

```
<li><a href="index.html">Home</a></li>
<li><a href="home-events.html">Home Events</a></li>
<li class="has-children"> <a href="#" class="current">Multiple Column Menu on Larger Viewports</a>
  <ul>
    <li><a href="tall-button-header.html">Tall Button Headers</a></li>
    <li><a href="image-logo.html">Image Logo</a></li>
    <li class="active"><a href="tall-logo.html">Tall Logo Images</a></li>
  </ul>
</li>
<li class="has-children"> <a href="#">Carousels</a>
  <ul>
    <li><a href="variable-width-slider.html">Variable Image Width Carousels</a></li>
  </ul>
</li>
```

# BEVEZETÉS A STÍLUSLAPOK HASZNÁLATÁBA

Abonyi-Tóth Andor,  
ELTE Informatikai Kar  
2020/2021/I.

# NÉHÁNY ALAPFOGALOM

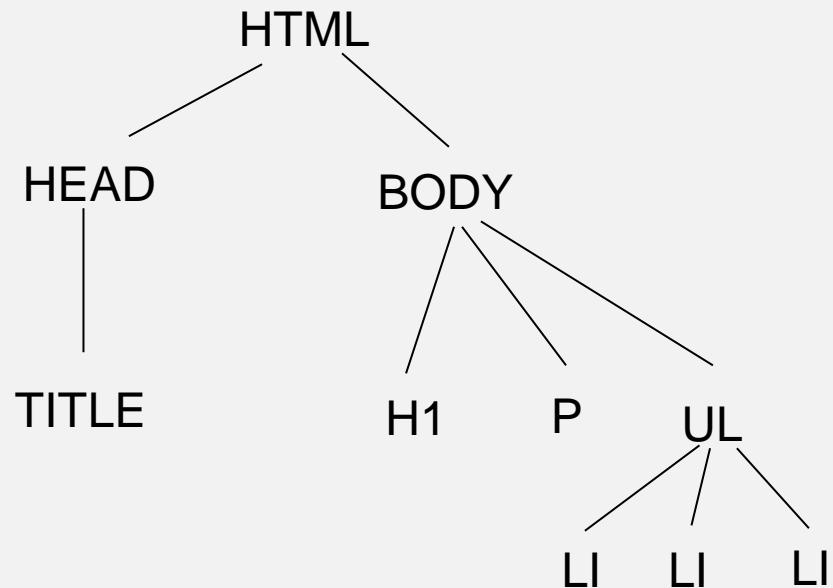
# HTML DOM (DOCUMENT OBJECT MODEL)

- W3C szabvány írja le (<https://www.w3.org/DOM>)
- Platform és nyelvfüggetlen interfész, amely a programok és szkriptek számára biztosítja a tartalom, struktúra és stílus dinamikus elérését és módosítását.
- DOM alapját az ún. DOM Core szabvány jelenti, amelyben egy csomópontokból álló általános fastruktúra ábrázolásához és működtetéséhez szükséges információk és műveletek vannak leírva.

# HTML DOM (DOCUMENT OBJECT MODEL)

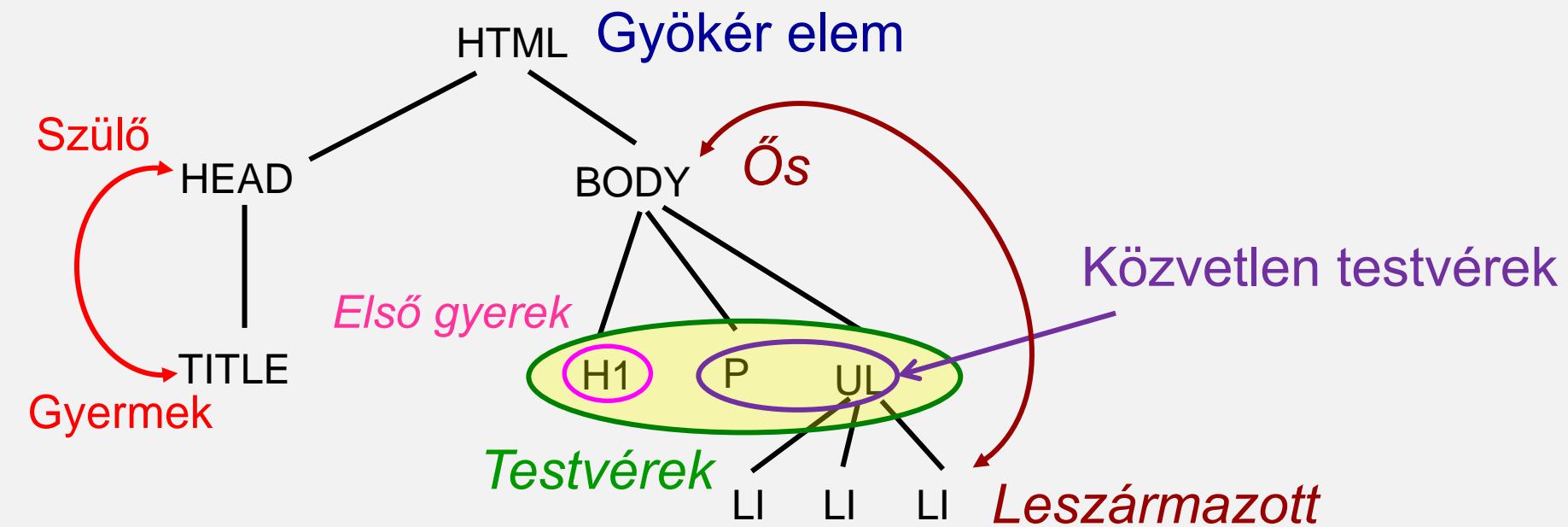
- A HTML dokumentum egy speciális elemekből álló fastruktúra, speciális többletfunkcionalitását a HTML DOM szabvány határozza meg.

```
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1>Címsor</h1>
    <p>Bekezdés</p>
    <ul>
      <li>listaelem</li>
      <li>listaelem</li>
      <li>listaelem</li>
    </ul>
  </body>
</html>
```



# HTML DOKUMENTUMFA

- Minden elemnek pontosan egy szülő eleme van, kivéve a gyökérelemet, amelynek nincs szülő eleme.



## DOKUMENTUMFA

- **Szülőnek** nevezük azokat az elemeket, amelyek tartalmaznak legalább egy másik tagot. Ezek lesznek a gyerek elemek.
- Azokat a gyermeket, amelyek közös szülővel rendelkeznek, **testvéreknek** nevezük.
- Azok a testvérek, amelyek közvetlenül egymás után helyezkednek el a dokumentumban, **közvetlen testvéreknek** nevezük.
- A fában egy elem alatti részfa elemeit az elem **leszármazottainak**, a fa az elem felett elhelyezkedő elemeit, pedig annak **őseinek** nevezük.

# A DOM MANIPULÁLÁSA JAVASCRIPTTEL

```
<!DOCTYPE html>
<html>
<body>
<p id="demo">Szöveg</p>
<script>
document.getElementById("demo").innerHTML =
"Átírtuk a bekezdés tartalmát!";
</script>

</body>
</html>
```

Átírtuk a bekezdés tartalmát!

```
<!DOCTYPE html>
<html>
<body>
<p id="p1">1. bekezdés</p>
<p id="p2">2. bekezdés</p>
<script>
document.getElementById("p2").style.color = "blue";
document.getElementById("p2").style.fontFamily = "Arial";
document.getElementById("p2").style.fontSize = "200%";
</script>

<p>A bekezdések kinézetét script segítségével
módosítottuk</p>

</body>
</html>
```

1. bekezdés

## 2. bekezdés

A bekezdések kinézetét script segítségével módosítottuk

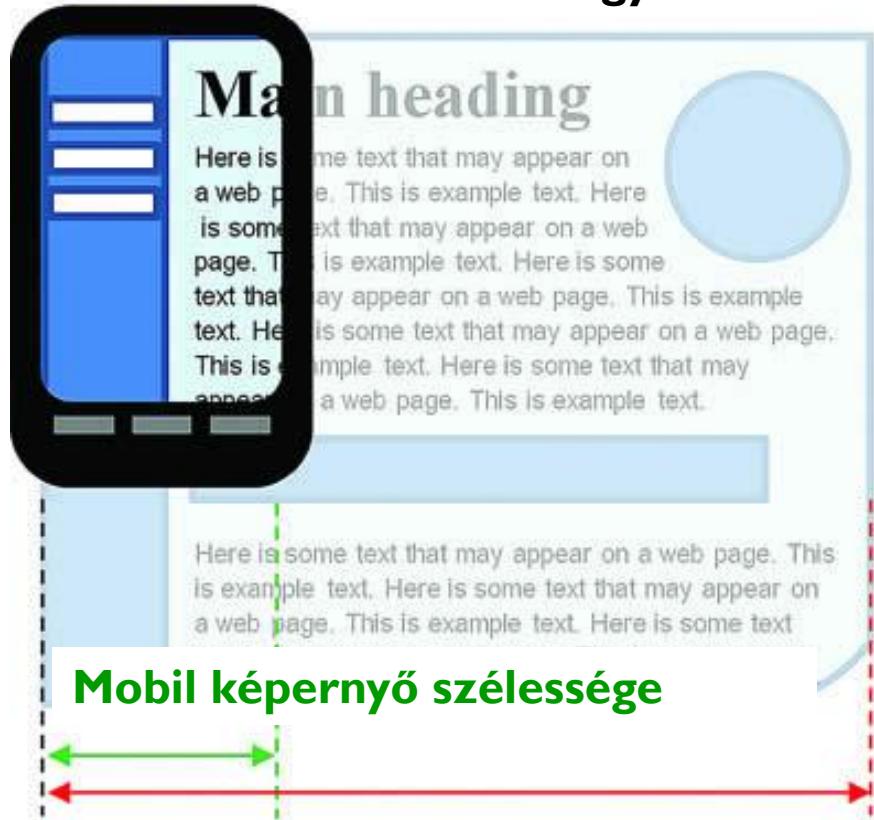
# A VIEWPORT FOGALMA

- A mobil eszközök böngészőprogramjai az oldalakat virtuális ablakokban jelenítik meg (ezt nevezzük **viewportnak**)
  - általában szélesebb, mint a képernyő, és ezen területen a felhasználók több irányban barangolhatnak, illetve egyes területekre ráközelíthetnek, illetve eltávolodhatnak.
- A **viewport** tulajdonságait `<meta>` taggel adjuk meg

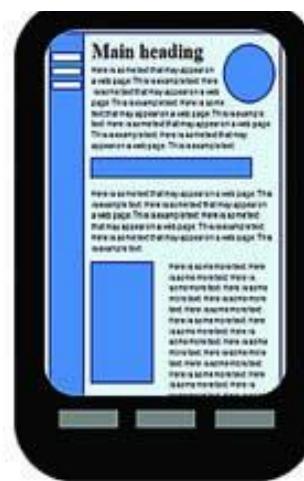
```
<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
```

# VIEWPORT PÉLDA

## Amikor ránagyítunk



## Amikor lekicsinyítjük



<https://msdn.microsoft.com/en-us/magazine/hh288079.aspx>

# VIEWPORT PARAMÉTEREK

- **width:** a viewport szélessége.
  - Ez lehet egy konkrét érték (pl. 500), de van egy speciális érték is (**device-width**), amellyel az aktuális eszköz kijelzőjének szélességét tudjuk megadni (normál nagyítási szintre vonatkozóan).
- **height:** a viewport magassága
  - A **device-height** érték is használható.
- **initial-scale:** alapértelmezett nagyítási szint, ami az oldal első betöltésekor lesz érvényes.

# BEVEZETÉS A STÍLUSLAPOK HASZNÁLATÁBA

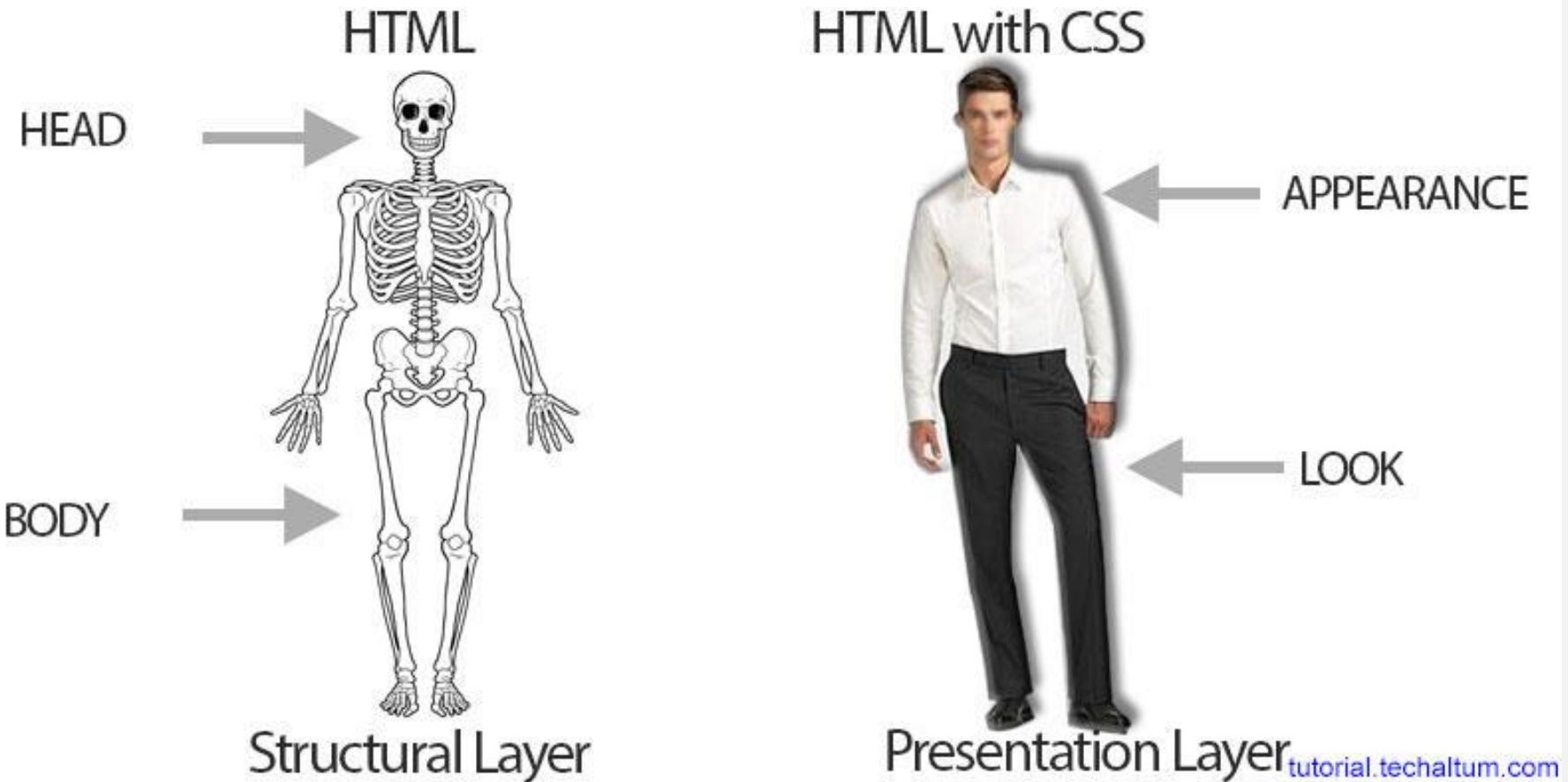
MI A CSS?



# Cascading Style Sheets

- W3C szabvány a weboldalak megjelenésének leírására
- Nyelv saját szintaxissal, szabályokkal.

# TARTALOM ↔ MEGJELENÉS



# HTML

# HTML + CSS



*Elvárás*

Akár ez is kisülhet  
belőle...  
Meg kell tanulni jól használni!

# CSS PÉLDA

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS példa</title>
<style>
    h1 {color: blue;
        background-color: lightgray}
</style>
</head>
<body>
<h1>Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
</body>
</html>
```

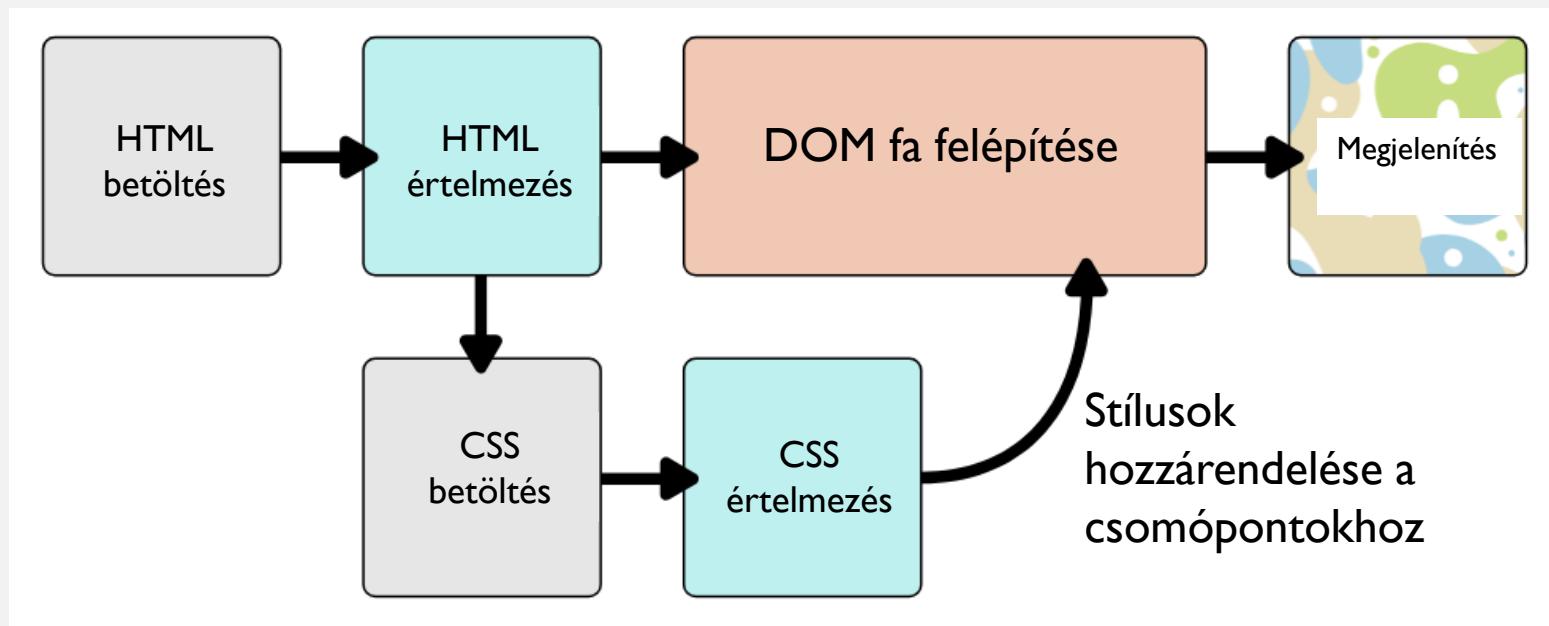
Ez a szabály azt írja le, hogy az 1-es címsorok (h1) kék színnel jelenjenek meg, és a háttérszínük világos szürke legyen.

Címsor 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

# HOGYAN MŰKÖDIK?

- Amikor egy böngésző megjelenít egy HTML oldalt, akkor legelőször betölti az oldalt a web-szerverről, majd értelmezi a jelöléseit (pl. HTML) és létrehozza a modellt a memóriában tárolt DOM formájában.
- A CSS betöltése után értelmezi azt, és az egyes stílusokat a DOM csomópontjaihoz rendeli, majd megjeleníti az oldalt.



# AZONOS TARTALOM (HTML KÓD), KÜLÖNBÖZŐ ARCULAT

- <http://www.csszengarden.com/>

This page is an archived list of Zen Garden designs added to date. You can also see the ones that [didn't make the cut](#). Total designs listed: 210

The screenshot shows the main page of the CSS Zen Garden. At the top, there's a large image of a modern building at night. Below it, the title "css zen garden" and subtitle "Beauty in CSS Design" are displayed. A message encourages users to "Select any style sheet from the list to load it into this page." Below this, a section titled "The Road to Enlightenment" discusses the goals of web enlightenment. A sidebar on the right lists various design names and their creators, such as "Under the Sea" by Eric Stoltz, "MAKE 'EM PROUD" by Michael McAgon and Scotty Reifsnyder, and "OCEANSCAPE" by Justin Gray. At the bottom, there's a section titled "So what is this about?" which provides a brief explanation of the project's purpose.



El Collar de Tomas  
MARIA STULTZ, Colombia



OrderedZen  
STEVE SMITH, United States



Bonsai  
MARTIN PLAZOTTA, Austria

The screenshot shows a different layout of the CSS Zen Garden homepage. It features a large logo at the top left and a "SELECT A DESIGN" button. Below the button, a list of designs is shown with their titles and creators. To the right, there's a section titled "The Road to Enlightenment" with a detailed explanation of web enlightenment. Further down, sections for "So What is This About?", "Participation", and "Archives" are visible. The overall design is more structured and modern than the first screenshot.

## MI AZ A „CASCADING”?



*„Az a folyamat, amikor különböző stíluslapokat kombinálunk és feloldjuk a köztük lévő konfliktusokat.”*

Håkon Wium Lie (CSS fejlesztő)

## MI AZ A „CASCADING”?



A cascade szó lépcsőzetes vizesést is jelent.

Így is lehet(ne) értelmezni:

*Hogyan csorognak le (jutnak érvényre) a felső szinten lévő stílus definíciók az alsó szinteken/lépcsőkön?*

## CSS ELŐNYEI

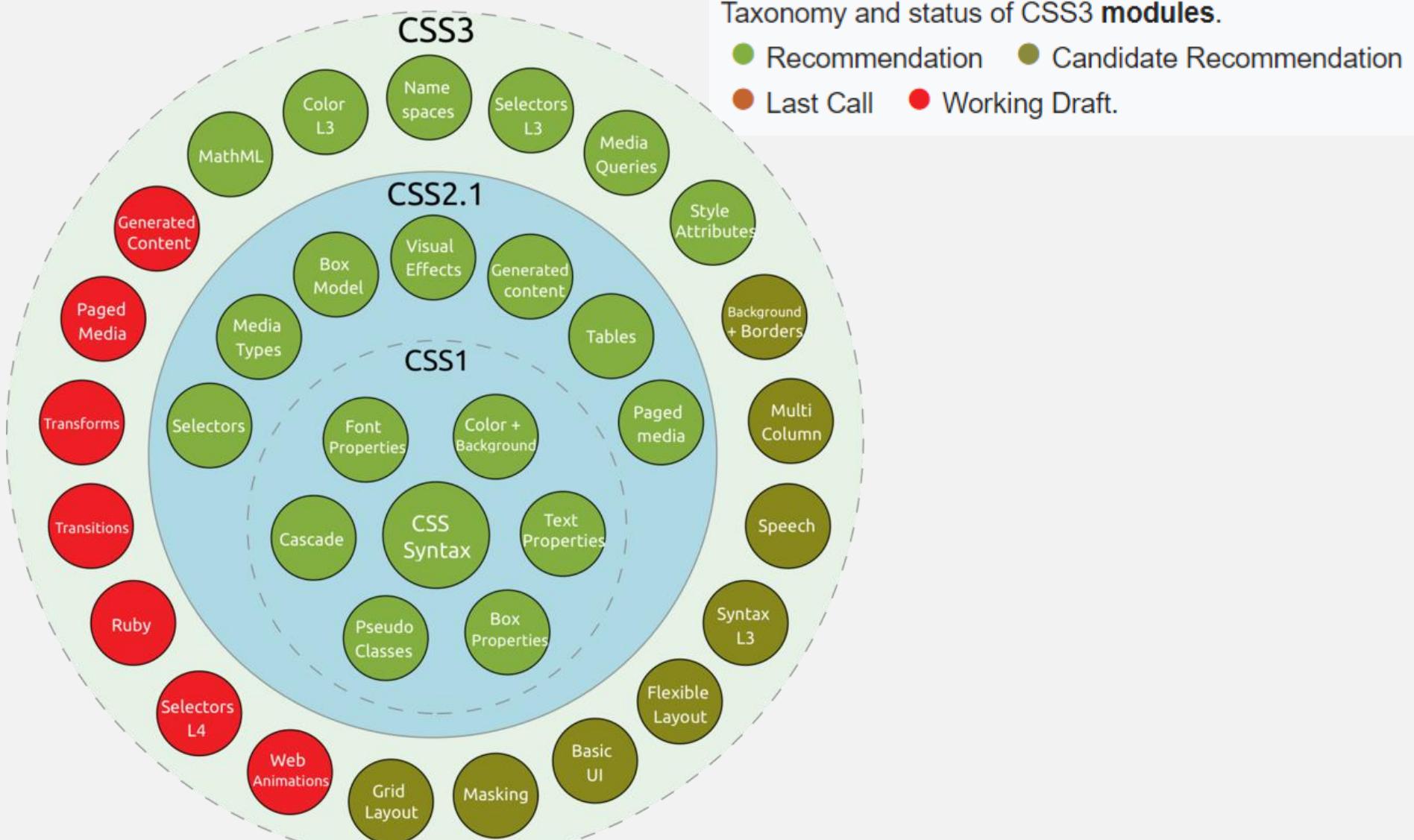


- Haladó formázási, elrendezési lehetőségek
- A website könnyebben karban tartható
- Haladó akadálymentességi technikák használata.
- Több kimenet (média típusok)
  - Képernyő (különböző felbontásokra), nyomtató, TV, projektor, stb.

# CSS VERZIÓK

- Cascading Style Sheets
  - Lépcsőzetes stíluslapok
- CSS 1. szint: 1996 decemberében jelent meg
- Cél, hogy elkülönítsük a dokumentum megjelenését és tartalmát
  - Növeli a használhatóságot, rugalmasságot, megjelenés kezelhetőségét és csökkenti a dokumentum komplexitását.
- CSS 2.szint: 1998 május
- CSS 2.1.: butított változat, igazodva a böngészők (gyenge) tudásához
- CSS 3.szint: Moduláris felépítésű. Egyes részei fejlesztés alatt...
  - Nagy részét a korszerű böngészőprogramok támogatják.

# CSS MODULOK



# CSS MODULOK

- <https://www.w3.org/Style/CSS/current-work>

## TABLE OF SPECIFICATIONS

Ordered from most to least stable:

Completed	Current	Upcoming	Notes	Info
CSS Snapshot 2010	NOTE			1 0
CSS Snapshot 2007	NOTE			1 0
CSS Color Level 3	REC	REC	See Errata	1 0
CSS Namespaces	REC	REC		1 0
Selectors Level 3	REC	REC		1 0
CSS Level 2 Revision 1	REC	REC	See Errata	1 0
CSS Level 1	REC		Unmaintained, see Snapshot	1 0
CSS Print Profile	NOTE			1 0
Media Queries	REC	REC		1 0
CSS Style Attributes	REC	REC		1 0
Stable	Current	Upcoming	Notes	Info
CSS Backgrounds and Borders Level 3	CR	PR		1 0
CSS Conditional Rules Level 3	CR	CR		1 0
CSS Multi-column Layout	CR	CR		1 0
CSS Values and Units Level 3	CR	PR		1 0
CSS Cascading and Inheritance Level 3	CR	PR		1 0
CSS Snapshot 2015	NOTE		Latest stable CSS	1 0

# STÍLUSLAP TÍPUSOK



- **A szerző által készített stíluslapok** (erről szól a kurzusunk)
- **Felhasználói stíluslapok**
  - Jellemzően fogyatékkal élő, vagy technológiailag megkülönböztetett emberek által használt lehetőség.
  - Pl. betűméret, kontraszt, színvilág felülbírálására gyengénlátó emberek számára
  - A böngészőprogramokban lehet beállítani ezeket az egyéni stíluslapokat. Sok esetben bővítményt kell/érdemes ehhez telepíteni.
- **Böngészőkben definiált stíluslapok**
  - Az elemek alapértelmezett megjelenítéséért felelősek.



# FELHASZNÁLÓI STÍLUSLAP

## NORMÁL MEGJELENÉS



## FELHASZNÁLÓI SZINTEN FELÜLÍRT MEGJELENÉS



Ha nem lenne gyengénlátóknak szóló stílusváltozat a honlapon, egy a böngészőben beállított, felhasználói stíluslappal akkor is testre tudnánk szabni a honlap megjelenését.

# CSS SZINTAXIS

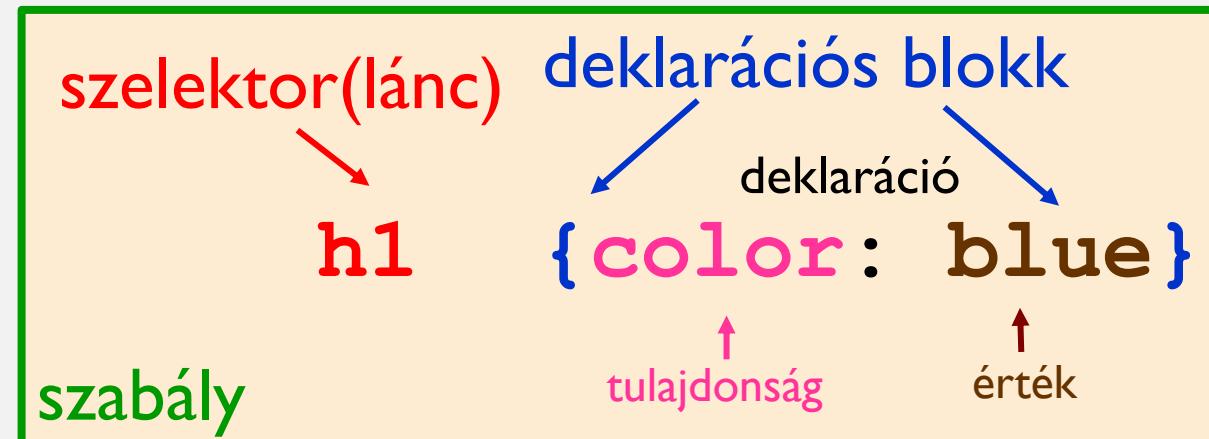
# CSS SZABÁLY (RULE)

A CSS szabály két részből áll

**A szelektor** tartalmazza a formázandó elem(ek)et.

Több elemet is fel lehet sorolni, ekkor már szelektortláncról beszélünk.

**A deklarációs blokk** tartalmazza az elemek formázásának leírását.



*Ez a szabály azt írja le, hogy az 1-es címsorok (h1) kék színnel jelenjenek meg.*

# TULAJDONSÁG: ÉRTÉK

- A tulajdonság – érték párokat kettősponttal választjuk el.
- minden deklaráció végét pontosvesszővel zárjuk le.
  - Az utolsó deklaráció végére nem kötelező kitenni a pontosvesszőt.
- A könnyebb olvashatóság/átláthatóság miatt érdemes szóközökkel, tabulátorokkal igazítani a kódot.

