



szit

< CSS

CSS

- **Szerző:** Sallai András
- Copyright © Sallai András, 2011-2020
- Licenc: CC Attribution-Share Alike 4.0 International [https://creativecommons.org/licenses/by-sa/4.0/]
- Web: https://szit.hu [https://szit.hu]

Bevezetés

A CSS a **Cascading Style Sheets** betűszavaiból alkotott mozaikszó. Magyarul **Lépcsős stíluslapok** néven fordítható.

A CSS egy egyszerű mechanizmus amellyel stílust (pl.: fontok, színek, helyek) határozhatunk meg webes dokumentumainkhoz.

A CSS legelterjedtebb használata a weboldalak alakításánál legismertebb. A webet Tim Berners-Lee eredetileg is úgy tervezte, hogy szétválasszák a szerkezetet és a kinézetet meghatározó elemeket. A kezdeti böngészők viszont erre nem voltak képesek. Kezdetben a HTML elemekkel adtak stílusinformációkat a weblapokhoz, ami sokáig nem is változott.

Tim Berners-Lee 1990 év, második felében elkészítette az első böngészőt, amit WorldWideWeb-nek nevezett el, majd később átnevezték Nexus névre. A böngésző NeXTSTEP operációs rendszerre készült.

Marc Andreessen és Eric Bina az NCSA-nál fejleszteni kezdik a Mosaic böngészőt Unix operációs rendszeren, X Window System felületre, ekkor még xmosaic a neve. Az első kiadás 1993-01-23-án készült el. A Mosaic böngészőt később portolják AmigaOS, OpenVMS, Classic Mac OS, MS Windowsra.

1994 október 10-én, Hákon Wiium Lie közzétette lépcsőzetes stíluslapok nevű nyelvet. 1994 október 13 Marc Andreessen bejelenti a Mosaic böngésző támogatja a lépcsőzetes stíluslapokat, amely kezdetben csak a közép-re igazítást támogatja. A Mosaic-ből későbbi tovább fejlesztés eredményeképpen létrejön a Netscape böngésző. A Mosaic-ot közben (1995) megvásárolja a Microsoft, akik Internet Explorer nevű böngészőt hozzák belőle létre.

Mára a böngészők CSS képessége messze túlhaladja a HTML elemekkel lehetséges stílusbeállításokat.

Stíluslapkészítés

A CSS használata során, elsőként meg kell adnunk melyik HTML elem tulajdonságait akarjuk beállítani, vagyis választanunk kell egy szelektort. A szelektorhoz egy vagy több tulajdonság-értékpárt rendelünk.

Szelektor lehet például a h1 elem, ha h1 fejezet címmel meghatározott tartalomra szeretnénk stílust meghatározni. Magát a stílusinformációt kapcsos zárójelek közé teszem. Pl.:

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

Láthatjuk, hogy a kapcsos zárójelen belül a tulajdonságok és az értékek között kettőspont (:) szerepel szeparátorként. Minden tulajdonság-érték párt pontosvesszővel (;) zárunk.

CSS-t több szereplő is megadhat.

A CSS lehet

- **Szerzői**
 - a webmester beállításai
- **Felhasználói**
 - a felhasználó saját beállításai
- **Kliens**
 - a böngésző beállításai

Mi a szerzői stíluslapokkal foglalkozunk.

A szerzői stíluslapokat három módon köthetjük HTML dokumentumunkhoz:

- **Inline**
 - attribútum stílus
 - helyben felülírva a általános stílus
- **Internal**
 - tag stílus
 - beágyazva a dokumentumba
- **External**
 - külső stílus
 - fájl a dokumentumból hivatkozva

Mi most a External definícióval foglalkozunk.

Externál stílusmeghatározásnál a stílusinformációk külön állományban vannak. A HTML dokumentumban csak egy hivatkozást teszünk rá. Ez a legelőnyösebb, ha egy több HTML oldalból álló webhelyet akarunk készíteni. Ha változtatni akarunk weblapjaink stílusán csak egyetlen helye kell a változtatást megtenni.

A stílusinformációkat tartalmazó állomány kiterjesztése:

.css

Ha csak egy ilyen állományunk van szokásos neve:

style.css


A HTML oldalba a head elemek közé kell elhelyezni a hivatkozást egy link elem segítségével:

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



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

Próbálja ki az alábbi szöveg színezését. A szín választása után, kattintsunk a a színválasztón kívül.

Lorem ipsum

színkód

Háttér


A következő táblázat tartalmazza, milyen lehetőségeink vannak:

Háttér beállítások	
Tulajdonság	Lehetséges értékek
background-color	#RGBszínkód
background-image	url(kepneve.kit)
background-repeat	no-repeat, repeat-x, repeat-y, repeat
background-position	left, center, right, top, bottom
background-attachment	scroll, fixed, local
background-size	méret pl.: 140px 300px vagy „cover”, „contain”
background-origin	padding-box, border-box, content-box
background-clip	padding-box, border-box, content-box

Háttérszín

```
background-color: #990000;
```

Próbálja ki az alábbi szöveg háttérének színezését.

Lorem ipsum

színkód

Kép beállítása

```
background-image: url(hatter.png);
```

Egy lehetséges háttérkép:



Ismétlés

Az ismétlést a backgorund-repeat tulajdonsággal állítjuk.

```
background-repeat: repeat-x;
```

Próbáljuk ki az ismétlési lehetőségeket:

no-repeat ▼

a beállított érték

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud
exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.
Duis aute irure dolor in
reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id
est laborum.

Pozíció

Pozíciót a background-position tulajdonsággal állítok.

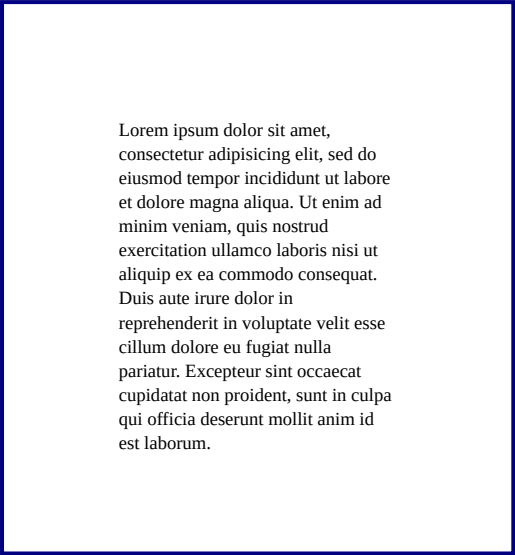
```
body {
  background-position: right;
}
```

Egyszerre két értéket is beállíthatok, de ennek csak akkor van értelme, ha a background-repeat tulajdonság például no-repeat;

```
body {
  background-repeat: no-repeat;
  background-position: right top;
}
```

left ▼

no-repeat ▼



Méret

A méret a background-size tulajdonsággal állítok.

