

Web-technológia

Cascading Style Sheets

Dr. Hatwágner F. Miklós

Széchenyi István Egyetem, Győr

https://github.com/wajzy/GKxB_INTM049.git

2020. március 3.

CSS: Cascading Style Sheets

- \approx lépcsőzetes/sorba kapcsolt stíluslapok
- *formázás, megjelenés* leírásának elválasztása a *tartalomtól* (HTML), előnyei:
 - külön fájlban tárolható, ami több weboldalhoz is használható, így csökken az összesített kódméret,
 - egységessé válik ezen oldalak megjelenése,
 - egymástól függetlenül, egyidejűleg lehet szerkeszteni a formát és a tartalmat,
 - gyorsabban módosítható a megjelenés, mert csak egy helyen kell változtatni,
 - hatékonyabbá válik a gyorstárazás,
- különféle médiára eltérő formázás lehetséges (pl. képernyő, nyomtatás)
- a CSS ereje
- hivatalos W3C oldal

Elavult módszer ([htmlFormazas.html](#))

```
1 <!DOCTYPE html>
2 <html lang="hu-HU">
3   <head>
4     <title>Elavult HTML formázás</title>
5     <meta charset="utf-8" />
6   </head>
7   <body>
8     <p><font size="+1"><i>Elavult HTML formázás</i></font></p>
9   </body>
10 </html>
```

```
<head>
  <title>CSS formázás</title>
  <link rel="stylesheet" type="text/css" href="cssFormazas.css">
  <meta charset="utf-8" />
</head>
<body>
  <p class="kiemelt">CSS formázás</p>
</body>
```

```
. kiemelt {
  font-size: larger;
  font-style: italic;
}
```

Szelektor (selector)

.kiemelt {
font-size: larger;
font-style: italic;
}

Érték (value)

Tulajdonság (property)

Deklaráció sablonja

```
szelektor {  
    tulajdonság1: érték(ek);  
    tulajdonság2: érték(ek);  
    ...  
    tulajdonságN: érték(ek);  
}
```

Szelektor

Mit akarunk formázni?

Tulajdonság

Milyen tulajdonságán változtassunk?

Érték

Milyen legyen az új állapot?

Megjegyzések a CSS-ben:

- `/* megjegyzes */`
- végleges kódból célszerű elhagyni
- Lehet több soros is

CSS ellenőrző

p { font-style: italic; }

p { font-style: italic; }

```
#lablec { font-size: 10pt; }
```

```
#lablec { font-size: 10pt; }
```

Az id nem kezdődhet számjegy karakterrel!

```
* { font-size: smaller; }
```

```
* { font-size: smaller; }
```

Osztály (class attribútum alapján)

```

.kisbetus { font-size: small; } /* bármilyen HTML elemhez */
.kisbetus { font-size: small; } /* bármilyen HTML elemhez, rövid
alak */

```

```
p.voros { color: red; } /* csak adott (pl. <p>) HTML elemhez */
```

A class értéke nem kezdődhet számjeggyel, de lehet egyszerre több, szóközzel elválasztott értéke:

Apróbetűs piros bekezdés

Elemek csoportosítása

h1, h2, h3 { font-family: Arial; }

egyszeruSzelektor1.html

14 `<p class="bevezeto">A magyarországi informatika kezdete a első`
`→ elektronikus számítógép elkészültétől eredeztethető. A szobányi m`
`→ éretű M-3-at szovjet tervek alapján építették meg, a magyar`
`→ szakembergárda pedig ezen kísérletezte ki, hogy miként lehet egy`
`→ számítógépet a népgazdaság érdekeit szolgáló mérnöki-, és gazdasá`
`→ gi számításokra felhasználni – például az Erzsébet-híd statikai`
`→ számításait ellenőrizték vele.</p>`

15 `<p>A mintegy kétévnyi építést követően hivatalosan 1959. január 21-én á`
`→ tadott <a href="https://itf.njszt.hu/termek_hardware/m-3-`
`→ szamitogep">M-3 számítógépre a <a href="https://ajovomultja.`
`→ hu/news/60-eves-az-elso-magyar-szamitogep">korabeli sajtóbeszámol`
`→ óban még számológépként hivatkoztak, mert a számítógép szó ké`
`→ söbb honosodott meg a magyar nyelvben.</p>`

egyszeruSzelektor1.css

```

1  h1, h2, h3, h4, h5, h6 {
2      font-family: Arial, Sans-Serif;
3  }
4  p { font-family: Garamond, "Times New Roman", Serif; }
5  a { color: navy; }
6  .forras {font-style: italic; }
7  .bevezeto { font-weight: bold; }

```

Ma van a magyar informatika születésnapja

Forrás: [NJSZT](#), 2020. január 21.

A magyarországi informatika kezdete a első elektronikus számítógép elkészültétől eredeztethető. A szabányi méretű M-3-at szovjet tervek alapján építették meg, a magyar szakembergárda pedig ezen kísérletezte ki, hogy miként lehet egy számítógépet a népgazdaság érdekeit szolgáló mérnöki-, és gazdasági számításokra felhasználni - például az Erzsébet-híd statikai számításait ellenőrizték vele.

A mintegy kétévnnyi építést követően hivatalosan 1959. január 21-én átadott [M-3 számítógépre](#) a [korabeli sajtóbeszámolóban](#) még számológépként hivatkoztak, mert a számítógép szó később honosodott meg a magyar nyelvben.

Bár ún. nulladik generációs számítógépet már az M-3 előtt néhány évvel is építettek a Budapesti Műszaki Egyetemen, Kozma László MESz-1 nevű gépe még telefonközpontokban használatos elektromágneses reléből épült fel, az M-3-mal ellentétben még nem felelt meg a Neumann-elveknek, amelyek előírják, hogy egy számítógépnek teljes egészében elektronikus módon kell működnie, és egy belső tárban kell tárolnia a programutasításokat.

