

Web-technológia

Cascading Style Sheets

Dr. Hatwágner F. Miklós

Széchenyi István Egyetem, Győr

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

2020. március 3.

CSS: Cascading Style Sheets

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

```

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?

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

45 11 6 2 1 10 3

[illegible]

Figure 1

```

.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; }


```
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; }
```

Forrás: NJSZT, 2020. január 21.

A magyarországi informatika kezdete a első elektronikus számítógép elkészültétől evezethető. A szobányi méretű M-3-at szovjet tervek alapján építették meg, a magyar szakemberek pedig ezen kísérletezték ki, hogy miként lehet egy számítógépet a népgazdaság érdekelt szolgáló mérnök-, és gazdasági számításokra felhasználni - például az Erzsébet-híd statikai számításait ellenőrizték vele.

A mintegy kétévnvi 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-as tudományos számítások emberről gyorsabb végrehajtására tervezték. Noha a gép csupán másodpercenként 30 számítási műveletet tud elvégezni (míg a mai gépek processzorai esetében már több tízezer milliomos utasítással számoltathunk másodpercenként), az M-3-ra írt programok révén így is számos gyakorlati feladatra használták fel. E gépen készült az épülő Erzsébet hidj statikai számításával 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!

```
<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>
```


utkoztes2.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.

How to Use

— 1 —

140 | / / / /

DL: noregonc@n: #f67f00, chel: #1d1e16, az: #mrendazorboli, el: #f67f00, yör: #f67f00

Pl. narancsszín: #ff7f00, ahol # jelzi a 10-es számrendszerbeli alakot, ff a vörös

(D) 1. $\frac{1}{2}$ 2. $\frac{1}{2}$ 3. $\frac{1}{2}$ 4. $\frac{1}{2}$ 5. $\frac{1}{2}$ 6. $\frac{1}{2}$ 7. $\frac{1}{2}$ 8. $\frac{1}{2}$ 9. $\frac{1}{2}$ 10. $\frac{1}{2}$ 11. $\frac{1}{2}$ 12. $\frac{1}{2}$ 13. $\frac{1}{2}$ 14. $\frac{1}{2}$ 15. $\frac{1}{2}$ 16. $\frac{1}{2}$ 17. $\frac{1}{2}$ 18. $\frac{1}{2}$ 19. $\frac{1}{2}$ 20. $\frac{1}{2}$ 21. $\frac{1}{2}$ 22. $\frac{1}{2}$ 23. $\frac{1}{2}$ 24. $\frac{1}{2}$ 25. $\frac{1}{2}$ 26. $\frac{1}{2}$ 27. $\frac{1}{2}$ 28. $\frac{1}{2}$ 29. $\frac{1}{2}$ 30. $\frac{1}{2}$ 31. $\frac{1}{2}$ 32. $\frac{1}{2}$ 33. $\frac{1}{2}$ 34. $\frac{1}{2}$ 35. $\frac{1}{2}$ 36. $\frac{1}{2}$ 37. $\frac{1}{2}$ 38. $\frac{1}{2}$ 39. $\frac{1}{2}$ 40. $\frac{1}{2}$ 41. $\frac{1}{2}$ 42. $\frac{1}{2}$ 43. $\frac{1}{2}$ 44. $\frac{1}{2}$ 45. $\frac{1}{2}$ 46. $\frac{1}{2}$ 47. $\frac{1}{2}$ 48. $\frac{1}{2}$ 49. $\frac{1}{2}$ 50. $\frac{1}{2}$ 51. $\frac{1}{2}$ 52. $\frac{1}{2}$ 53. $\frac{1}{2}$ 54. $\frac{1}{2}$ 55. $\frac{1}{2}$ 56. $\frac{1}{2}$ 57. $\frac{1}{2}$ 58. $\frac{1}{2}$ 59. $\frac{1}{2}$ 60. $\frac{1}{2}$ 61. $\frac{1}{2}$ 62. $\frac{1}{2}$ 63. $\frac{1}{2}$ 64. $\frac{1}{2}$ 65. $\frac{1}{2}$ 66. $\frac{1}{2}$ 67. $\frac{1}{2}$ 68. $\frac{1}{2}$ 69. $\frac{1}{2}$ 70. $\frac{1}{2}$ 71. $\frac{1}{2}$ 72. $\frac{1}{2}$ 73. $\frac{1}{2}$ 74. $\frac{1}{2}$ 75. $\frac{1}{2}$ 76. $\frac{1}{2}$ 77. $\frac{1}{2}$ 78. $\frac{1}{2}$ 79. $\frac{1}{2}$ 80. $\frac{1}{2}$ 81. $\frac{1}{2}$ 82. $\frac{1}{2}$ 83. $\frac{1}{2}$ 84. $\frac{1}{2}$ 85. $\frac{1}{2}$ 86. $\frac{1}{2}$ 87. $\frac{1}{2}$ 88. $\frac{1}{2}$ 89. $\frac{1}{2}$ 90. $\frac{1}{2}$ 91. $\frac{1}{2}$ 92. $\frac{1}{2}$ 93. $\frac{1}{2}$ 94. $\frac{1}{2}$ 95. $\frac{1}{2}$ 96. $\frac{1}{2}$ 97. $\frac{1}{2}$ 98. $\frac{1}{2}$ 99. $\frac{1}{2}$ 100. $\frac{1}{2}$