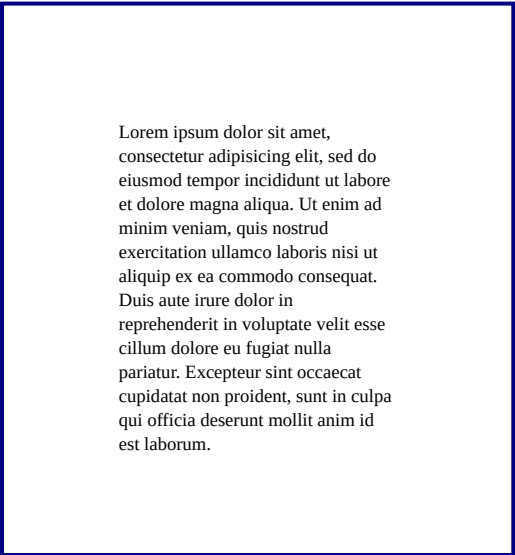
```
body {
  background-size: cover;
}
```

auto	Alapértelmezett érték. Eredeti méret mutatása.
hossz	Szélesség beállítása. Ha van második paraméter is, akkor az magasság.
százalék	Szélesség és magasság. Az első szélesség, a második magasság.
cover	Átméretezi a háttérképet, hogy kitöltsse az egész tárolót, még akkor is ha le kell vágni a képből.
contain	A kép legyen teljesen látható.
initial	Alapértelmezett érték beállítása.
inherit	Örökölt méret.

100px ▼

no-repeat ▼

virag2.png ▼



Két háttér

Két háttérkép is betehető egyszerre.

```
body {  
  background-image: url(kep001.jpg), url(kep003.jpg);  
  background-size: 50px 50px;  
  background-repeat: repeat-x, repeat-y;  
}
```

A példában az egyiket vízszintesen, másikat függőlegesen ismételtük.

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud
exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.
Duis aute irure dolor in
reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id
est laborum.

Pozíció megadásával:

```
body {  
  background-image:  
    url(kep001.jpg),  
    url(kep003.jpg);  
  background-size: 50px 50px;  
  background-repeat: no-repeat, no-repeat;  
  background-position: left top, right top;  
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua. Ut enim ad
minim veniam, quis nostrud
exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.
Duis aute irure dolor in
reprehenderit in voluptate velit esse
cillum dolore eu fugiat nulla
pariatur. Excepteur sint occaecat
cupidatat non proident, sunt in culpa
qui officia deserunt mollit anim id
est laborum.

Karakterkódolás beállítása

Általános szintaxis:

```
@charset "<IANA által definiált karakterkészlet neve>";
```

UTF-8 kódolás beállítása:

```
@charset "UTF-8";
```

Font

Stílus

```
body { font-style: italic; } /* normal, italic, oblique, inherit */
```

Italic:

Lorem ipsum dolor est amet.

Oblique:

Lorem ipsum dolor est amet.

2021. 10. 27. 10:59

oktatas:web:css:css_nyelv [szit]

Az Oblique lenne a kurzív, de tipográfiában a kurzív betűt úgy írják, hogy a tollat nem emelik meg, amit pedig mi kapunk nem valódi kurzív. A kurzív tehát egy külön fontkészlet lenne.

Az oblique itt azt jelenti a számítógép mesterségesen, fonttól függetlenül megdönti a betűket. A italic pedig a dőlt betűs fontkészletet tölti be.

Vastagság

```
body {
  /* normal, bold, bolder, lighter, 100, 200, 300, ..., 900, inherit */
  font-weight: bold;
}
```

Méret

```
body { font-size: 14px; }
```

font-size	
medium	Alapértelmezett érték.
xx-small	
x-small	
small	
large	
x-large	
xx-large	
smaller	Kisebb mint a szülőé
larger	Nagyobb mint a szülőé
hossz	Hossz számmal, mértékegységgel
százalék	Százalékos megadás

Fontcsalád

```
font-family: serif;
```

Font család	Megjelenés	Leírás
serif	serif	Talpas fontok
sans-serif	sans-serif	Talpatlan fontok
monospace	monospace	Fix betűszélességű fontok
cursive	cursive	Kurzív fontok
fantasy	fantasy	Fantázia fontok

Lorem ipsum dolores amet - serif

Lorem ipsum dolores amet - sans-serif

Lorem ipsum dolores amet - monospace

Lorem ipsum dolores amet - cursive

Lorem ipsum dolores amet - fantasy

Csoportban

```
body {
  font: italic bold 14px Arial, Helvetica, sans-serif;
}
```

Fontvariációk

```
font-variant: small-caps;
```

normal	Normál font. Alapértelmezett
small-caps	Kiskapitális
inherit	A szülőelemtől örökölt fontvariáció

```
/* kis-kapitális */
font-variant-caps: small-caps;

/* minden betű kis-kapitális méretű nagybetű */
font-variant-caps: all-small-caps;
```

- <https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant-caps> [https://developer.mozilla.org/en-US/docs/Web/CSS/font-variant-caps] (2021)

Néhány ötlet

001

Lorem ipsum dolores amet

```
color: #333333;
font-family: 'Lucida Sans Unicode', 'Lucida Grande', sans-serif;
```

002

Lorem ipsum dolores amet

color: #333333;
font-family: Century Gothic, sans-serif;

003

Lorem ipsum dolores amet

color: #333333;
font-family: Tahoma, Geneva, sans-serif;

004

Lorem ipsum dolores amet

color: #333333;
font-family: 'Palatino Linotype', 'Book Antiqua', Palatino, serif;

004

Lorem ipsum dolores amet

color: #333333;
font-family: 'Lucida Console', Monaco, monospace;

005

Lorem ipsum dolores amet

color: #333333;
font-family: 'Courier New', Courier, monospace;

006

Lorem ipsum dolores amet

color: #333333;
font-family: Arial, Helvetica, sans-serif;

007

Lorem ipsum dolores amet

color: #333333;
font-family: Georgia, Serif;

Szövegbeállítások

color	szöveg színe	
direction	szöveg iránya	ltr, rtl, inherit
letter-spacing	betűk közötti távolság	
line-height	sor magassága	
text-align	szöveg vízszintes igazítása	left, center, right, justify
text-decoration	szöveg dekoráció hozzáadása	none, overline, line-through, underline
text-indent	az első sor behúzása	pl.: 50px
text-shadow	árnyék	
text-transform	kis és nagybetűs	uppercase, lowercase, capitalize
unicode-bidi	A direction tulajdonsággal együtt használjuk. Felül kell-e ími a szöveget, hogy több nyelvet támogasson.	normal, embed, bidi-override, isolate, isolate-override, plaintext, initial, inherit
vertical-align	függőleges igazítás	baseline, hossz, %, super, top, text-top, middle, bottom, text-bottom, initial, inherit
white-space	a white-space karakterek viselkedése	normal, nowrap, pre, pre-line, pre-wrap, initial, inherit
word-spacing	szavak közötti távolság	

Dekoráció

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

Lehetséges értékei

none
underline
overline

line-through
underline overline

szöveg - text-decoration: underline

szöveg - text-decoration: overline

szöveg - text-decoration: line-through

szöveg - text-decoration: underline overline

szöveg - text-decoration: underline overline wavy blue;

