Web-technológia

Cascading Style Sheets, I. rész

Dr. Hatwagner F. Miklós

Széchenyi István Egyetem, Győr

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

2020. április 28.

SZÉCHENYI EGYETEM UNIVERSITY OF GYŐR

INFORMATIKA TANSZÉK DEPARTMENT OF COMPUTER SCIENCE

CSS: Cascading Style Sheets

- ≈ 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?

Megjegyzések a CSS-ben:

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

CSS ellenőrző

JSFiddle - Code Playground

Böngésző Webfejlesztő menüje (F12), stílusszerkesztő

Szelektorok

HTML elem neve

```
p { font-style: italic; }
```

Egyedi azonosító (id attribútum) alapján

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

Az `id` nem kezdődhet számjegy karakterrel!

Univerzális szelektor, mindenre illeszkedik

```
* { 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:

```
<p class="kisbetus voros">Apróbetűs piros bekezdés</p>
```

Elemek csoportosítása

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

egyszeruSzelektor1.html

```
3   <head>
4     <title>Egyszerű szelektorok</title>
5     <link rel="stylesheet" type="text/css" href="egyszeruSzelektor.css">
6     <meta charset="utf-8" />
7   </head>
8   <body>
9     <article>
10      <header>
11        <h1>Ma van a magyar informatika születésnapja</h1>
12        <p class="forras␣bevezeto">Forrás: <a href="https://njszt.hu/hu/news
            ↪ /2020-01-21/ma-van-magyar-informatika-szuletesnapja">NJSZT</a>,
            ↪  <time datetime="2020-01-21␣00:00">2020. január 21.</time></p>
13      </header>
```

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 <a href="https://ajovomultja.
            ↪ hu/news/60-eves-az-elso-magyar-szamitogep">korabeli sajtóbeszámol
            ↪ óban</a> még számológépként hivatkoztak, mert a számítógép szó ké
            ↪ sőbb honosodott meg a magyar nyelvben.</p>
```

egyszeruSzelektor1.css

```
h1, h2, h3, h4, h5, h6 {
  font-family: Arial, Sans-Serif;
}
p { font-family: Garamond, "Times New Roman", Serif; }
a { color: navy; }
.forras {font-style: italic; }
.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 szobányi méretű M-3-at szovjet tervek alapján építették meg, a magyar szakembergárda pedig ezen kísérletezte ki, hogy miként lehet egy számítógépet a népgazdaság érdekelt szolgáló mérnöki-, és gazdasági számításokra felhasználni - például az Erzsébet-híd statikai számításait ellenőrizték vele.

A mintegy kétévnyi 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 relékből 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.

Egy elembe tetszőleges mélységben beágyazott másik elemek kiválasztása

leszarmazott.html

```
7        section p { background-color: yellow; }

11     <section>
12       <div>
13         <p>Első bekezdés.</p>
14         <p>Második bekezdés.</p>
15       </div>
16       <p>Harmadik bekezdés.</p>
17     </section>
18     <p>Negyedik bekezdés.</p>
```

Első bekezdés.

Második bekezdés.

Harmadik bekezdés.

Negyedik bekezdés.

Egy elembe közvetlenül beágyazott gyermek elemek kiválasztása

gyermek.html

```
7        section > p { background-color: yellow; }

11     <section>
12       <div>
13         <p>Első bekezdés.</p>
14       </div>
15       <p>Második bekezdés.</p>
16       <p>Harmadik bekezdés.</p>
17     </section>
18     <p>Negyedik bekezdés.</p>
```

Első bekezdés.

Második bekezdés.

Harmadik bekezdés.

Negyedik bekezdés.

Egy elemet közvetlenül követő testvér elem kiválasztása

testver.html