A Magyar Tudományos Akadémia Kibernetikai Kutatócsoportjában megépített, első generációs M-3-ast tudományos számítások embernél gyorsabb végrehajtására tervezték. Noha a gép csupán másodpercenként 30 számítási műveletet tudott elvégezni (míg a mai gépek processzorai esetében már több tízezer millió utasítással számolhatunk másodpercenként), az M-3-ra írt programok révén így is számos gyakorlati feladatra használták fel. E gépen készült az épülő Erzsébet híd statikai számításaival kapcsolatos program, de például szállítási költségek optimalizálásához, részecskék vizsgálatához szükséges számításokat is végeztek vele.

- 1 Külső fájlban (css kiterjesztés, `<link>` elem)
- 2 A `<head>` elembe ágyazott `<style>` elemben. Csak akkor ajánlott, ha egyetlen HTML fájlt kívánunk formázni ezekkel a stílusokkal.
- 3 Soron belül: a HTML elemek `style` attribútumának értékeként. Ismét **keveredik a tartalom a stílussal**, ezért általában **nem ajánlott** a használata!

egyszeruSzelektor2.html

```
3  <head>
4    <title>Egyszerű szelektorok</title>
5    <meta charset="utf-8" />
6    <style>
7      p { font-family: Garamond, "Times New Roman", Serif; }
8      a { color: navy; }
9      .forras {font-style: italic; }
10     .bevezeto { font-weight: bold; }
11   </style>
12 </head>

16     <h1 style="font-family: Arial, Sans-Serif;">Ma van a magyar
    ↪ informatika születésnapja</h1>
```

Ha több előírás is vonatkozik ugyanannak az objektumnak a formázására, elsőként a forrás prioritása dönt (csökkenő sorrendben):

- 1 soron belüli formázások
- 2 külső és belső (<link>, <style> elemek) formázások
- 3 böngésző alapértelmezése

Azonos prioritás (pl. két külső stíluslap) esetén a később betöltött szabály felülírja a korábbi.

utkozes1.html

```
6      <style>
7          p { color: red; }
8      </style>
9      <link rel="stylesheet" type="text/css" href="utkozes1.css" />
10 </head>
11 <body>
12     <p>Első bekezdés.</p>
13     <p style="color: green;">Második bekezdés</p>
```

utkozes1.css

```
1 p { color: blue; }
```

Első bekezdés.

Második bekezdés

utkozes2.html

```
6      <link rel="stylesheet" type="text/css" href="utkozes1.css" />
7      <style>
8          p { color: red; }
9          p { color: orange; }
10     </style>
11 </head>
12 <body>
13     <p>Első bekezdés.</p>
```

utkozes1.css

```
1 p { color: blue; }
```

Első bekezdés.

Számtalan dolognak beállítható a színe CSS tulajdonságokkal, pl.:

`color`

Szöveg írásszíne

`background-color`

Háttérszín

Szín, mint a tulajdonság értéke megadható:

`kulcsszavakkal`

Pl. red (vörös), green (zöld), blue (kék), white (fehér), black (fekete), ...

140 szabványos színkód

`Hexadecimálisan, RGB összetevőkkel`

Pl. narancsszín: #ff7f00, ahol # jelzi a 16-os számrendszerbeli alakot, ff a vörös (Red), 7f a zöld (Green) és 00 a kék (Blue) összetevő intenzitása 8 biten előjel nélkül, fixpontosan. Additív színkeverés.

rgb() függvény

`rgb(red, green, blue)`, ahol mindhárom összetevő lehet 0-255 közötti decimális egész, vagy 0-100%. Pl. `rgb(255,0,0)` vagy `rgb(100%, 0%, 0%)` vörös színt eredményez.

rgba() függvény

`rgb(red, green, blue, alpha)`, ahol a színösszetevőket egy átlátszóság érték követi ([0, 1]).



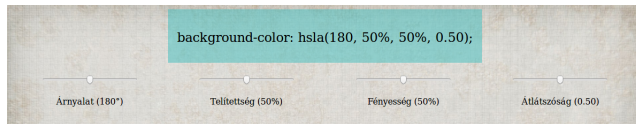
[szinek1.html](#)

hsl() függvénnyel

`hsl(hue, saturation, lightness)`, ahol `hue` az árnyalat, $[0, 360]$ fok közötti elfordulás a színceréken. Pl. 0° a vöröshöz, 120° a zöldhöz, 240° a kékhez tartozik. `saturation` a telítettség, százalékban. A 0% a színinformáció hiányát (szürkeség) jelzi, 100% a teljes színezettséget. `lightness` a fényesség, szintén százalékban. A 0% mindig fekete, a 100% mindig fehér színt ad.

hsla() függvénnyel

A fentiek kiegészülnek átlátszósággal.



szinek2.html

Induljon ki a **szinezes.html** fájlból!
Kapcsolja ezt össze egy külső stíluslappal,
majd érje el, hogy a jobb oldali ábrának
megfelelő színekben pompázzon! Próbáljon
minél több féle szín megadási módszert
alkalmazni! Törekedjen a lehető
legtömörebb CSS szabályok megalkotására!

szinezes-mo.html, szinezes-mo.css

Az első szintű címsorok legyenek barnák!

Ez a bekezdés legyen vörös!

Ez pedig zöld, de az utolsó szava legyen viágoszöld!

Az első szintű címsorok legyenek barnák!

A második és harmadik szintű címsorok legyenek kékek!

Ez egy szürke stílusú bekezdés.

Ez itt egy harmadik szintű címsor.