```
h1 {  
    color: blue;  
    font-size: 300%  
}
```

*Ez a szabály azt írja le, hogy az 1-es címsorok (h1) kék színnel jelenjenek meg, és a betűméret háromszoros legyen.*

# MEGJEGYZÉSEK

- Nyugodtan tegyünk megjegyzéseket a CSS kódba a /\* és \*/ karakterek közé.
- A kommentek hasznosak, ha később nekünk (vagy másnak) módosítania kell a kódot.

```
/* Címsorok beállítása */
h1  {
    color: blue;
    font-size: 300%
}
```



# STÍLUSLAP CSATOLÁSI LEHETŐSÉGEK

Külső stíluslap  
belinkelése HTML  
taggel

CSS beágyazása a  
HTML kódba

CSS importálása a  
CSS-en belül

Beágyazott (inline)  
stíluslap az adott  
elemre vonatkozóan

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>CSS példa</title>
    <link rel="stylesheet" type="text/css"
          href="css/alap.css" media="screen">
    <style>
      @import url("css/kontrasztos.css");
      h1 {color: blue}
    </style>
  </head>

  <body>
    <h1>Címsor 1</h1>
    <h2>Címsor 2</h2>
    <p style="color:green">Ez egy zöld
      bekezdés</p>
  </body>
</html>
```

## INLINE (BEÁGYAZOTT) MEGADÁS

```
<p style="color:green">  
  Ez egy zöld bekezdés  
</p>
```

- A **style** paraméterrel közvetlenül az adott elemet tudjuk formázni. A szabály csak az adott elemre (illetve azok leszármazottaira) lesz érvényes.
- Nem szerencsés a használata, nem hatékony, körülményes a változtatások átvezetése.

# INLINE (BEÁGYAZOTT) MEGADÁS

```
<p style="color:green">  
  Ez egy zöld bekezdés  
</p>
```

## Néhány esetben hasznos lehet

- Ha pl. egy CMS rendszerben nem módosíthatjuk a központi stíluslapot, de a megjelenést módosítani szeretnénk
- HTML formátumú e-mailek küldésénél
- Dinamikus tartalom módosításkor  
(pl. JavaScript módosítja az oldal megjelenését)

# INLINE (BEÁGYAZOTT) PÉLDA

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title>Inline példa</title>
</head>
<body>
<h1 style="color:blue">Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<h1 style="color:blue">Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<h1 style="color:blue">Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```



Ha minden 1-es címsort kékre akarunk állítani, akkor NE így csináljuk!

# MÁR NEM INLINE PÉLDA

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <title> Már nem Inline példa</title>
<style>
    h1 {color:blue}
</style>
</head>
<body>
    <h1>Címsor 1</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <h1>Címsor 1</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
    <h1>Címsor 1</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

A lapon belüli definícióval megoldottuk, hogy minden 1-es címsor kék színű legyen.

A lapon belüli definíciót külső állományba vihetjük, amit belinkelhetünk, vagy importálhatunk.

# KÜLSŐ STÍLUSLAP ALKALMAZÁSA

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
    <title>Külső CSS</title>
    <link rel="stylesheet" type="text/css" href="pelda.css">
</head>
<body>
<h1>Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<h1>Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
<h1>Címsor 1</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>
</body>
</html>
```

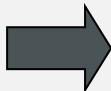
Hozzunk létre egy css állományt (pl. pelda.css) a következő tartalommal a HTML állományunk mappájában:

```
h1 {color:blue}
```

Ezzel a megoldással ugyanazt a stíluslapot több oldalhoz is hozzárendelhetjük. Ha úgy döntünk, hogy a címsor mégsem kék színű, akkor csak egy helyen kell megváltoztatni a tulajdonságot...

# DEKLARÁCIÓK CSOPORTOSÍTÁSA

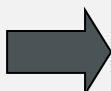
```
h1 {font-family: verdana}  
h2 {font-family: verdana}  
h3 {font-family: verdana}
```



```
h1, h2, h3 {font-family: verdana}
```

A szelektorokat vesszővel választjuk el.

```
h1 {font-family: helvetica}  
h1 {font-size: 12pt}  
h1 {font-style: normal}
```



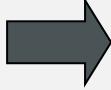
```
h1 {
```

```
    font-family: helvetica;  
    font-size: 12pt;  
    font-style: normal;
```

```
}
```

A tulajdonság:érték párok után pontosvessző áll.

```
h1 {font-weight: bold}  
h1 {font-size: 12pt}  
h1 {font-family: helvetica}
```



```
h1 {font: bold 12pt helvetica}
```

Itt az egyes értékek között szóköz van.  
(shorthand megadásnak nevezi az angol szabvány)

# ISMERETLEN TULAJDONSÁGOK

- A böngészők figyelmen kívül hagyják az ismeretlen vagy hibás tulajdonságokat.

```
img { float: left }          /* helyes CSS 2.1 deklaráció */  
img { float: left here }    /* a „here” hibás érték */  
img { background: "red" }    /* az értéket nem szabad idézőjelbe tenni */  
img { border-width: 3 }      /* hiányzik a mértékegység */
```

Amit a fentiekből a CSS értelmező figyelembe vesz:

```
img { float: left }  
img {}  
img {}  
img {}
```

# LEZÁRATLAN STRING

A szóközt tartalmazó betűcsaládok nevét aposztrófok közé kell zárni!

Azonban ebben a példában hiányzik a záró aposztróf a Times szöveg után.

```
p { color: green;  
    font-family: 'Courier New Times  
    font-style: italic;  
    background-color: yellow; }
```

A megnyitott, de le nem zárt idézőjelek, aposztrófok problémát okoznak. Amit a fentiekből a CSS értelmező figyelembe vesz:

```
p { color: green;  
    background-color: yellow; }
```

# STÍLUSLAPOK VALIDÁLÁSA

- <http://jigsaw.w3.org/css-validator/>

The screenshot shows the W3C CSS Validation Service interface. At the top, there is a blue header bar with the W3C logo and the text "CSS Validation Service" and "Cascading Style Sheets (CSS) és stíluslapokkal rendelkező (X)HTML dokumentumok ellenőrzése". Below the header, there are three tabs: "URI segítségével" (selected), "File feltöltéssel", and "Közvetlen bevitellel". A large input field labeled "Érvényesítés URI segítségével" contains the placeholder text "Adja meg a dokumentum (HTML CSS-sel vagy csak CSS) URI-jét, melyet érvényesíteni szeretne:". Below this input field is a "Cím:" label followed by an empty text input. A blue button labeled "További lehetőségek" is located below the input field. At the bottom center is a rounded rectangular button labeled "Ellenőrzés".

# MÉRTÉKEGYSÉGEK

## RELATÍV MÉRTÉKEGYSÉGEK

Egy másik tulajdonság értékétől függnek, például a szülő elem méretétől, vagy az ott alkalmazott betűmérettől.

**300%**

## ABSZOLÚT MÉRTÉKEGYSÉGEK

Az érték nem függ más elemek értékétől, konkrét méretet jelöl (pl. cm)

**2cm**

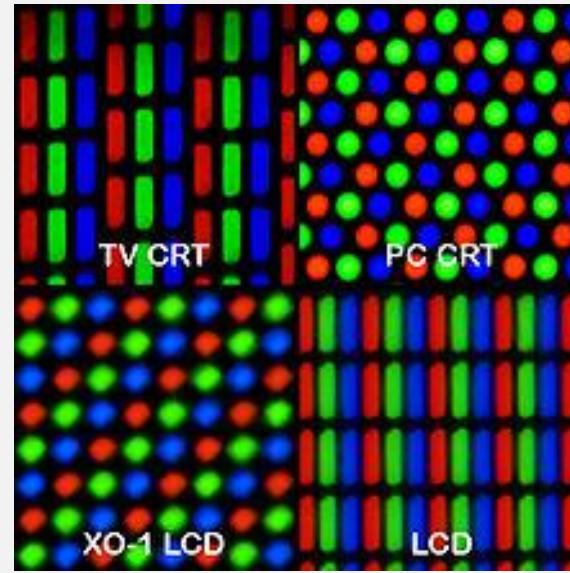
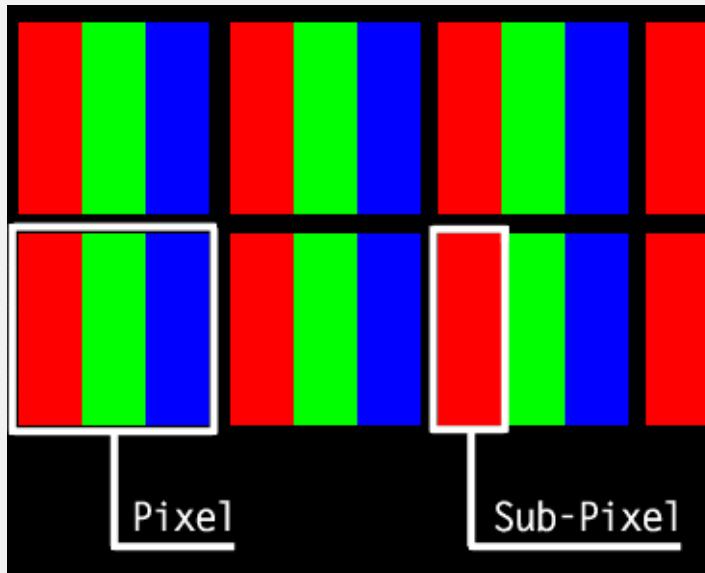
## MÉRTÉKEGYSÉGEK (CSS3)

A számokat egybeírjuk a mértékegységekkel! Vagyis 12px és nem 12 px.

Abszolút	Relatív
<ul style="list-style-type: none"><li>• <b>in</b> (=2.54 cm)</li><li>• <b>cm</b></li><li>• <b>mm</b></li><li>• <b>pt</b> (=1/72in)</li><li>• <b>pc</b> (=12pt)</li><li>• <b>px</b> (képpont)</li></ul>	<ul style="list-style-type: none"><li>• <b>em</b> (adott környezet betűméretéhez képest történő megadás. pl. 1.2em)</li><li>• <b>%</b> (relatív értéket jelent, de hogy minek az értékét, az változó. Lehet: azonos elem más tulajdonsága, szülő elem egy tulajdonsága, a formázott tartalom egy tulajdonsága (pl. tartalmazó elem szélessége). pl. 150%)</li><li>• <b>ex</b> (x-magasság - többnyire az adott font x betűjének magasságához képest történő megadás. pl. 2ex)</li><li>• <b>rem</b> (a gyökér elem betűméretéhez képest történő relatív méretmegadás pl. 1.5rem)</li><li>• <b>ch</b> (Az adott betűcsalád 0 karakterének szélességéhez történő relatív megadás. pl. 2ch)</li><li>• <b>vw</b> (A viewport szélességéhez igazodik. 1vw a viewport szélesség 1%-ának felel meg.)</li><li>• <b>vh</b> (A viewport magasságához igazodik. 1vh a viewport magasság 1%-ának felel meg.)</li></ul>

# A PIXEL RELATÍV VAGY ABSZOLÚT MÉRTÉKEGYSÉG?

- A px (pixel, képpont) a CSS2 szabványban relatívként szerepelt, a CSS3-ban abszolút mértékegység.
- Hardveres felfogásban a pixel a kijelző legkisebb egysége, amely rendszerint RGB (vörös, zöld, kék) alpixelekből áll.



Alpixelek különböző elrendezése

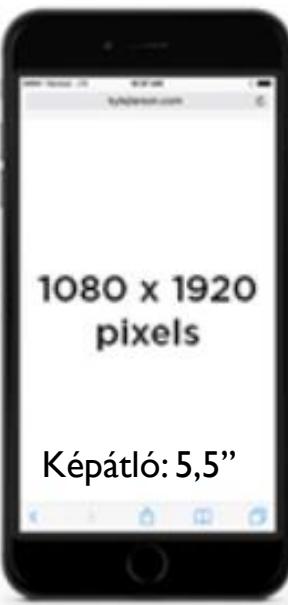
# NE A FIZIKAI FELBONTÁSBÓL INDULJUNK KI!

IPHONE 6 PLUS



**Physical Pixels**

the number of actual pixels in the display



SAMSUNG SYNCMASTER  
S24D330H



Képátló

24"

Felbontás

1920x1080

**Pixelsűrűség (pixel per inch)**

**400ppi**

**92ppi**

# IPHONE FELBONTÁSOK ÉS VIEWPORT MÉRETEK

## iPhone felbontások

1st Gen, 3G & 3GS

**320×480**

4 & 4S

**640×960**

5, 5C & 5S

**640×1136**

6

**750×1334**

6 Plus

**1242×2208**

downsampled to

**1080×1920**

More about the crazy  
downsampling business [here](#)  
and [here](#).

## Viewport méretek

320x480

320x480

320x568

375x667

414x736



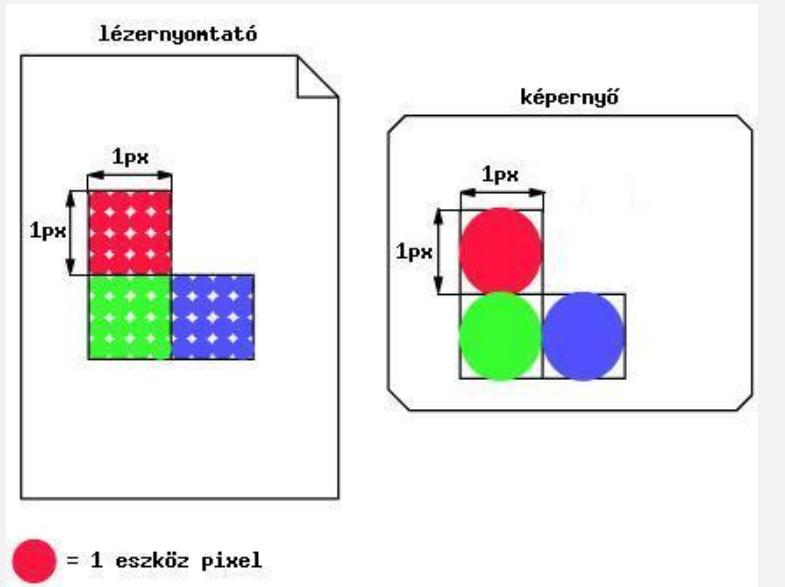
# REFERENCIA PIXEL

- A CSS szabvány **nem a hardveres pixellel** számol, hanem úgynévezett **referencia pixellel**.
- Amennyiben az eszköz képpontsűrűsége nagy mértékben eltér a tipikus monitor képpontsűrűségétől, a böngészőnek át kell méreteznie a pixel értékeket.
- A referencia pixelt egy 96dpi képpontsűrűségi eszköz egy képpontjának szokásos olvasási távolságához (kb. 71 cm) tartozó látószöggel definiálják.

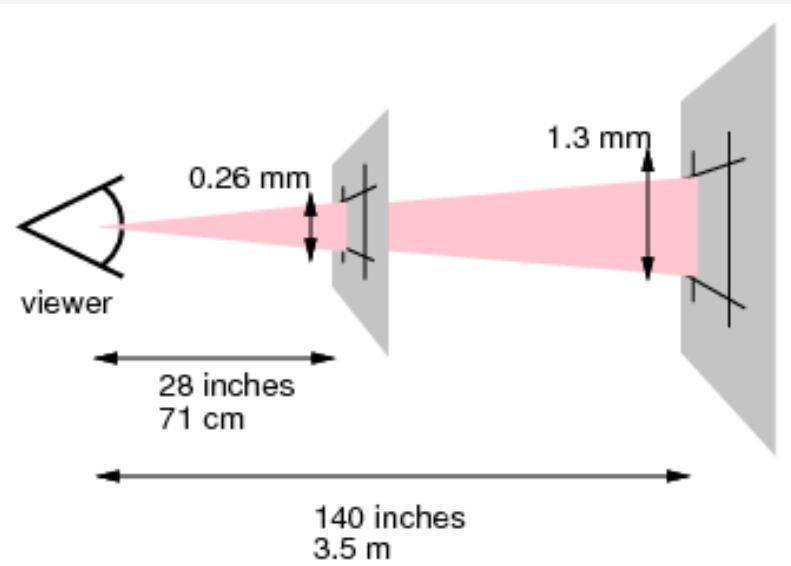
<https://www.w3.org/TR/css-values/#reference-pixel>,

<https://alistapart.com/article/a-pixel-identity-crisis/>

# REFERENCIA PIXEL



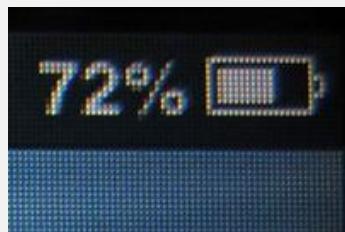
Egy lézernyomtató a képernyő pixelt több pontból fogja nyomtatni, attól függően, hogy mekkora a pontsűrűsége.



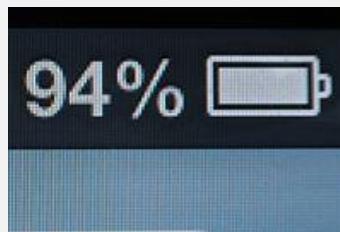
A kb. 71 cm-es olvasási távolság 0,26 mm pixelméretet eredményez, a 3,5 méteres távolság már 1,3 mm-t.

# RETINA MEGJELENÍTŐ

A retina megjelenítő képpontsűrűsége jóval nagyobb, mint elődjeié. Itt már a pixeleket (tipikus olvasási távolságból) az emberi szem nem képes megkülönböztetni.



Nem retina  
megjelenítő



Retina  
megjelenítő

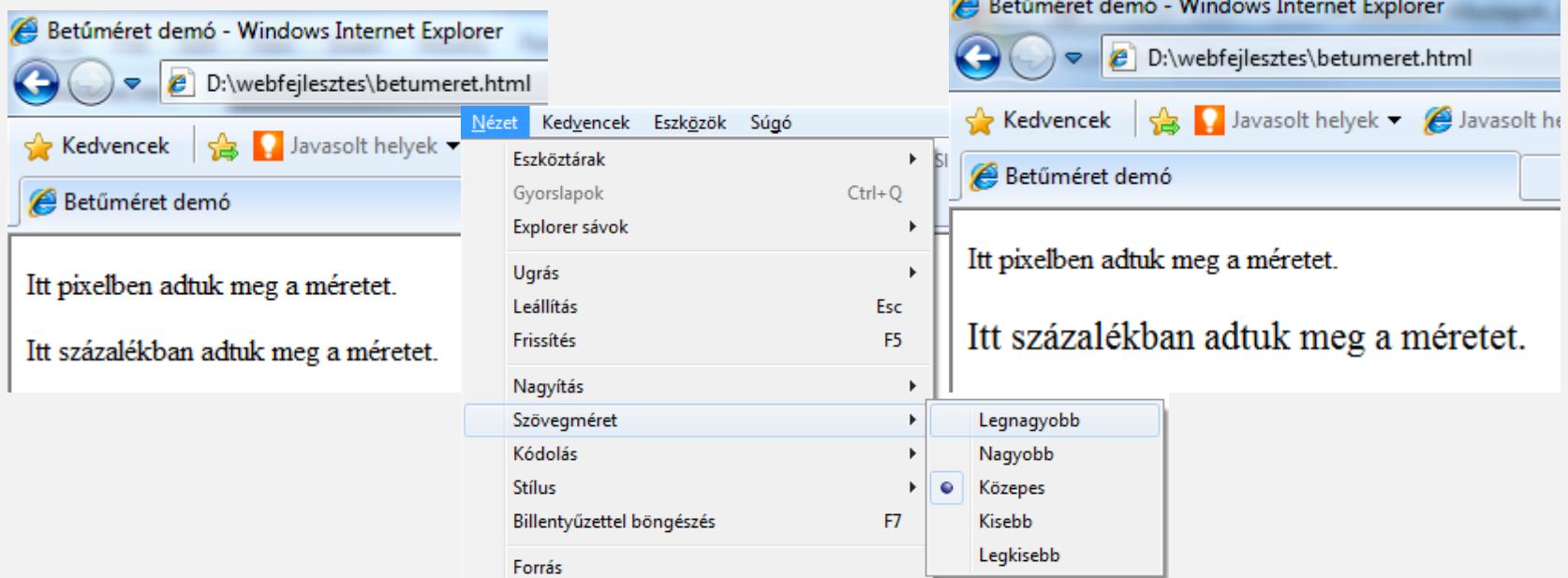


[https://en.wikipedia.org/wiki/Retina\\_display](https://en.wikipedia.org/wiki/Retina_display)

<https://deviceatlas.com/blog/mobile-viewport-size-statistics-2017>

# BETŰMÉRET MEGADÁS PIXELBEN

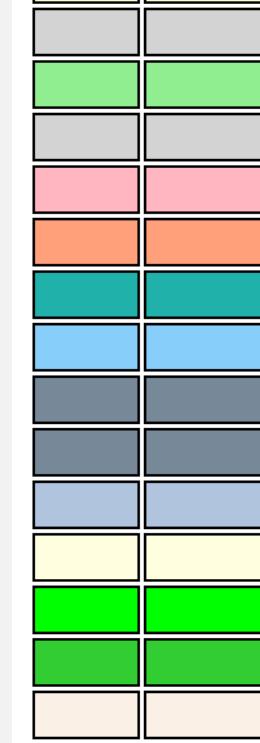
- Betűméret megadásnál lehetőleg ne használjuk, mert nem lesz ( minden böngészőben ) nagyítható a betűméret, ami a gyengénlátó felhasználóknak hátrányos.



# SZÍNJELÖLÉSEK, SZÍN NEVEK

A színnevek teljes lista a <https://www.w3.org/TR/css-color-3/#svg-color> címen elérhető.

Named	Numeric	Color name	Hex rgb	Decimal
		<i>black</i>	#000000	0,0,0
		<i>silver</i>	#C0C0C0	192,192,192
		<i>gray</i>	#808080	128,128,128
		<i>white</i>	#FFFFFF	255,255,255
		<i>maroon</i>	#800000	128,0,0
		<i>red</i>	#FF0000	255,0,0
		<i>purple</i>	#800080	128,0,128
		<i>fuchsia</i>	#FF00FF	255,0,255
		<i>green</i>	#008000	0,128,0
		<i>lime</i>	#00FF00	0,255,0
		<i>olive</i>	#808000	128,128,0
		<i>yellow</i>	#FFFF00	255,255,0
		<i>navy</i>	#000080	0,0,128
		<i>blue</i>	#0000FF	0,0,255
		<i>teal</i>	#008080	0,128,128
		<i>aqua</i>	#00FFFF	0,255,255

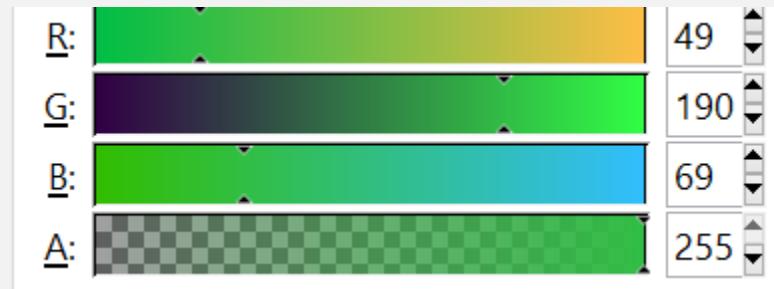
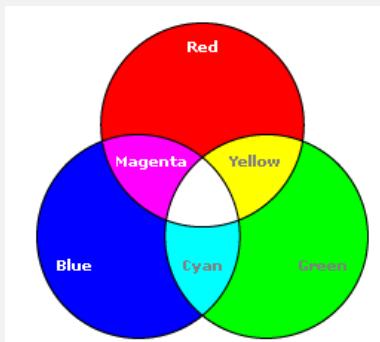


<i>lightgray</i>	#D3D3D3	211,211,211
<i>lightgreen</i>	#90EE90	144,238,144
<i>lightgrey</i>	#D3D3D3	211,211,211
<i>lightpink</i>	#FFB6C1	255,182,193
<i>lightsalmon</i>	#FFA07A	255,160,122
<i>lightseagreen</i>	#20B2AA	32,178,170
<i>lightskyblue</i>	#87CEFA	135,206,250
<i>lightslategrey</i>	#778899	119,136,153
<i>lightslategrey</i>	#778899	119,136,153
<i>lightsteelblue</i>	#B0C4DE	176,196,222
<i>lightyellow</i>	#FFFFE0	255,255,224
<i>lime</i>	#00FF00	0,255,0
<i>limegreen</i>	#32CD32	50,205,50
<i>linen</i>	#FAF0E6	250,240,230