```
body {
  text-decoration-line: underline;
}
```

Árnyék

```
body {
  text-shadow: 5px 5px 3px #222222;
}
```

Dekorációs színe

```
p {
  text-decoration: underline;
  text-decoration-color: red;
}
```

A dekoráció stílusa

```
div {
  text-decoration-style: solid;
}
```

text-decoration-style	
solid	Alapértelmezett
double	Dupla vonal
dotted	Pontozott vonal
dashed	Szaggatott vonal
wavy	Hullám

Lorem ipsum - text-decoration-style: solid;

Lorem ipsum - text-decoration-style: double;

Lorem ipsum - text-decoration-style: dotted;

Lorem ipsum - text-decoration-style: dashed;

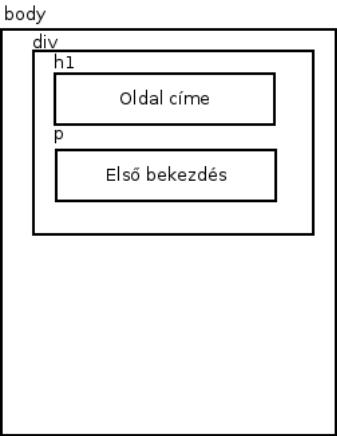
Lorem ipsum - text-decoration-style: wavy;

A white-space karakterek viselkedése

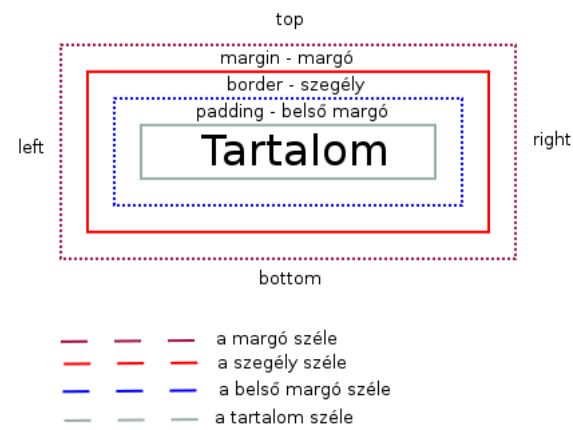
normal	A whitespacek összvonása. A sordobozt kitölti.
pre	Csak ott lesz sortörés, ahol a forrásban is sortörés van. Közök nem lesznek összevonva
nowrap	Közök összevonása. Szövegen belül nincs sortörés
pre-wrap	Közök nem lesznek összevonva. Sortörés, ahol forrásban is van, vagy ahol a sordoboz kitöltése megköveteli.
pre-line	Közök összevonása. Sortörés, ahol a forrásban új sor kezdődik, vagy ahol a sordoboz kitöltése megköveteli.

Dobozmodell

Minden HTML elemet doboznak tekintünk.



Minden doboznak lehet szöveges, vagy más más tartalma. A szövegnek állíthatjuk a színét, nagyságát, fontját. Minden doboz háttérét beállíthatjuk, színnel, képpel. Minden doboznak van szegélye. Minden doboznak állíthatunk belső margót (a tartalom és a doboz belső széle közötti távolság. Minden doboznak van margója. A dobozok ezen tulajdonságát külön tudjuk állítani, felül, alul, jobbra és balra.



Overflow

Egyszerű túlfolyás-kezelés

A szöveg túlfolyik a dobozon. Ezt kezelhetjük az overflow tulajdonsággal. Következő értékeket veheti fel: visible, hidden, scroll, auto. Az alapértelmezett a visible.

```
overflow: hidden;
```

A hidden érték esetén a túlfolyt szöveg egyszerűen eltűnik.

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit.
Etiam
condimentum

```
overflow: scroll;
```



```
overflow: none;
```

Az alapértelmezett "none" érték esetén a szöveg túlfolyik ha nem fér el a dobozban:

Lorem ipsum
dolor sit amet,
consectetur
adipiscing elit.
Etiam
condimentum
blandit sapien
non volutpat.

Az x és y tag

```
overflow-x: hidden;  
overflow-y: hidden;
```

Érték	Leírás
visible	A tartalmat nem bántjuk a görgető sáv a dobozon kívül lesz.
hidden	A tartalom nyírva lesz, a görgető sáv rejtve
scroll	A tartalom nyírva lesz, a görgető sáv megjelenik
auto	A görgetősáv megjelenése a tartalomtól függ
no-display	Ha a tartalom nem tölti ki a dobozt, akkor doboz törölve lesz.
no-content	Ha a tartalom nem tölti ki a dobozt, akkor tartalom elrejtésre kerül

Csak függőleges scroll:

```
overflow-y: scroll;
```



Keret tulajdonságok

border

```
.doboz {
  border: 2px solid black;
}
```

style

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

Lehetséges értékei

- none
- dotted
- dashed
- solid
- double
- groove
- ridge
- inset
- outset



border-width

Lehetséges értékei

- thin
- medium
- thick
- érték

border-collapse

Táblázat szegélyek esetén használható.

- separated
- collapse

Lista tulajdonság

A számozott és számozatlan listák többféle tulajdonságát tudjuk beállítani CSS-ből.

Megadhatjuk, hogy számozott legyen vagy számozatlan. A számozott milyen alakzattal jelenjen meg, esetleg milyen képpel. A számozott listánál, megadhatjuk római vagy latin számokat, ábécét, kis vagy nagy betűvel.

list-style-type

```
ul {
  list-style-type: circle;
}
```

- disc
- circle
- square
- decimal
- lower-roman
- upper-roman
- lower-alpha
- upper-alpha
- none

```
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;
}
```

Próbálja ki a következő beépített HTML részlettel:

- alma
- körte
- barack
- szilva
- málna
- meggy
- cseresznye

disc

list-style-image

```
ul {
  list-style-image: url(listajel.png);
}
```

list-style-position

```
ul {
  list-style-position: inside;
}
```

- inside
- outside

Az inside esetén a lista előtti alakzatot a lista tartalmazza. Az outside esetén azon kívül van. Ez a gyakorlatban úgy látszik, hogy outside esetén egy kicsit előrébb kezdődik, mint egy inside elrendezéssel formázott lista.

Megjelenítés

A dobozok megjelenését két tulajdonsággal lehet szabályozni:

- visibility
- display

```
h1.hidden {
  visibility:hidden;
}
```

A visibility:hidden elrejtí az elemet, de az elem helye **megmarad**.

```
h1.hidden {
  display:none;
}
```

A display:none viszont teljesen **törli** a szövegfolyamból az elemet.

A block és az inline elemek

Egy blokkelem kitölti a rendelkezésre álló teljes szélességet, a következő elem már új sorban fog megjelenni.

Blokk elemekre példa:

- <h1>
- <p>
- <div>

<p>

Egy inline elem csak a számára szükséges helyet foglalja el, a következő elem azonos sorban jelenik meg, ha elfér. Az inline elemek nem reagálnak a width, height, margin, padding tulajdonságok beállításaira.

Inline elemekre példa:

-
- <a>

A **display** tulajdonsággal megcserélhetjük az elem láthatóságának tulajdonságát.

Köztes állapot az inline-block elem:

```
display: inline-block;
```

Az inline-block elemek sem töltik ki a rendelkezésre álló helyet, mint az inline elemek, viszont használhatók vele a width, height, margin és padding tulajdonságok.