Ez is szürke.

Ennek meg a háttére legyen citromsárga!

HTML elemek háttérével kapcsolatos tulajdonságok:

`background-color`

A háttér színe. Alapértelmezetten `transparent`, azaz átlátszó.

`background-image`

Háttérkép, amivel alapértelmezés szerint kicsempézi az elem teljes területét (margókat nem). Alapértéke `none`, nincs háttérkép. Az `url()` függvény paramétereként adható meg a képfájl, pl.

```
background-color: url("hatter.png");
```

Megadhatók [színátmenetek](#) is.

A szöveg maradjon **olvasható** a háttéren!

background-repeat

Háttérkép csempézési iránya

- repeat mindkét irányban, túlnyúló részek levágásával, alapértelmezés
- repeat-x csak vízszintesen
- repeat-y csak függőlegesen
- no-repeat csak egyszer, alapértelmezetten a bal felső sarokban
- round torzítja a képet a vágás elkerülésére
- space csak annyiszor ismételi, ami vágás nélkül elfér, közöttük helyet hagy

Két érték megadásakor az első a vízszintes, második a függőleges irányra vonatkozik.



background-repeat:

- ☒ Mindkét irányban csempéz (repeat)
- ☐ Vízszintesen csempéz (repeat-x)
- ☐ Függőlegesen csempéz (repeat-y)
- ☐ Csak egyszer jelenik meg (no-repeat)
- ☐ Elosztja térközökkel (space)
- ☐ Torzít (round)

hatter.html

background-position

Igazítás, a *vízszintes* és a *függőleges* pozíciót várja. Ha egyet kap, a másik center lesz.

- Függőlegesen: left, center, right
- Vízszintesen: top, center, bottom
- Mindkettőnél lehet százalékot, vagy egyéb CSS mértékegységet (pl. képpont) használni.

pozicio1.html

```
7      p {
8          background-image: url("css3.svg");
9          background-size: 10%;
10         background-repeat: no-repeat;
11     }

15     <h1>Balra fent</h1>
16     <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.

24     <h1>Jobbra lent</h1>
25     <p style="background-position: right bottom">Aenean ut suscipit

37     <h1>Középen</h1>
38     <p style="background-position: center">Ut semper, ante pretium

53     <h1>Jobb alsó saroktól 10%-ra</h1>
54     <p style="background-position: 90% 90%">Lorem ipsum dolor sit
```

background-attachment

- `scroll` a háttér együtt gördül az oldallal, alapértelmezés
- `fixed` rögzített háttér
- `local` az elem tartalmával együtt gördül a háttér

A logo mindig a jobb alsó sarokban: [rogzites1.html](#)

Két bekezdés között kilátszik a háttérben rögzített logo: [rogzites2.html](#)

background

Rövidítés: egy összetett tulajdonsággal sok egyszerű tulajdonság értéke állítható be.

Értékek sorrendje rögzített, de tetszőleges számú érték elhagyható!

`background: background-color background-image background-repeat
background-attachment background-position`

pozicio1.html

```
p {  
  background-image: url("css3.svg");  
  background-size: 10%;  
  background-repeat: no-repeat;  
}
```

pozicio2.html

```
p {  
  background: url("css3.svg") no-repeat;  
  background-size: 10%;  
}
```

background-size

- auto: Alapértelmezés, eredeti méret.
- *szélesség, magasság*: utóbbi elhagyásával auto-t feltételez. Használhatók CSS mértékegységek és százalékok (**a szülő elem mérete a 100%**, nem a sajátja!).
- cover Addig nyújt és vág, amíg le nem fedi a szülő elem teljes területét.
- contain Addig nyújt, amíg egyszer bele nem fér a háttér a szülő elembe.



background-size:

- ☐ Eredeti méret (auto)
 - ☐ Teljes lefedés (cover)
 - ☒ Maximális méret vágás nélkül (contain)
- 100%

[meret.html](#)

Induljon ki a [rogzites2.html](#) fájlból, és alakítsa át a jobb oldali ábrának megfelelően!

- Az írásszín legyen világos szürke!
- A teljes oldal háttere legyen kék (RGB-összetevők: 0, 145 és 190)!
- A `<div>` elem háttereként állítsa be a [HTML5sticker.png](#) fájlt!
- Ennek helyzete ne függjön a görgetéstől!
- Helyezze el azt a képernyő közepén!
- A képet méretezze aránytartó módon úgy, hogy éppen kitöltse a rendelkezésre álló helyet!
- Próbálja mindezt a lehető legkevesebb CSS tulajdonság felhasználásával elérni!

rogzites3-mo.html, HTML5sticker.png

volutpat. Praesent congue, augue ultricies placerat auctor, neque libero blandit dolor, eu suscipit ante quam ac eros. Vivamus id massa mattis, fermentum tortor in, dignissim ex. Morbi faucibus, sapien non maximus hendrerit, enim lectus suscipit ex, sed volutpat erat odio eget tortor. Curabitur eu ullamcorper dolor. Phasellus vehicula sit amet urna finibus tempor. Nulla in sodales magna.