```
<p style="color:navy">
```

# RGB SZÍN-KOORDINÁTARENDSZER

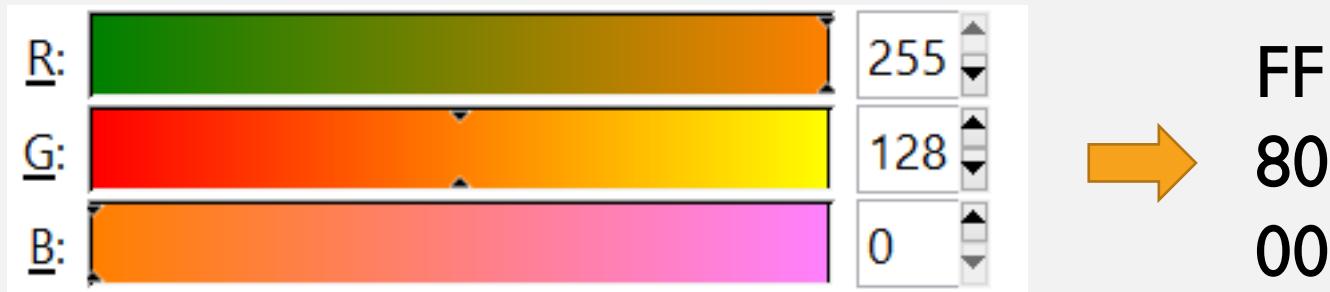
- A műszaki életben leggyakrabban használt színt koordinátarendszer, mert a színeknek a képernyőn történő megvalósításával kapcsolatos.
- Képernyőn minden színt az RGB alapszínek additív keverésével állítanak elő. A képernyőn a három alapszín úgy van értelmezve, hogy
  - $R + G + B = 1$  : fehér szín
  - $R + G + B = 0$  : fekete szín



Az opcionális „A” érték az Alfa csatornára utal, vagyis az átlátszatlanság mértékére.

## RGB KÓD 16-OS (HEXADECIMÁLIS) SZÁMRENDSZERBEN

- Az R,G,B értékek 0 és 255 között adhatóak meg.
- Ha tömörebb formában akarjuk megadni, akkor a decimális kódot át lehet számolni 16-os számrendszerbe. Ilyenkor a kód elé # jelet kell tenni.

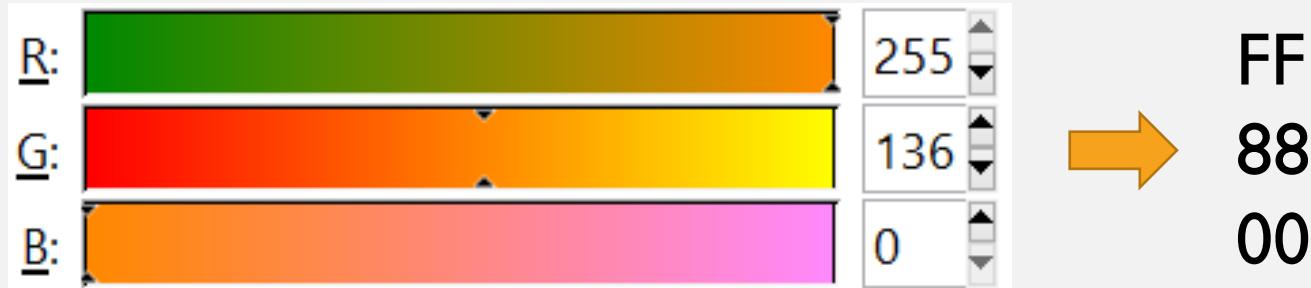


```
color:  
rgb(255,128,0)
```

```
color: #ff8000
```

## RGB KÓD 16-OS (HEXADECIMÁLIS) SZÁMRENDSZERBEN

- Ha az R,G,B kódban ugyanazok a számjegyek ismétlődnek, akkor azokat lehet rövidíteni, vagyis egyszer leírni.



color:  
rgb(255, 136, 0)

color: #ff8800  
color: #f80

## PÉLDA (EGY HEXA SZÍNMEGADÁS)



```
#chucknorris {  
color: #BADA55;  
}
```

<https://speckyboy.com/css-puns-jokes/>



<https://www.teepublic.com/t-shirt/23190-css-shirt>

# ÁTLÁTSZATLANSÁG (ALFA CSATORNA)

- Az alfa csatorna értéke 0 és 1 közti szám lehet
  - 0: teljesen átlátszó
  - 1: teljesen átlátszatlan

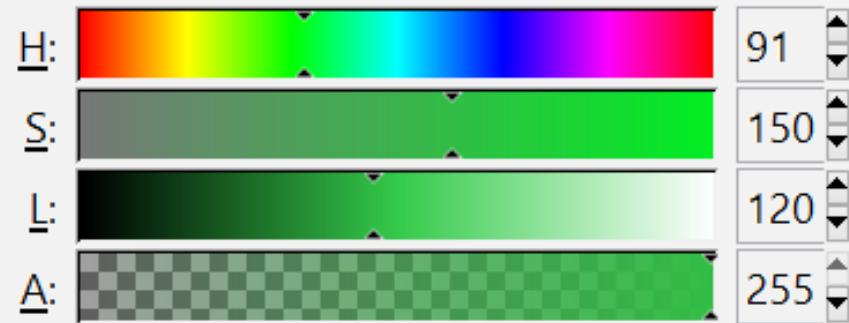


# HEADING 1

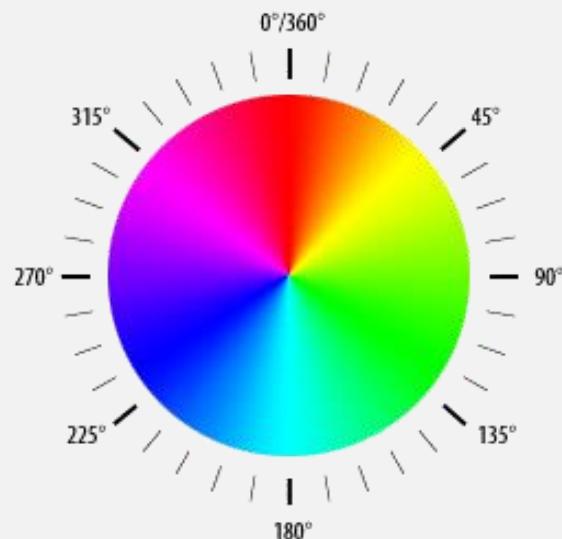
```
h1 {  
    color: rgb(0, 0, 0);  
    background-color: rgba(255, 255, 255, 0.5);  
}
```

# HSL SZÍN-KOORDINÁTARENSZER

- Jóval intuitívebb, mint az RGB
- HSL/HSLA:
  - H: árnyalat
  - S: a telítettség,
  - L: fényesség



Az „A” érték az Alfa csatornára utal, vagyis az átlátszatlanság mértékére.



A színkör ellentétes oldalain a kiegészítő (komplementer) színek találhatóak.

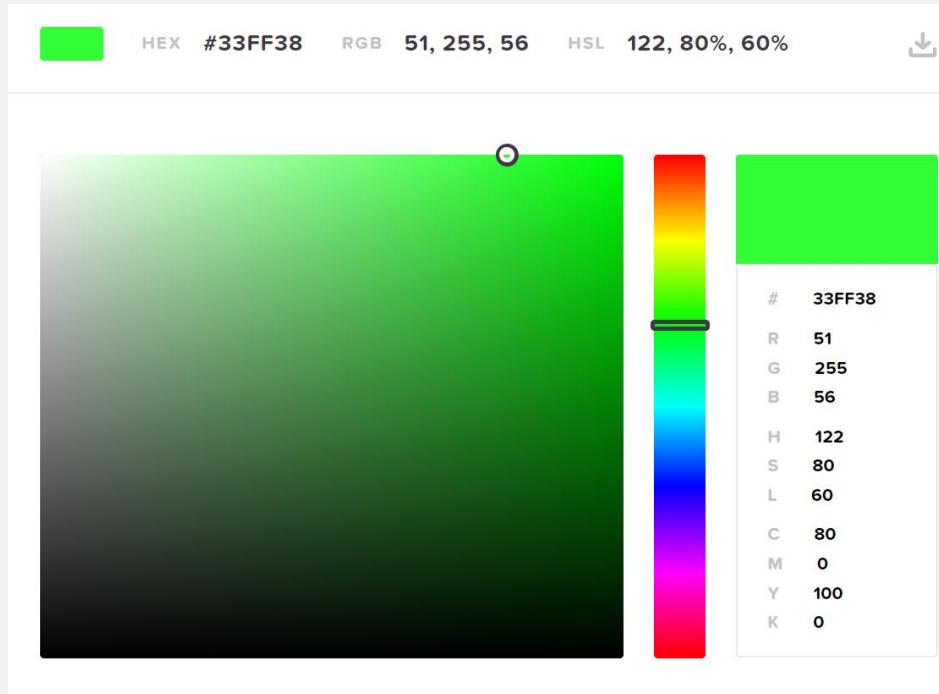
# SZÍNJELÖLÉSEK

- Szín megadás **decimális RGB/RGBA** kóddal
  - <p style="color:rgb(51, 71, 28)>
  - <p style="color:rgb(51, 71, 28, 0.5)>
- Szín megadás **hexadecimális RGB** kóddal
  - <p style="color:#f66">
  - <p style="color:#BADA55">
- Szín megadás **decimális HSL/HSLA** kóddal
  - <p style="color:hsl(300, 50, 70)>
  - <p style="color:hsla(300, 50, 70, 1)>
- Szín megadás **százalékosan**
  - <p style="color:rgb(50%, 70%, 30%)>

<http://html-color-codes.com/>

# SZÍNKÓDOK MEGHATÁROZÁSA

- Szinte bármelyik grafikus rajzoló, képmanipuláló program alkalmas a színek meghatározására.
- Hasznos online eszköz:
  - <https://htmlcolorcodes.com/color-picker/>



# FALLBACK

- Érdemes gondoskodnunk arról, hogy az átlátszóságot nem ismerő (nagyon régi) böngészőprogramok is megjelenítsenek egy színt.
- A lenti példában az rgba-t nem ismerő böngészőkben beállításra kerül egy szín, amelyet felülírunk később, de ezt csak az rgba-t ismerő böngészők veszik figyelembe.

## CSS

```
div {  
    background: rgb(200, 54, 54); /* The Fallback */  
    background: rgba(200, 54, 54, 0.5);  
}
```

<http://css-tricks.com/rgba-browser-support/>

# URL MEGADÁS

- URL megadására olyan esetben lesz szükség, amikor például háttérképet, külső stíluslapot akarunk beállítani.
- Használata: url ('url') ; vagy url ("url") ;

## Példa

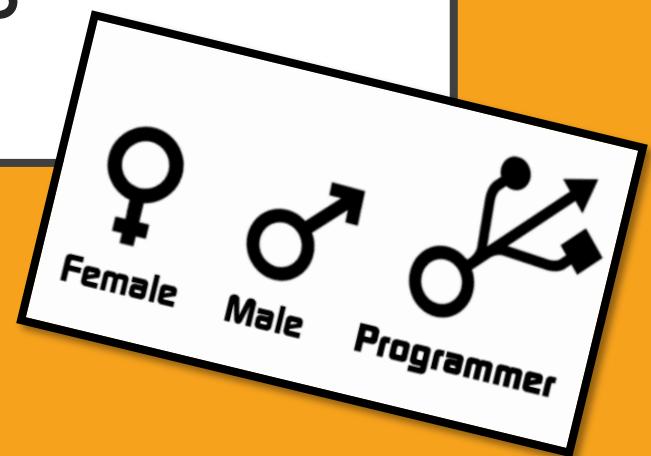
```
body {  
background-image: url ('kepek/hatter.jpg');  
}  
  
li {  
list-style: url ('http://v.hu/kep.png') disc  
}
```

## A KEZDETI (INITIAL) ÉRTÉK

- minden tulajdonságnak van egy kezdeti (initial) értéke, amely egy tulajdonság definíciós táblában van elhelyezve.
- Ha vissza akarjuk állítani az alapértéket, de nem tudjuk, hogy az pontosan micsoda, használjuk az initial értéket.

```
div {min-width:initial; }
```

# ÖRÖKLŐDÉS



# ÖRÖKLŐDÉS

- A tulajdonságok (egy része) öröklődik a szülő elem(ek)től

```
<h1 style="color:blue">  
A címsor mindig  
<em>fontos</em>.</h1>
```



A címsor mindig **fontos**.

Az em elem tartalma is kék lett, mert  
örökölte a színt a szülőtől...

- Ebből következik, hogy a teljes lapra vonatkozó tulajdonságokat a BODY elemhez érdemes kötni.

```
body { font-size:12pt;  
       font-family: Arial, Verdana, sans-serif  
 }
```

# ÖRÖKLŐDÉS

- A gyermek elemek nem a szülőkben megadott relatív értékeket öröklik, hanem azok számított értékét!

```
body
{
    font-size: 12px;
    text-indent: 3em; ←
}
h1 { font-size: 25px }
```

A bekezdés első sorának behúzását jelenti.  
Itt ez most 36px-nek felel meg.

A h1 örökli a text-indent értéket a szülőtől (body), de csak a számított értéket. Vagyis a behúzás nem  $3 \cdot 25\text{px} = 75\text{px}$ , hanem 36px lesz.

# HA NEM ÍGY LENNE...

## Címsor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in purus augue. In hac habitasse platea dictumst. Sed fringilla, velit sed commodo feugiat, mi leo porttitor sapien, non rutrum est nisi vitae purus. Donec eget elit non ipsum vulputate imperdiet. Proin scelerisque porta mi, in dignissim sapien interdum vitae. Sed non ante sit amet tellus accumsan placerat non in elit.

## Címsor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in purus augue. In hac habitasse platea dictumst. Sed fringilla, velit sed commodo feugiat, mi leo porttitor sapien, non rutrum est nisi vitae purus. Donec eget elit non ipsum vulputate imperdiet. Proin scelerisque porta mi, in dignissim sapien interdum vitae. Sed non ante sit amet tellus accumsan placerat non in elit.

A számított érték  
öröklése miatt az első  
sor behúzása független  
lesz a betűmérettől...

Ha nem számított értéket  
örökölne a gyerek, akkor a  
betűméret hatással lenne az  
első sor behúzására is, ami  
nem kívánt eredménnyel  
járna.

## Címsor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in purus augue. In hac habitasse platea dictumst. Sed fringilla, velit sed commodo feugiat, mi leo porttitor sapien, non rutrum est nisi vitae purus. Donec eget elit non ipsum vulputate imperdiet. Proin scelerisque porta mi, in dignissim sapien interdum vitae. Sed non ante sit amet tellus accumsan placerat non in elit.

## Címsor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in purus augue. In hac habitasse platea dictumst. Sed fringilla, velit sed commodo feugiat, mi leo porttitor sapien, non rutrum est nisi vitae purus. Donec eget elit non ipsum vulputate imperdiet. Proin scelerisque porta mi, in dignissim sapien interdum vitae. Sed non ante sit amet tellus accumsan placerat non in elit.

# ÖRÖKLŐDÉS ÉS AZ INHERIT ÉRTÉK

- Nem minden beállítás öröklődik.
  - Pl. a szegély (border) nem öröklődik, de ezt felülbírálhatjuk az `inherit` értékkel.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<style type="text/css">
  div {border:1px solid blue;padding:5px;}
  p {border:inherit;}
</style>
</head>
<body>
<div>
  <p>Lorem ipsum dolor sit amet...</p>
  <p>Nam id nisl vitae neque posuere
interdum...</p>
</div>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean in dolor a tellus feugiat molestie sed quis nisi. Vivamus augue nisi, imperdiet id consequat id, luctus ac velit.

Nam id nisl vitae neque posuere interdum. Quisque pulvinar fermentum metus, at imperdiet purus accumsan blandit. Nunc bibendum, nibh vitae lobortis varius, erat eros elementum felis, a tempor nisi lectus at turpis. Praesent sodales mauris ac massa dignissim fermentum.

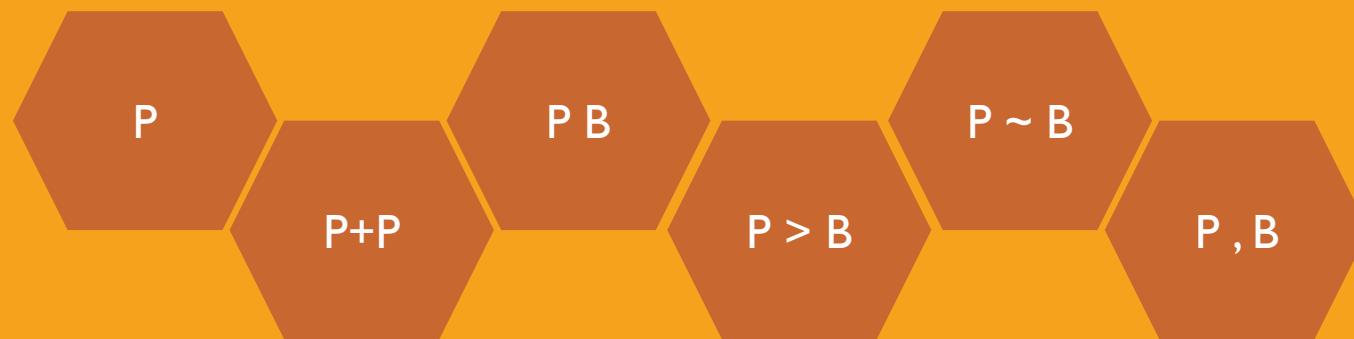
A bekezdéseken alapból nem lenne szegély.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean in dolor a tellus feugiat molestie sed quis nisi. Vivamus augue nisi, imperdiet id consequat id, luctus ac velit.

Nam id nisl vitae neque posuere interdum. Quisque pulvinar fermentum metus, at imperdiet purus accumsan blandit. Nunc bibendum, nibh vitae lobortis varius, erat eros elementum felis, a tempor nisi lectus at turpis. Praesent sodales mauris ac massa dignissim fermentum.

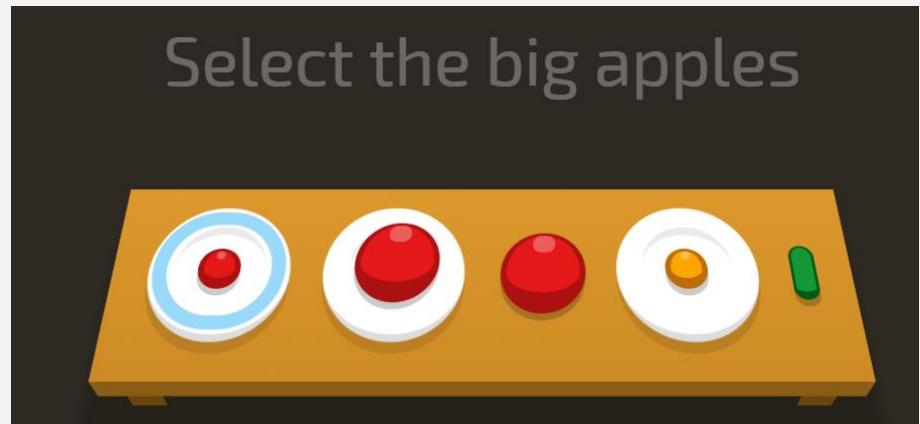
Az `inherit` érték miatt a bekezdés is örökli a szülő elem szegély beállításait.

# SZELEKTOROK (KIJELÖLŐK)



# SZELEKTOROK GYAKORLÁSA

- Játékos gyakorlás  
(CSS diner)
- <https://flukeout.github.io/>



CSS Editor	style.css	HTML Viewer	table.html
1 Type in a CSS selector			
2 {			
3 /* Styles would go here. */			
4 }			
5			
6 /*			
7 Type a number to skip to a level.			
8 Ex → "5" for level 5			
9 */			
10			
11			
12			
13			

The CSS Editor shows a placeholder 'Type in a CSS selector' with an 'enter' button. The style.css file contains a single brace {}, followed by a note about skipping levels with a comment /\* and an example Ex → "5" for level 5. The table.html file shows the HTML structure of the game board, including a table element containing several plate and apple elements, along with an orange and a pickle.

# ÖSSZEKAPCSOLT SZELEKTOR (LESZÁRMAZÁS)

- Ha más tulajdonságokat akarok adni az elemnek, attól függően, hogy milyen elembe van beágyazva, összekapcsolhatom a szelektorokat
- Mód: **szelektor1 szelektor2 {definíció}**

## Példa

```
h1 {color: blue}  
em {color: blue}
```

```
<h1>Ez nagyon <em>fontos</em> dolog</h1>  
<p> Ez is <em>fontos</em></p>
```

Ez nagyon fontos dolog  
Ez is fontos.