(Red), (T a z o l d (Green) es 00 a kek (Blue) összetevő intenzitása 8 biten előle

()

né|kij| fixpontosan Additív színkeverés

rgba() 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.

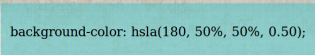
`rgb(red, green, blue, alpha)`, ahol a színösszetevőket egy átlátszóság érték követi ($[0, 1]$).



szinek1.html

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.

A fentiek kiegészülnek átlátszósággal.



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!

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 ismétél, ami vágás nélkül elfér, közöttük helyet hagy

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



background-repeat:

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

hatter.html

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 fedti 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

- 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!

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.



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 amicit 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

- () , () ,

“ “ ”

```
body {
    margin: 0px;
}
p {
    background-color: yellow;
    padding: 20px;
    border: 5px solid red;
    margin: 10px;
}
span {
    background-color: orange;
}
```

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.

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

`<p style="width: 200px; height: 100px;">`Ez egy 200x100 képpont méretű bekezdés.

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

Kivéve, ha a `` megjelenítés módját `` blokkszintűre módosítjuk.

content-box

Csak a tartalom méretét

border-box

Tartalom + kitöltés + szegély

Kényelmes:

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

Név:

Pl. Gipsz Jakab

Végzettség:

középiskola/gimnázium ▾

[meretezes.html](#)

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


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

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

max-width

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

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

A túllógó részeket levágják.

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

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

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

Minimális és maximális szélesség az elektronikus világban fogadott tény, miszerint egy olvasót vagy egy felhasználót az olvasni kívánt anyag azonos elrendezéseket nézi. A Lorem Ipsum szövegben természetesen nem lehet olyan elemek szerepeljenek, amelyek a szövegben betűket tartalmaz, ellentétben a hagyományos nyomtatással, ahol a betűk és a formázás egyaránt fontos szerepet játszik a szöveg megértésében. A Lorem Ipsum szöveg a leggyakoribb szöveg a weboldalak és a nyomtatott anyagok tervezésében. A Lorem Ipsum szöveg a leggyakoribb szöveg a weboldalak és a nyomtatott anyagok tervezésében. A Lorem Ipsum szöveg a leggyakoribb szöveg a weboldalak és a nyomtatott anyagok tervezésében.

Downloaded from <http://ajph.org/> on November 10, 2015

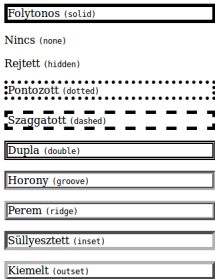
A szegélyeknek állítható a

- 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.

```
<p style="border-style: _solid">Folytonos <code>(solid)</code></p>
<p style="border-style: _none">Nincs <code>(none)</code></p>
<p style="border-style: _hidden">Rejtett <code>(hidden)</code></p>
<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)

felül-alul jobb-bal

ertéke van

felül jobb-bal alul

ertéke van

felül jobb alul bal (óramutató járása)

szerint)

Hasonlóképpen lehet oldalanként szabályozni a margókat és kitöltéseket is.

A1	B1
A2	B2

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

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

szegelyek3.html