I've seen the
FUTURE
It's in my
BROWSER



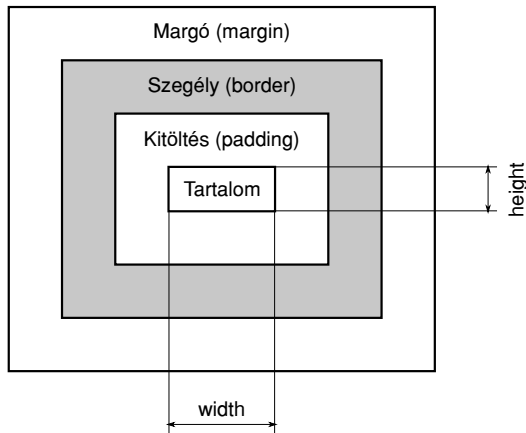
Ut semper, ante pretium aliquam commodo, libero turpis gravida metus, eu semper justo quam at risus. Maecenas vitae libero massa. Proin sodales erat ex, sit amet consequat massa venenatis sit amet. Sed et luctus tortor, eu vestibulum libero. Curabitur euismod nisi sem, vel elementum nibh posuere id. Pellentesque tincidunt sed dui non viverra. Sed suscipit mollis lobortis. Vestibulum facilisis volutpat justo in feugiat. Pellentesque sit amet

Minden HTML elemet egy *doboznak* tekintünk. Ezek szerkezete belülről kifelé:

- Az elem tartalma (szöveg, kép, ...)
- Kitöltés (padding; átlátszó)
- Szegély (border)
- Margó (margin; átlátszó)

Megjegyzések

- A szélesség (width) és magasság (height) tulajdonságok a tartalmi rész méreteire vonatkoznak.
- Soron belüli elemek méretét a böngésző határozza meg, nem méretezhetők át.



dobozMeret.html

```

7  body {
8      margin: 0px;
9  }
10 p {
11     background-color: yellow;
12     padding: 20px;
13     border: 5px solid red;
14     margin: 10px;
15 }
16 span {
17     background-color: orange;
18 }

```

Ez a bekezdés a szülő (body) szélességéhez igazodik, magassága a tartalom függvénye.

Ez egy 200x100 képpont méretű bekezdés.

A bekezdésben lévő **soron belüli elemek** mérete nem módosítható!

Kivéve, ha a megjelenítés módját blokkszintűre módosítjuk.

```

22 <p>Ez a bekezdés a szülő (<code>body</code>) szélességéhez igazodik , magassága a tartalom
    ↳ függvénye.</p>
23 <p style="width: 200px; height: 100px;">Ez egy 200x100 képpont méretű bekezdés.</p>
24 <p>A bekezdésben lévő <span style="width: 120px; height: 50px;">soron belüli elemek</span>
    ↳ mérete nem módosítható!</p>
25 <p>Kivéve, ha a <span style="display: block; width: 120px; height: 50px;">megjelenítés mó
    ↳ dját</span> blokkszintűre módosítjuk.</p>

```

Mit számol bele a böngésző a méret
(width, height) adatokba? →
box-sizing

content-box

Csak a tartalom méretét

border-box

Tartalom + kitöltés + szegély

Kényelmes:

```
* { box-sizing: border-box; }
```

Név:

Pl. Gipsz Jakab

Végzettség:

középiskola/gimnázium ▾

meretezes.html

```
11     input[type=text] {  
12         width: 200px;  
13         box-sizing: border-box;  
14     }  
15     select {  
16         width: 200px;  
17     }
```


Blokk szintű elemek szélessége (`width`) és magassága (`height`) megadható:

- `auto`: alapértelmezett
- valós világbeli, relatív vagy megjelenítőtől függő mértékegység (pl. `cm`, `ex`, `px`)
- a tartalmazó blokk %-ában megadva
- `initial`: az alapértelmezett értéket használja az örökölt helyett
- `inherit`: a szülőtől örökölt értéket használja

A merev méretek helyett megadhatók intervallumok is:

`max-width`

Ennél csak keskenyebb lehet. Felülírja `width` értékét.

`min-width`

Ennél csak szélesebb lehet. Ha a tartalom ennél szélesebb, nem veszik figyelembe. A szélesség változásával a magasság is változhat.

`max-height`

Ennél csak alacsonyabb lehet. Ha a tartalom ennél magasabb, a viselkedés az overflow-tól függ. Felülírja `height` értékét.

`min-height`

Ennél csak magasabb lehet. Ha a tartalom ennél alacsonyabb, akkor ekkorra növeli meg a magasságot.

Túlcsoordulások kezelése: overflow

visible

A túllógó részek is megjelennek, esetleg rálógva más tartalmakra. Alapértelmezés.

hidden

A túllógó részeket levágják.

scroll

Görgetősávok jelennek meg a túllógó részek megjelenítéséhez. Némelyik böngésző mindig mutatja, mások csak akkor, ha szükséges.

auto

Csak akkor jelennek meg görgetősávok, ha nem fér el a tartalom.

Léteznek overflow-x és overflow-y tulajdonságok csak az egyik irány viselkedésének megadásához.

tulnyulas.html – Ellenőrizze a méretezés hatását, túlcsordulásokat!

```

11 <p style="width: 500px; height: 105px;"><mark>Szélesség és
12 magasság</mark>A Lorem Ipsum egy egyszerű szövegrészlete ,

20 legutóbb softwarekkel mint például az Aldus Pagemaker.</p>
21 <p style="min-width: 400px; max-width: 750px;"><mark>Minimális és
22 maximális szélesség</mark>Ez egy régóta

```

Szélesség és magasság A Lorem Ipsum egy egyszerű szövegrészlete, szövegutánpótló betűszedő és nyomdaiparnak. A Lorem Ipsum az 1500-as évek óta standard szövegrészletként szolgált az iparban; mikor egy ismeretlen nyomdász összeállította a betűkészletét és egy példa-könyvet vagy szöveget nyomott papírra, ezt használta.

Minimális és maximális szélesség Ez egy régóta elfogadott tény, miszerint egy olvasót az a szöveg érdekli, amely a legkevesebb betűt tartalmaz. A Lorem Ipsum egyben nem tartalmaz semmilyen betűt, amely a legkevesebb betűt tartalmaz, ellentétben a ma használt szövegekkel, amelyek a legkevesebb betűt tartalmaz. Sok esetben a Lorem Ipsumot mint alapbeállítást szövegmodellként, és egy keresés a lorem ipsum-ra sok félkész weboldalt fog eredményezni.