```
7        div + p { background-color: yellow; }

11     <section>
12       <p>Első bekezdés.</p>
13       <div>
14         <p>Második bekezdés.</p>
15       </div>
16       <p>Harmadik bekezdés.</p>
17       <p>Negyedik bekezdés.</p>
18     </section>
19     <p>Ötödik bekezdés.</p>
```

Első bekezdés.

Második bekezdés.

Harmadik bekezdés.

Negyedik bekezdés.

Ötödik bekezdés.

Egy elemet közvetlenül követő összes testvér kiválasztása

testver2.html

```
7        div ~ p { background-color: yellow; }

11     <section>
12       <p>Első bekezdés.</p>
13       <div>
14         <p>Második bekezdés.</p>
15       </div>
16       <p>Harmadik bekezdés.</p>
17       <p>Negyedik bekezdés.</p>
18     </section>
19     <p>Ötödik bekezdés.</p>
```

Első bekezdés.

Második bekezdés.

Harmadik bekezdés.

Negyedik bekezdés.

Ötödik bekezdés.

Látszólagos osztályok (pseudo-class): egy elem adott állapota esetén alkalmazandó formázása, referencia

Az egér alatti elem kiválasztása

hover.html

```
7    p:hover { background-color: yellow; }

11   <p>Mozgassa ide az egeret!</p>
```

Mozgassa ide az egeret!

Mozgassa ide az egeret!

Azon elemek kiválasztása, melyek a szülőjük első gyermekei

firstchild.html

```
7        p:first-child { background-color: yellow; }

10   <body>
11     <p>Első bekezdés.</p>
12     <p>Második bekezdés.</p>
13     <p>Harmadik bekezdés.</p>
14   </body>
```

Első bekezdés.

Második bekezdés.

Harmadik bekezdés.

Nyelvfüggő beállítások alkalmazása

lang.html

```
7        q:lang(hu) { quotes: "\0201E" "\0201D"; }

11     <p><q>Magyar idézet.</q></p>
12     <p><q lang="en-US">Angol idézet.</q></p>
```

„Magyar idézet.”

“Angol idézet.”

Adott típus n-edik előfordulása (részleteket ld. később `nth-child()`-nál)

nthoftype.html

```
7        li:nth-of-type(2n) { background-color: yellow; }

11     <ul>
12       <li>Első elem</li>
13       <li>Második elem</li>
14       <li>Harmadik elem</li>
15       <li>Negyedik elem</li>
16     </ul>
```

- Első elem
- Második elem
- Harmadik elem
- Negyedik elem

Látszólagos elemek (pseudo-elements): elemek bizonyos részeinek kiválasztása

Első sor kiválasztása: `::first-line`

- Csak blokkszintű elemekkel használható
- Csak bizonyos tulajdonságokkal használható:
  `font`, `word-spacing`, `letter-spacing`, `line-height`, `text-decoration`, `text-transform`, `vertical-align`, `color`, `background`, `clear`.

CSS3 előtt `:` állt ott, ahol most `::`.

Szöveg első betűjének kiválasztása: `::first-letter`

- Csak blokkszintű elemekkel használható
- Csak bizonyos tulajdonságokkal használható:
  `font`, `line-height`, `text-decoration`, `text-transform`, `color`, `background`, `margin`, `border`, `padding`, `vertical-align` (ha nincs lebegtetés), `float`, `clear`.

Kijelölt szöveg: `::selection`

- Csak bizonyos tulajdonságokkal használható:
  `color`, `background`, `cursor`, `outline`.

Tartalom valami előtt: `::before`, mögött: `::after`

Tartalom megadása: `content` tulajdonsággal, értéke:

- idézőjelek/aposztrófok közötti szöveg
- karakterek unicode kódja, pl. `'\02192'` $\equiv \rightarrow$
- `url()` függvénnyel adott kép

pseudoelements.html

```
7   h1::before {
8     content: "\02192";
9     color: red;
10  }
11  h1::after {
12    content: "\02190";
13    color: red;
14  }
```

