

Web-technológia

Cascading Style Sheets, I. rész

Dr. Hatwagner F. Miklós

Széchenyi István Egyetem, Győr

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

2020. március 30.

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>
```

Formázás CSS-sel ([cssFormazas.html](#))

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

Formázás CSS-sel ([cssFormazas.css](#))

```
1 .kiemelt {
2   font-size: larger;
3   font-style: italic;
4 }
```

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?

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

Abstract

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; }
```

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

```
*.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

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

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 kétévnnyi 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 <a href="https://ajovomultja.
    ↪ hu/news/60-eves-az-elso-magyar-szamitogep">korabeli sajtóbeszámol
    ↪ óban</a> 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!

Széchenyi István Egyetem, Győr

```

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

utkoztes2.html

```
6      <link rel="stylesheet" type="text/css" href="utkoztes1.css" />
7      <style>
8          p { color: red; }
9          p { color: orange; }
10     </style>
11 </head>
12 <body>
13     <p>Első bekezdés.</p>
```

utkoztes1.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]).



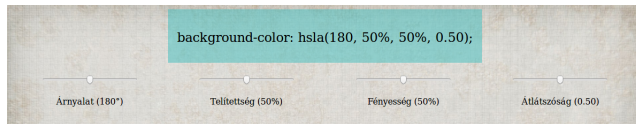
szinek1.html

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 háttere legyen citromsárga!

background-color

background-image

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

A szöveg maradjon **olvasható** a háttéren!

Háttérkép csempézési iránya

- 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

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



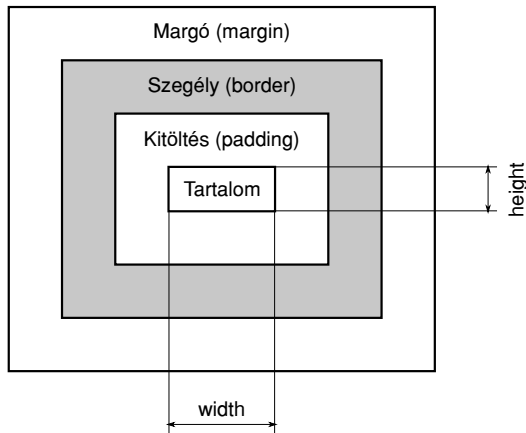
Ut semper, ante pretium aliquam commodo, libero turpis gravida metus, eu semper justo quam at risus. Maecenas vitae libero massa. Proin sodales erat ex, sit amet consequat massa venenatis sit amet. Sed et luctus tortor, eu vestibulum libero. Curabitur euismod nisi sem, vel elementum nibh posuere id. 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:

Végzettség:

meretezes.html

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


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 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.

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:

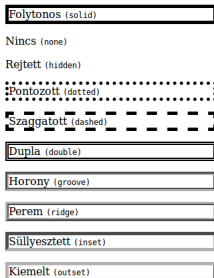
- Utóbbi kettő csak a stílus beállítása esetén működik.
- Minden paraméter állítható külön az egyes oldalakra is.

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-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>
  
```


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:

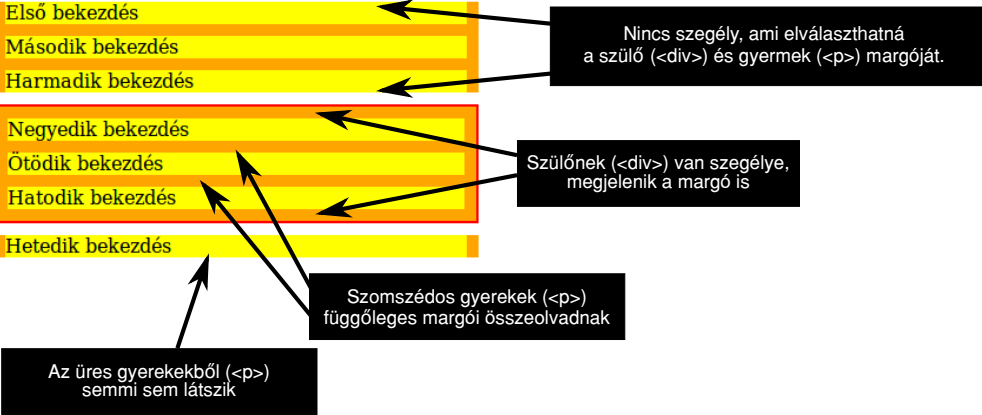
- szülő szomszédos gyerekei között (szélső gyerekek margói túlnyúlnak a szülőn)
- ha nincs olyan megjeleníthető szegély, kitöltés, stb., ami elválasztaná a szülő és valamely gyerekének alsó/felső margóját
- üres blokkok alsó és felső margóját is összevonják

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`.

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

outline-style

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

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

Serif

„Talpas” betűkészletek; főleg bekezdések szövegéhez, mert „vezeti a szemet” az alapvonalon, de képernyőn sokan nehezen olvassák

Sans-serif

„Talp nélküli” betűkészletek, főleg címsorokhoz

Monospace

„Egyenközü”, azonos szélességű betűkből álló betűkészletek, főleg forrásszövegekhez

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

- Széchenyi István Egyetem, Győr

normal

Álló betűk, alapértelmezés

Dólt betűk

„Kevésbé dőlt”, gyenge támogatás

normal

Normál betűk, alapértelmezett.

Kisbetűt, a kisbetűket kicsinyített nagybetűkkel helyettesíti.

Abszolút méretekben

latív méretekben

lcsszavakkal

Átméretezés: smaller, larger.

normal

bold

bolder, lighter

100, 200, 300, ..., 900

Különféle vastagságok, de többnyire csak a normál és a félkövér támogatott.