```
h1 em {color: red}
```

Ez nagyon fontos dolog  
Ez is fontos.

## OSZTÁLYOK (CLASS) HASZNÁLATA

- Akkor használjuk, ha bizonyos tulajdonságokat több elemnél is be akarunk állítani. Pl. szegélyezés, igazítások.
- Létrehozhatunk általános osztályt, amelyet bármilyen elemnél felhasználhatunk.
- Adott szelektorhoz is köthetjük az osztályt.

# OSZTÁLYOK (CLASS) HASZNÁLATA

- **CSS szinten**

- `.osztálynév {definíció}`
  - Ez egy általános osztály, amelyet több elemnél is felhasználhatunk.
- `szelektor.osztálynév {definíció}`
  - Ekkor egy adott szelektorhoz lesz kötve az osztály

```
.fontos {color:red;}  
  
p.szegely  
{border:1px solid black;}
```

- **HTML szinten**

- **class attribútum használata**
- **Egy elem több osztályba is tartozhat**

```
<p class="fontos">  
<p class="szegely">  
<p class="szegely fontos">
```

# OSZTÁLYOK (CLASS) HASZNÁLATA

```
<style type="text/css">  
p {padding:5px; margin:10px; }  
.fontos {color:red;}  
p.szegely {border:1px solid black;}  
h1.szegely {border:2px dotted blue;}  
</style>
```

...

```
<h1>Címsor 1</h1>  
<p class="fontos">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.</p>  
<p class="szegely">Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.</p>  
<p class="szegely fontos">Lorem ipsum dolor sit  
amet, consectetur adipiscing elit.</p>  
<h1 class="szegely fontos">Címsor 1</h1>  
<p>Lorem ipsum dolor sit amet, consectetur  
adipiscing elit.</p>
```

## Címsor 1

  Lorem ipsum dolor sit amet, consectetur adipiscing elit.

  Lorem ipsum dolor sit amet, consectetur adipiscing elit.

  Lorem ipsum dolor sit amet, consectetur adipiscing elit.

## Címsor 1

  Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Több osztályba is tartozhat  
ugyanaz az elem!

# TÖBB OSZTÁLY HASZNÁLATA

```
<style type="text/css">  
h1.szegely {border:2px dotted blue;}  
.fontos {color:red;}  
h1.szegely.fontos {border:1px solid red; color:black;}  
</style>
```

...

```
<h1 class="szegely">Címsor 1</h1>
```

```
<h1 class="fontos">Címsor 1</h1>
```

```
<h1 class="szegely fontos">Címsor 1</h1>
```

Címsor 1

Címsor 1

Címsor 1



Alap esetben kék szegéllyel és piros színnel jelenne meg.  
A .szegely.fontos CSS definíció miatt más lesz a megjelenése.

# EGYEDI AZONOSÍTÓK (ID) HASZNÁLATA

- CSS szinten
  - **#azonosító**, vagy **szelektor#azonosító**

```
div#fejlec {margin:10px}
```
- HTML szinten
  - id attribútum használata 

```
<div id="fejlec"></div>
```
  - Egyedinek kell lennie az oldalon belül!
  - Fontos a kis- és nagybetű közti különbség.
  - HTML 4.01 szerint:
    - Betűvel kezdődik, amelyet számok, betűk és a következő jelek követhetnek: - \_ :.
    - HTML 5: legalább 1 karakter, nem tartalmazhat szóközt.

# ID ÉS CLASS ALKALMAZÁSA UGYANAZON ELEMEN

```
<style type="text/css">  
#oldalfejlec {height:150px;  
               background-color:lightgray;}  
  
#tartalom {height:auto;  
            background-color:white;}  
  
#oldallablec {height:50px;  
               background-color:lightgray;}  
  
.kozep {text-align:center;}  
</style>
```

**id** és **class** besorolás is lehet egyszerre. Ilyenkor a rangsorolási szabályok döntenek az ellentmondó deklarációk érvényre jutásáról.

```
<header id="fejlec" class="kozep">Fejléc</header>  
<main id="tartalom">Tartalom Lorem ipsum... </main>  
<footer id="oldallablec" class="kozep">Lábléc</div>
```

Fejléc

Tartalom Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis in purus augue. In hac habitasse platea dictumst. Sed fringilla, velit sed commodo feugiat, mi leo porttitor sapien, non rutrum est nisi vitae purus. Donec eget elit non ipsum vulputate imperdiet. Proin scelerisque porta mi, in dignissim sapien interdum vitae. Sed non ante sit amet tellus accumsan placerat non in elit. Nulla ante elit, laoreet sit amet faucibus non, dignissim quis nibh. Integer augue odio, pharetra at eleifend euismod, dapibus ultrices mauris. Ut arcu diam, mattis id auctor at, malesuada ac nibh. Vivamus posuere nisi non quam porta sed cursus eros faucibus. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In a scelerisque lorem. Aliquam ac eros sapien.

Lábléc

# PSZEUDÓ ELEMEK ÉS OSZTÁLYOK

# PSZEUDÓ ELEMEK ÉS PSZEUDÓ OSZTÁLYOK

- Pszeudó elem (látszólagos elem)
  - A dokumentumfában nem jelölt elemek elérését teszi lehetővé. Pl. egy szónak az első betűje, vagy egy bekezdésnek az első sora, egy elem elő- illetve után generált elemek
    - :first-letter, :first-line, :before, :after
- Pszeudó osztály (látszólagos osztály)
  - Az elemek egyes állapotaihoz rendelt osztályok

## PSZEUDÓ ELEMEK ÉS PSZEUDÓ OSZTÁLYOK

- A szintaxisuk két részből áll, egy szelektorból és annak egy látszólagos osztályából, ami egy kettősponttal kezdődik

**szelektor:látszólagos\_osztály {def}**

# LINK ÁLLAPOTOK

```
a:link      {}      /* (nem látogatott) link */
a:visited   {}      /* látogatott link */
a:active    {}      /* aktív link */
a:hover     {}      /* ha fölé visszük az egeret */
```

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="utf-8">
    <style type="text/css">
      body {background-color:black;}
      a:link {color:yellow; }
      a:visited {color:#ff9900; }
      a:hover {color:white; }
    </style>
  </head>
  <body>
    <p><a href="p1.html">Normál link</a></p>
    <p><a href="p2.html">Ezt már meglátogattam</a></p>
    <p><a href="p3.html">Fölé viszem az egeret</a></p>
  </body>
</html>
```

Normál link

Ezt már meglátogattam

Fölé viszem az egeret

## LÁTSZÓLAGOS OSZTÁLYOK

- Kombinálhatók normál osztályokkal és akár más látszólagos osztályokkal

```
a.kulso:visited {color: blue}
```

```
a:link:active {color: red }
```

## LÁTSZÓLAGOS ELEMEK

A HTML **elemek tartalmának kitüntetett részei**, például egy bekezdés első sora, vagy az első betűje. Ilyen elemek a következők:

- :first-letter      **egy szöveg első betűje.**
- :first-line        **egy szöveg első sora.**
- :after             **generált tartalom közvetlenül az elem után.**
- :before            **generált tartalom közvetlenül az elem előtt.**
- ::selection        **a kijelölés stílusát határozhatjuk meg.**

A látszólagos elemeket kezdhetjük **szimpla vagy dupla kettősponttal** is, jelentésük változatlan marad. Kivéve a ::selection-t ami csak dupla kettősponttal működik.

# :FIRST-LETTER

- Az első karakter formázására alkalmas.

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8">
<style type="text/css">
  p {font-size: 12pt; line-height: 12pt}
  p:first-letter {font-size: 200%;}
  p.iniciale:first-letter {font-size: 300%;}
  float:left; padding-top:5px; padding-right:5px;}
</style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet...</p>
  <p>Morbi sit amet felis eros, ut blandit
  ligula. Vestibulum feugiat facilisis
  ullamcorper... </p>
  <p class="iniciale">Vivamus neque orci,
  viverra vitae posuere at, laoreet varius
  metus. Mauris blandit cursus feugiat... </p>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut ipsum sit amet mi tristique semper in sodales mi. Integer dictum dapibus arcu eu commodo. Donec nisl lorem, lobortis ac viverra a, tristique id lacus. Proin pharetra risus felis, at malesuada nisl. Donec congue libero in nibh malesuada eu molestie orci porttitor.

Morbi sit amet felis eros, ut blandit ligula. Vestibulum feugiat facilisis ullamcorper. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nulla faucibus volutpat ipsum et semper. Suspendisse quam massa, luctus semper condimentum nec, luctus nec arcu.

Vivamus neque orci, viverra vitae posuere at, laoreet varius metus. Mauris blandit cursus feugiat. Fusce sed purus eu tellus pharetra euismod. Ut eleifend pretium nisl id placerat. Aenean id condimentum ligula. Maecenas a odio eget neque dictum lacinia. Suspendisse auctor imperdiet bibendum. Fusce a dignissim dolor. Curabitur scelerisque erat accumsan felis placerat varius fermentum ipsum ultrices.

# :FIRST-LINE

- Első sor formázása alkalmas

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<meta charset="utf-8">
<style type="text/css">
  p:first-line {font-size: 130%;}
</style>
</head>
<body>
  <p>Lorem ipsum dolor sit amet,
  consectetur adipiscing elit... </p>
  <p>Morbi sit amet felis eros, ut blandit
  ligula. Vestibulum feugiat facilisis
  ullamcorper ...</p>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ut ipsum sit amet mi tristique semper in sodales mi. Integer dictum dapibus arcu eu commodo. Donec nisl lorem, lobortis ac viverra a, tristique id lacus. Proin pharetra risus felis, at malesuada nisl. Donec congue libero in nibh malesuada eu molestie orci porttitor.

Morbi sit amet felis eros, ut blandit ligula. Vestibulum feugiat facilisis ullamcorper. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Nulla faucibus volutpat ipsum et semper. Suspendisse quam massa, luctus semper condimentum nec, luctus nec arcu.

# :BEFORE, :AFTER, ::SELECTION

```
<style type="text/css">  
p {padding:10px;  
 border:1px solid black;  
/* szegélyt rakunk a bekezdésekre */}
```

```
p.megjegyzes:before  
{content: "Ez a szöveg elő kerül. ";  
font-style:italic; }
```

```
p.pelda:after  
{content: "Ez a szöveg után kerül.";  
font-weight:bold; display:block; }
```

```
::selection {background-color:yellow; }  
  
</style>
```

```
<p class="megjegyzes">Lorem ipsum dolor...</p>  
<p class="pelda">Proin in massa dapibus...</p>
```

Ez a szöveg ki lett jelölve  
a böngésző-programban.



Ez a szöveg elő kerül. Lorem ipsum dolor sit amet, consectetur adipiscing elit

Proin in massa dapibus neque commodo placerat a a ligula.  
Ez a szöveg után kerül.

# DINAMIKUS LÁTSZÓLAGOS OSZTÁLYOK

- :link                      olyan link, ami **még nem volt kiválasztva**.
- :visited                  olyan link, **ami már ki volt kiválasztva**, vagyis rákattintottunk.
- :hover                    olyam elem, **amelyre az egérkurzor mutat**.
- :active                    éppen aktív elem. Pl. egy betöltődő link.
- :focus                    olyan elem, **amelyiken a fókusz van**.
- :target                    az **anchor (könyvjelző) linkek célpontjainak stílusaiért felel**.
- :enabled                 annak az **elemnek** (többnyire űrlapelemnek) a stílusaiért felel, amelyik **engedélyezve van**.
- :disabled                annak az **elemnek** a stílusaiért felel, amelyik **le van tiltva**.
- :checked                 annak az **checkbox**-nak a stílusaiért felel, amely **ki van jelölve**.

Később ezekre is nézünk példákat!

# STRUKTURÁLIS LÁTSZÓLAGOS OSZTÁLYOK

- **:root** elem, amely **nem rendelkezik szülővel**.
- **:empty** az az elem, ami **nem rendelkezik** egyetlen **gyermekkel** sem.
- **:only-child** az az elem, amely **nem rendelkezik testvérrel**.
- **:only-of-type** az az elem, amelyből **nincs hasonló** típusú a **testvérei között**.
- **:first-child** az az elem, amely az **összes testvére közül** az **első**.
- **:first-of-type** az az elem, amely a szelektor típusából a **testvérei között az első**.
- **:nth-child(n)** a **testvérei között az n.** elem.
- **:nth-of-type(n)** a **testvérei között az n.** az **adott szelektorra illeszkedő** elemek közül.
- **:nth-last-of-type(n)** a **testvérei között hátulról az n.** az **adott szelektorra illeszkedő** elemek közül.
- **:last-child** a **testvérei közül az utolsó** elem.
- **:last-of-type** az az elem, amely a **szelektor típusából a testvérei között az utolsó**.

Később ezekre is nézünk példákat!

# EGYÉB LÁTSZÓLAGOS OSZTÁLYOK

## Nyelvfüggőség

- `:lang`
  - segítségével **nyelvfüggővé** tehetjük stílusszabályainkat. A nyelvet a lang HTML attribútum segítségével állíthatjuk be.

## Tagadás

- `:not()`
  - **tagadást fogalmazhatunk meg.** Erre az összes olyan szelektor illeszkedik, amelyekre nem igaz a paraméterként adott tulajdonság.

Később ezekre is nézünk példákat!

# CSS I ÉS CSS 2 SZELEKTOROK

Minta	Jelentés	
*	Megfelel bármelyik elemnek.	CSS 2.
E	Egyezik bármely E elemmel (bármely E típusú elemmel).	
E F	Egyezik bármely F elemmel, ami egy E elem leszármazottja.	
E > F	Egyezik bármely F elemmel, ami egy E elem gyermek-eleme	CSS 2.
E:first-child	Megegyezik az E elemmel, ha az E a szülő elemének első gyermeke.	CSS 2.
E:link E:visited	Megegyezik az E elemmel, ha az E elem egy olyan link, amelynek célját még nem látogatták (:link), vagy már látogatták (:visited).	
E:active E:hover E:focus	Megegyezik az E elemmel, bizonyos felhasználói beavatkozások alatt.	CSS 2. CSS 2. CSS 2.
E:lang(c)	Megegyezik az E típusú elemmel, ha az a (beszélt) c nyelven íródott.	CSS 2.

# CSS I ÉS CSS 2 SZELEKTOROK

Minta	Jelentés	
<b>E + F</b>	Megegyezik bármely F elemmel, amelyet közvetlenül megelőz egy E elem.	CSS 2.
<b>E[valami]</b>	Megegyezik bármely E elemmel, amelynek "valami" attribútuma be van állítva (tekintet nélkül annak értékére).	CSS 2.
<b>E[valami="proba"]</b>	Megegyezik azzal az E elemmel, amelynek "valami" attribútumának értéke pontosan "proba".	CSS 2.
<b>E[valami~="proba"]</b>	Megegyezik azzal az E elemmel, amelynek "valami" attribútumának értékei szóközzel elválasztott értékek, és egyik közülük pontosan "proba".	CSS 2.
<b>E[lang ="en"]</b>	Megegyezik azzal az E elemmel, amelynek "lang" attribútuma pontosan az „en”, vagy „en”-nel kezdődik és kötőjel követi.	CSS 2.
<b>DIV.warning</b>	Csak HTML. Ugyanaz, mint a DIV[class~="warning"].	
<b>E#azonosito</b>	Megegyezik azzal az E elemmel, amelynek ID-je "azonosito".	

# \* SZELEKTOR

A \* szelektort leggyakrabban arra használjuk, hogy minden elem margóját és kitöltését lenullázzuk. Ezen értékeket aztán minden általunk használt elem esetén újra beállítjuk, így egységesek lesznek a margók, nem az adott böngésző alapbeállításán múlnak.

## Kód részlet

```
<body>
  <p>Lorem ipsum dolor ... </p>
  <p>Nullam a posuere turpis...</p>
</body>
```

## CSS

```
* { margin:0; /* margó lenullázása */
  padding:0; /* kitöltés lenullázása */
}
p {margin:10px; padding:10px;}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque turpis lacus, viverra vitae sodales vel, iaculis at nunc. Ut lobortis dapibus leo sed semper. Vestibulum pellentesque eros at risus volutpat at interdum nunc aliquet. Praesent lacus leo, vulputate sed sagittis et, vulputate sit amet nibh. Praesent quis neque id arcu accumsan imperdiet. Duis placerat mauris id quam eleifend elementum. Pellentesque tincidunt auctor mattis. Maecenas sed ornare risus.

Nullam a posuere turpis. Ut placerat viverra eros at condimentum. Sed nisl ipsum, commodo id hendrerit ac, vestibulum sed erat. Aliquam feugiat fermentum justo, dapibus porta odio interdum ac. Nullam facilisis pretium leo eget fringilla. In hac habitasse platea dictumst. Nulla facilisi. In id leo nec diam sodales egestas. Vestibulum facilisis sem at mi consequat gravida. Maecenas accumsan, magna in tincidunt bibendum, nisi odio molestie velit, sit amet tincidunt ligula magna porttitor arcu. Curabitur hendrerit libero id nisi dictum sed pretium elit tempor. Sed ac risus arcu. Integer condimentum erat sit amet est scelerisque cursus.

# LESZÁRMAZÁS (E F)

Itt most azt látjuk, hogy az `em` elem zöld színű lesz, ha a `h1` elem leszármazottjaként használjuk.

## Kódrészlet

```
<h1>CSS, avagy <em>lépcsőzetes  
stíluslapok</em></h1>  
<p>A CSS (angolul Cascading Style  
Sheets) a számítástechnikában egy  
<em>stílusleíró nyelv</em>, mely a  
HTML vagy XHTML típusú strukturált  
dokumentumok megjelenését írja le.  
Ezen kívül használható bármilyen XML  
alapú dokumentum stílusának  
leírására is, mint például az SVG,  
XUL stb.</p>
```

## CSS

```
em {color:blue}  
h1 {color:blue}  
/* ha az em elem a h1  
leszármazottja  
Legyen zöld színnel megjelenítve*/  
h1 em {color:green;}
```

## CSS, avagy *lépcsőzetes stíluslapok*

A CSS (angolul Cascading Style Sheets) a számítástechnikában egy *stílusleíró nyelv*, mely a HTML vagy XHTML típusú strukturált dokumentumok megjelenését írja le. Ezen kívül használható bármilyen XML alapú dokumentum stílusának leírására is, mint például az SVG, XUL stb.

# LESZÁRMAZÁS (E F) PÉLDA

## Kódrészlet

```
<ul id="pelda">
  <li>1. elem
    <ul>
      <li>1. elem 1. alelem</li>
      <li>1. elem 2. alelem
        <ul>
          <li>1. elem 2. alelem 1. al-alelem</li>
          <li>1. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>2. elem
    <ul>
      <li>2. elem 1. alelem</li>
      <li>2. elem 2. alelem
        <ul>
          <li>2. elem 2. alelem 1. al-alelem</li>
          <li>2. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>3. elem</li>
</ul>
```

## CSS

```
ul#pelda li {border:1px solid red;}
```

Minden `<li>` elemben lévő elem szegélyt kap.

- 1. elem
  - 1. elem 1. alelem
  - 1. elem 2. alelem
    - 1. elem 2. alelem 1. al-alelem
    - 1. elem 2. alelem 2. al-alelem
- 2. elem
  - 2. elem 1. alelem
  - 2. elem 2. alelem
    - 2. elem 2. alelem 1. al-alelem
    - 2. elem 2. alelem 2. al-alelem
  - 2. elem 3. alelem
- 3. elem

# GYEREK ELEM (E>F) PÉLDA

## Kódrészlet

```
<ul id="pelda">
  <li>1. elem
    <ul>
      <li>1. elem 1. alelem</li>
      <li>1. elem 2. alelem
        <ul>
          <li>1. elem 2. alelem 1. al-alelem</li>
          <li>1. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>2. elem
    <ul>
      <li>2. elem 1. alelem</li>
      <li>2. elem 2. alelem
        <ul>
          <li>2. elem 2. alelem 1. al-alelem</li>
          <li>2. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>3. elem</li>
</ul>
```

## CSS

```
ul#pelda > li {border:1px solid red;}
```

Csak azon listaelem kap szegélyt, ami gyerekeleme a külső <ul> elemnek.

- 1. elem
  - 1. elem 1. alelem
  - 1. elem 2. alelem
    - 1. elem 2. alelem 1. al-alelem
    - 1. elem 2. alelem 2. al-alelem
- 2. elem
  - 2. elem 1. alelem
  - 2. elem 2. alelem
    - 2. elem 2. alelem 1. al-alelem
    - 2. elem 2. alelem 2. al-alelem
  - 2. elem 3. alelem
- 3. elem

# ELSŐ GYEREK (:FIRST-CHILD)

## Kódrészlet

```
<ul id="pelda">
  <li>1. elem
    <ul>
      <li>1. elem 1. alelem</li>
      <li>1. elem 2. alelem
        <ul>
          <li>1. elem 2. alelem 1. al-alelem</li>
          <li>1. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>2. elem
    <ul>
      <li>2. elem 1. alelem</li>
      <li>2. elem 2. alelem
        <ul>
          <li>2. elem 2. alelem 1. al-alelem</li>
          <li>2. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
      <li>2. elem 3. alelem</li>
    </ul>
  </li>
  <li>3. elem</li>
</ul>
```

## CSS

```
ul#pelda > li:first-child {border:1px solid red;}
```

Csak azon listaelem kap szegélyt, ami gyerekeleme a külső `<ul>` elemnek, és maga első gyermeknek számít.

- 1. elem
  - 1. elem 1. alelem
  - 1. elem 2. alelem
    - 1. elem 2. alelem 1. al-alelem
    - 1. elem 2. alelem 2. al-alelem
- 2. elem
  - 2. elem 1. alelem
  - 2. elem 2. alelem
    - 2. elem 2. alelem 1. al-alelem
    - 2. elem 2. alelem 2. al-alelem
  - 2. elem 3. alelem
- 3. elem

# ELSŐ GYEREK (:FIRST-CHILD)

## Kódrészlet

```
<ul id="pelda">
  <li>1. elem
    <ul>
      <li>1. elem 1. alelem</li>
      <li>1. elem 2. alelem
        <ul>
          <li>1. elem 2. alelem 1. al-alelem</li>
          <li>1. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
    </ul>
  </li>
  <li>2. elem
    <ul>
      <li>2. elem 1. alelem</li>
      <li>2. elem 2. alelem
        <ul>
          <li>2. elem 2. alelem 1. al-alelem</li>
          <li>2. elem 2. alelem 2. al-alelem</li>
        </ul>
      </li>
      <li>2. elem 3. alelem</li>
    </ul>
  </li>
  <li>3. elem</li>
</ul>
```

## CSS

```
ul#pelda li:first-child {border: 1px solid red;}
```

Itt minden olyan listaelem szegélyt kap, ami első gyerek, függetlenül attól, hogy ki a szülője.

- 1. elem
  - 1. elem 1. alelem
  - 1. elem 2. alelem
    - 1. elem 2. alelem 1. al-alelem
    - 1. elem 2. alelem 2. al-alelem
- 2. elem
  - 2. elem 1. alelem
  - 2. elem 2. alelem
    - 2. elem 2. alelem 1. al-alelem
    - 2. elem 2. alelem 2. al-alelem
  - 2. elem 3. alelem
- 3. elem

# NYELVFÜGGŐ STÍLUS (:LANG)

## Kódrészlet

```
<p lang="en">Friend: a person who can turn a horrible day into an absolutely perfect one.</p>
<p lang="hu">Barát az, aki egy rettenetes napból is tökéleteset tud varázsolni.</p>
<p lang="en">If you can't laugh at yourself, friend, i'd be glad to do it for you.</p>
<p lang="hu">Ha nem tudsz magadon nevetni, barátom, én örömmel megteszem helyetted. </p>
```

## CSS

```
p:lang(en) {color:blue}
p:lang(hu) {color:red; margin-left:20px;}
```

A cél, hogy az angol szövegek kék, a magyarok piros színnel, beljebb jelenjenek meg.

Friend: a person who can turn a horrible day into an absolutely perfect one.

Barát az, aki egy rettenetes napból is tökéleteset tud varázsolni.

If you can't laugh at yourself, friend, i'd be glad to do it for you.

Ha nem tudsz magadon nevetni, barátom, én örömmel megteszem helyetted.

# KÖZVETLEN MEGELŐZÉS (TESTVÉR ELEMÉKNÉL) (E+F)

## Kódrészlet

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<style type="text/css">
p {text-indent:20px;}
h1+p {text-indent:0;}
</style>
</head>
<body>
<h1>Címsor</h1>
<p>Lorem ipsum dolor sit amet...</p>
</body>
</html>
```

## Címsor

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris viverra sem ut purus vehicula eget lobortis mi luctus. Aliquam at luctus lectus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris viverra sem ut purus vehicula eget lobortis mi luctus. Aliquam at luctus lectus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris viverra sem ut purus vehicula eget lobortis mi luctus. Aliquam at luctus lectus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris viverra sem ut purus vehicula eget lobortis mi luctus. Aliquam at luctus lectus.