```
<style>
span {
  display: inline-block;
  width: 200px;
}
</style>
```


Interaktív példa:

elem

block

Pozicionálás

A position lehetséges értékei:

- static (alapértelmezett)
- relative
- absolute
- fixed

Értékek

A static érték

Az alapértelmezett érték. A doboz a tartalom folyamában kerül elhelyezésre. Nem használhatók a left, right, top, bottom tulajdonságok.

A relative érték

A doboz a statichoz hasonlóan kerül elhelyezésre tartalom-folyamban, de a folyamhoz képest igazítható a helyzete. A következő elem ehhez képest fog igazodni.

Az absolute érték

A doboz a tartalom-folyamtól függetlenül kerül elhelyezésre. A következő elem nem veszi figyelembe ennek helyét.

A fixed érték

A doboz a tartalom-folyamtól függetlenül kerül elhelyezésre, hasonlóan az absolute értékhez. A következő elem nem veszi figyelembe ennek helyét. A különbség az absoluthoz képest, hogy a doboz rögzítésre kerül a képernyőhöz képest. A tartalom görgetéskor elgördül a dobozhoz képest.

```
div.osztalynev {
    position: absolute;
    top: 0;
    left: 0;
}
```

Statikus példa

A következő CSS fájlban beállítottam top és left tulajdonságot is, de azok hatástalanok, ha a position: static érték van beállítva.

static.css

```
div.doboz {
    position: static;
    background-color: rgba(255, 255, 0, 0.9);
    top: -20px;
    left: 40px;
    display: inline-block;
    width: 200px;
}
```

static.html

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" href="static.css">
</head>
<body>

  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
  </div>

  <div class="doboz">
    Relatív pozicionálás
  </div>

  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
  </div>

</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Láthatjuk, hogy a szövegdoboz a helyén maradt.

Relative példa

relative.css

```
div.doboz {
    position: relative;
    background-color: rgba(255, 255, 0, 0.9);
    top: -20px;
    left: 40px;
    display: inline-block;
    width: 200px;
}
```

relative.html

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" href="relative.html">
</head>
<body>

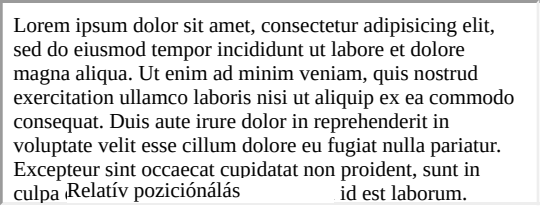
<div>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
</div>

<div class="doboz">
  Relatív pozicionálás
</div>

<div>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
</div>

</body>
</html>
```

A következő példa, beépített weblapon mutatja a relatívan pozicionált szöveget, amelyet kicsit fel, és jobbra eltolunk.



Látható, hogy megmaradt az elemek folyamatban a doboz helye, és saját magához képest lett eltolva fel és jobbra irányba.

Absolute példa

Az absolute megegyezik a relatív beállításokkal, kivéve, hogy honnan számítjuk a top vagy left értékeket. Az absolute beállításnál a top tulajdonságot a body elemhez képest számoljuk. A szöveget így kivettük a tartalomfolyamból.

absolute.css

```
div.doboz {
  position: absolute;
  background-color: rgba(255, 255, 0, 0.9);
  top: 20px;
  left: 40px;
  display: inline-block;
  width: 200px;
}
```

absolute.html

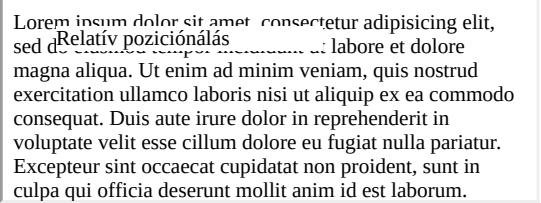
```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" href="absolute.css">
</head>
<body>

<div>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
</div>

<div class="doboz">
  Relatív pozicionálás
</div>

<div>
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
</div>

</body>
</html>
```



Görgetéskor jól látszik, hogy a szövegdoboz gördül a tartalommal együtt.

Fixed példa

A fixed megegyezik az absolute beállítással, kivéve, hogy az adott elem által létrehozott doboz a képernyőhöz lesz rögzítve, azaz görgetéskor nem mozdul.

fixed.css

```
div.doboz {
  position: fixed;
  background-color: rgba(255, 255, 0, 0.9);
  top: 20px;
  left: 40px;
```

```
display: inline-block;
width: 200px;
}
```

fixed.html

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" href="fixed.css">
</head>
<body>

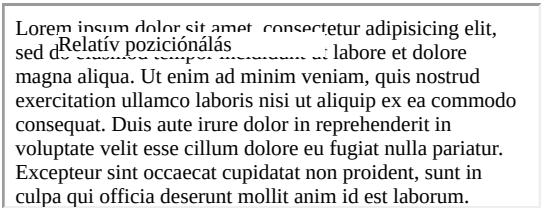
  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
  </div>

  <div class="doboz">
    Relatív pozicionálás
  </div>

  <div>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
  </div>

</body>
</html>
```

Figyeljük meg, hogy görgetéskor a szövegdoboz kezdeti pozícióban marad.



Pozicionálás példa 2

Középre igazított doboz láthatunk az alábbi példában.

kozepre.html

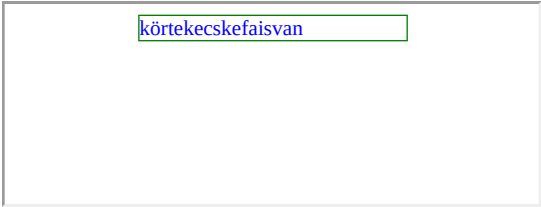
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title></title>
  <link href="kozepre.css" rel="stylesheet">
</head>
<body>

  <div id="doboz1">körtekecskefaisvan</div>

</body>
</html>
```

style.css

```
div#doboz1 {
  color: blue;
  border: 1px solid green;
  width: 50%;
  left: 25%;
  position: absolute;
  top: 5%;
}
```



Lebegtetés

A lebegtetés vagy másként úsztatás a float tulajdonsággal lehetséges.

A float

Egy elemet jobbra vagy balra tudunk lebegtetni, vagy másként úsztatni.