```
15  p::first-line {
16    font-variant: small-caps;
17    font-weight: bold;
18  }
19  p::first-letter {
20    font-size: 200%;
21    color: blue;
22  }
23  p::selection {
24    color: red;
25    background-color: yellow;
26  }
```

```
30  <h1>Nemes Tihamér</h1>
31  <p>Nemes Tihamér (Budapest, 1895. április 29. – Budapest, 1960.
32  március 30.) feltaláló, gépészmérnök, MTA doktora, villamosmérnök,
33  a műszaki tudományok doktora, a kibernetika egyik hazai
34  úttörője.</p>
```

→Nemes Tihamér←

Nemes Tihamér (Budapest, 1895. április 29. - Budapest, 1960. március 30.) feltaláló, gépészmérnök, MTA doktora, villamosmérnök, a műszaki tudományok doktora, a kibernetika egyik hazai úttörője.

Szülei Nemes Mihály és Auguszt Mária voltak. Oklevelét 1917-ben Budapesten szerezte a Műegyetemen. Rövid ideig dolgozott a Lloyd repülőgépgyárnál. 1921-től a Telefonhírmondóhoz került, majd annak főmérnökének nevezték ki. 30 éves korában az Elektromos és Finommechanikai Rt. igazgatójává léphetett elő. 1929-től a Posta Kísérleti Állomásán dolgozott. Az 1930-as évek közepétől megkezdte a televíziózás elméleti és gyakorlati tanulmányozását. Részt vett az első hazai kísérletekben. 1938-ban szabadalmi bejelentést tett színes televízió megvalósítására. 1950-ben a Távközlési Kutató Intézet tudományos munkatársa.

Érdeklődési körébe tartoztak az: elektrotechnika, logika, szimuláció, modellezés, de leginkább a logikai gép. Nemes Tihamér sakkozó- és sakkfeladványokat megoldó gépei is elsősorban az emberi gondolkozás modellezésére szolgáltak. Számtalan televíziós szabadalma szimulálta az emberi szem funkcióit. Postamérnökként egyike volt azoknak, akik a magyar televíziózást megteremtették. 1953-ban készítette az első magyar kísérleti tv-adó berendezést.

pseudoelements.html

Kiválasztás attribútumok alapján

Adott attribútummal rendelkező elemek kiválasztása

attributum1.html

```
7       a[target] { background-color: yellow; }

12    <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tihamér">Nemes Tihamér</a></p>
13    <p><a href="https://hu.wikipedia.org/wiki/Neumann_János" target="_blank">Neumann János</a></p>
```

Néhány hivatkozás híres magyar tudósok Wiki oldalaira

Nemes Tihamér

Neumann János

Pontosan az adott értékű attribútummal rendelkező elemek kiválasztása

attributum2.html

```
7       a[target="_blank"] { background-color: yellow; }

12    <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" target="_self">
          ↪ Nemes Tihamér</a></p>
13    <p><a href="https://hu.wikipedia.org/wiki/Neumann_J%C3%A1nos" target="_blank">
          ↪ Neumann János</a></p>
```

Néhány hivatkozás híres magyar tudósok Wiki oldalaira

Nemes Tihamér

Neumann János

Adott teljes szót (is) tartalmazó értékű attribútummal rendelkező elemek kiválasztása

attributum3.html

```
7   a[title~="Nemes"] { background-color: yellow; }

12  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" title="Nemes Tihamér">Nemes Tihamér</a></p>
13  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Nagy_%C3%81gnes" title="Ágnes, Nemes Nagy">Nemes Nagy Ágnes</a></p>
14  <p><a href="https://hu.wikipedia.org/wiki/Nemesg%C3%A1zok" title="Nemesgázok">Nemesgázok</a></p>
```

Érdekes Wiki oldalak

Nemes Tihamér

Nemes Nagy Ágnes

Nemesgázok

Kiválasztja az elemet, ha az adott attribútumnak a megadott szó az egyetlen értéke, esetleg ezzel kezdődik és kötőjellel folytatódik

attributum4.html