Azon bekezdések első sora nincs behúzva,  
amelyek közvetlenül I-es címsor után következnek.

# ATTRIBÚTUM SZELEKTOR (PARAMÉTER)

## Kódrészlet

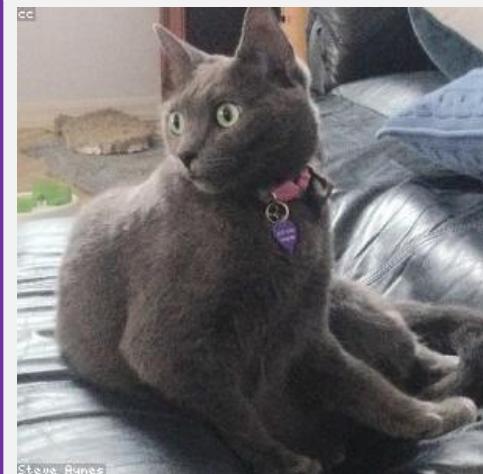
```
  
  

```

## CSS

```
img[title] {border:2px solid purple;}
```

Ebben a példában csak az a kép kap lila szegélyt, amelynek van title paramétere.



# ATTRIBÚTUM SZELEKTOR (PARAMÉTER=„ÉRTÉK”)

## Kódrészlet

```
<form action="#">
  <label>Név</label>
  <input type="text" >
  <label>Jelszó</label>
  <input type="password" >
</form>
```

## CSS

```
input[type="password"]
{
  background-color: yellow;
}
```

Név

Jelszó

A jelszó (password) típusú input mező háttere sárgára lett állítva.  
A szöveg (text) típusú mező háttere nem változott.

# ATTRIBÚTUM SZELEKTOR (PARAMÉTER~="ÉRTÉK")

## Kódrészlet

```
<ul>
<li><a href="http://www.inf.elte.hu" title="ELTE Informatikai
Kar">ELTE IK</a></li>
<li><a href="http://www.w3c.hu" title="W3C konzorcium">W3C</a></li>
<li><a href="http://www.elte.hu" title="Az ELTE
weblapja">ELTE</a></li>
</ul>
```

## CSS

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

- [ELTE IK](http://www.inf.elte.hu "ELTE Informatikai Kar")
- [W3C](http://www.w3c.hu "W3C konzorcium")
- [ELTE](http://www.elte.hu "Az ELTE weblapja")

Csak azon linkek háttere lesz sárga, amelyek title attribútumában szerepel az ELTE szó.

# E[lang="en"] TÍPUSÚ SZELEKTOR

## Kódrészlet

```
<p lang="hu">A Clash egy közismert együttes.</p>
<p lang="en-gb">The Clash are a well-known band.</p>
<p lang="en-us">The Clash is a well-known band.</p>
```

## CSS

```
p[lang="en"] {color:blue;}
```

A Clash egy közismert együttes.

The Clash are a well-known band.

The Clash is a well-known band.

Mindegy, hogy amerikai, vagy brit angol nyelvjárás szerint írtunk, kék lesz a szöveg színe.

# CSS3-AS SZELEKTOROK

Minta	Jelentés
E ~ F	Olyan F elem, amelyet korábban megelőzött egy E elem. (hasonló mint E + F, de nem kell, hogy közvetlenül egymás után következzenek az elemek)
E[attr]	<p>p[testAttr^=foo] bármely elem, amely „testAttr” paraméterének van értéke, és az a „foo” karakterláncjal kezdődik.</p> <p>p[testAttr\$=foo] bármely elem, amely „testAttr” paraméterének van értéke, és az a „foo” karakterláncjal végződik.</p> <p>p[testAttr*=foo] bármely elem, amely „testAttr” paraméterének van értéke, és abban szerepel „foo” karakterlánc.</p>
:empty	Tartalom nélküli elemre illeszkedik
:enabled :disabled :checked	Enabled, disabled, checked állapotú (űrlap)elemekre illeszkedik

Minta	Jelentés
:first-of-type :last-of-type :only-of-type	X:first-of-type olyan X elem, amely a szülőjének első gyereke X:last-of-type olyan X elem, amely a szülőjének utolsó gyereke X:only-of-type olyan X elem, amely a szülőjének egyetlen gyereke
:last-child	Utolsó gyermekelem
:not	Tagadás p#test *:not(em) {text-decoration: underline;} A test azonosítójú bekezdés leszármazottai, kivéve az em elemeket.

# PROBLÉMA FELVETÉS

## Kódrészlet

```
<div id="pelda">
  <p>Lorem ipsum dolor sit amet, consectetur
  adipiscing elit.</p>
  <div>
    <p>Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. </p>
  </div>
<div>
  <p>Lorem ipsum dolor sit amet, consectetur
  adipiscing elit. </p>
  <div>
    <p>Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. </p>
  </div>
</div>
</div>
```

Hogyan tudnánk csak ezen blokkot pirosra állítani,  
anélkül, hogy megváltoztatnánk a HTML kódot?

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

# MEGOLDÁS

## Kódrészlet

```
<div id="pelda">
  <p>Lorem ipsum dolor sit amet, consectetur
  adipiscing elit.</p>
  <div>
    <p>Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. </p>
  </div>
<div>
  <p>Lorem ipsum dolor sit amet, consectetur
  adipiscing elit. </p>
  <div>
    <p>Lorem ipsum dolor sit amet,
    consectetur adipiscing elit. </p>
  </div>
</div>
</div>
```

Hogyan tudnánk csak ezen blokkot pirosra állítani, anélkül, hogy megváltoztatnánk a HTML kódot? Ez a div adott típusból az első, vagyis first-of-type a jó látszólagos osztály.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

# :ONLY-OF-TYPE HATÁSA

## Kódrészlet

```
<div id="content">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing
elit.</p>
<div>
    <p>Lorem ipsum dolor sit amet, consectetur
        adipiscing elit.</p>
</div>
<div>
    <p>Lorem ipsum dolor sit amet, consectetur
        adipiscing elit.</p>
    <div>
        <p>Lorem ipsum dolor sit amet,
            consectetur adipiscing elit.</p>
    </div>
</div>
</div>
```

## CSS

```
div#content >:only-of-type {color:red;}
```

Csak a bekezdés tartalma lesz piros, mert bekezdésből csak egy van, divből viszont van kettő is.

**Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.**

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

Lorem ipsum dolor sit amet,  
consectetur adipiscing elit.

# CSS3 SZELEKTOROK

Minta	Jelentés
<b>:nth-child()</b>	Egy formulával meghatározható sorszámú gyerek <code>div#test &gt; *:nth-child(3n+1) { color: red; }</code>
<b>:nth-of-type()</b>	Egy formulával meghatározható sorszámú, adott típusú gyerek <code>div#test div:nth-of-type(5n+2) { background-color: #9999ff; }</code>
<b>:nth-last-child()</b> <b>:nth-last-of-type()</b>	Hasonló a fentiekhez, de az utolsó elemre vonatkozik. (A sorszám az utolsótól számítódik)
<b>:only-child</b>	Olyan elem, amely a szülő elem egyetlen gyermeke
<b>:root</b>	Minden blokk gyökéreleme (HTML-ben ez a <code>&lt;html&gt;</code> elem lesz.)
<b>::selection</b>	A felhasználó által kijelölt szöveg
<b>:target</b>	Azon elem, ami link célként van beállítva (egyedi neve van és mutat rá link a <code>#egyedenév</code> formátumban)

# :NTH-CHILD() – VALAHÁNYADIK GYEREK

```
<style type="text/css">
td {
    padding:10px;
    border:1px solid blue;
}
table {
    border:1px solid green;
    width:250px;
    border-collapse:collapse;
}
tr:nth-child(3n+1) {
    background-color:lightgreen;
}
tr:nth-child(3n+2) {
    background-color:lightblue;
}
</style>
...
<table>
<tr><td>1</td><td>2</td></tr>
<tr><td>3</td><td>4</td></tr>
<tr><td>5</td><td>6</td></tr>
<tr><td>7</td><td>8</td></tr>
<tr><td>9</td><td>10</td></tr>
</table>
```

1	2
3	4
5	6
7	8
9	10

A táblázatsorok közül a 3-mal oszthatók az alap (fehér) háttérét kapják, a következő világoszöld, a rákövetkező világoskék lesz.

# RANGSOROLÁS

Teljes dokumentáció

<https://www.w3.org/TR/css-cascade-3/>

# PROBLÉMA FELVETÉS

Az alábbi kód milyen színű 1-es címsort eredményez?

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <meta charset="utf-8">
    <style type="text/css">
      div.szegely h1 {color:green}
      h1 {color:blue;}
    </style>
  </head>
  <body>
    <div class="szegely">
      <h1 style="color:red">Címsor</h1>
    </div>          Emiatt piros lenne a színe
  </body>
</html>
```

Emiatt zöld lenne a színe  
Emiatt kék lenne a színe

# Címsor

Piros lesz a rangsorolási  
szabályok miatt

# RANGSOR

- Ellentmondó definíciók esetén melyik fog érvényesülni? => **súly** hozzárendelése
- **Rangsor felállítás szabályai**
  - Kérdéses elem/tulajdonság párosra alkalmazott összes deklaráció előkeresése
    - A kérdéses elem szelektorként szerepel?
      - Igen: OK
      - Nem: Van örökölt érték?
        - Igen: OK
        - Nem: Alaptulajdonságok jut(hat)nak érvényre

# RANGSOR

- Meghatározás sorrendje szerint
  - Későbbi deklaráció > korábbi deklaráció

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
    <style type="text/css">
      h1 {color: blue}
      h1 {color: green}
    </style>
  </head>
```

Ez egy 1-es címsor

```
<body>
  <h1>Ez egy 1-es címsor</h1>
</body>
</html>
```

# RANGSOR

- explicit súly szerinti rendezés
  - ! important > normál

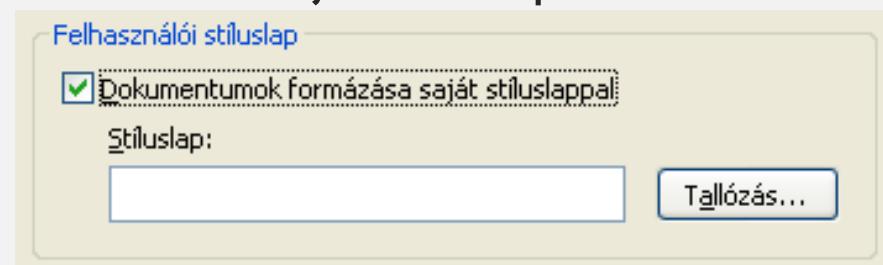
```
<!DOCTYPE html>
<html>
<head>
<title></title>
<style type="text/css">
    body {background-color: #c0c0c0}
    h1 {color: black ! important; background: white ! important }
    h1 {color: blue}
</style>
</head>
<body>
<h1>Ez egy 1-es címsor</h1>
</body>
</html>
```

Az !important attribútummal ellátott deklarációk nagyobb súlyt kapnak, mint a normál deklarációk.

Ez egy 1-es címsor

# RANGSOR

- **Eredet szerinti rendezés**
  - Szerző definíciói > Olvasó definíciói (felhasználói stíluslap) > Kliens program beállításai
  - Mi van, ha olvasóként felül akarom bírálni a szerző definícióit => **! Important** attribútum használata a saját stíluslapban
- Példa
  - `a:link {color: green ! important}`
  - Mi van ha a szerző is használta?
    - Szerző (important) > olvasó (important) [CSS1]
    - Szerző (important) < olvasó (important) [CSS2+]



# RANGSOR

- Szelektor egyedisége szerint. Képzeljünk el egy táblázatot a következő oszlopokkal (a,b,c,d)

a	b	c	d
style	id	attrib	(pszeudó) elemek

- a: Ha a style attribútum lett használva, akkor 1-es az értéke.
- b: ID attribútumok száma
- c: Egyéb attribútumok száma (osztály, pszeudó osztály)
- d: Elemek és pszeudó elemek száma
  - A abcd számokat össze kell fűzni, így megkapjuk az egyediséget

**a: Ha a style attribútum lett használva, akkor 1-es az értéke.**

**b: id attribútumok száma**

**c: egyéb attribútumok száma (osztály, pszeudó osztály)**

**d: elemek és pszeudó elemek száma**

A abcd számokat össze kell fűzni, így megkapjuk az egyediséget

szelektor	a	b	c	d	egyediség
*	0	0	0	0	0
li {...}	0	0	0	1	1
li:first-letter {...}	0	0	0	2	2
ul li{...}	0	0	0	2	2
ol ul li {...}	0	0	0	3	3
ol ul+li {...}	0	0	0	3	3
li.red {...}	0	0	1	1	11
li.red.level {...}	0	0	2	1	21
ol ul li.red {...}	0	0	1	3	13
#x34y {...}	0	1	0	0	100
style=""	1	0	0	0	1000

# RANGSOR ÉRTÉKE

`li:first-child h2 .title`

0

Inline styles

0

IDs

2

Classes, attributes  
and pseudo-classes

2

Elements and  
pseudo-elements

A <https://specificity.keegan.st/> weboldalon ki lehet próbálni, hogy melyik szelektorláncnak, mekkora a rangsora

# KIÉRTÉKELÉSI SORREND (CSÖKKENŐ PRIORITÁSI SORRENDBEN)

## I. Erebet és fontosság

1. Átmenetek (Transitions)
2. Fontos (Important) böngésző deklarációk
3. Fontos (Important) felhasználói deklarációk
4. Fontos (Important) szerzői deklarációk
5. Animációs deklarációk
6. Normál szerzői deklarációk
7. Normál felhasználói deklarációk
8. Normál böngésző deklarációk

## 2. Egyediség

- Az ismertetett számítás szerint

## 3. Meghatározás sorrendje

- Az utolsó meghatározás győz
- Importált stíluslapok esetén úgy kell tekinteni, mintha az importált stíluslap tartalma az adott helyre lenne beágyazva
- A linkelt stíluslapokat úgy kell tekinteni, mintha egymás után lennének fűzve

# CSS EGYEDISÉG EGY ÁBRÁZOLÁSA

<http://www.standardista.com/css3/css-specificity/>

# CSS SPECIFICITY

WITH PLANKTON, FISH AND SHARKS

*		div		li > ul		body div ... ul li p a	
universal selector	0 - 0 - 0	1 element	0 - 0 - 1	2 elements	0 - 0 - 2	12 elements	0 - 0 - 12
.myClass		* .myClass		[type=checkbox]		:only-of-type	
1 class	0 - 1 - 0	1 universal selector 1 class	0 - 1 - 0	1 attribute selector	0 - 1 - 0	1 pseudo-class	0 - 1 - 0
li.myClass		li [attr]		li:nth-of-type(3n) ~ li		form input[type=email]	
1 element 1 class	0 - 1 - 1	1 element 1 attribute	0 - 1 - 1	2 elements 1 pseudo-class	0 - 1 - 2	2 elements 1 attribute	0 - 1 - 2
li.class:nth-of-type(3n)		input[type]:not(.class)		clnth-chj[~^~] chk[type] ...		#myDiv	
1 element 1 class 1 pseudo-class	0 - 2 - 1	1 element 1 class 1 attribute	0 - 2 - 1	10 class/attribute/pseudo-classes	0 - 10 - 0	ID Selector	1 - 0 - 0
#myDiv li.class a[href]		#divitis #myDiv a		style=""		!important	
2 types 2 class/attribute 1 ID Selector	1 - 2 - 2	2 ID Selectors 1 type selector	2 - 0 - 1	inline style	1 - 0 - 0 - 0	!important	1 - 0 - 0 - 0 - 0

X-0-0: The number of ID selectors

0-Y-0: The number of class selectors, attributes selectors, and pseudo-classes

0-O-Z: The number of type selectors and pseudo-elements

\*, +, >, ~ : Universal selector and combinators do not increase specificity

:not(x): Negation selector has no value. Argument increases specificity

ESTELLE WEYL \* @ESTELLEVW \* WWW.STANDARDISTA.COM \* 2104



# CSS EGYEDISÉG EGY ÁBRÁZOLÁSA

[https://stuffandnonsense.co.uk/archives/css\\_specificity\\_wars.html](https://stuffandnonsense.co.uk/archives/css_specificity_wars.html)



**a**  
1 x element selector



**p a**  
2 x element selectors



**.foo**  
1 x class selector \*



**a.foo**  
1 x element selector  
1 x class selector

Sith power: 0,0,1

Sith power: 0,0,2

Sith power: 0,1,0

Sith power: 0,1,1



**p a.foo**  
2 x element selectors  
1 x class selector



**.foo .bar**  
2 x class selectors



**p.foo a.bar**  
2 x element selectors  
2 x class selectors



**#foo**  
1 x id selector

Sith power: 0,1,2

Sith power: 0,2,0

Sith power: 0,2,2

Sith power: 1,0,0



**a#foo**  
1 x element selector  
1 x id selector



**.foo a#bar**  
1 x element selector  
1 x class selector  
1 x id selector



**.foo .foo #foo**  
2 x class selectors  
1 x id selector



**style**  
1 x style attribute

Sith power: 1,0,1

Sith power: 1,1,1

Sith power: 1,2,0

Sith power: 1,0,0



\* Same specificity  
class selector =  
attribute attribute =  
pseudo-classes

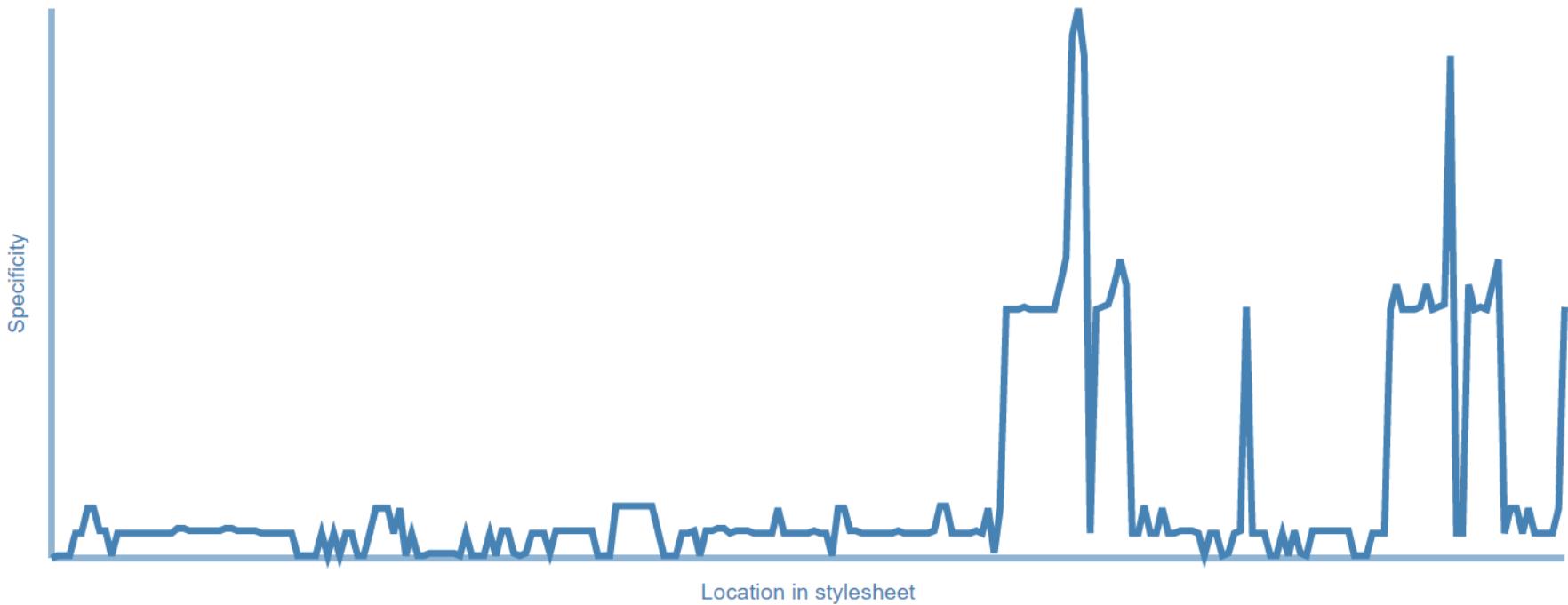


!important

# CSS EGYEDISÉG GRAFIKON

## CSS Specificity Graph Generator

Spikes are bad, and the general trend should be towards higher specificity later in the stylesheet. More info



# MÉDIATÍPUSOK

<https://www.w3.org/TR/mediaqueries-4/>

## HASZNÁLHATÓ MÉDIATÍPUSOK (MEDIA QUERIES LEVEL 4)

- all                    összes eszköz
- print                lapokra tördelt megjelenés (pl. nyomtató)
- screen              képernyők
- speech             beszédszintetizátorok

# (NÉHÁNY) MÉDIA JELLEMZŐ

- width Viewport szélessége
- height Viewport magassága
- aspect-ratio Szélesség-magasság oldalarány
- orientation Viewport tájolása  
(portrait, landscape)
- resolution Felbontás (dpi)
- scan Pásztázás módja  
(interlace, progressive))
- grid Rács rendszerű megjelenítő (pl. tty)
- update Milyen gyakran frissíti a képernyőt  
(none, slow, fast)
- color Színmélység

# MÉDIATÍPUSOK HASZNÁLATA HTML KÓDBAN

- Médiatípus hozzárendelés a *media* paraméterrel

- A külső stíluslap (<link>) belinkelése esetén

```
<link rel="stylesheet" type="text/css" media="print, handheld" href="nyomt.css">
```

- A <source> tag esetén

```
<picture>
  <source media="(min-width: 650px)" srcset="img_pink_flowers.jpg">
  <source media="(min-width: 465px)" srcset="img_white_flower.jpg">
  
</picture>
```

- A <style> paraméter esetén

```
<style type="text/css" media="screen">
```

# MÉDIATÍPUSOK HASZNÁLATA CSS-BEN

- Cél média megadása @media vagy @import szabályokkal.
  - `@import url("fancyfonts.css") screen;`
  - `@media print { /* nyomtatásra vonatkozó def. */ }`

# A @MEDIA SZABÁLY - PÉLDA

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title></title>
<style type="text/css">
@media screen
{
    body {
        background-color:darkblue;
        color:white;
        font-family:sans-serif;
    }
}
@media print
{
    body {
        background-color:white;
        color:black;
        font-family:serif;
    }
    div#menu {display:none} /* ne látszódjon */
}
</style>
</head>
<body>
<div id="menu">Menüpont1 Menüpont2 </div>
<h1>Címsor 1</h1>
<p>Szöveg</p>
</body>
</html>
```

Menüpont1 Menüpont2

**Címsor 1**

Szöveg

Megjelenés a képernyőn

**Címsor 1**

Szöveg

Megjelenés nyomtatásban  
(nem látszik a menü,  
talpas betűtípus, más színek)

# TOVÁBBI PÉLDÁK

```
/* ----- Kindle Fire HD 8.9" ----- */  
  
/* Portrait and Landscape */  
@media only screen  
and (min-device-width: 1200px)  
and (max-device-width: 1600px)  
and (-webkit-min-device-pixel-ratio: 1.5) {  
  
}
```



```
/* ----- Moto 360 Watch ----- */  
@media  
(max-device-width: 218px)  
and (max-device-height: 281px) {  
  
