol az elembe tartozó szöveg, kép vagy egyéb adat megjelenik. Háttérszíne adja az egész dobozét is.
- **padding:** az elem tartalma és a szegély közötti térköz. Színe megegyezik a doboz háttérszínével.
- **border:** az elem tartalmát és az akörül térközt körülvevő szegély. Színe a doboz háttérszínéhez igazodjon!
- **margin:** a határvonal körüli szabad rész méretét adja meg. Háttérszíne nincs, azaz átlátszó (=transparent).

Doboz-méretezés:

```
{width:100px;  
height:100px;  
padding:100px;  
border:100px solid green;  
margin:100px;}
```

A doboz össz-szélessége ill. -magassága a Firefox-hoz hasonló böngészőkben (valamint IE-ben, megfelelő DOCTYPE mellett) =

elem-méret + bal térköz + jobb térköz + bal szegély + jobb szegély + bal margó + jobb margó.

Ehhez megfelelő DOCTYPE-declaration (melyet a dokumentum fejlécszele elé írunk):

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

A doboz össz-szélessége ill. -magassága az Internet Explorerhez hasonló böngészőkben (más DOCTYPE mellett) =

elem-méret (beleértve a térközt és szegélyt is) + bal margó + jobb margó.

II. CSS – szegélyek

Szegélytípus (border-style):

Nincs keret: {border-style:none;}
Pontozott: {border-style:dotted;}
Kötőjeles: {border-style:dashed;}
Vonalas: {border-style:solid;}
Kettős vonal: {border-style:double;}
Vájt: {border-style:groove;}
Domború: {border-style:ridge;}
Süllyesztett: {border-style:inset;}
Kiemelt: {border-style:outset;}
Elrejtett (nem látszó) keret: {border-style:hidden;}

Oldalspecifikus szegélytípus:

Felső szegély: {border-top-style:típusnév;}
Jobb szegély: {border-right-style:típusnév;}
Alsó szegély: {border-bottom-style:típusnév;}
Bal szegély: {border-left-style:típusnév;}

Szegélytípus-gyorsírás:

Minden oldal: {border-style:típusnév;}
Vízszintes/függőleges oldalak: {border-style:típusnév típusnév;}
Teteje/olalai/alja: {border-style:típusnév típusnév típusnév;}
Teteje/jobbjaja/alja/balja: {border-style:típusnév típusnév típusnév típusnév;}

Vastagság: {border-width:*thin/medium/thick/100px*;}
Megadásához mindig állítsuk be a **border-style**-t is!

Oldalspecifikus szegélyvastagság:

Felső szegély: {border-top-width:*méret*;}
Jobb szegély: {border-right-width:*méret*;}
Alsó szegély: {border-bottom-width:*méret*;}
Bal szegély: {border-left-width:*méret*;}
}

Szegélyvastagság-gyorsírás:

Minden oldal: {border-width:*méret*;}
Vízszintes/függőleges oldalak: {border-width:*méret méret*;}
Teteje/olalai/alja: {border-width:*méret méret méret*;}
Teteje/jobbjaja/alja/balja: {border-width:*méret méret méret méret*;}
}

Szín: {border-color:*black/rgb(255,255,255)/#ffffff/transparent*;}
Megadásához mindig állítsuk be a **border-style**-t is!

Oldalspecifikus szegélyszín:

Felső szegély: {border-top-color:*szín*;}
Jobb szegély: {border-right-color:*szín*;}
Alsó szegély: {border-bottom-color:*szín*;}
Bal szegély: {border-left-color:*szín*;}
}

Szegélyszín-gyorsírás:

Minden oldal: {border-color:*szín*;}
Vízszintes/függőleges oldalak: {border-color:*szín szín*;}
Teteje/olalai/alja: {border-color:*szín szín szín*;}
Teteje/jobbjaja/alja/balja: {border-color:*szín szín szín szín*;}
}

Szegély-jellemzők gyorsírása (border):

Minden oldal: {border:*vastagság/típus/szín*}
Teteje: {border-top:*vastagság/típus/szín*}
Jobbjaja: {border-right:*vastagság/típus/szín*}
Alja: {border-bottom:*vastagság/típus/szín*}
Balja: {border-left:*vastagság/típus/szín*}
}

III. CSS – kontúrozás

Kontúr-típus:

Nincs kontúr: {outline-style:none;}
Pontozott: {outline-style:dotted;}
Kötőjeles: {outline-style:dashed;}
Vonalas: {outline-style:solid;}
Kettős vonal: {outline-style:double;}
Vájt: {outline-style:groove;}
Domború: {outline-style:ridge;}
Süllyesztett: {outline-style:inset;}
Kiemelt: {outline-style:outset;}
Elrejtett (nem látszó) kontúr: {outline-style:hidden;}
}

Kontúr-szín: {outline-color:*green/rgb(255,255,255)/#ffffff*;}
Megadásához mindig állítsuk be az **outline-style**-t is!

Kontúr-vastagság: {outline-width:*thin/medium/thick/100px*;}
Megadásához mindig állítsuk be az **outline-style**-t is!

Kontúr-jellemzők gyorsírása(outline):

{outline:*szín típus méret*;}
}

A CSS kontúr-jellemzők csak DOCTYPE megadása mellett működnek!

Példa egy DOCTYPE-ra:

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

IV. CSS – margók

Egyedi (ill. „oldalspecifikus”) margó-beállítás:

Teteje: {margin-top:auto/100px/100pt/100em/100cm/100%;}
Jobbja: {margin-right:auto/100px/100pt/100em/100cm/100%;}
Alja: {margin-bottom:auto/100px/100pt/100em/100cm/100%;}
Balja: {margin-left:auto/100px/100pt/100em/100cm/100%;}

Csoportos margó-beállítás (a margó-jellemzők gyorsírása):

Minden oldal: {margin:méret;}
Vízszintes/függőleges oldalak: {margin:méret méret;}
Teteje/olalai/alja: {margin:méret méret méret;}
Teteje/jobbja/alja/balja: {margin:méret méret méret méret;}

V. CSS – térköz

Egyedi (ill. „oldalspecifikus”) térköz-beállítás:

Teteje: {padding-top:100px/100pt/100em/100cm/100%;}
Jobbja: {padding-right:100px/100pt/100em/100cm/100%;}
Alja: {padding-bottom:100px/100pt/100em/100cm/100%;}
Balja: {padding-left:100px/100pt/100em/100cm/100%;}

Csoportos térköz-beállítás (a térköz-jellemzők gyorsírása):

Minden oldal: {padding:méret;}
Vízszintes/függőleges oldalak: {padding:méret méret;}
Teteje/olalai/alja: {padding:méret méret méret;}
Teteje/jobbja/alja/balja: {padding:méret méret méret méret;}