```
7   a[class|="nemes"] { background-color: yellow; }

12  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" class="nemes tihamer">Nemes Tihamér</a></p>
13  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Nagy_%C3%81gnes" class="nemes-nagy-agnes">Nemes Nagy Ágnes</a></p>
14  <p><a href="https://hu.wikipedia.org/wiki/Nemesg%C3%A1zok" class="nemesgazok">Nemesgázok</a></p>
15  <p><a href="https://hu.wikipedia.org/wiki/Magyar_nemess%C3%A9g" class="fold nelkuli nemes">Magyar nemesség</a></p>
16  <p><a href="https://hu.wikipedia.org/wiki/Nemess%C3%A9g" class="nemes">Nemesség</a></p>
```

Érdekes Wiki oldalak

Nemes Tihamér
Nemes Nagy Ágnes
Nemesgázok
Magyar nemesség
Nemesség

Kiválasztja az elemet, ha attribútumának értéke az adott karaktersorozattal kezdődik

attributum5.html

```
7   a[class^="nemes"] { background-color: yellow; }

12  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" class="nemes tihamer">Nemes Tihamér</a></p>
13  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Nagy_%C3%81gnes" class="nemes-nagy-agnes">Nemes Nagy Ágnes</a></p>
14  <p><a href="https://hu.wikipedia.org/wiki/Nemesg%C3%A1zok" class="nemesgazok">Nemesgázok</a></p>
15  <p><a href="https://hu.wikipedia.org/wiki/Magyar_nemess%C3%A9g" class="fold nelkuli nemes">Magyar nemesség</a></p>
16  <p><a href="https://hu.wikipedia.org/wiki/Nemess%C3%A9g" class="nemes">Nemesség</a></p>
```

Érdekes Wiki oldalak

Nemes Tihamér
Nemes Nagy Ágnes
Nemesgázok
Magyar nemesség
Nemesség

Kiválasztja az elemet, ha attribútumának értéke az adott karaktersorozattal fejeződik be

attributum6.html

```
7   a[class$="nemes"] { background-color: yellow; }

12  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" class="nemes tihamer">Nemes Tihamér</a></p>
13  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Nagy_%C3%81gnes" class="nemes-nagy-agnes">Nemes Nagy Ágnes</a></p>
14  <p><a href="https://hu.wikipedia.org/wiki/Nemesg%C3%A1zok" class="nemesgazok">Nemesgázok</a></p>
15  <p><a href="https://hu.wikipedia.org/wiki/Magyar_nemess%C3%A9g" class="fold nelkuli nemes">Magyar nemesség</a></p>
16  <p><a href="https://hu.wikipedia.org/wiki/Nemess%C3%A9g" class="nemes">Nemesség</a></p>
```

Érdekes Wiki oldalak

Nemes Tihamér
Nemes Nagy Ágnes
Nemesgázok
Magyar nemesség
Nemesség

Kiválasztja az elemet, ha attribútumának értéke tartalmazza az adott karaktersorozatot

attributum7.html

```
7   a[class*="nemes"] { background-color: yellow; }

12  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Tiham%C3%A9r" class="nemes tihamer">Nemes Tihamér</a></p>
13  <p><a href="https://hu.wikipedia.org/wiki/Nemes_Nagy_%C3%81gnes" class="nemes-nagy-agnes">Nemes Nagy Ágnes</a></p>
14  <p><a href="https://hu.wikipedia.org/wiki/Nemesg%C3%A1zok" class="nemesgazok">Nemesgázok</a></p>
15  <p><a href="https://hu.wikipedia.org/wiki/Magyar_nemess%C3%A9g" class="fold nelkuli nemes">Magyar nemesség</a></p>
16  <p><a href="https://hu.wikipedia.org/wiki/Nemess%C3%A9g" class="nemes">Nemesség</a></p>
```

Érdekes Wiki oldalak

Nemes Tihamér

Nemes Nagy Ágnes

Nemesgázok

Magyar nemesség

Nemesség

Stílusok forrása

Háromféle helyen lehet stílusokat megadni:

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>
```

Ütközések feloldása

- 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
- Ha forrás alapján nem lehet különbséget tenni, megvizsgálják a *specifikusságot*.
- Ha ez sem segít, a később betöltött szabály felülírja a korábbit.