A float lehetséges értékei:

- left
- right
- none

```
div.osztalynev {
  float: right;
```

}

Lássunk egy példát:

lebegtetes.html

```
<!DOCTYPE html>
<html lang="hu">
<head>
  <meta charset="utf-8">
  <title></title>
  <link rel="stylesheet" href="lebegtetes.css">
</head>
<body>

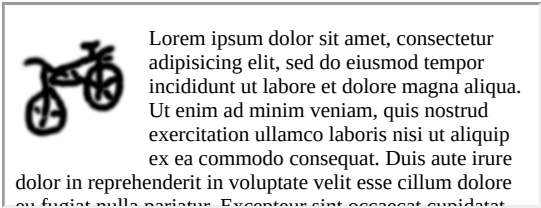
  <p>
    
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veni
  </p>

</body>
</html>
```

A CSS fájl:

lebegtetes.html

```
p img {
  float:left;
}
```



Ahogy láthatjuk a képet balra úsztatjuk a szövegben.

A clear

Az objektum nem úszthatató az előtte lévő objektumon hiába van beállítva float érték.

Lehetséges értékei:

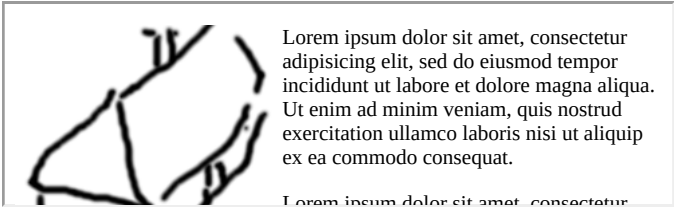
left	A balra úsztatás törlése
right	A jobbra úsztatás törlése
both	Egyik oldalra sem lehet úsztatni
none	Úsztatás mindkét oldalon megengedett

Adott két bekezdés. Az első bekezdésben van egy úsztatott kép. Az úsztatás miatt a második bekezdés is körbefolyik a kép körül.

```
<p>
  
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, qu
</p>
<p class="ketto">
  Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, qu
</p>

p img {
  float:left;
}
```

Az eredmény:

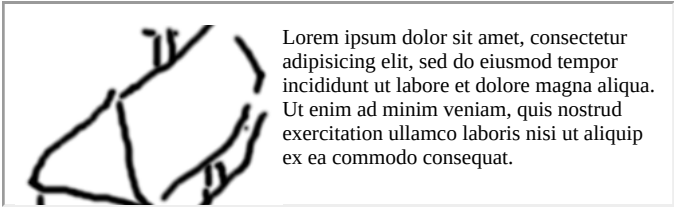


Most töröljük a második bekezdésről az úsztatást:

leveg2.css

```
p img {
  float: left;
}

.ketto {
  clear: left;
}
```



Blokk elemek igazítása

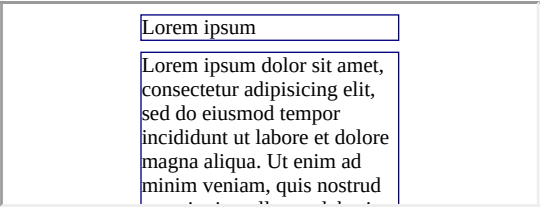
Blokk elemek:

- <h1>
- <p>
- <div>

A blokk elemek igazítását a jobb és bal margók megadásával végezhetjük el.

Példa a középre igazításra:

```
.kozepre
{
    margin-left:auto;
    margin-right:auto;
    width:70%;
    background-color:#b0e0e6;
}
```



A középre igazítást a beépített weblon is láthatjuk:

Jobbra igazítás

```
.jobbra
{
    position: absolute;
    right: 0;
    width: 300px;
    background-color: #b0e0e6;
}
```

Jobbra floattal:

```
.jobbra
{
    float:right;
    width:300px;
    background-color:#b0e0e6;
}
```

Vágás

A láthatóság határainak beállítása.

Egy képet adott formára vágunk. Jelenleg téglalap alakúra, bár jelenleg nem támogatott más forma:

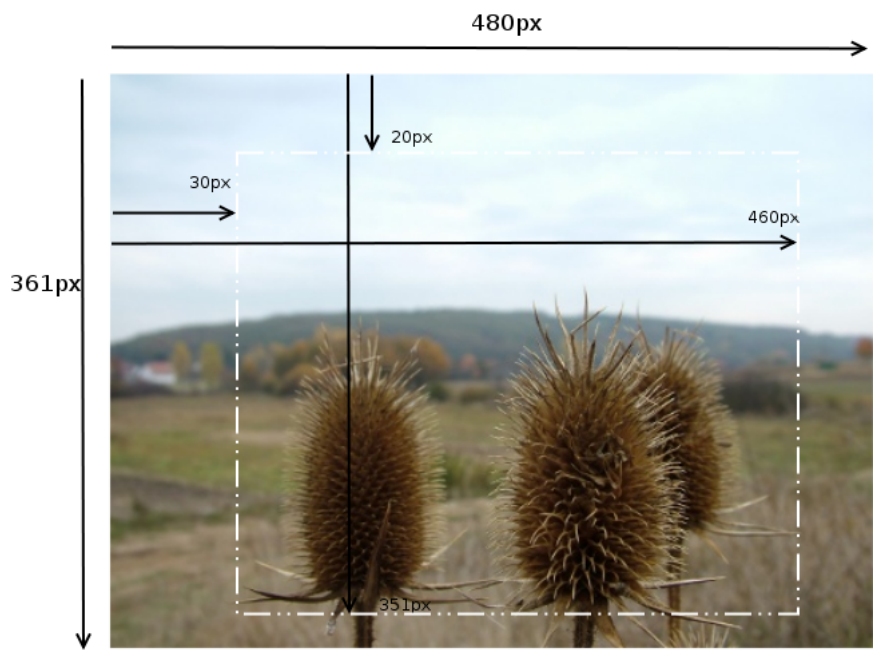
```
img {
    position: absolute;
    clip: rect(20px, 460px, 351px, 30px);
}
```

A vágást csak abszolút pozícionálású elemen lehet végrehajtani.

Tetejét, jobboldalát, alját és baloldalát adjuk meg.

480x361 kép

clip: rect(20px, 460px, 351px, 30px);



Ál-osztályok

Az ál-osztályokkal néhány elemhez speciális effektusokat rendelhetünk.

Az ál-osztályok szintaxisa

szelektor:ál-osztály {beállítás:érték;}

De használható így is:

szelektor.osztály:álosztály {beállítás:érték;}

Példa

```
a:link {color:#FF0000;} /* még nem látogatott linkek */
a:visited {color:#00FF00;} /* látogatott linkek */
a:hover {color:#FF00FF;} /* ha az egér felé megy */
a:active {color:#0000FF;} /* aktuális linkek */
```

- Megjegyzés
 - Az a:hover az a:link és az a:visited után kell következzen a CSS definícióban!
 - Az a:active az a:hover után kell következzen.
 - Az álosztályok nem kisbetű-nagybetű érzékenyek

Az összes ál-osztály

Szelektor	Példa	A példa leírása
:link	a:link	Meg nem látogatott linkek.
:visited	a:visited	Látogatott linkek.
:active	a:active	Aktív linkek.
:hover	a:hover	link amely felett az egér tartózkodik.
:focus	input:focus	Az input elem, ha fókuszba került.
:first-letter	p:first-letter	A p elem első betűje.
:first-line	p:first-line	A p elem első sora.
:first-child	p:first-child	A p elem közül az, amely a szülő első eleme.
:before	p:before	Minden p elem elé tartalmat szúr be.
:after	p:after	Minden p elem után tartalmat szúr be.
:lang(nyelv)	p:lang(hu)	Minden p elem a lang tulajdonsággal és hu értékkel lesz beállítva.