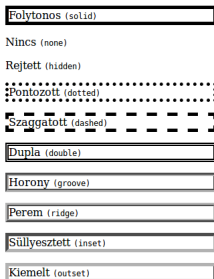
Az első bekezdés tartalma rálóg a másodikra, túl alacsony a blokk.
A második bekezdés maximális méreten. Ha keskenyre állítjuk az ablakot, vízszintes görgetősáv jelenik meg a böngészőablak alján.

szegelyek1.html

```

14 <p style="border-style: _solid">Folytonos <code>(solid)</code></p>
15 <p style="border-style: _none">Nincs <code>(none)</code></p>
16 <p style="border-style: _hidden">Rejtett <code>(hidden)</code></p>
17 <p style="border-style: _dotted">Pontozott <code>(dotted)</code></p>

```



Oldalankénti szegélystílusok megadhatók:

- 1-4 érték megadásával, pl.
border-style: dotted dashed solid none;
- Oldalakra vonatkozó tulajdonságokkal:
border-*-style, ahol * helyén állhat top, right, bottom, left.

1 értéke van

felül-jobb-alul-bal (minden oldalra
ugyanazt a stílust állítja)

2 értéke van

felül-alul jobb-bal

3 értéke van

felül jobb-bal alul

4 értéke van

felül jobb alul bal (óramutató járása
szerint)

Hasonlóképpen lehet oldalanként szabályozni a margókat és kitöltéseket is.

border-style: solid

border-style: solid none

```
border-style: double none solid
```

```
border-style: dotted dashed double solid
```

border-left-style: solid

```
border-top-style: dotted
```

szegelyek2.html

Ha táblázatok szomszédos cellái közös, de eltérő stílusú szegélyeket használnak, akkor

`none`

ha a szomszédnak be van állítva a szegélye, az fog megjelenni

`hidden`

még ha be is van állítva a szomszéd szegélye, akkor sem fog megjelenni

A1	B1
A2	B2

szegelyek3.html

```

21      <td style="border-style: none"
      ↪  ">A1</td>

26      <td style="border-style: none"
      ↪  B2</td>
  
```


M: 1 111 1011101 11 11 10101 11 101 101

[illegible]

A margók mindig átlátszók, csak a szélességük állítható:

- 1-4 érték megadásával, pl.

```
margin: 10px 20px 30px 40px;
```

(Fent, jobbra, lent, balra; további esetek mint `border-style`-nál.)

- Oldalakra vonatkozó tulajdonságokkal:

`margin-*`, ahol `*` helyén állhat `top`, `right`, `bottom`, `left`.

A margó szélessége lehet:

- `auto`: a tartalom által fel nem használt helyet felosztja egyenlően a bal és jobb oldal közt → középre igazít
- `inherit`: a befoglaló, szülő elem beállításait örökli
- CSS mértékegységgel (pl. `px`, `cm`) adott
- `%`: a szülő elem méretének százaléka

Negatív értékek is használhatók.

A blokkok felső és alsó margói időnként összeolvadnak, és a kettő közül csak a nagyobb marad meg:

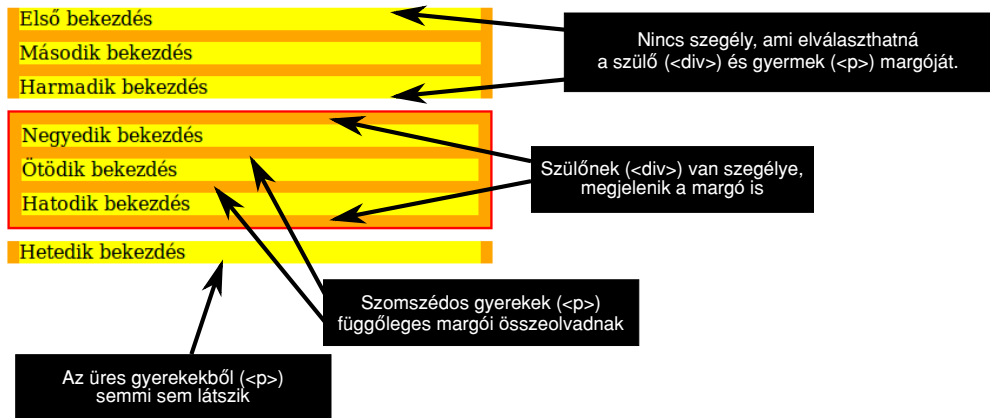
- szülő szomszédos gyerekei között (szélső gyerekek margói túlnyúlnak a szülőn)
- ha nincs olyan megjeleníthető szegély, kitöltés, stb., ami elválasztaná a szülő és valamely gyerekének alsó/felső margóját
- üres blokkok alsó és felső margóját is összevonják

További részletek

margok.html

```
7      div {  
8          background-color: orange;  
9      }  
10     div.keretes {  
11         border: 2px solid red;  
12     }  
13     p {  
14         background-color: yellow;  
15         margin: 10px;  
16     }
```

```
20  
21     <p>Első bekezdés</p>  
22     <p>Második bekezdés</p>  
23     <p>Harmadik bekezdés</p>  
24 </div>  
25 <div class="keretes">  
26     <p>Negyedik bekezdés</p>  
27     <p>Ötödik bekezdés</p>  
28     <p>Hatodik bekezdés</p>  
29 </div>  
30 <div>  
31     <p>Hetedik bekezdés</p>  
32     <p></p>  
33     <p></p>  
34 </div>
```



A kitöltések mindig átlátszók, csak a szélességük állítható:

- 1-4 érték megadásával, pl.
`padding: 10px 20px 30px 40px;`
(Fent, jobbra, lent, balra; további esetek mint `border-style`-nál.)
- Oldalakra vonatkozó tulajdonságokkal:
`padding-*`, ahol `*` helyén állhat `top`, `right`, `bottom`, `left`.