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.

Specifikusság meghatározása

- A = 1, ha a stílus a `<style>` elemben található, egyébként 0 (pl. `<p style="color: red;">...</p>`)
- B = a szelektorban lévő ID attribútumok száma (pl. `#bekezdes`)
- C = a szelektorban lévő osztályok, attribútumok és látszólagos osztályok száma (pl. `.bekezdes`, `a[target="_blank"]`, `a:hover`)
- D = az elemnevek és látszólagos elemek száma (pl. `p`, `p::first-line`)

Az univerzális szelektor (`*`) és ami ebből származik (pl. `body *`) 0 specifikusságú.

Ütközések feloldása

Szelektor	B	C	D	Specifikusság
`*`	0	0	0	0
`p`	0	0	1	1
`div p`	0	0	2	2
`ul ol + li`	0	0	3	3
`h1 + *[title]`	0	1	1	11
`ul ol li.voros`	0	1	3	13
`li.voros.szint`	0	2	1	21
`#oszlopfej`	1	0	0	100

Az érték részei:

1. elhagyható előjel (+, −)
2. kulcsszó (pl. `red`, `thin`), szám, függvény (pl. `url()`, `rgb()`, stb.) vagy karakterlánc
3. mértékegység (`%`, `cm`, `pt`, `px`, `em`, stb.)

Használat:

- fenti három elem között nem lehet fehér karakter
- 0 érték esetén a mértékegység elhagyható
- rövidítések esetén (pl. `border` a `border-width`, `border-style` és `border-color` helyett) az értékeket fehér karakterek szeparálják

Abszolút mennyiségek

- `cm`, centiméter
- `mm`, milliméter
- `in`, inch (1in = 2,54cm)
- `px`, képpont (1px = 1 fizikai képpont a legfeljebb 96dpi-nél felbontású eszközökön, nagyfelbontású kijelzőkön több)
- `pt`, nyomdai pont (1pt = 1/72in)
- `pc`, pica (1pc = 12pt)

Relatív mennyiségek

- `em`, a karakterkészlet betűmérete
- `rem`, a `<html>` elem betűmérete
- `ex`, az x betű magassága
- `ch`, a 0 karakter szélessége
- `vw`, a viewport szélességének %-a
- `vh`, a viewport magasságának %-a
- `vmin`, a viewport kisebbik méretének 1%-a
- `vmax`, a viewport nagyobbik méretének 1%-a
- `%`, a szülő elem méretének %-a

Általános érték kulcsszavak

- `inherit`: az értéket meg kell örökölni a szülő elemtől.
- `initial`: a tulajdonság eredeti értéke.

Fekete bekezdésben piros szó.

Kék bekezdésben kék szó.

Zöld bekezdésben fekete szó.

inherit.html

```
7        span { color: red; }
8        .kek { color: blue; }
9        .zold { color: green; }
10       .kek span { color: inherit; }
11       .zold span { color: initial; }

15     <p>Fekete bekezdésben <span>piros</span> szó.</p>
16     <p class="kek">Kék bekezdésben <span>kék</span> szó.</p>
17     <p class="zold">Zöld bekezdésben <span>fekete</span> szó.</p>
```

Színek

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.