}
```



<https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

# ALTERNATÍV STÍLUSLAPOK MEGADÁSA

- <link rel="stylesheet" type="text/css" href= "[uri](#)" title="Normál" media="screen" >
- <link rel="alternate stylesheet" type="text/css" href="[uri](#)" title="Fekete/fehér" media="screen" >

The screenshot illustrates how different stylesheets are applied to a website using the Firefox 'Nézet' (View) menu.

**Left Side (Original View):** Shows the standard view of the website with a colorful landscape painting as the header image. The navigation bar includes tabs for 'Fekete/Fehér' (Black/White), 'Fehér/Fekete' (White/Black), and 'Normál' (Normal).

**Middle Column:** A vertical stack of three screenshots showing the effect of different stylesheets:

- Top:** 'Fekete/Fehér' (Black/White) style applied. The background is black, and the text and images are white or light-colored.
- Middle:** 'Fehér/Fekete' (White/Black) style applied. The background is white, and the text and images are dark.
- Bottom:** 'Normál' (Normal) style applied. This is identical to the original view.

**Bottom Left:** A large callout box labeled 'Firefox Nézet menü' (Firefox View menu) points to the open menu on the left side of the screen. The menu is expanded to show the 'Eszközök' (Tools) section, which contains options like 'Eszköztárak' (Toolbars), 'Oldalsáv' (Tab), 'Leállítás' (Stop), 'Frissítés' (Refresh), 'Nagyítás' (Zoom), 'Oldalstílus' (Page Style), 'Karakterkódolás' (Character Encoding), and 'Teljes képernyős üzemmód' (Full Screen Mode). The 'Oldalstílus' option is currently selected.

**Bottom Right:** A circular page number '130' is located in the bottom right corner.

# VÁLTÁS A STÍLUSOK KÖZÖTT

- Javascript segítségével is válthatunk stílust
  - Az alábbi megoldásokban cookie tárolja el a választást.
- <http://alistapart.com/article/alternate>
- <https://www.thesitewizard.com/javascripts/change-style-sheets.shtml>
- <https://www.sitepoint.com/creating-simple-style-switcher/>

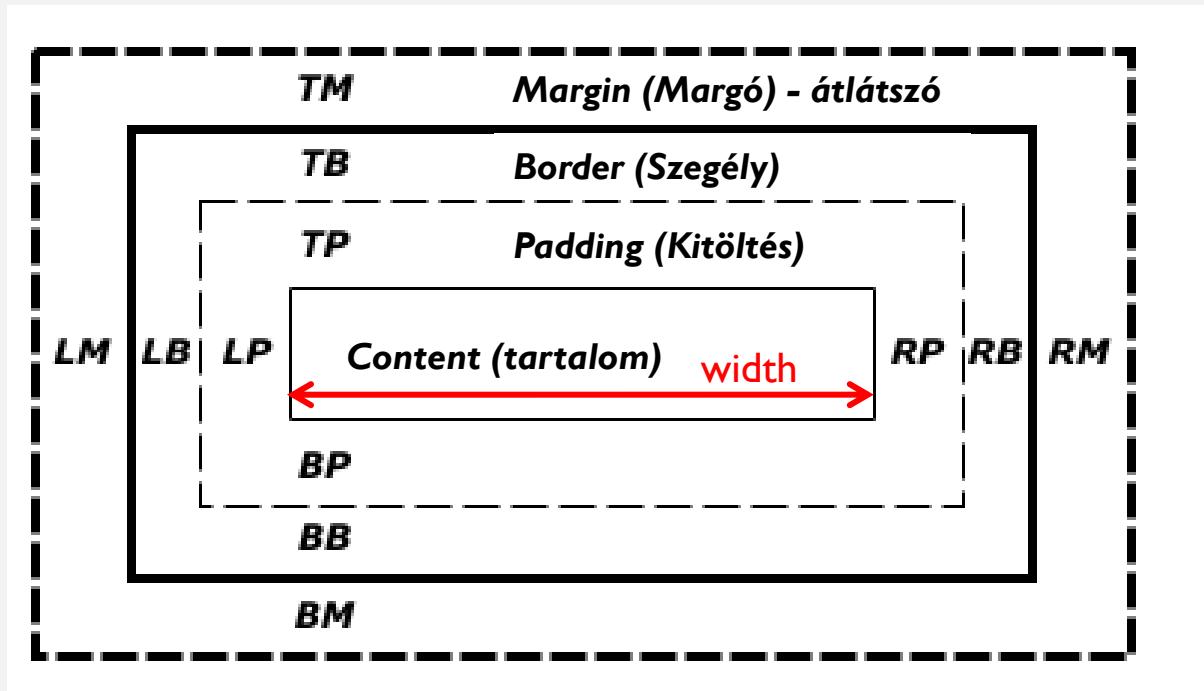
# FORMÁZÁSMODELL (DOBOZMODELL)

# FORMÁZÁSMODELL

**TOP (felső)**

**LEFT (bal)**

**RIGHT (jobb)**



**BOTTOM (alsó)**

A tartalom (content) mérete a megjelenített tartalomtól függ, de beállítható a width és height (szélesség és magasság) paraméterekkel.

# SZEGÉLYEK

- **Mind a négy oldali szegélyre vonatkozik az alábbi:**
  - border-style: stílus [none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset]
  - border-width: vastagság [thin, medium, thick, méret]
  - border-color: szín vagy transparent (átlátszó)
- **Kompakt (shorthand) megadás mind a négy oldalra szimmetrikusan:**
  - border: 1px solid green;

# SZEGÉLYEK

- A tulajdonságok külön is megadhatóak a négy oldalnak

**Szín**

border-top-color;  
border-right-color;  
border-bottom-color;  
border-left-color;

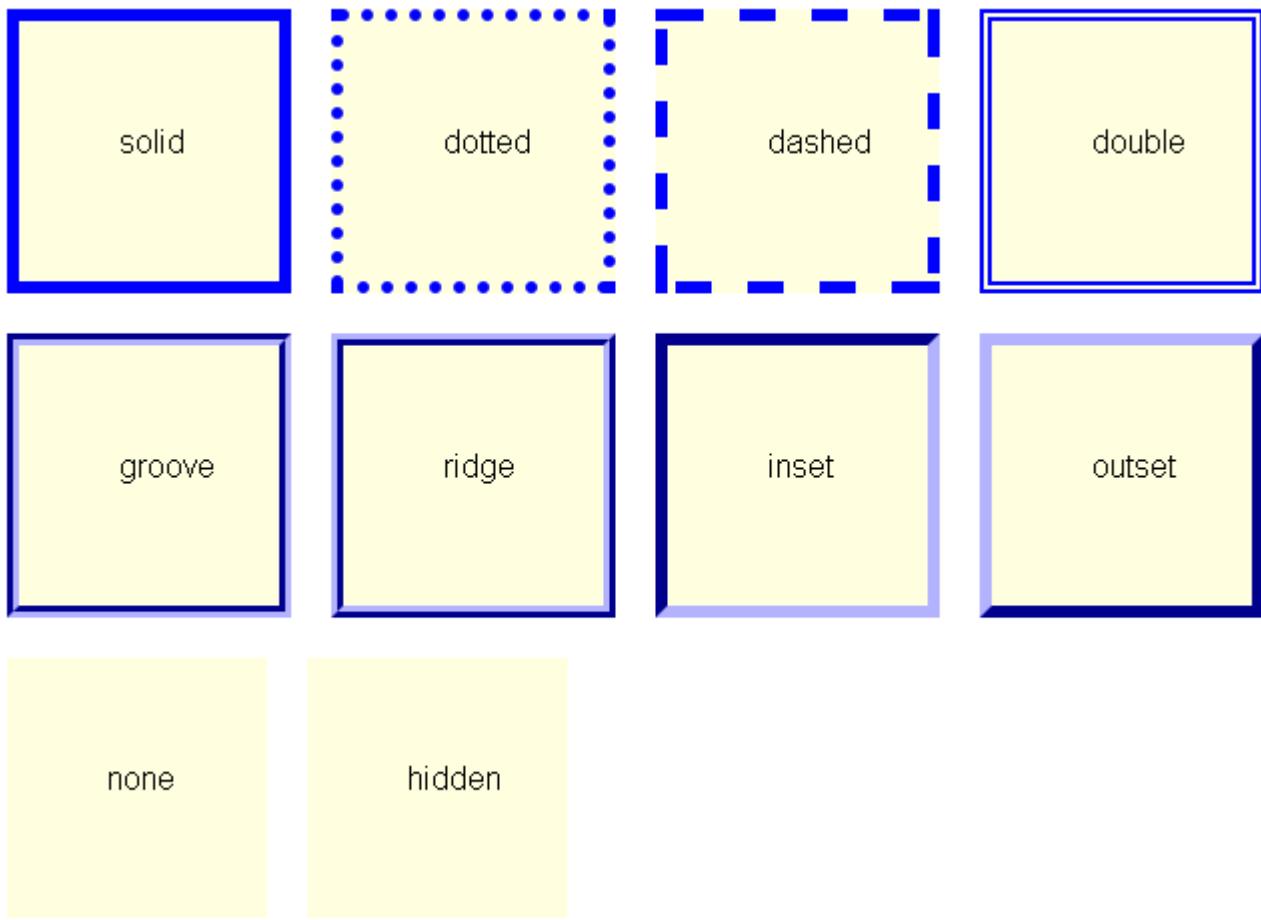
**Stílus**

border-top-style;  
border-right-style;  
border-bottom-style;  
border-left-style;

**Vastagság**

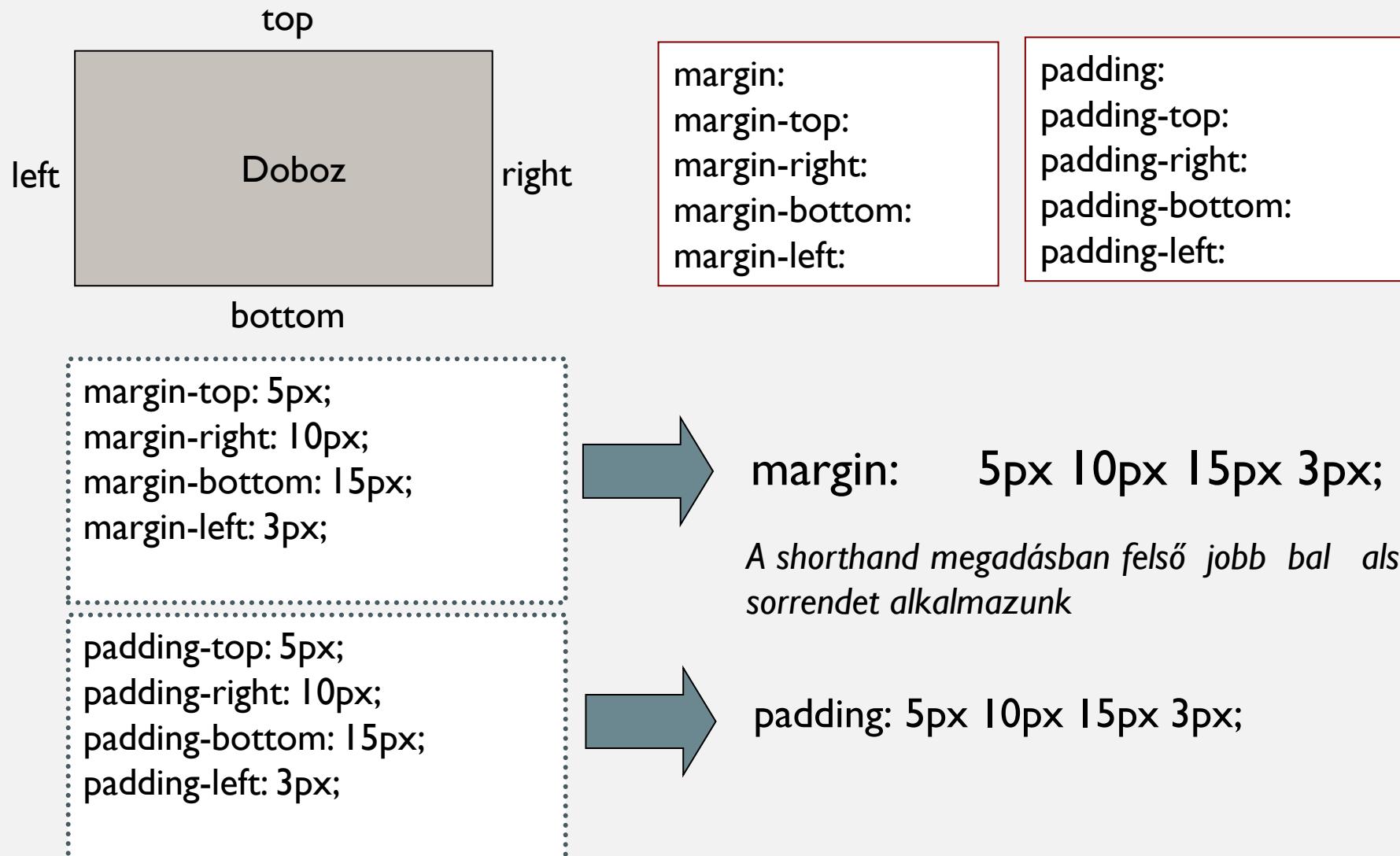
border-top-width;  
border-right-width;  
border-bottom-width;  
border-left-width;

# SZEGÉLY STÍLUSOK

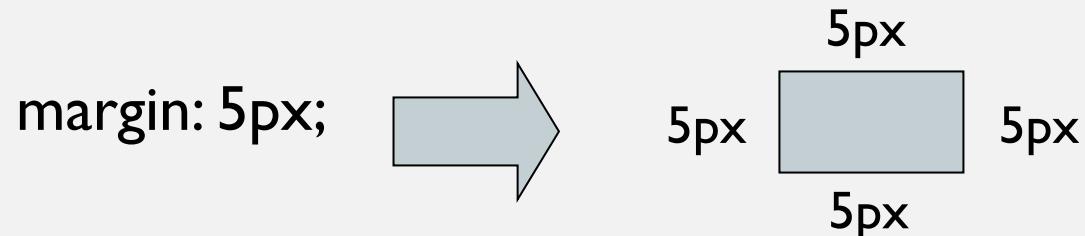
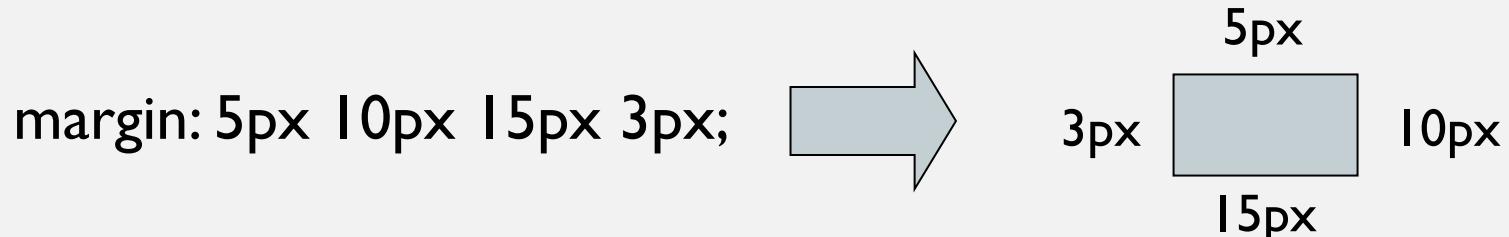


Mi a különbség a hidden és none között? Speciális esetekben (pl. táblázatok formázásánál) szükséges.

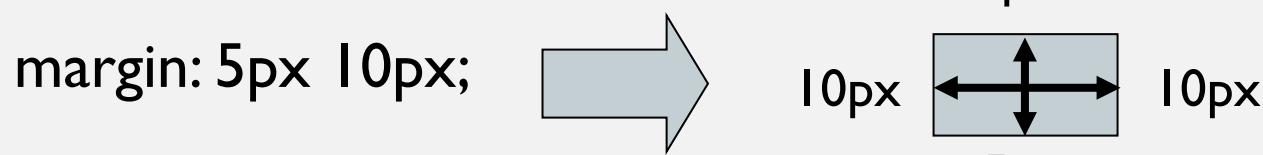
# MARGÓ, KITÖLTÉS MEGADÁSA



# MARGÓ SHORTHAND MEGADÁSA



A következő példákban az érték öröklődik a szemben lévő oldalról



# KÖZÉPRE IGAZÍTÁS MARGÓ BEÁLLÍTÁSSAL

- Ha egy blokkot középre akarunk igazítani, akkor a bal-, és jobboldali margóját is auto értékre kell állítani.

```
.kozepre {  
    margin-left:auto;  
    margin-right:auto;  
    margin-top:20px;  
    margin-bottom:20px;  
    width: 50%;  
}
```

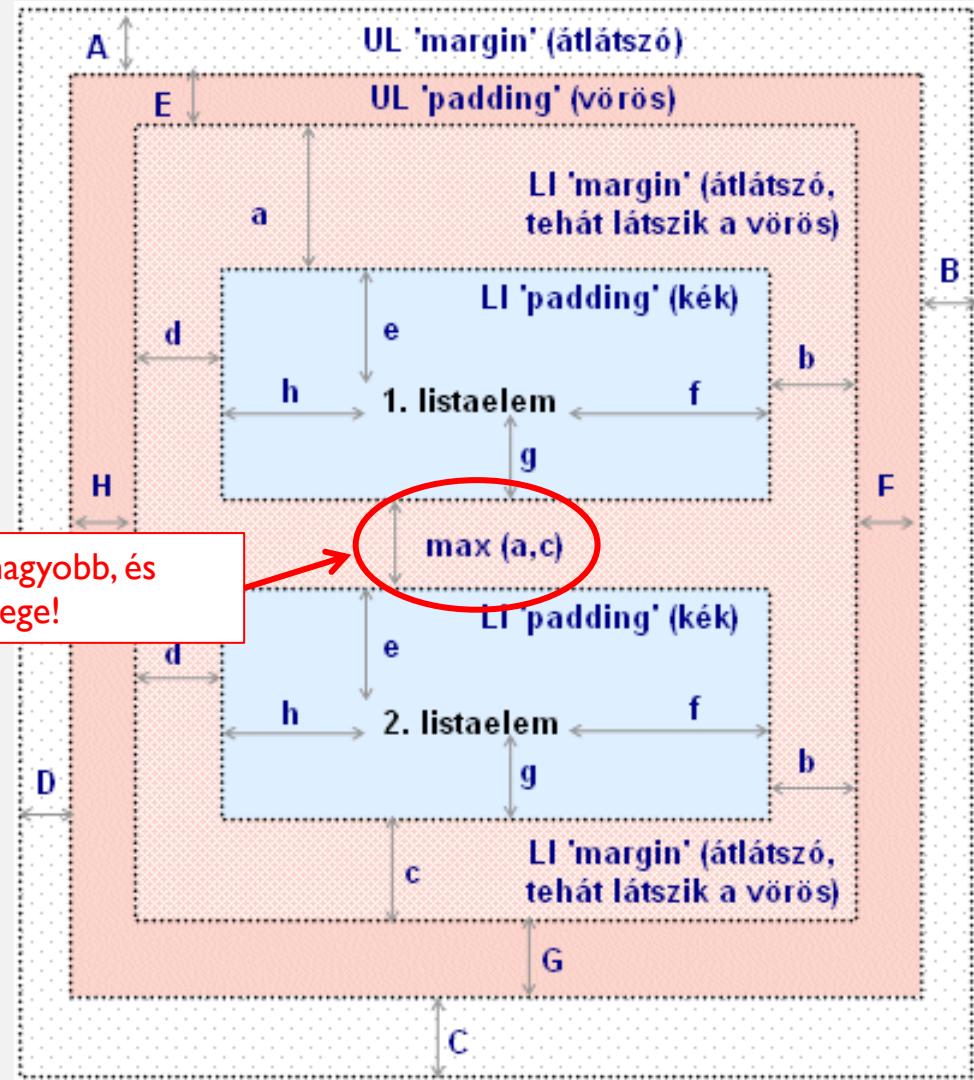
Így is írhattuk volna:  
margin:20px auto;

# MARGÓ ÖSSZEVONÁS

- A szomszédos függőleges margók összevonásra kerülnek bizonyos feltételek fennállása esetén.
- Bővebb info:
  - <http://www.w3.org/TR/CSS21/box.html>

# LISTAELEMEK MARGÓ ÖSSZEVONÁSA

```
<style type="text/css">
ul {
    background: red;
    margin: A B C D;
    padding: E F G H;
}
li {
    color: white;
    background: blue;
    margin: a b c d;
    padding: e f g h;
}
</style>
...
<ul>
    <li>Első elem</li>
    <li>Második elem</li>
</ul>
```



# PÉLDA MARGIN, PADDING ÉRTÉKRE

p {border:1px solid blue;}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aenean in dolor a tellus feugiat molestie sed quis nisi.  
Vivamus augue nisi, imperdiet id consequat id, luctus ac  
velit.

↑ Alapértelmezett margó

Nam id nisl vitae neque posuere interdum. Quisque  
pulvinar fermentum metus, at imperdiet purus accumsan  
blandit. Nunc bibendum, nibh vitae lobortis varius, erat  
eros elementum felis, a tempor nisi lectus at turpis.  
Praesent sodales mauris ac massa dignissim fermentum.

p {border:1px solid blue;  
margin:0;padding:10px;}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aenean in dolor a tellus feugiat molestie sed quis nisi.  
Vivamus augue nisi, imperdiet id consequat id, luctus ac  
velit.

Nam id nisl vitae neque posuere interdum. Quisque  
pulvinar fermentum metus, at imperdiet purus accumsan  
blandit. Nunc bibendum, nibh vitae lobortis varius, erat  
eros elementum felis, a tempor nisi lectus at turpis.  
Praesent sodales mauris ac massa dignissim fermentum.

p {border:1px solid blue; margin:0;}

Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Aenean in dolor a tellus feugiat molestie sed quis nisi.  
Vivamus augue nisi, imperdiet id consequat id, luctus ac  
velit.

Nam id nisl vitae neque posuere interdum. Quisque  
pulvinar fermentum metus, at imperdiet purus accumsan  
blandit. Nunc bibendum, nibh vitae lobortis varius, erat  
eros elementum felis, a tempor nisi lectus at turpis.  
Praesent sodales mauris ac massa dignissim fermentum.

p {border:1px solid blue;  
padding:10px;margin:5px;}

Lorem ipsum dolor sit amet, consectetur adipiscing  
elit. Aenean in dolor a tellus feugiat molestie sed quis  
nisi. Vivamus augue nisi, imperdiet id consequat id,  
luctus ac velit.

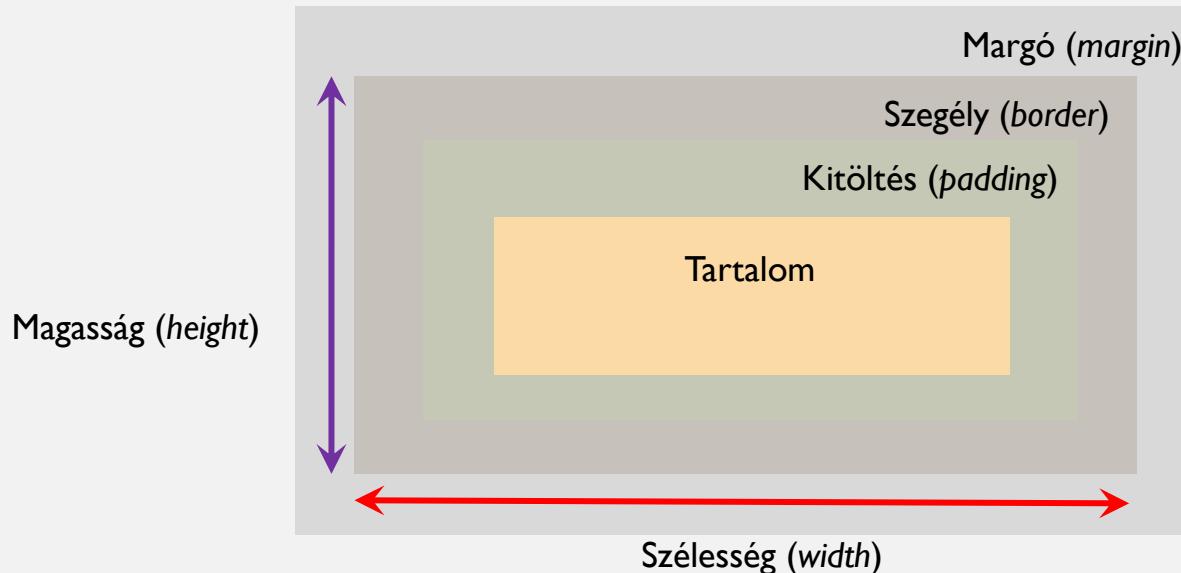
Nam id nisl vitae neque posuere interdum. Quisque  
pulvinar fermentum metus, at imperdiet purus  
accumsan blandit. Nunc bibendum, nibh vitae lobortis  
varius, erat eros elementum felis, a tempor nisi lectus  
at turpis. Praesent sodales mauris ac massa dignissim  
fermentum.

# DOBOZOK MÉRETE

- Méret megadása
  - width: szélesség
  - height: magasság
- Minimális és maximális méretek megadása
  - min-width: minimális szélesség
  - max-width: maximális szélesség
  - min-height: minimális magasság
  - max-height: maximális magasság

# DOBOZMÉRETEZÉS MÓDOSÍTÁSA

Egyes esetekben kényelmesebb lenne a méretmegadás, ha a szélesség (width) és magasság (height) értékekbe a kitöltés és a szegély mérete is beleszámítana. Ehhez a módosításhoz a **box-sizing**: **border-box**; paramétert kell használni. (Az alapértelmezett érték a **content-box**)



# DOBOZMÉRETEZÉSEK ÖSSZEHASONLÍTÁSA

## Példa

### CSS kód

```
<style>
div {
    width:200px;
    border:10px solid blue;
    padding:10px;
    margin:10px;
}
```

```
div#blokk1 {box-sizing: content-box;}
```

```
div#blokk2 {box-sizing: border-box;}
</style>
```

### HTML kód

```
<body>
    <div id="blokk1">Ez az 1.blokk</div>
    <div id="blokk2">Ez a 2. blokk</div>
</body>
```

Ez az 1. blokk

Ez a 2. blokk

# HÁTTÉRKÉP MEGADÁSA

CSS-ben nem csak mozaikszerű háttér elrendezésre van mód

A háttérképet igazíthatjuk

**background-position**

vízszintesen (left, center, right, %, hossz) függőlegesen (top, center, bottom, %, hossz)

Beállíthatjuk az ismétlődés módját

**background-repeat**

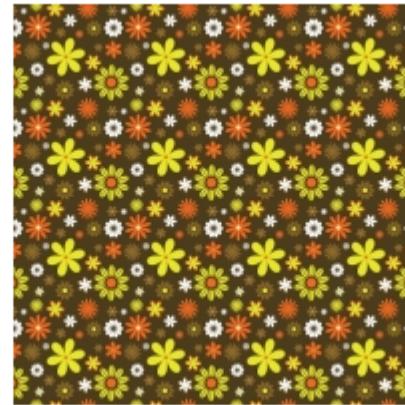
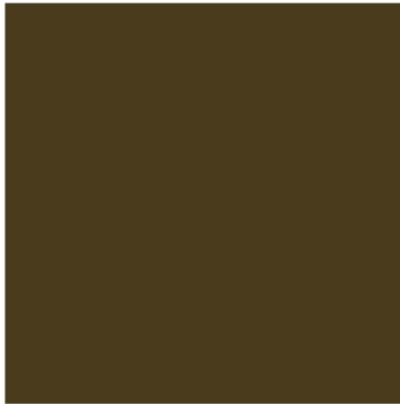
Nincs (no-repeat), mozaikszerű (repeat), vízszintesen (repeat-x),  
függőlegesen (repeat-y)

Csatolás módja

**background-attachment**

Fix (fixed) vagy tartalommal gördíthető (scroll)

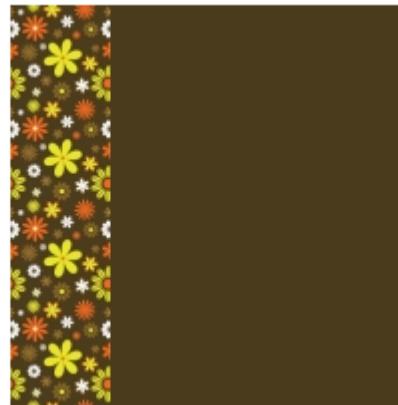
# PÉLDÁK



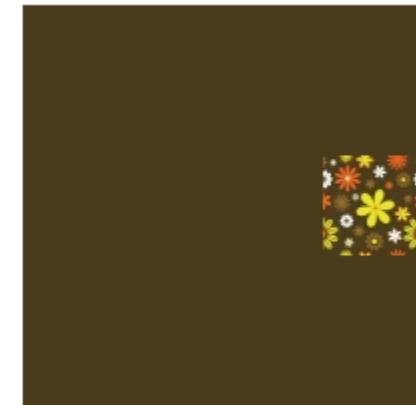
Egy **div elem**, beállított háttérszínnel.  
Ennél próbáljuk ki a különböző háttér  
elhelyezéseket.



