

Web-technológia

Cascading Style Sheets

Dr. Hatwágner F. Miklós

Széchenyi István Egyetem, Győr

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

2020. február 25.

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

```

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;
}
```

Tulajdonság (property)

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

Milyen legyen az új állapot?

Megjegyzések a CSS-ben:

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

CSS ellenőrző

p { font-style: italic; }

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

Az id nem kezdődhet számjegy karakterrel!

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

10/10/2017

1. *Journal of Management Studies*, 1997, 34, 1, 1-14.

—


```
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; }
```

1. *Journal of the American Medical Association*, 1997; 278: 1019-1024.


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

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



```
<style>
  p { color: red; }
</style>
<link rel="stylesheet" type="text/css" href="utkozes1.css" />
</head>
<body>
  <p>Első bekezdés.</p>
  <p style="color: green;">Második bekezdés</p>
```

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

Első bekezdés.

Második bekezdés

```
<link rel="stylesheet" type="text/css" href="utkozes1.css" />
<style>
  p { color: red; }
  p { color: orange; }
</style>
</head>
<body>
  <p>Első bekezdés.</p>
```

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


color

background-color

Szín, mint a tulajdonság értéke megadható:

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

140 szabványos színekód

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.

rgba() függvényrel

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.

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



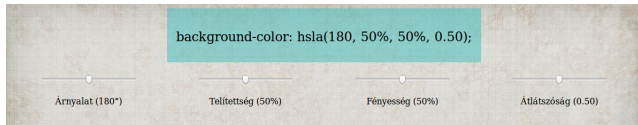
szinek1.html

hsla() függvényrel

hsl(hue, saturation, lightness), ahol hue az árnyalat, $[0, 360]$ fok közötti elfordulás a színkeréken. 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() függvényrel

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 háttere legyen citromsárga!

background-color

background-image

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

Megadhatók **színátmenetek** is.

A szöveg maradjon **olvasható** a háttéren!

11/11/2019 11:11 AM

- 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 ismétel, ami vágás nélkül elfér, közöttük helyet hagy

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



- ☒ 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.


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

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

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

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

<h1>Jobb alsó saroktól 10%-ra</h1>
<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](#)

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
```

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

```
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 fedí 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



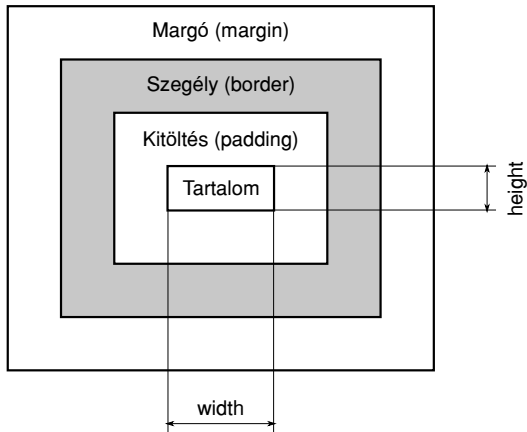
Ut semper, ante pretium aliquam commodo, libero turpis gravida metus, eu semper justo quam at risus. Maecenas vitae libero massa. Proin sodales erat ex, sit amet consequat massa venenatis sit amet. Sed et luctus tortor, eu vestibulum libero. Curabitur euismod nisi sem, vel elementum nibh posuere id. 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
23         ↪ függvénye.</p>
24     <p style="width: 200px; height: 100px;">Ez egy 200x100 képpont méretű bekezdés.</p>
25     <p>A bekezdésben lévő <span style="width: 120px; height: 50px;">soron belüli elemek</span>
26         ↪ mérete nem módosítható!</p>
27     <p>Kivéve, ha a <span style="display: block; width: 120px; height: 50px;">megjelenítés mó
28         ↪ dját</span> blokkszintűre módosítjuk.</p>

```

box-sizing

Csak a tartalom méretét

Tartalom + kitöltés + szegély

Kényelmes:

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

Pl. Gipsz Jakab

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

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 ekkorára növeli meg a magasságot.

visible

hidden

scroll

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.

```
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
```

Minimális és maximális szélesség az elektronikus szöveg fogadott tény, miszerint egy olvasó bizonyos arányban egy adott minimális és maximális szélességű szöveget nézi. A Lorem Ipsum szöveg nem szélességi kód, azaz nem tartalmaz szélességi betűket tartalmaz, ellentétben a táblázatos szövegekkel, amelyekben a táblázatok és a formázások megőrzésére szolgál. Szöveg szerkesztő és weboldal szerkesztő használja a Lorem Ipsum-ot mint alapbeállítási szövegmodellt, és egy keresés a Lorem Ipsum-ra szöveg szerkesztő fogadott eredményt.

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.

A szegélyeknek állítható a

- stílusa (`border-style`),
- szélessége (`border-width`), és a
- színe (`border-color`).

Megjegyzések:

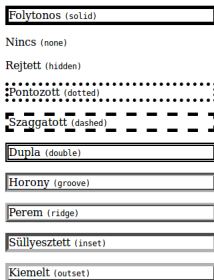
- Utóbbi kettő csak a stílus beállítása esetén működik.
- Minden paraméter állítható külön az egyes oldalakra is.

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

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" style="width: 50%; height: 50px; vertical-align: top;">  
    ↪ ">A1</td>
```

```
26      <td style="border-style: none" style="width: 50%; height: 50px; vertical-align: top;">  
    ↪ B2</td>
```


border: width style color

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.


```
div {
    background-color: orange;
}
div.keretes {
    border: 2px solid red;
}
p {
    background-color: yellow;
    margin: 10px;
}
```

```

20      <div>
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`.

- 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 háterszín 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

outline-style

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

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

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

outline-offset

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

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 beszédszintetizátor segítségével olvassa fel a weblapok szövegét), vagy Braille-készüléken megjelenítve.