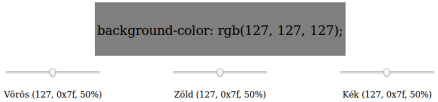
Színek

rgb() függvénnyel

`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() függvénnyel

`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 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() függvénnyel

A fentiek kiegészülnek átlátszósággal.

szinek2.html

Az elemek átlátszósága az opacity tulajdonsággal is állítható [0, 1] intervallumban; a kisebb értékekhez nagyobb átlátszóság tartozik.

atlatszosag1.html

```
7        img       { opacity: 0.4; }
8        img:hover { opacity: 1; }

12     <p><a href="https://uni.sze.hu"><img src="sze_logo.svg" alt="SZE logo"
          ↪ width="400" /></a>
13        <a href="https://it.sze.hu"><img src="it_logo.svg" alt="IT logo" width=
             ↪ "400" /></a></p>
```

Színek

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!

HTML elemek hátterével 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 ismétel, ami vágás nélkül elfér, közöttük helyet hagy

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

Háttér

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 százelékot, 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

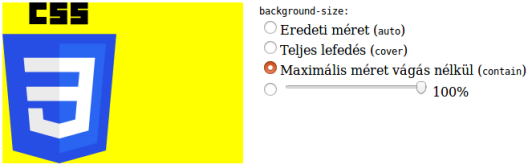
```
p {
  background-image: url("css3.svg");
  background-size: 10%;
  background-repeat: no-repeat;
}
```

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.



meret.html

Háttér

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 hátteréként állítsa be a HTML5sticker.png fájl!
- 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

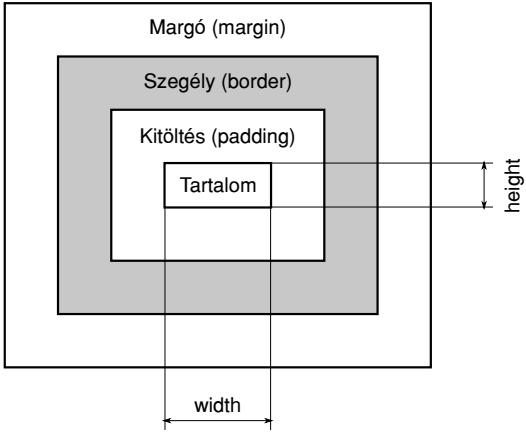
Dobozmodell

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 méretezhetőek 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
- Érdekes lehetőség az abszolút és relatív méretek vegyítésére: `calc()` függvény. Négy alapműveletes kifejezések kiértékelhetők vele.

calc.html

```
7        body { background-color: red; }
8        p {
9          width: calc(100% - 100px);
10         background-color: yellow;
11       }

14   <body>
15     <p>calc függvény használata</p>
16   </body>
```

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 ekkorára növeli meg a magasságot.

Túlcsordulások 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.

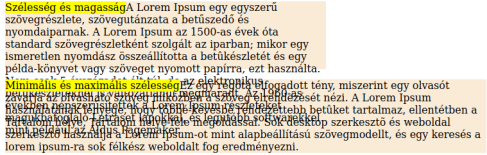
`auto`
: Csak akkor jelennek meg görgetősávok, ha nem fér el a tartalom.