```
background-image:url('hatter.jpg');  
background-repeat:repeat-x;
```

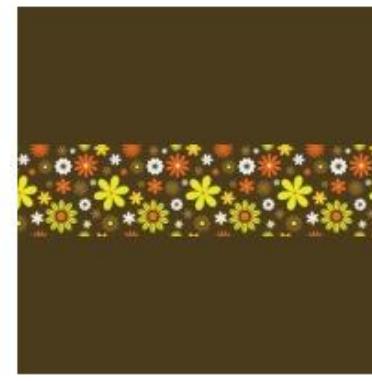
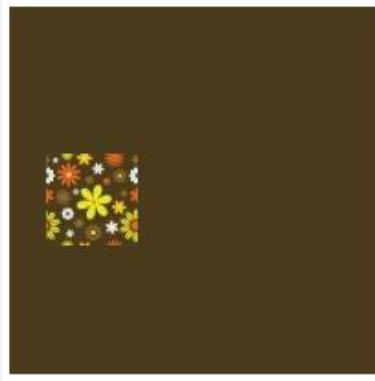


```
background-image:url('hatter.jpg');  
background-repeat:repeat-y;
```



```
background-image:url('hatter.jpg');  
background-repeat:no-repeat;  
background-position:center right;
```

# PÉLDÁK



```
background-image:url('hatter.jpg');  
background-repeat:no-repeat;  
background-position:20px 80px;
```

```
background-image:url('hatter.jpg');  
background-repeat:no-repeat;  
background-position:50% 50%;
```

```
background-image:url('hatter.jpg');  
background-repeat:repeat-x;  
background-position:50% 50%;
```

# TÖBB HÁTTÉRKÉP MEGADÁSA (CSS3)



```
background-image:url('hatter.jpg'), url('hatter2.jpg');  
background-repeat:no-repeat, no-repeat;  
background-position:left top, right bottom;
```



```
background-image:url('hatter.jpg'), url('hatter2.jpg');  
background-repeat:repeat-y, repeat-x;  
background-position:left top, right bottom;
```

A többöt gyakorlaton megnézzük!

```
background-image:url('hatter.jpg'), url('hatter2.jpg');  
background-repeat:repeat-x, repeat;  
background-position:left top, right bottom;
```

## SHORTHAND MEGADÁS SPECIALITÁSA

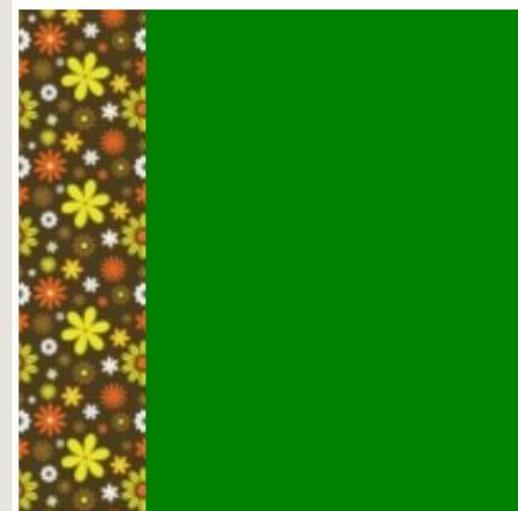
- A rövidített (shorthand) paraméter megadásnál ügyelni kell arra, hogy ezzel minden olyan altulajdonságot is alapértelmezett állapotba tudunk hozni, amit explicit módon nem adtunk meg.
- Például háttérként egyaránt beállíthatnánk színt és képet. Mindkettő egyszerre tudna érvényesülni, ahol nincs kép (mert pl. a kép függőlegesen ismétlődik), ott a háttérszín látszódna.

# HÁTTÉRSZÍN FELÜLÍRÁSA

```
<style type="text/css">
div {
  width:200px;
  height:200px;
  background-color:#4a3b1c;
  background-image:url('hatter.jpg');
  background-repeat:repeat-y;
}
</style>
```



```
<style type="text/css">
div {
  width:200px;
  height:200px;
  background-color:#4a3b1c;
  background-image:url('hatter.jpg');
  background-repeat:repeat-y;
}
div {background-color:green}
</style>
```



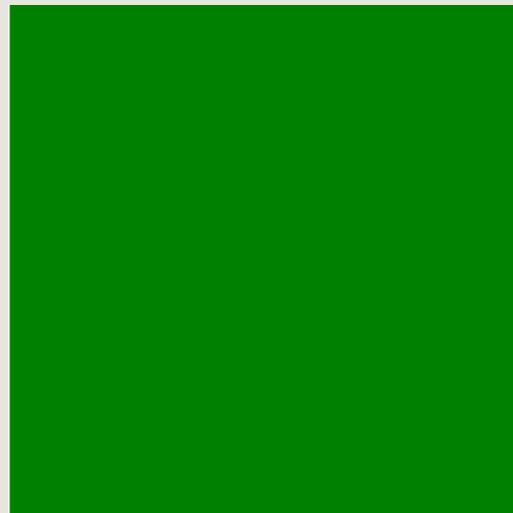
# HÁTTÉRSZÍN FELÜLÍRÁSA SHORTHAND MEGADÁSSAL

```
<style type="text/css">
div {
width:200px;
height:200px;
background-color:#4a3b1c;
background-image:url('hatter.jpg');
background-repeat:repeat-y;
}
</style>
```



```
<style type="text/css">
div {
width:200px;
height:200px;
background-color:#4a3b1c;
background-image:url('hatter.jpg');
background-repeat:repeat-y;
}

</style>
```



# VIZUÁLIS FORMÁZÁSMODELL

<https://www.w3.org/TR/css-display-3/>

# ALAPFOGALMAK

- A dokumentumfa elemei a doboz modellnek megfelelően jelennek meg, ahol alapvetően fontos a(z)
  - a doboz mérete és a doboz típusa (blokk, inline),
  - a pozícionálási séma,
  - az elemek viszonya a dokumentumfában,
  - külső információk (viewport [képmező] mérete)

# BLOKKSZINTŰ ÉS SORON BELÜLI (INLINE) ELEMEK

- Blokkszintű elemek
  - Vizuálisan blokként jelennek meg (pl. bekezdés, lista, div)
  - A display tulajdonság következő értékei tartoznak ide: block, list-item, table
- Soron beüli elemek
  - Vizuálisan nem alkotnak blokkot, soron belül rendeződnek (pl. img, em, strong, span)
  - A display tulajdonság következő értékei tartoznak ide: inline, inline-table, inline-block

# DIV ÉS SPAN TAGEK HASZNÁLATA A HTML NYELVBEN

<div></div>

- Ezzel az elemmel több más elemet körbekeríthetünk. Így egy adott div elemre beállított tulajdonságok öröklődnek a leszármazott elemekre.
- Gyakran használjuk a taget arra, hogy pozícionált blokkokat hozzunk létre.
  - HTML5-ben sok olyan strukturális tag jelenik meg, ami miatt a div-et ritkábban kell használnunk...

<span></span>

- Ez az elem soron belüli elem, vagyis főként karakterszintű formázások elvégzéséhez szoktuk használni.

# DISPLAY TULAJDONSÁG

**Display:** Egy elem megjelenítési módját adja meg két szempontból

- Külső megjelenési típus: hogyan vesz részt az adott elem a „flow layout” megjelenítésben
- Belső megjelenési típus: az elem gyermeké hogyan legyen elrendezve.

```
display: block    flow;  
display: inline  table;  
display: flex    run-in;
```

# DISPLAY TULAJDONSÁG

A paraméterek az alábbi kulcsszavakból adhatóak meg

- <display-outside> block, inline, run-in
- <display-inside> flow, flow-root, table, flex, grid, ruby
- <display-listitem> list-item
- <display-internal> table-row-group, table-header-group, table-footer-group, table-row, table-cell, table-column-group, table-column, table-caption, ruby-base, ruby-text, ruby-base-container, ruby-text-container
- <display-box> contents, none

<https://www.w3.org/TR/css-display-3>

<https://developer.mozilla.org/en-US/docs/Web/CSS/display>

# GYAKRAN HASZNÁLT DISPLAY TULAJDONSÁGOK

- **display: none**
  - Nem jelenik meg. Fókuszt nem kap (javascript segítségével igen).
- **display: inline**
  - Az elem soron belüli elem lesz (előtte és utána nem lesz sortörés)
- **display: inline-block**
  - Az elem soron belüli elem lesz (előtte és utána nem lesz sortörés), azonban beállítható a szélessége és magassága

## GYAKRAN HASZNÁLT DISPLAY TULAJDONSÁGOK

- **display: block**
  - Blokkszintű elemként viselkedik az elem
- **display: list-item**
  - A lista elemek két egymás mellett álló téglalapként jelennek meg, a bal oldali behúzásként is felfogható, a jobb oldaliban pedig a lista tartalma jelenik meg.

# DISPLAY TULAJDONSÁG - PÉLDA

```
<div id="menu">  
<ul>  
<li><a href="#">Kezdőlap</a></li>  
<li><a href="#">Magamról</a></li>  
<li><a href="#">Hobby</a></li>  
<li><a href="#">Linkek</a></li>  
</ul>  
</div>
```

- [Kezdőlap](#)
- [Magamról](#)
- [Hobby](#)
- [Linkek](#)



CSS nélkül

```
<style type="text/css">  
div#menu ul {padding:0; margin:0;}  
div#menu li {  
    list-style-type:none;  
    display:inline;  
    padding: 5px 20px;  
    border-right:1px dotted black;  
    margin:0;  
}  
div#menu li:last-child() {border:0}  
</style>
```

CSS alkalmazásával



[Kezdőlap](#)

[Magamról](#)

[Hobby](#)

[Linkek](#)

## GRID VS. FLEXBOX

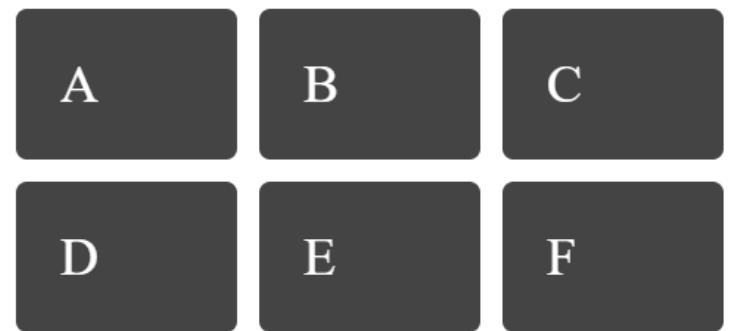
A CSS3 bevezeti a Grid, illetve Flexbox alapú elrendezést

- Grid
  - az oldal egy rácsként, vagyis oszlopok és sorok együtteseként fogható fel
- Flexbox
  - Itt a dobozok elrendezhetőek tetszőleges irányban, lineárisan egy fő tengely mentén, illetve keresztirányú tengely mentén
  - Automatikusan változik a dobozok mérete

# GRID PÉLDA

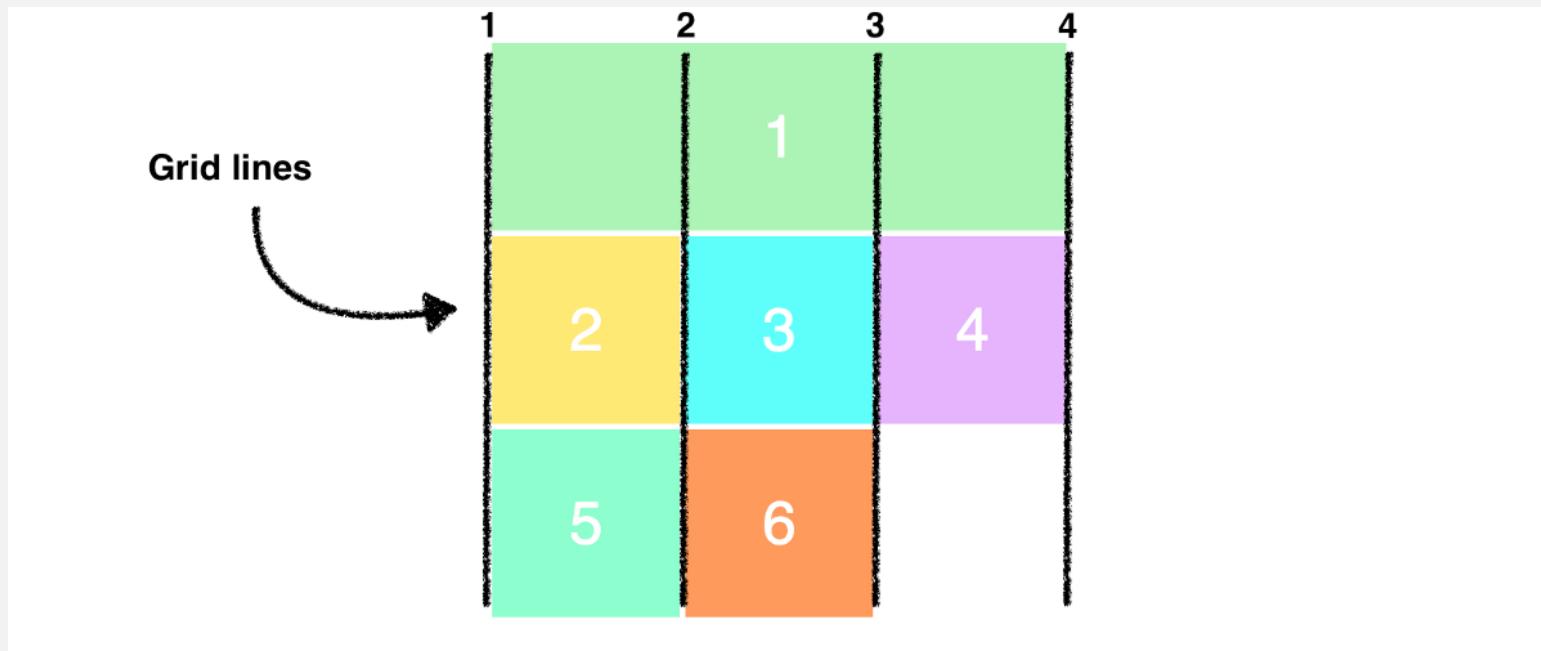
```
body {  
margin: 40px;  
}  
  
.wrapper {  
display: grid;  
grid-template-columns: 100px 100px 100px;  
grid-gap: 10px;  
background-color: #fff;  
color: #444;  
}  
  
.box {  
background-color: #444;  
color: #fff;  
border-radius: 5px;  
padding: 20px;  
font-size: 150%;  
}
```

```
<div class="wrapper">  
<div class="box a">A</div>  
<div class="box b">B</div>  
<div class="box c">C</div>  
<div class="box d">D</div>  
<div class="box e">E</div>  
<div class="box f">F</div>  
</div>
```



# JÓ LEÍRÁSOK A GRID MEGISMERÉSÉHEZ

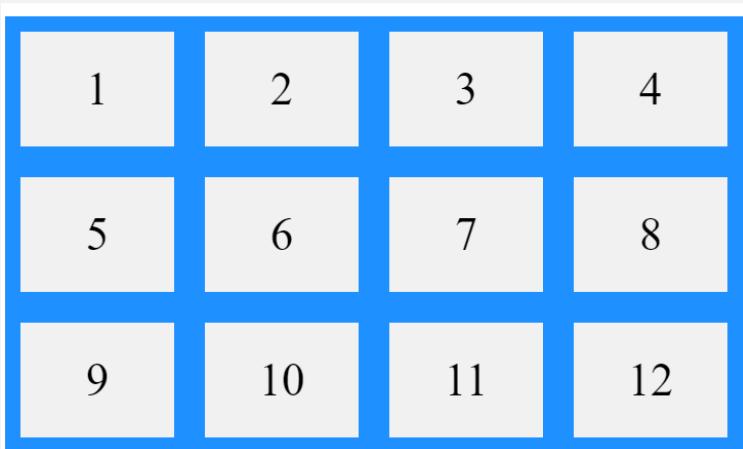
- <https://css-tricks.com/snippets/css/complete-guide-grid/>
- <https://medium.freecodecamp.org/learn-css-grid-in-5-minutes-f582e87b1228>



# FLEX PÉLDA

```
.flex-container {  
    display: flex;  
    flex-wrap: wrap;  
    background-color: DodgerBlue;  
}  
  
.flex-container > div {  
    background-color: #f1f1f1;  
    width: 100px;  
    margin: 10px;  
    text-align: center;  
    line-height: 75px;  
    font-size: 30px;  
}
```

```
<div class="flex-container">  
    <div>1</div>  
    <div>2</div>  
    <div>3</div>  
    <div>4</div>  
    <div>5</div>  
    <div>6</div>  
    <div>7</div>  
    <div>8</div>  
    <div>9</div>  
    <div>10</div>  
    <div>11</div>  
    <div>12</div>  
</div>
```



# FLEXBOX TANULÁSA JÁTÉKOSAN

- <https://flexboxfroggy.com/#hu>

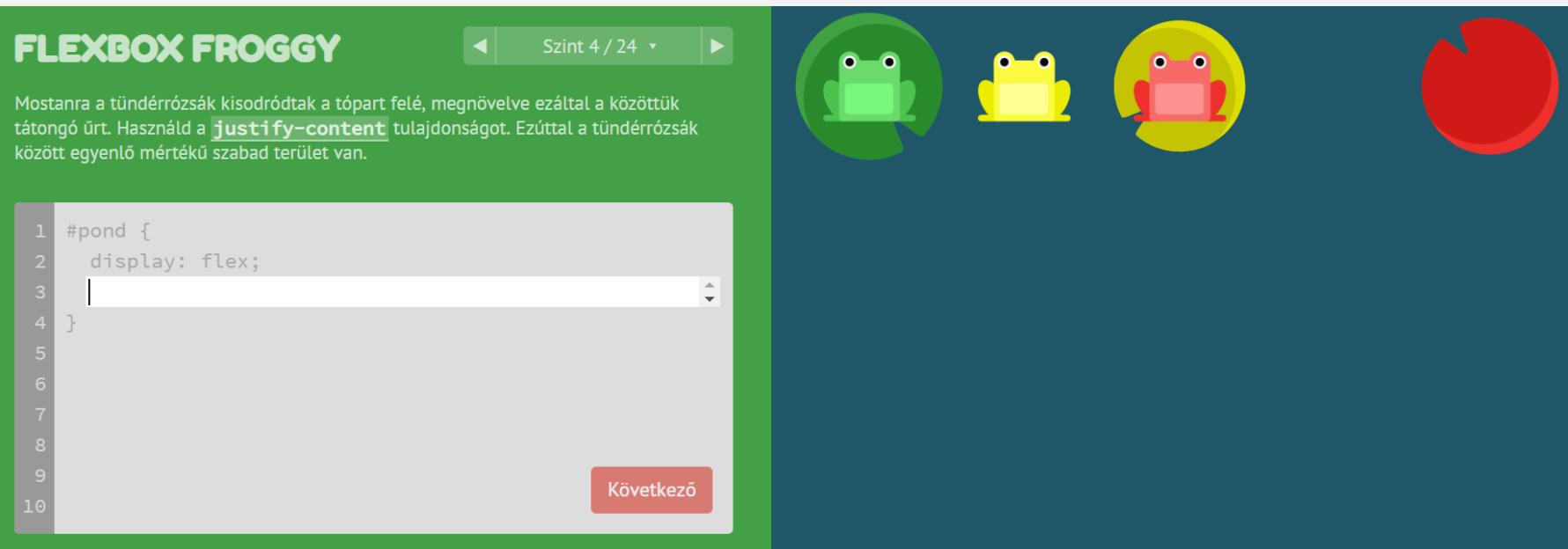
## FLEXBOX FROGGY

Szint 4 / 24 ▶

Mostanra a tündérrózsák kisodródtak a tópart felé, megnövelve ezáltal a közöttük tátongó űrt. Használ a **justify-content** tulajdonságot. Ezúttal a tündérrózsák között egyenlő mértékű szabad terület van.

```
1 #pond {  
2   display: flex;  
3 }  
4  
5  
6  
7  
8  
9  
10
```

Következő



## ELEMÉK LÁTHATÓ RÉSZE

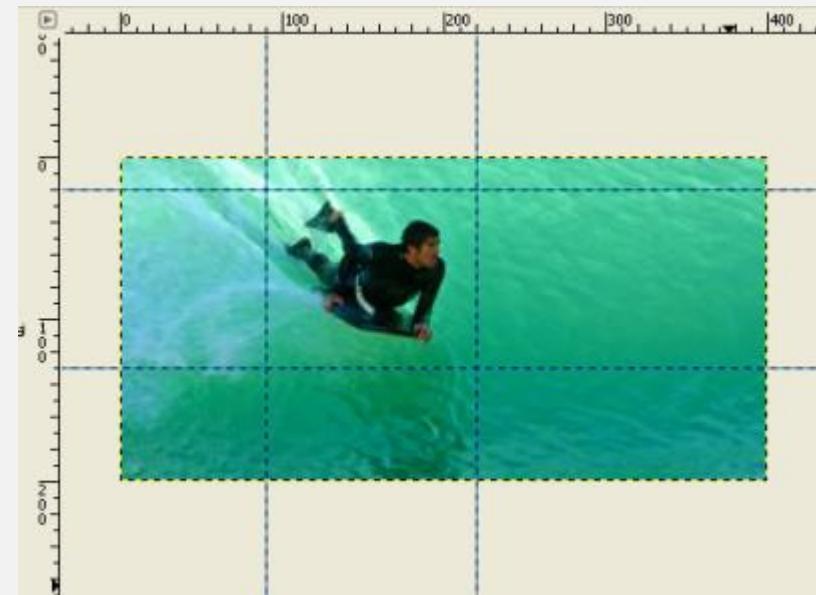
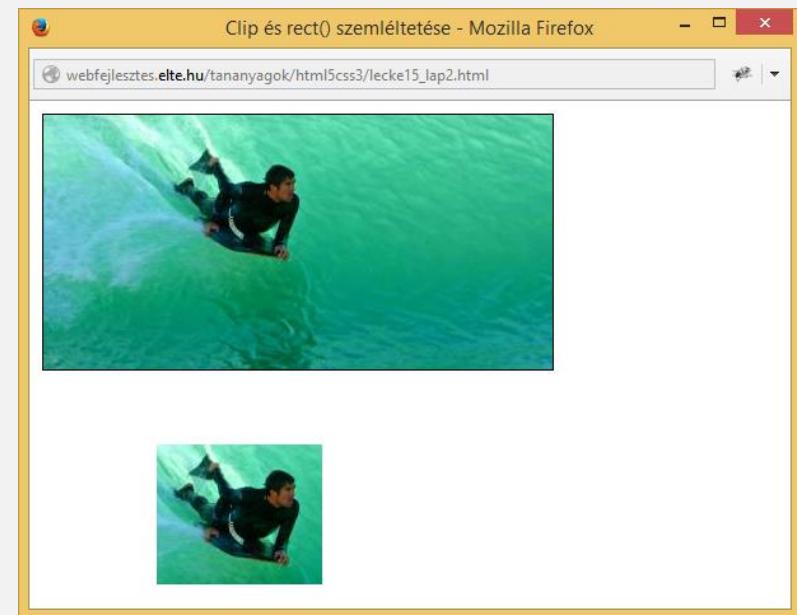
- A clip tulajdonság segítségével beállíthatjuk, hogy egy elemnek csak egy meghatározott **részét jelenítsük meg**.
- A rect() érték segítségével megadhatjuk azt a **téglalapot, amit látni szeretnénk** az elemből.
- A zárójelek között **négy, vesszővel elválasztott hosszúságértéket** adunk meg, melyek sorrendben a fenti, a jobb, az alsó és a bal értékeket határozzák meg. Ezek az értékek az elem bal felső sarkától számolódnak.