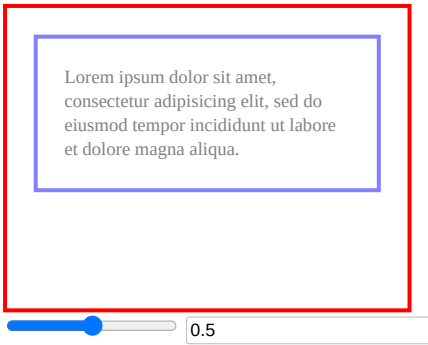
Átlátszatlanság

Az opacity tulajdonság

```
.doboz {
  opacity: 0.6;
}
```

```
.doboz {
  filter: opacity(0.6);
}

.doboz {
  height: 150px;
  width: 220px;
  background-color: #ffffff;
  opacity: 0.6;
}
```



Az rgba színek

Az rgba, az rgb kiterjesztése egy alfa csatomával, amely egy objektum átlátszóságát adja meg.

Egy rgba szín megadása a következő szintaktika szerint történik.

rgba(piros, zöld, kék, alfa)

Az alfa paraméter egy szám 0.0 (teljesen átlátszó) és 1.0 (teljesen átlátszatlan) között.

Példa:

```
.doboz {
  background-color: rgba(255, 0, 0, 0.3);
}
```

Táblázatok

Vegyünk egy táblázatot, amelyen szeretnénk beállításokat végezni.

```
<table>
  <tr>
    <th>Név</th>
    <th>Település</th>
  </tr>
  <tr>
    <td>Peres János</td>
    <td>Szeged</td>
  </tr>
  <tr>
    <td>Timp István</td>
    <td>Szeged</td>
  </tr>
  <tr>
    <td>Lazi Ferenc</td>
    <td>Miskolc</td>
  </tr>
</table>
```

Táblázat beállítások

A szegély beállítása a táblázaton is értelmezhető művelet. Ha viszont csak a table elemet jelölöm meg, akkor a cellák között nem lesz szegély.

```
table {
  border: 1px solid black;
}
```

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 1: Táblázat

A következő példában a fejléc és az adatcellák között is beállítom a szegélyt:

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

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 2: Táblázat szegélyezése cellákkal

Alapértelmezetten a cellák egymástól elszeparáltan kis távolságra jelennek meg. Ha szeretem az egybeolvasztásukat a border-collapse (kicsit lentebb többet is megtudhatunk erről) tulajdonságot kell beállítanunk:

```
table {
  border-collapse: collapse;
}

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

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 3: Cellák összeomlasztása

A border-collapse lehetséges értékei	
collapse	A szegély összeomlik egyetlen vonallá, amikor lehetséges. (border-spacing és a empty-cells beállítások nem működnek)
separate	A szegély különálló (border-spacing és empty-cells beállítások működnek). Alapértelmezett
inherit	A border-collapse tulajdonság a szülő elemtől öröklött.

A változatosság kedvéért a td cellák szélességét állítsuk 100px-re, színüket orange színre.

```
td {
  width: 100px;
  background-color: orange;
}
```

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 4: Színezett td elemek

Cella közötti, cella és tartalma közötti tér

Cellák közötti tér

Ha a cellák összeomlasztása helyett, szeretnénk növelni azok közötti teret, akkor használjuk a border-spacing tulajdonságot.

```
table {
  border-spacing: 20px;
}
```

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 5: A cellák közötti tér

A cella és tartalma közötti tér

```
td {
  padding: 30px;
}
```

A teljes CSS fájl jelenleg így néz ki:

```
table.css
.....
table {
  border-spacing: 20px;
}

table, th, td {
  border: 1px solid black;
}

td {
  width: 100px;
  background-color: orange;
  padding: 30px;
}
```

Név	Település
Peres János	Szeged
Timp István	Szeged
Lazi Ferenc	Miskolc

Figure 6: A cella és tartalma közötti távolság állítása

Cella tartalmának igazítása

```
td {
  text-align: center;
}
```

Z-index

```
<div id="egy" >z-index: 1</div>
<div id="ketto">z-index: 2</div>
<div id="harom">z-index: 3</div>

#egy {
  position: relative; left: 0px; top:0px; width:100px; height:100px;
  background: #f00;
  z-index:3;
}

#ketto {
  position: relative; left: 50px; top:-50px; width:100px; height:100px;
  background: #0f0;
  z-index:2;
}

#harom {
  position: relative; left: 100px; top:-100px; width:100px; height:100px;
  background: #00f;
  z-index:1;
}

z-index: 3
```

2

1

z-index: 1

z-index: 2

z-index: 3

Az utóbbi példában a dobozok z-indexét megfordítottuk:

```
#egy {
  position: relative; left: 0px; top:0px; width:100px; height:100px;
  background: #f00;
}
```

```
    z-index:1;
}

#ketto {
    position: relative; left: 50px; top:-50px; width:100px; height:100px;
    background: #0f0;
    z-index:2;
}

#harom {
    position: relative; left: 100px; top:-100px; width:100px; height:100px;
    background: #00f;
    z-index:3;
}
```

Mértékegységek

Relatív hosszérték	
em	A releváns karakter mérete
px	képpont; eszközfüggő
ex	A releváns karakter x-magassága
%	
ch	A „0” (zero) szélességhez relatívan
rem	A root elem fontszélességéhez relatívan
vw	A viewport szélességének 1%-hoz relatívan
vh	A viewport magasságának 1%-hoz relatívan
vmin	A viewport kisebb dimenziójához képest 1%-al
vmax	A viewport nagyobb dimenziójához képest 1%-al

A vmax értéket az Inernet Exploler a nem általános vm névvel támogatja, a Safari nem támogatja.

Abszolút hosszérték	
in	inch; hüvelyk; 25,4 mm
cm	centiméter
mm	milliméter
pt	pt; pont; 1/72 hüvelyk; kb.: 0,352 mm
pc	pica; 12 pont, 3/18 hüvelyk; kb.: 4,23 mm

Példák:

```
h1 {
    margin: 0.3in;
    line-height: 2cm;
    word-spacing: 5mm;
    font-size: 10pt;
}
```

Mikor ajánlott hasznli			
	Ajánlott	Alkalmanként	Nem ajánlott
Képernyő	px, em, %	ex	mm, cm, in, pt, pc
Nyomtatott	mm, cm, in, pt, pc, em, %	px, ex	

Újabb mértékegységek:

1vw	1%-a a viewport szélességének
1vh	1%-a a viewport magasságának
1vmin	1%-a a viewport rövidebbik oldalának
1vmax	1%-a a viewport hosszabbik oldalának

At-szabályok