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
```



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.

Szegélyek

A szegélyeknek állítható a

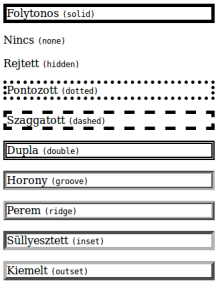
- stílusa (`border-style`),
- szélessége (`border-width`), és a
- színe (`border-color`).

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`.

Ha a `boder-style`-nak

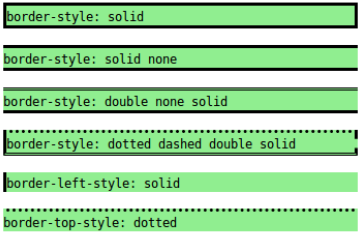
1 értéke van
: `felül-jobb-alul-bal` (minden oldalra ugyanazt a stílust állítja)

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.



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: hidden
    ↪ ">A1</td>

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

Rövidítések

`border: width style color`

Minden oldalon beállítja a szegély szélességét, stílusát, színét.

`border-*: width style color`

A * lehet `top`, `right`, `bottom` és `left`; csak ezekre állítja a fenti három tulajdonságot.

Szegélyek lekerekítése

`border-top-left-radius`
: Bal felső

`border-top-right-radius`
: Jobb felső

`border-bottom-right-radius`
: Jobb alsó

`border-bottom-left-radius`
: Bal alsó

`border-radius`
: Rövidítés fentiek gyors beállításához

`border-radius` paraméterezése

1 érték megadásával

mind a négy sarkot állítja

2 érték megadásával

bal felső + jobb alsó, jobb felső + bal alsó

3 érték megadásával

bal felső, jobb felső + bal alsó, jobb alsó

4 érték megadásával

bal felső, jobb felső, jobb alsó, bal alsó

lekerekites.html

```
7        p {
8          border-style: solid;
9          border-width: 3px;
10         padding: 10px;
11       }
12       #mind {
13         border-color: red;
14         border-radius: 10px;
15       }
16       #szemkozti {
17         border-color: green;
18         border-radius: 10px 20px;
19       }
20       #jobbalso {
21         border-color: blue;
22         background-color: navy;
23         color: white;
24         border-radius: 5px;
25         border-bottom-right-radius: 30px;
26       }
```

Minden sarok kerekítve

Szemközti sarkok eltérő lekerekítési sugárral

Jobb alsó sarok erősen lekerekítve

Margók

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`.

Margók

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.

Margók

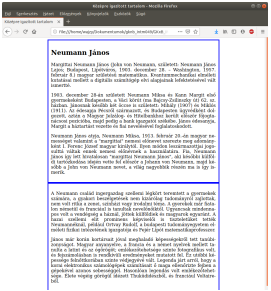
Az ablak keskenyebb, mint az elem *rögzített* szélessége? → gördítősáv. *Maximális* szélesség → csökkenthető. Középre igazítás: `margin: auto`-val.

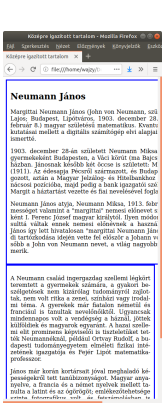
kozepre.html

```
7       section {

12        margin: auto;
13      }
14      section.fixszel { width: 600px; }
15      section.maxszel { max-width: 600px; }

19    <section class="fixszel">
20      <h1>Neumann János</h1>
21      <p>Margittai Neumann János (John von
          ↪ Neumann, született: Neumann
```





Margók

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

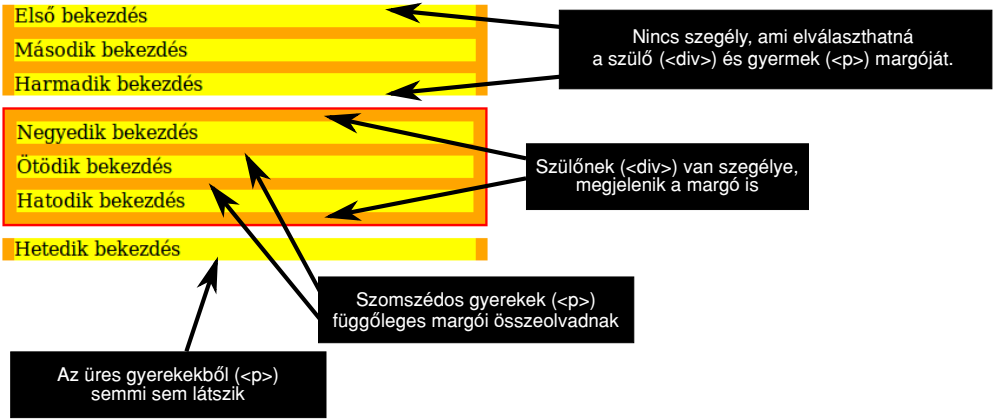
További részletek

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  <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>
```