A kitöltés szélessége lehet:

- `inherit`: a befoglaló, szülő elem beállításait örökli
- CSS mértékegységgel (pl. `px`, `cm`) adott
- `%`: a szülő elem méretének százaléka

Negatív értékek **nem** használhatók.

Próbálja meg
elkészíteni az ábrának
megfelelően a
dobozokat!

Ezt a részt azzal emeljük ki a szövegből, hogy 15 képpont széles kitöltése van a bal oldalon, mellette egy ugyanolyan széles, narancssárga, egyszínű szegély található.

A héttérszín szürkéjének színösszetevői 240 értékűek.

Fekete alapon fehérrel írva, körben 5 képpontnyi kitöltéssel valami nagyon súlyos aggodalomra okot adó tényt közlünk.

A fejléc írásszín komponensei: 237, 234, 142. A háttérszín összetevők: 173, 116, 36, kitöltés körben 5 képpont, felső margó 10 képpont, a többi oldalon nincs. A szegély körben 1 képpont széles, folytonos, 107, 89, 65 színösszetevőjű vonal.

Az elemek háttérszín-összetevői: 242, 201, 143. A kitöltés körben 5 képpont, margók nincsenek.

A szegélyek itt is ugyanolyan szélességűek és színűek, mint a fejlécben, de csak a két oldalon és alul vannak megrajzolva.

dobozok.html

Körvonal (outline): az elemet a szegélyen kívülről körülöleli, kiemeli környezetéből. Rálóghat más elemekre.

outline-style

Stílus, mint border-style, pl. solid, dotted, double, ...

A többi tulajdonság beállítása **hatástalan** a stílus megadása nélkül!

outline-color Körvonal színe. Értéke lehet invert, ami minden háttéren látható.

outline-width Szélesség CSS mértékegységekben, vagy thin, medium, thick.

Rövidítés:

```
outline: outline-width outline-style outline-color
```

Sorrend tetszőleges, bármelyik érték elhagyható.

```
outline-offset
```

A körvonal távolsága a szegélytől. Ez a terület áttetsző.

korvonal.html

```

7      p {
8          background-color: antiquewhite;
9          border: 1px solid black;
10         width: 90%;
11         margin: auto;
12     }

20     leírására is, mint például az SVG, XUL stb.</p>
21     <p style="outline: 15px solid red; outline-offset: 5px;">A CSS
22     specifikációját a World Wide Web Consortium felügyeli.</p>
23     <p>A CSS-t a weblapok szerkesztői és olvasói egyaránt

```

A CSS (Cascading Style Sheets, magyarul: lépcsőzetes stíluslapok) a számítástechnikában egy stílusleíró nyelv, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezenkívül használható bármilyen XML alapú dokumentum stílusának leírására is,

A CSS specifikációját a World Wide Web Consortium felügyeli.

megjelenéshez kapcsolódó elemeit. A tervezése során a legfontosabb szempont az volt, hogy elkülönítsék a dokumentumok struktúráját (melyet HTML vagy egy hasonló leíró nyelvben lehet megadni) a dokumentum megjelenésétől (melyet CSS-sel lehet megadni). Az ilyen elkülönítésnek több haszna is van, egyrészt növeli a weblapok használhatóságát, rugalmasságát és a megjelenés kezelhetőségét, másrészt csökkenti a dokumentum tartalmi struktúrájának komplexitását. A CSS ugyancsak alkalmas arra, hogy a dokumentum stílusát a megjelenítési módszer függvényében adja meg, így elkülöníthető a dokumentum formája a képernyőn, nyomtatási lapon, hangos böngészőben (mely beszéd szintetizátor segítségével olvassa fel a weblapok szövegét), vagy Braille-készüléken megjelenítve.

text-align

Vízszintes igazítás: left (balra), center (középre), right (jobbra), justify (sorkizárt)

vizzszintes.html

```

7      .balra { text-align: left }
8      .kozepre { text-align: center }
9      .jobbra { text-align: right }
10     .sorkizart { text-align: justify }

15     <p class="balra">
16     A HTML (angolul: HyperText Markup Language=hipersz
      ↪ öveges jelölőnyelv) egy leíró nyelv ,
      ↪ melyet weboldalak készítéséhez
      ↪ fejlesztettek ki, és mára már internetes
      ↪ szabvánnyá vált a W3C (World Wide Web
      ↪ Consortium) támogatásával.

```

Balra igazított szöveg

A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

Középre igazított szöveg

A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

Jobbra igazított szöveg

A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

Sorkizárt szöveg

A HTML (angolul: HyperText Markup Language=hiperszöveges jelölőnyelv) egy leíró nyelv, melyet weboldalak készítéséhez fejlesztettek ki, és mára már internetes szabvánnyá vált a W3C (World Wide Web Consortium) támogatásával.

none

manual

auto

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

elvalasztas.html

```

8      <p>Nincs elválasztás. Nincs elválasztás. Nincs elválasztás.
12     <p style="hyphens: manual;">El&hyphen;vá&hyphen; lasz&hyphen;tás
13     kézzel megjelölt he&shy;lye&shy;ken. El&hyphen;vá &hyphen; lasz
16     <p style="hyphens: auto;">Automatikus elválasztás. Automatikus

```

Nincs elválasztás. Nincs elválasztás. Nincs elválasztás. Nincs elválasztás. Nincs elválasztás. Nincs elválasztás.
Nincs elválasztás. Nincs elválasztás. Nincs elválasztás. Nincs elválasztás.