```
font: font-style font-variant font-weight font-size/line-height
      font-family | caption | icon | menu | message-box |
      small-caption | status-bar | initial | inherit;
```

A méret és a karakterkészlet megadása kötelező. A caption, icon, ... kulcsszavakkal lehet a böngésző által valamilyen célra már használt beállításokat kérni egy adott helyen.

karakter.html

```

6      <style>
7          h1 { font-family: Arial , Helvetica , sans-serif; }
8          p { font-family: "Times␣New␣Roman", Times, serif; }
9          p.prg { font-family: "Courier␣New", Courier , monospace; }
10     </style>

13     <h1>Ez egy címsor</h1>
14     <p>Ez egy bekezdés.</p>
15     <p class="prg">printf("Hello␣World!\n");</p>
16     <p>Ebben a bekezdésben <span style="font-style:␣italic;">döntött
17     </span> és <span style="font-style:␣oblique;">kissé döntött</span>
18     részek is előfordulnak. <span style="font-variant:␣small-caps;">
19     Néha kiskapitális formázást használunk.</span>
20     <span style="font-size:␣20px;">20 px</span>,
21     <span style="font-size:␣20pt;">20 pt</span>,
22     <span style="font-size:␣1.5em;">1.5em</span>,
23     <span style="font-size:␣5vw;">5vw</span>,
24     <span style="font-size:␣smaller;">smaller</span>,
25     <span style="font-size:␣larger;">larger</span>.</p>

```

Ez egy bekezdés.

```
printf("Hello World!\n");
```

Ebben a bekezdésben *döntött* és *kissé döntött* részek is előfordulnak. NÉHA KISKAPITÁLIS FORMÁZÁST HASZNÁLUNK.

20px, 20pt, 1.5em, 5vw, smaller, larger.

- egyedi megjelenést kölcsönöz
- mindenki ugyanazt a készletet használja, garantáltan azonos megjelenés mindenhol (sok eszközön hiányosak a készletek, főleg a ritkán használt karakterek)

- TrueType Font (TTF)
- OpenType Font (OTF)
- Web Open Font Format (WOFF)

font-family

src

font-stretch

font-style

font-weight

◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡

webfont.html

```
7      @font-face {  
8          font-family: devil;  
9          src: url("devil/Devil-East-Free-Font.woff");  
10     }  
11     p {  
12         font-family: devil;  
13         font-size: 80pt;  
14     }  
15     p.felkover {  
16         font-weight: bold;  
17     }  
  
21     <p>Devil East Free Font</p>  
22     <p class="felkover">Devil East Free Font</p>
```

Devil East Free Font
Devil East Free Font

Google Fonts

- Több száz ingyenes karakterkészlet
- Könnyű kereshetőség
- Egyszerű integráció a weboldalba

googleFonts.html

```
6  <link href="https://fonts.googleapis.com/css?  
    ↪ family=Baloo+2&display=swap" rel="  
    ↪ stylesheet" />  
7  <style>  
8    p {  
9      font-family: 'Baloo 2', cursive;  
10     font-size: 20pt;  
11   }  
12 </style>  
  
15 <p>Google Fonts – Baloo</p>
```

Google Fonts - Baloo

Készítse el Semmelweis Ignác oldalát a [Wiki](#) oldal szövegét felhasználva!

- Töltse le a [Ballerina](#) karakterkészletet!
- Használja ezt az első szintű címsorban szereplő név kiírására, 42 nyomdai pont méretben!
- A bekezdések szövegét írja [Libre Baskerville](#) karakterkészlettel, 12 nyodai pont mérettel!
- Készítsen stílusokat a félkövér és dőlt betűs részek megjelöléséhez!

[semmelweis.html](#)

Semmelweis Ignác

Semmelweis Ignác Fülöp (Buda, 1818. július 1. – Döbling, 1865. augusztus 13.) magyar orvos, „az anyák megmentője”.

1818. július 1-jén a tabáni Palota utca (ma Apród utca) 1–3. alatti Meindl-házban látta meg a napvilágot, ahol édesapjának – a hienc (nyugat-magyarországi német) ősökkel rendelkező Semmelweis Józsefnek (1778–1846) – jól menő fűszerüzlete is volt. Édesanyja Müller Teréz, Müller Fülöp módos krisztinavárosi sváb kocsigyártó és Anderl Teréz leánya. A szülők 1810. január 14-én kötöttek házasságot a Krisztinavárosban. Ignác (a tízből) ötödik gyerekként született, két bátyja ismert pesti kereskedő, Károly öccse pedig a Belvárosi római katolikus plébánia lelkésze volt. Ignácot a közeli Tabáni Plébániatemplomban[1] keresztelték meg.[2] (A kor következtelen anyakönyvezésére jellemző, hogy a tíz testvér vezetéknevét tízféleképpen írták; Semmelweiss, Semmelweis, Semelveisz stb.)