```
@import "masik.css";
```

Az import sorok meg kell, hogy előzzék az összes szabályt.

Az import segítségével, más CSS fájlokat emelhetünk a jelenlegihez.

Függelék

Értékek

Egész és valós számok

Az egész számok 0-9 közötti számjegyek.

A valós számok lehetnek egész számok, 0 vagy több számjegy, amelyet egy pont, azt pedig újabb számjegyek követnek.

Az egész és a valós számokat megelőzhetik (-) vagy (+) karakterek.

Relatív hosszértékek

- em: a releváns karakter mérete
- ex: a releváns karakter x magassága
- px: pixel vagy képpont

Például:

```
p {
  font-size: 1em
}
```

Abszolút hosszérték

- cm: centiméter
- mm: milliméter
- in: inch - hüvelyk 25,4 mm
- pc: pica - 12 pont, vagyis 3/18 hüvelyk, kb. 4,23 mm
- pt: point - 1/72 hüvelyk, kb. 0,352 mm

Százalékérték

```
p {
  font-size: 120%;
}
```

Számlálók

Fejezetek számozása

```
<h2>A haladás előnye</h2>
<p>Lorem ipsum dolor est amet</p>
```

```
<h2>A haladás veszélyei</h2>
<p>Lorem ipsum dolor est amet</p>
```

```
<h2>Érintkezés</h2>
<p>Lorem ipsum dolor est amet</p>
```

```
<h2>Társalgás</h2>
<p>Lorem ipsum dolor est amet</p>
```

```
h2 {
  counter-increment: section;
}
h2:before {
  content:counter(section, upper-roman)". ";
}
```

I. A haladás előnye

Lorem ipsum dolor est amet

II. A haladás veszélyei

Lorem ipsum dolor est amet

III. Érintkezés

Lorem ipsum dolor est amet

IV. Társalgás

Lorem ipsum dolor est amet

Bekezdések számozása

```
p {
  counter-increment: par-num;
}
p:before {
  content:counter(par-num, upper-roman)". ";
}
```

Fejezetenként újraszámozva:

```
p {
  counter-increment: par-num;
}
h2 {
  counter-reset: par-num;
}
p:before {
  content:counter(par-num, upper-roman)". ";
}
```

Bekezdések számozása:

```
<p>Első</p>
<p>Második</p>

body {
  counter-reset: section;
}
p:before {
  counter-increment: section;
```

```
    content: counter(section) ". " ;
}
```

A haladás előnye

- I. Lorem ipsum dolor est amet
- II. Lorem ipsum dolor est amet
- III. Lorem ipsum dolor est amet

A haladás veszélyei

- I. Lorem ipsum dolor est amet
- II. Lorem ipsum dolor est amet
- III. Lorem ipsum dolor est amet

Átlátszó szegély

Szükséges hozzá valamilyen heterogén háttér. Például a GIMP-ben mintával kiöntött kis kép.

```
szelektor {
  border-radius: 5px;
  box-shadow:
    0 2px 2px rgba(0,0,0,0.2),
    0 1px 5px rgba(0,0,0,0.2),
    0 0 12px rgba(255,255,255,0.4);
}
```

Számított érték

A calc() egy CSS függvény, amely paraméterként a következőket várja: <length>, <frequency>, <angle>, <time>, <number> vagy <integer>.

A calc() segítségével egy CSS tulajdonság értéke számítható ki. A calc() függvényen belül lehet egy másik calc() függvény, vagyis egymásba ágyazhatók.

Szintaktika:

```
calc(kifejezés)
```

Használható operátorok:

+	Összeadás
-	Kivonás
*	Szorzás
/	Osztás

Használatára példa:

```
width: 90%; /* ha egy böngésző nem támogatja a calc() függvényt */
width: calc(100% - 80px);

width: 130px; /* ha egy böngésző nem támogatja a calc() függvényt */
width: calc(100% / 6);

width: 98%; /* ha egy böngésző nem támogatja a calc() függvényt */
width: calc(100% - 1em);
```

Konkrét példa:

index.html

```
<!doctype html>
<html>
<head>
<meta charset="utf-8" />
<title>0szlopok</title>
<style>
#oszlopdoboz {
  width: 200px;
  height: 200px;
}
.col {
  width: calc(100% / 3);
  width: -moz-calc(100% / 3);
  height: 200px;
  float:left;
}
#col1 {
  background-color: blue;
}
#col2 {
  background-color: yellow;
}
#col3 {
  background-color: #bbbbbb;
}

</style>
</head>
<body>

<div id="oszlopdoboz">
<div id="col1" class="col"></div>
<div id="col2" class="col"></div>
<div id="col3" class="col"></div>
</div>
```

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

Függőleges igazítás

Tekintsük át mit tehetünk egy fix méretű 100×100 pixeles dobozban, a függőleges igazításért.

Amit szoktak ajánlani az a display: table-cell és vertical-align: middle használata. Ez azonban nem minden böngészőben működik az elvártaknak megfelelően, hiszen a vertical-align eredetileg táblázat celláján belüli igazításra és szövegközi igazításokra tervezték.

A vertical-align használata

```
#doboz {
  background-color: blue;
  color: white;
  height: 100px;
  width: 100px;
  display: table-cell;
  vertical-align: middle;
  text-align: center;
  border-radius: 5px;
}
```

Ügyeljünk, hogy be legyen állítva a „display: table-cell;” tulajdonság.

```
<div id="doboz">
  alma
</div>
```



Működik: Chrome verzió: 31, Firefox 25.0.1

Belső 50%-os doboz

A következő példában a szöveget egy belső dobozba teszem és ennek a doboznak állítok 50%-os felső távolságot. Így persze nem lesz középen, de ha kivonom a felét, azaz 0.5em-et akkor majdnem a felénél lesz.

Ez persze csak akkor működik, ha szöveg egy soros. Ha már két sor van, akkor a 1em mértéket kell kivonni. Szükségünk lesz a calc() függvényre is, amelyet esetleg régebbi böngészők még nem támogatnak.