```
21 <td style="border-style: hidden  
    ↪ ">A1</td>
```

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


border: width style color

A margók mindig átlátszók, csak a szélességük állítható:

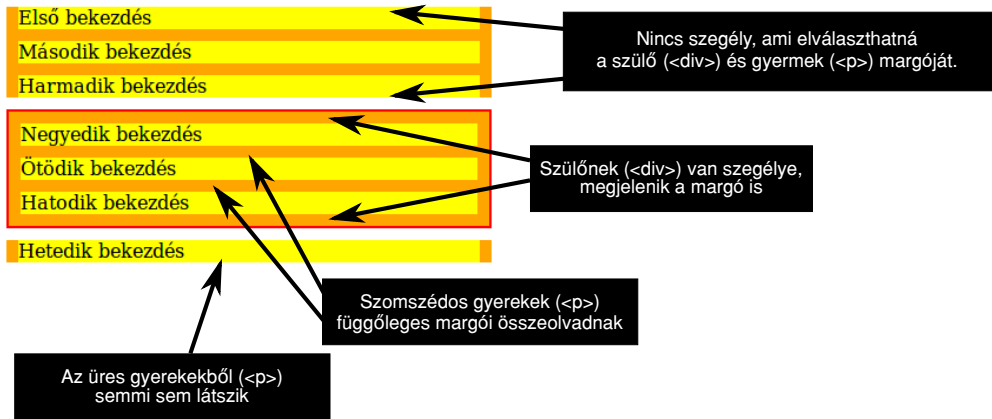
- 1-4 érték megadásával, pl.
`margin: 10px 20px 30px 40px;`
(Fent, jobbra, lent, balra; további esetek mint `border-style`-nál.)
- Oldalakra vonatkozó tulajdonságokkal:
`margin-*`, ahol `*` helyén állhat `top`, `right`, `bottom`, `left`.

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;
}
```



A kitöltések mindig átlátszók, csak a szélességük állítható:

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

A kitöltés szélessége lehet:

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

Negatív értékek **nem** használhatók.

Próbálja meg elkészíteni az ábrának megfelelően a dobozokat!

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

A héttérszín szürkéjének színösszetevői 240 értékűek.

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

A fejléc írásszín komponensei: 237, 234, 142. A 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

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.

```

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 nyelvből 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-készülékben megjelenítve.

text-align

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

vizszintes.html

```

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

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

```

Balra igazított szöveg

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

Középre igazított szöveg

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

Jobbra igazított szöveg

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

Sorkizárt szöveg

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

baseline

távolság

%

sub

super

szülő felső indexéhez

sor legmagasabb eleméhez

szülő elem szövegének tetejéhez

szülő közepéhez

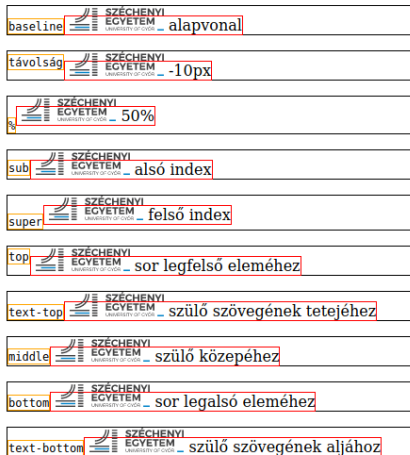
sor legalsó eleméhez

szülő szövegének aljához

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



```
<p><code>baseline</code><span style="vertical-align:
    ↪ align:␣baseline;">
    
    alapvonal</span></p>
```



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

3 9 3 3 3 3

minden fehér karaktert megőriz, és tördel, ha szükséges

Nem választ magától monospace karakterkészletet!

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

<p>
 Fahrenheit — Celsius

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

normal

szokásos távolság, alapértelmezés

távolság

betűk közötti távolság, negatív érték is elfogadott

normal

szokásos távolság (betűmagasság negyede), alapértelmezés

távolság

szavak közötti távolság, negatív érték is elfogadott

```
.betusuru { letter-spacing: -.1em }
.betunormal { letter-spacing: normal }
.beturitka { letter-spacing: .5em }
.szoritka { word-spacing: .5cm }
.szosuru { word-spacing: -.2em }
```

<p>Néhány szóban a betűk távolságát megnöveltük, másutt

Néhány szóban a b e t ű k t á v o l s á g á t m e g n ö v e l t ű k , másutt lecsökkentettük, de néhány helyen megőriztük az eredeti állapotot. Hasonlóan a szavak köztiszóközöket is lehetséges sűríteni, de ritkítani is.