El-vá-lasz-tás kézzel megjelölt helyeken. El-vá-lasz-tás kézzel megjelölt helyeken. El-vá-lasz-tás kézzel megjelölt helyeken.

Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás. Automatikus elválasztás.

vertical-align: tetszőleges elem függőleges igazítása

baseline

szülő szövegének alapvonalához

távolság

tetszőleges mértékű süllyesztéshez/emeléshez, negatív érték is elfogadott

%

sormagasság %-ában megadott emelés/süllyesztés, negatív érték is elfogadott

sub

szülő alsó indexéhez

super

szülő felső indexéhez

`top`

sor legmagasabb eleméhez

`text-top`

szülő elem szövegének tetejéhez

`middle`

szülő közepéhez

`bottom`

sor legalsó eleméhez

`text-bottom`

szülő szövegének aljához

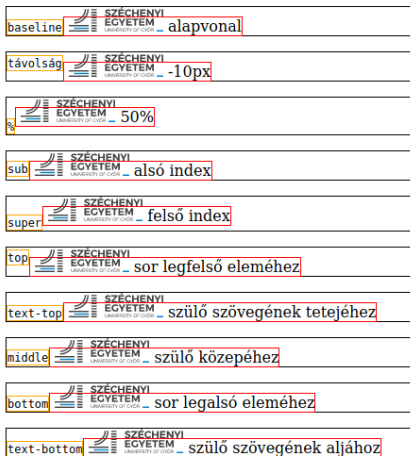
fuggoleges.html

```

7   p, span, code { border: 1px solid black; }
8   span { border-color: red; }
9   code { border-color: orange; }

13  <p><code>baseline</code><span style="vertical-align:
    ↪ align: baseline;">
14    
15    alapvonal</span></p>

```



text-indent

Első sor behúzása: *távolság* (a bekezdés bal szélétől számított behúzás), % (szülő elem szélességének százalékában adott behúzás)

behuzas.html

```

7      p {
8          text-indent: 1cm;
9          text-align: justify;
10     }

14     <p>
15     A HTML (angolul: HyperText Markup Language=hiperszo-
16     veses jelölőnyelv) egy leíró nyelv,
        melyet weboldalak készítéséhez fejlesztettek ki, és
        mára már internetes szabvánnyá vált a W3C (World Wide Web
        Consortium) támogatásával.
    </p>

```

A HTML (angolul: HyperText Markup Language=hiperszo-
veges jelölőnyelv) egy leíró nyelv,
melyet weboldalak készítéséhez fejlesztettek ki, és
mára már internetes szabvánnyá vált a W3C (World
Wide Web Consortium) támogatásával.

Az aktuális változata a 4.01, mely az SGML
általános jelölőnyelv egy konkrét alkalmazása (azaz
minden 4.01-es HTML dokumentum egyben az SGML
dokumentumszabványnak is meg kell, hogy feleljen).

normal

nowrap

```
pre
```

```
pre-line
```

```
pre-wrap
```

minden fehér karaktert megőriz, és tördel, ha szükséges

Nem választ magától monospace karakterkészletet!

fahrcels2.html

```
p {  
  white-space: pre;  
  font-family: monospace;  
}
```

<p>

Fahrenheit — Celsius

Fahrenheit - Celsius

300	148.9
280	137.8
260	126.7
240	115.6
220	104.4
200	93.3
180	82.2
160	71.1
140	60.0
120	48.9
100	37.8
80	26.7
60	15.6
40	4.4
20	-6.7
0	-17.8

letter-spacing: betűk közötti távolság

normal

szokásos távolság, alapértelmezés

távolság

betűk közötti távolság, negatív érték is elfogadott

word-spacing: szavak közötti távolság

normal

szokásos távolság (betűmagasság negyede), alapértelmezés

távolság

szavak közötti távolság, negatív érték is elfogadott

tavolsag.html

```

7      .betusuru { letter-spacing: -.1em }
8      .betunormal { letter-spacing: normal }
9      .beturitka { letter-spacing: .5em }
10     .szoritka { word-spacing: .5cm }
11     .szosuru { word-spacing: -.2em }

15     <p>Néhány szóban a <span class="beturitka">betűk távolságát
16     megnöveltük</span>, <span class="betusuru">másutt

```

Néhány szóban a b e t ű k t á v o l s á g á t m e g n ö v e l t ű k , másutt
 l e c s ö k e n t e t t ű k , de néhány helyen megőriztük az eredeti állapotot. Hasonlóan a szavak
 k ö z t i s z ó k ö z ö k e t i s l e h e t s ű r í t e n i , d e r i t k í t a n i i s .

normal

capitalize

uppercase

lowercase

csupa kisbetűvel nyomtat

nagybetu.html

```
7      .nagykezdő { text-transform: capitalize }
8      .nagybetű { text-transform: uppercase }
9      .kisbetű { text-transform: lowercase }
```



```
13     <p>Ebben a bekezdésben nincsen semmi különös.</p>
14     <p class="nagykezdő">Itt viszont minden szó első betűje nagybetűs.</p>
```

Itt Viszont Minden Szó Első Betűje Nagybetűs.

vagy úgy járni, mint szegény nemecsek.

`text-decoration-line`: vonal húzása a szöveggel párhuzamosan

`none`

nincs vonalazás, alapértelmezés. Pl. hivatkozások aláhúzásának eltávolításához használható.

`underline`

aláhúzza a szöveget; **félrevezetheti az olvasót**, ha nem csak a hivatkozások jelennek meg aláhúzással!