Margók


Első bekezdés
Második bekezdés
Harmadik bekezdés
Negyedik bekezdés
Ötödik bekezdés
Hatodik bekezdés
Hetedik bekezdés
Nincs szegély, ami elválaszthatná a szülő ( ) és gyermek (<p>) margóját.
Szülőnek ( ) van szegélye, megjelenik a margó is
Szomszédos gyerekek (<p>) függőleges margói összeolvadnak
Az üres gyerekekből (<p>) semmi sem látszik

Kitöltés

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`.

Kitöltés

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.

Kitöltés

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

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.

Körvonal

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 stíluslelíró 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.

Általános fontcsaládok: nagyon hasonló megjelenésű karakterkészletek

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

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

`font-family`: karakterkészlet kiválasztása

- Karakterkészletek listája; ha valamelyik nincs telepítve, a következővel próbálkozik → érdemes egy általános fontcsalád nevét tenni a végére
- Ha a névben szóköz van, idézőjelek közé kell tenni
- Jól bejáratott kombinációk, pl.
  - `"Times New Roman", Times, serif`
  - `Arial, Helvetica, sans-serif`
  - `"Courier New", Courier, monospace`

`font-style`: álló és dőlt betűk

`normal`
: Álló betűk, alapértelmezés

`italic`
: Dőlt betűk

`oblique`
: „Kevésbé dőlt”, gyenge támogatás

`font-variant`: változatok

`normal`
: Normál betűk, alapértelmezett.

`small-caps`
: Kiskapitális, a kisbetűket kicsinyített nagybetűkkel helyettesíti.

`font-size`: méretezés

Abszolút méretekben

A felhasználó nem méretezheti át. Pl. `px` (CSS képpont), `pt` (nyomdai pont).

Relatív méretekben

Felhasználó átméretezheti. Pl. `em` (1em a bekezdések alapértelmezett mérete = `16px`), `%` (a szülő elem betűkészletének méretéhez viszonyítva), `vw` (`1vw` = a *viewport* szélességének 1%-a; átméreteződik az ablak méretezésével)

Kulcsszavakkal

Előre definiált méretek: `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`.
Átméretezés: `smaller`, `larger`.

`font-weight`: „vastagság”, „súly”

`normal`
: Normál szélesség (400), alapértelmezett.

`bold`
: Félkövér (700)

`bolder`, `lighter`
: Növeli, csökkenti a vastagságot

`100, 200, 300, ..., 900`
: Különféle vastagságok, de többnyire csak a normál és a félkövér támogatott.

Rövidítés:

```
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;">20px</span>,
21  <span style="font-size: 20pt;">20pt</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 címsor

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.

Karakterkészletek letölthetők a hálózatról: `@font-face`

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

Megbízhatóan használható formátumok:

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

A `@font-face` szabályban használható tulajdonságok:

`font-family`
: Ezzel a névvel lehet majd hivatkozni a karakterkészletre később, kötelező.

`src`
: A fájl forrását adja meg `url()` CSS függvénnyel, kötelező.

`font-stretch`
: Ha a karakterkészletnek készültek különféle sűrűségű változatai, ezzel lehet kiválasztani, hogy valamelyiket milyennek tekintsen a böngésző (pl. `normal`, `condensed`, `expanded`). Ennek hiányában a böngészőnek kell előállíttatnia a speciális formákat a normálból kiindulva.

`font-style`
: Hasonlóan, dőlt változatokhoz (pl. `normal`, `italic`).

`font-weight`
: Hasonlóan, a „kövérséghez” (pl. `normal`, `bold`).

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 következetlen anyakönyvezésére jellemző, hogy a tíz testvér vezetéknevét tízféleképpen írták; Semmelweiss, Semmelweis, Semelveisz stb.)