```
#kulso {
  background-color: blue;
  color: white;
  height: 100px;
  width: 100px;
  border-radius: 5px;
}
#doboz {
  background-color: blue;
  position: relative;
  top: calc(50% - 0.5em);
  text-align: center;
}
```

```
<div id="kulso">
  <div id="doboz">
    alma
  </div>
</div>
```

Működés:



Span elem 50%-al

Talán a legkorrektebb megoldás ha span elemként kezeljük a középre igazítandó szöveget, majd mellé teszünk egy másik span elemet, amelynek a magasságát 50%-ra állítjuk be. Pixelre még így sincs pont középen, ezért a középre igazítandó span elemen beállítom a vertical-align: middle értéket.

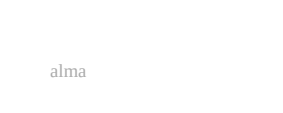
Kódolva:

```
#kulso {
  background-color: blue;
  color: white;
```



```
        height: 100px;
        width: 100px;
        border-radius: 5px;
        text-align: center;
    }
    #doboz {
        background-color: blue;
    }
    .hely {
        display: inline-block;
        height: 50%;
        width: 0;
        margin-right: -4px;
    }
<div id="kulso">
  <span class="hely"></span>
  <span id="doboz">alma</span>
</div>
```

Működés:



Ha vízszintesen is középre igazítunk mint a példában, a span elem extra margója miatt nem lesz pixelre pontosan középen. Ezen segíthetünk, ha margin-right: -4;-t állítunk be, vagy ha lehet a szülődoboz számára font-size:0; értéket állítunk.

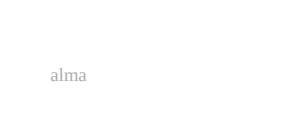
Transzformáció

```
#kulso {
    background-color: blue;
    color: white;
    height: 100px;
    width: 100px;
    border-radius: 5px;
    text-align: center;
    position: relative;
}
#doboz {
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
}
<div id="kulso">
  <div id="doboz">
    alma
  </div>
</div>
```



Flex doboz

```
<div class="doboz">
  almā
</div>
.doboz {
    border-radius: 5px;
    width: 100px;
    height: 100px;
    background-color: blue;
    color: white;
    display: flex;
    justify-content: center;
    align-items: center;
}
```



Rugalmas magasság esetén

Ha doboz magassága nem fix méretű, vagyis a tartalomtól függ, akkor használható a padding felül-alul beállítva.

```
.doboz {
    padding: 50px 0;
}
```

Sor magasság

Egy soros szöveg esetén a sor magasságát beállíthatjuk a doboz magasságára.

```
.doboz {
  line-height: 100px;
}
```

Szövegdobozok egymás mellett

Dobozok egymás mellett:

```
.dobozEgyDotted {
  color: #656565;
  height: 200px;
  width: 150px;
  display: table-cell;
  vertical-align: middle;
  text-align: justify;
  font-size: 11px;
  border: 1px dotted #656565;
  padding: 10px;
}

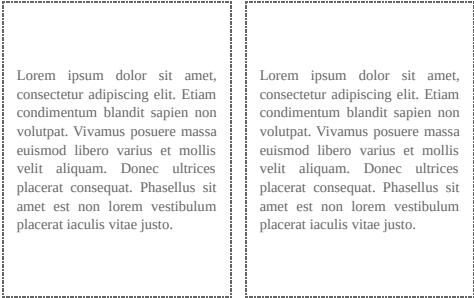
.dobozKetto {
  height: 200px;
  width: 10px;
  display: table-cell;
}

<div class="dobozEgyDotted">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam condimentum blandit sapien non volutpat.
  Vivamus posuere massa euismod libero varius et mollis velit aliquam. Donec ultrices placerat consequat.
  Phasellus sit amet est non lorem vestibulum placerat iaculis vitae justo.
</div>

<div class="dobozKetto"></div>

<div class="dobozEgyDotted">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam condimentum blandit sapien non volutpat.
  Vivamus posuere massa euismod libero varius et mollis velit aliquam. Donec ultrices placerat consequat.
  Phasellus sit amet est non lorem vestibulum placerat iaculis vitae justo.
</div>
```

Működés közben:



Extra margó

A li és a span elemek alapértelmezésként 2 pixeles extra margót kapnak. Ez kényelmetlen lehet, ha például ebből szeretnék menüt készíteni, és nem szeretnénk azokat az extra margókat.

Az extra margók akkor jelentkeznek a fenti elemeknél, ha azokat inline-block elemmé tesszük, és szeretnénk elkerülni a float:left beállítást. Mi tehetünk?

Negatív jobb margó

Az egyik lehetőség az igazítandó elemek számára egy -4 pixeles jobb-margót állítása.

```
span {
  background-color: navy;
  color: white;
  padding: 4px;
  margin-right: -4px;
}

<span>Egy</span>
<span>Kettő</span>
```

Egy Kettő

Azért nem a legjobb megoldás, mert a Firefox újabban 5px extra margót használ (2015).

jQuery megoldása

Eltávolíthatjuk egy jQuery kóddal:

```
<p class="removeTextNodes">
  <span> Foo </span>
  <span> Bar </span>
</p>

$('.removeTextNodes').contents().filter(function() {
  return this.nodeType === 3;
}).remove();
```

Szülő elem fontmérete

Hatékony megoldás, ha a szülőelem fontméretét 0 értékre állítom, a span vagy a li elem méretét pedig a kívánra. Alapértelmezett méret a böngészőkben 16px.

```
<div id="ez">
  <span>egy</span>
  <span>kettő</span>
  <span>három</span>
</div>
```

```
#ez {
  font-size: 0;
}
#ez span {
  font-size: 16px;
}
```

Flex doboz használata

Rugalmas méretű dobozokkal is megoldható.

```
<div id="amaz">
  <span>egy</span>
  <span>kettő</span>
  <span>három</span>
</div>

#amaz {
  display: flex;
}
```

Régebbi típusú böngészők esetén:

-webkit-box	-moz-box	-ms-flexbox	-webkit-flex
-------------	----------	-------------	--------------

```
display: flex;          /* Opera, Firefox, Chrome */
display: -moz-box;      /* Firefox */
display: -webkit-box;   /* iOS, Safari, Chrome */
display: -webkit-flex;  /* Későbbi Chrome */
display: -ms-flexbox;   /* IE 10 */
```

inline-block elemek közötti white-space

Ha van két inline-block elemünk, és azok két külön sorban vannak, közöttük lesz egy white-space karakter. Ha a két inline-block dobozt közvetlenül egymás mell szeretnénk tenni, ez nem várt hatást eredményez:

```
egy  kettő  három
```

Többféle megoldás létezik. Lehet float vagy flex dobozt használni. Itt most két olyan megoldást nézünk meg, ahol megmaradunk az inline-block tulajdonság mellett.

A szülő elem fontmérete

Az egyik megoldás lehet, ha a szülő elem fontméretét 0-ra állítjuk.

```
<div class="doboz">
  <div>egy</div>
  <div>kettő</div>
  <div>három</div>
</div>

.doboz {
  font-size: 0;
}
.doboz div {
  display: inline-block;
  background-color: navy;
  padding: 3px;
  color: white;
  font-size: 1rem;
}
```

```
egy kettő három
```

Megjegyzések beszúrása

Beszúrhatunk a white-space helyére üres megjegyzéseket.

```
<div class="doboz">
  <div>egy</div><!--
  --><div>kettő</div><!--
  --><div>három</div>
</div>

egy kettő három
```

Külső hivatkozások

- <http://www.w3.org/International/questions/qa-css-charset> [http://www.w3.org/International/questions/qa-css-charset]
- <http://caniuse.com> [http://caniuse.com] (Kompatibilitási összehasonlítások)
- <http://www.sitepoint.com/css-position-sticky-introduction-polyfills/> [http://www.sitepoint.com/css-position-sticky-introduction-polyfills/] (Pozicionálás sticky példa; csak Firefox)