`overline`

a szöveg fölött húz vonalat

`line-through`

áthúzza a szöveget

```
text-decoration-style: a vonal stílus
```

solid

folytonos vona

double

dupla vonal

dotted

pontvonal

dashed

szaggatott vonal

wavy

hullámos vonal

szín

Rövidítés:

text-decoration-line text-decoration-color text-decoration-style

Akár többféle vonal is megadható, tetszőleges rész elhagyható, sorrend tetszőleges

dekoracio.html

```

7      .ala { text-decoration: underline; }
8      .fole { text-decoration: overline; }
9      .at { text-decoration: line-through; }
10     .helytelen { text-decoration: wavy red underline; }
11     .kombinalt { text-decoration: overline underline blue dotted; }

15     <p>A szavak <span class="ala">alá</span> és <span class="fole">
16     fölé</span> is lehet vonalat húzni, de <span class="at">ki is
17     húzhatjuk</span>, a <span class="helytelen">hejtelenül</span>

```

A szavak alá és föle is lehet vonalat húzni, de ki is húzhatjuk, a hejtelenül írt szavakat megjelölhetjük és mindenféle kombinációkat is kitalálhatunk.

text-shadow: szöveg árnyéka

h-shadow v-shadow blur-radius color

vízszintes eltolás, függőleges eltolás, elmosás mértéke, szín.

Az elmosás mértéke elhagyható, a többi kötelező. Az eltolásoknál negatív értékek megengedettek. Vesszővel elválasztva több árnyék is megadható egyszerre.

none nincs árnyék, alapértelmezés

arnyek.html

```
7      .dupla { text-shadow: #666666 1px 2px, #AAAAAA 2  
      ↪ px 4px; }  
8      .neon { text-shadow: 0px 0px 10px red; }  
9      .lathatatlan {  
10         color: white;  
11         text-shadow: 2px 2px 10px black;  
12     }
```

```
16     <p class="dupla">Dupla árnyéket vet a szöveg.</p>  
17     <p class="neon">Tiszta '80-as évek!</p>  
18     <p class="lathatatlan">Csak az árnyék miatt  
    ↪ olvasható.</p>
```

Dupla árnyéket vet a szöveg.

Tiszta '80-as évek!

Csak az árnyék miatt olvasható.

line-height: sormagasság

normal

betűméretből következő sormagasság, alapértelmezett

szám

az aktuális betűméretet ezzel szorozva kapja meg a sormagasságot

távolság

rögzített sormagasság, CSS mértékegységben

%

az aktuális betűméret %-a

sormagassag.html

```
7      p { border: 1px solid black; }  
8      .masfeles { line-height: 1.5; }  
9      .dupla { line-height: 200%; }
```

```
13     <p>Normál sormagasság.</p>  
14     <p class="masfeles">Másfeles sormagasság.</p>  
15     <p class="dupla">Kétszeres sormagasság.</p>
```

Normál sormagasság.

Másfeles sormagasság.

Kétszeres sormagasság.

Többféle írásirány támogatott egyazon oldalon

irasirany.html

```

7      .jobbrolBalra {
8          direction: rtl;
9          unicode-bidi: bidi-override;
10     }

```

```

14     <p>Balról jobbra.</p>

```

```

15     <p class="jobbrolBalra">Jobbról balra.</p>

```

Balról jobbra.

.arlab lórbboJ

Készítse el az ábrán látható weboldalt!

- Induljon ki a **szoveg.txt** fájlból!
- A címsor betűi között hagyjon 5-5 képpontnyi helyet,
- írja csupa nagybetűvel, és
- jelenítsen meg alatta 3-3 képpontnyival jobbra és lefelé eltolt árnyékot, mely kék színű, és elmosásának sugara 10 képpont!
- A bekezdés legyen sorkizárt igazítású,
- a sormagasság másfélszeres,
- az első sor behúzása 20 képpontnyi,
- és automatikusan elválasztott!
- Az emberek neveit emelje ki zöld színű, dupla aláhúzással!

szoveg.html

ATARI

Az Atari egy amerikai vállalat, a világ egyik leghíresebb videójáték-konzol, illetve videójáték-fejlesztő/gyártó cége. 1972-ben [Nolan Bushnell](#) és [Ted Dabney](#) alapította. Megalapításától videójátékokkal foglalkozott, ekkor jelentek meg a híres, népszerű Computer Space és Pong játékok. (A kettő közül a Pong számít a legforradalmibbnak, hiszen mérőföldkőnek számít a videójáték-történelemben.) Első konzoljuk az Atari VCS (Video Computer System) volt, amely később az Atari 2600 nevet kapta. 1976-tól otthoni számítógépeket is gyártottak (Atari 400/800, később XL-sorozat). Az 1983-as videójáték-válság alaposan kereszbe tette a cég működését. A nehéz helyzetbe került céget 1984-ben [Jack Tramiel](#), a Commodore alapítója és korábbi tulajdonosa vásárolta meg. Az ő vezetése alatt mutatták be 1985-ben a 16 bites, Motorola 68000 processzoron alapuló Atari ST számítógépcsaládot, amely a nyolcvanas-kilencvenes évek fordulóján igen népszerű volt. Az olcsó PC-k elterjedésével az Atari folyamatosan piacot veszített, és hiába mutatta be az igen fejlett, Motorola 68030 alapú Atari Falcon számítógépet és a 64 bites játékkonzolok előfutárát, az Atari Jaguar, a csődöt nem kerülhette el. Az Atari Corporation 1996-ban átalakult, majd megszüntették. Az értékes Atari márkanévét 1998-ban a Hasbro játékgyártó világcég vette meg. 2001-ben az Infogrames szoftvercég vette meg a Hasbro Interactive nevű leányvállalatot az Atari márkához kapcsolódó jogokkal együtt. A cég jelenleg Atari SA. néven működik.