```

<style type="text/css">
div {
    width:400px; height:200px;
    border:1px solid black;
    padding:0; position: absolute;
}
div#doboz1 {
    left: 10px;
    top: 10px;
}

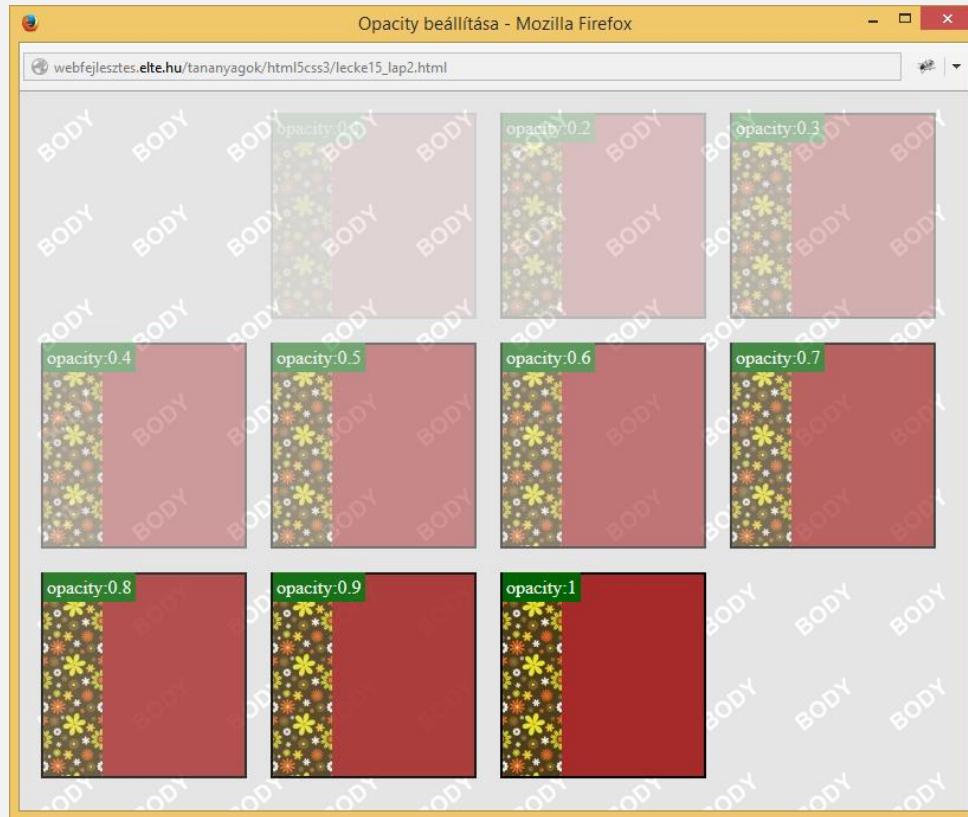
div#doboz2 {
    left: 10px;
    top: 250px;
    clip: rect(20px,220px,130px,90px);
}
</style>
...
<div id="doboz1"></div>
<div id="doboz2"></div>

```



# ÁTLÁTSZÓSÁG (OPACITY)

- Megadhatunk egy tetszőleges 0.0 (teljesen átlátszó) és 1.0 (teljesen átlátszatlan) közötti **valós számot**.



# DOBOZ ÁRNYÉK (BOX-SHADOW)

- Először két tetszőleges hosszúságértékkel adjuk meg a vízszintes és függőleges eltolás mértékét.
- Ezután egy pozitív hosszúságérték következik, ami meghatározza, hogy mennyire legyen elmosódott az árnyék. 0 érték esetén nincs elmosódás.
- Majd opcionális módon egy pozitív hosszúságértéket megadva meghatározhatjuk az árnyék (megadott színnel egyező) részének méretét. Ha nem adjuk meg, akkor 0-nak tekinti a bőngésző az értékét.
- Ezután egy tetszőleges színértéket megadva beállítjuk az árnyék színét (akár átlátszóság megadásával is, lásd rgba).
- Opcionálisan megadva az inset hatására az árnyék az elemen belülre kerül.

# DOBOZ ÁRNYÉKOK

```
box-shadow:  
5px 5px 10px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
-5px 5px 10px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
5px -5px 10px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
-5px -5px 10px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
10px 10px 10px 0  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
10px 10px 10px 5px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
10px 10px 10px 0  
rgba(0, 0, 0, 0.6);
```

```
box-shadow:  
inset 5px 5px 5px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
inset 5px -5px 5px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
inset -5px 5px 5px  
rgba(0, 0, 0, 1);
```

```
box-shadow:  
inset -5px -5px 5px  
rgba(0, 0, 0, 1);
```

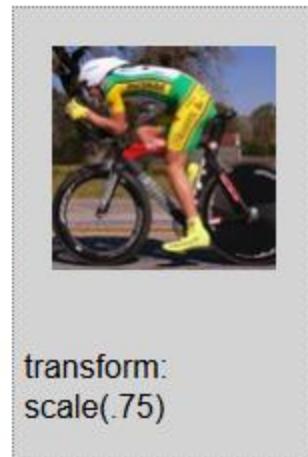
```
box-shadow:  
5px 5px 5px rgba(0, 0, 255, 1),  
inset 5px 5px 5px rgba(255, 0,  
0, 1);
```

# TRANSZFORMÁCIÓK ÉS ÁTMENETEK

- A rotate() érték **elforgatja** az elemet a képernyő síkján, mégpedig a paraméterül kapott szöggel.
- A rotateX() és a rotateY() értékek a paraméterül kapott szöggel **forgatják el** az elemet a képernyő **vízszintes**, és **függőleges** tengelye körül.
- A scale() érték a paraméterül kapott egy vagy (vesszővel elválasztva) két számérték szorosára **méretezi át** az elemet. Az első paraméter a vízszintes, az opcionális második a függőleges átméretezésért felel.
- A scaleX()/scaleY() értékek a paraméterül kapott számérték szeresére **méterezi át** az elemet **vízszintesen/függőlegesen**.



Eredeti kép



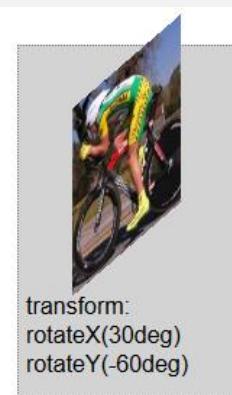
transform:  
scale(.75)



transform:  
rotate(10deg)

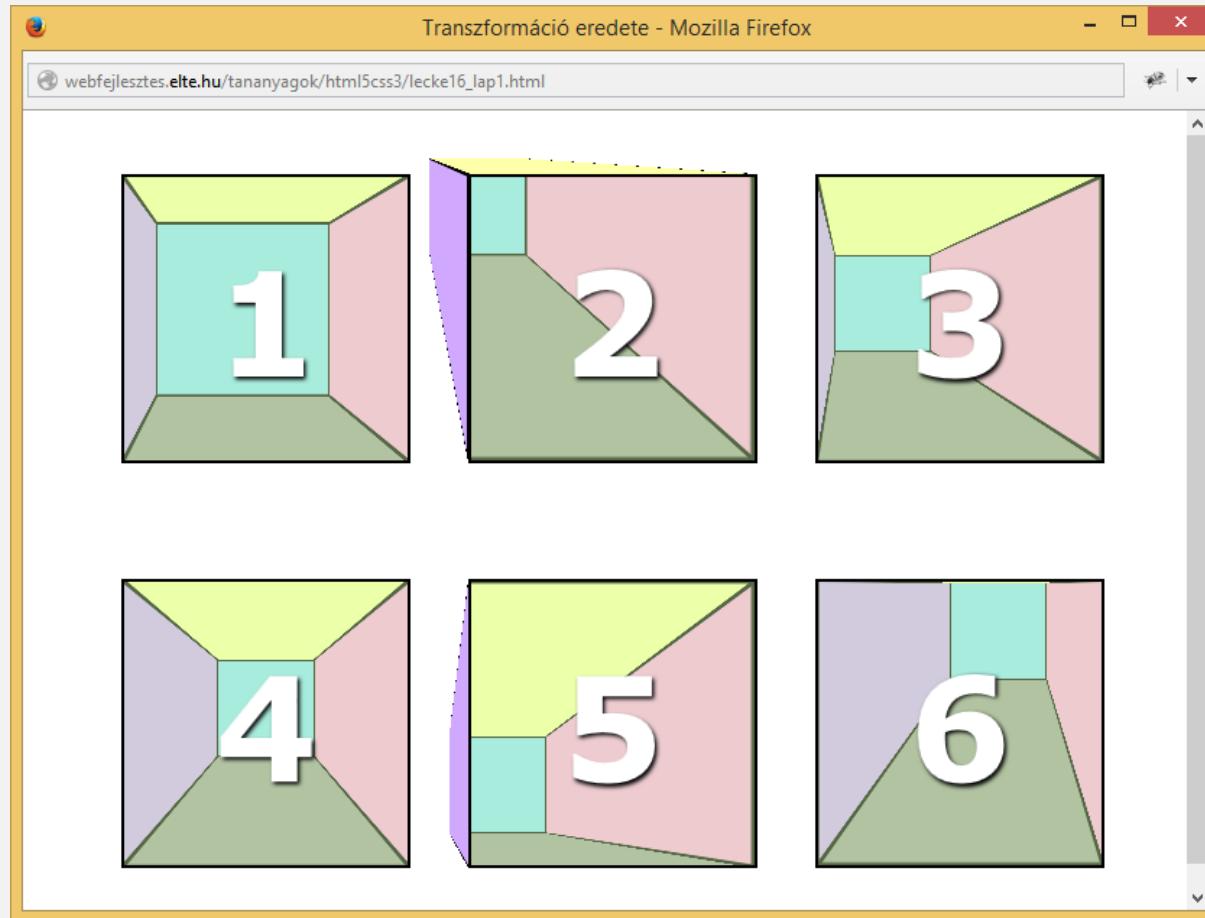
# TRANSZFORMÁCIÓK ÉS ÁTMENETEK

- A skew() érték a paraméterül kapott egy vagy (vesszővel elválasztva) két szög mértékével **ferdítí** az elemet. Az első paraméter a vízszintes, az opcionális második a függőleges ferdítésért felel.
- A skewX()/skewY() értékek a paraméterül kapott szöggel **ferdítik** el az elemet **vízszintesen/függőlegesen**.
- A translate() érték a paraméterül kapott egy vagy (vesszővel elválasztva) két hosszúságérték mértékével **tolja el** az elemet. Az első paraméter a vízszintes, az opcionális második a függőleges eltolásért felel. A pozitív értékek jobbra/le a negatív értékek balra/fel történő eltolást eredményeznek.
- A translateX()/translateY() értékek a paraméterül kapott hosszúságértékkkel **tolják el** az elemet **vízszintesen/függőlegesen**.



# 3D TRANSZFORMÁCIÓK

- Részletesebben a web-es tananyagban térünk ki...



# POZÍCIONÁLÁSI SÉMÁK

<https://www.w3.org/TR/css-position-3/>

# KI ITT BELÉPSZ, HAGYJ FEL MINDEN REMÉNNYEL

## Humanity's victories:



**land probe perfectly on a comet 310 million miles away, using science**



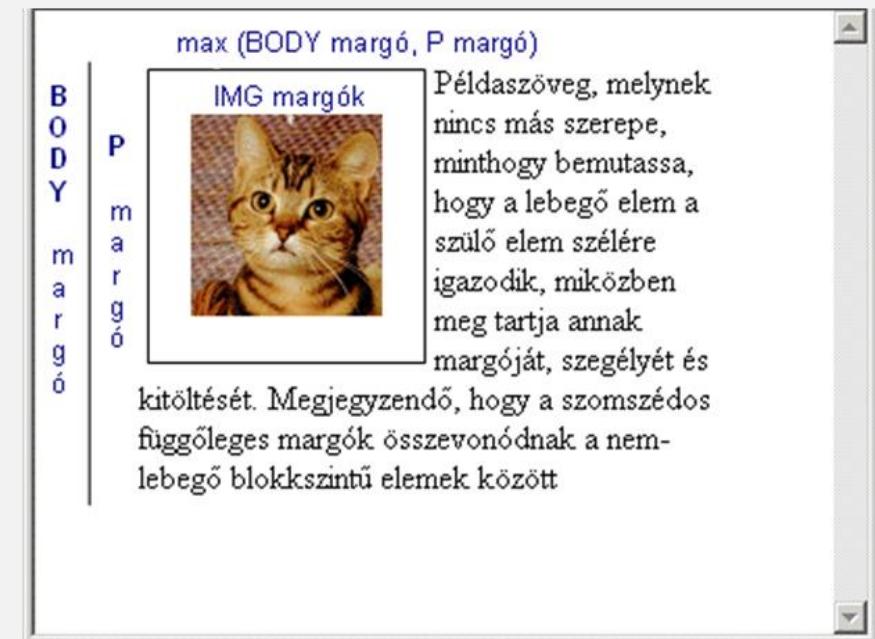
**get stuff on a web page to align properly  
using CSS**

# CSS 3 POZÍCIONÁLÁSI SÉMÁK

- Három pozícionálási séma
  - **Normál elrendezés:** blokk formázás, soron belüli formázás, relatív és sticky pozícionálás
  - **Lebegés (float):** Balra, illetve jobbra történhet a lebegtetés. A lebegtetett elemet a többi tartalom körülveszi/körülveheti.
  - **Abszolút pozícionálás:** A normál elrendezésből kikerül, a tartalmazó blokk pozíciójához képest történik az elrendezés.

# LEBEGŐ ELEMEK (FLOAT: LEFT | RIGHT | NONE)

```
<!DOCTYPE html>
<html>
<head>
<title>stíluslap példa</title>
<style type="text/css">
    img {float: left}
    body, p, img {margin: 25px}
</style>
</head>
<body>
    <p>
        
        Példaszöveg, melynek nincs más
        szerepe, minthogy bemutassa, hogy a
        lebegő elem a szülő elem szélére
        igazodik, miközben megtartja annak
        margóját, szegélyét és kitöltését.
        megjegyzendő, hogy a szomszédos
        függőleges margók összevonódnak a nem-
        lebegő blokkszintű elemek között
    </p>
</body>
</html>
```

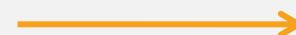


# OVERFLOW TULAJDONSÁG

- Mi történik, ha a tartalom mérete nagyobb, mint a tartalmazó elemé?

## **overflow: visible**

Ez az alapbeállítás, ekkor a tartalom kilóghat a tartalmazó elemből.



## **overflow:hidden**

A túlnyúlás el lesz rejtve.



## **overflow:auto**

A túlnyúlás el lesz rejtve, de megjelenik egy gördítősáv, ha szükséges.



## **overflow:scroll**

A túlnyúlás el lesz rejtve, és mindenképpen megjelenik a gördítősáv..



# OVERFLOW TULAJDONSÁG

- Ha nem szeretnénk, hogy egy lebegtetett elem kilógjon a dobozból, akkor is sikерrel használhatjuk az `overflow:hidden`, vagy `overflow:auto;` tulajdonságot.



`div.doboz {  
background-color:lightyellow;  
border:1px solid green;  
padding:5px;  
}`

`div.doboz {  
background-color:lightyellow;  
border:1px solid green;  
padding:5px;  
overflow:hidden;  
}`

`div.doboz {  
background-color:lightyellow;  
border:1px solid green;  
padding:5px;  
overflow:hidden;  
}`



`div.doboz {  
background-color:lightyellow;  
border:1px solid green;  
padding:5px;  
overflow:hidden;  
}`



# LEBEGŐ ELEMEK ÉS A CLEAR TULAJDONSÁG

- Clear: **none** | **left** | **right** | **both** | **inherit**
  - az elem melyik oldalán engedélyezi lebegő elemek megjelenését.
- Ha a **clear** értéke **leftre** van beállítva, az elem, amelyre alkalmaztuk, a bal oldalán levő bármely lebegő elem alá kerül.

# LEBEGŐ ELEMÉK (CLEAR: LEFT|RIGHT|BOTH|INHERIT)

```
<!DOCTYPE html>
<html>
<head>
<title></title>
<style type="text/css">
    img      {margin:5px;}
    img.bal  {float:left;}
    img.jobb {float:right;}
    p.tores  {clear:both}
</style>
</head>
<body>

<p>
    Lorem ipsum dolor sit amet, consectetur
    adipiscing elit...
<p class="tores">Cum sociis natoque
    penatibus et magnis dis parturient
    montes...

<p>
    Phasellus tortor nisi, venenatis sit amet
    tincidunt ut, ...
</body>
</html>
```

Hiába van még hely a kép mellett, a clear tulajdonság miatt a bekezdés a kép alá kerül...



Donec elit libero, feugiat ac euismod sit amet, accumsan a mi. Curabitur mi urna, consectetur vel elementum eget, vestibulum id ligula.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Mauris vestibulum gravida ligula, eget semper ligula tincidunt in. In vitae leo ac metus consectetur feugiat. Nullam pellentesque massa ut ipsum fringilla tristique. In tellus lectus, posuere sed rutrum sit amet, auctor et nunc. Sed a sapien nulla, ut ullamcorper ipsum. Praesent eu nibh a mi convallis consequat vitae sit amet odio.

Phasellus tortor nisi, venenatis sit amet tincidunt ut, fringilla nec diam. Vestibulum commodo, ante non dictum mattis, leo velit blandit nisl, convallis pellentesque arcu justo vel libero. Ut erat augue, eleifend et lobortis eget, dignissim quis lectus. Cras felis magna, condimentum ut porta a, pretium nec nulla. Maecenas faucibus leo in ante venenatis eu molestie orci dapibus. Sed pharetra laoreet euismod. Nulla facilisi. Aliquam ultrices mollis elementum. Curabitur urna metus, eleifend in malesuada eget, hendrerit vel urna. Aenean blandit lectus ac justo dignissim tempus. Quisque faucibus mollis pharetra. Vivamus justo eros, condimentum a semper sed, ornare sed justo. Fusce id dui tortor accumsan sollicitudin urna. Curabitur ac augue libero nec mattis lorem.



# POZÍCIONÁLÁS

Position: static | relative | absolute | fixed | inherit

## Static

Nem pozícionált, a kódban elfoglalt helye határozza meg a váznon való elhelyezkedését. A top, right, bottom, left paramétereknek nincs hatása az elemre.

## Relative

A statikushoz hasonló helyre kerül az elem, de a bal és felső pozíció megadásával eltolhatjuk a top, right, bottom, left paraméterek használatával. (pl. kerüljön lejjebb és jobbra)

## Absolute

Ez az elem kikerül a megjelenítési folyamatból. Az abszolút pozícionálású elemek a (dokumentumfában) legközelebb eső olyan szülő elemhez képest lesznek pozícionálva, amelyikre szintén meg van adva a position tulajdonság MÁS értékkel, mint a static.

## Fixed

A fix pozicionálással a görgetés ellenére is adott pozíció marad az elem. (képernyőkoordinátához rögzül)

## Sticky

A legközelebbi gördíthető ōs elemhez képest történő relatív megadást tesz lehetővé-

# STATIKUS POZÍCIONÁLÁS

```
<style type="text/css">  
div#doboz1 {  
    position: static;  
}
```

```
div#doboz2 {  
    position: static;  
}  
</style>
```

...

```
<div id="doboz1">Lorem ipsum dolor  
sit amet...</div>  
<div id="doboz2">Lorem ipsum dolor  
sit amet...</div>
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

A dobozok pozíciója a kódban elfoglalt sorrendtől függ.

# RELATÍV POZÍCIONÁLÁS

```
div#doboz1 {position: static;}  
div#doboz2 {  
    position: relative;  
    width: 90%;  
    left: 50px;  
    top: 10px;  
}  
div#doboz3 {  
    position: relative;  
    width: 400px;  
    left: 20px;  
    top: 20px;  
}  
</style>  
...
```

```
<div id="doboz1">Lorem ipsum dolor sit  
amet...</div>  
<div id="doboz2">Lorem ipsum dolor sit  
amet...</div>  
<div id="doboz3">Lorem ipsum dolor sit  
amet...</div>
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz3: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

A dobozt eltolhatjuk ahhoz a pozícióhoz képest, ahova amúgy (statikus poz. esetén) kerülne.

# ABSZOLÚT POZÍCIONÁLÁS (I.)

```
<style type="text/css">  
div#doboz1 {  
    position: absolute;  
    width: 150px;  
    left: 10px;  
    top: 10px;  
}  
  
div#doboz2 {  
    position: absolute;  
    width: 50%;  
    left: 170px;  
    top: 30px;  
}
```

```
</style>
```

```
<div id="doboz1">Lorem ipsum dolor sit  
amet...</div>  
<div id="doboz2">Lorem ipsum dolor sit  
amet...</div>
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Az 1.doboz fix, a 2. doboz változó szélességű.

# ABSZOLÚT POZÍCIONÁLÁS (2.)

```
<style type="text/css">
div#doboz1 {
    position: absolute;
    width: 250px;
    left: 10px;
    top: 10px;
}

div#doboz2 {
    position: absolute;
    width: 250px;
    left: 200px;
    top: 40px;
}
</style>
...
<div id="doboz1">Lorem ipsum dolor
sit amet...</div>
<div id="doboz2">Lorem ipsum dolor
sit amet...</div>
```

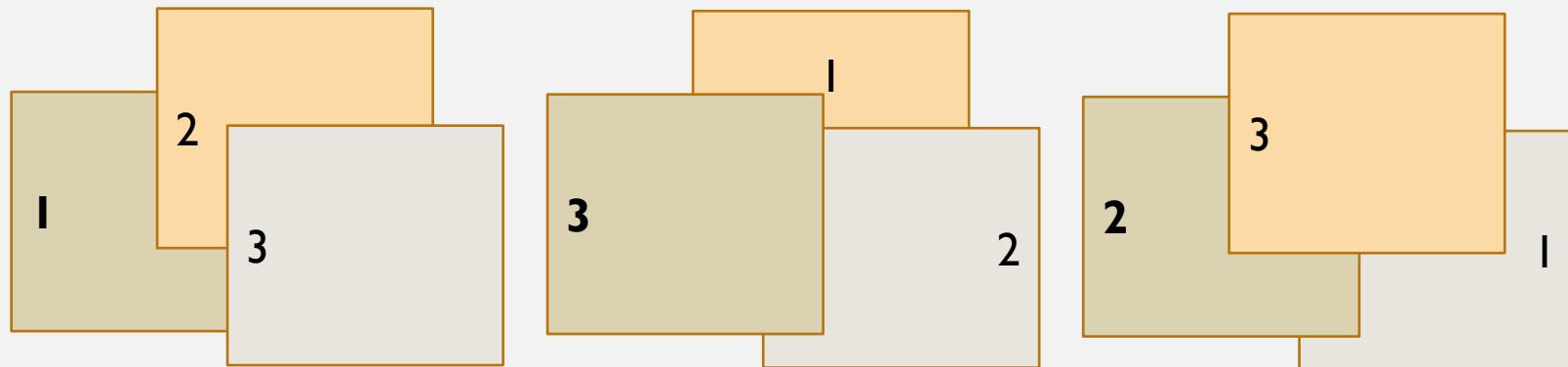
doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis tempor mattis, bibendum ut, ligula.

doboz2: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

A két doboz akár egymásra is kerülhet. A 2. doboz takarja az elsőt, mert később szerepel a kódban. Ez felülbírálható a z-index paraméterrel. Lásd köv. dia.

# RÉTEGEK

- A CSS alapú megjelenítéskor minden abszolút-ként pozícionált elemnek van egy pozíciója (z-index) a harmadik dimenzióban, a "z" tengely mentén. Minél nagyobb a szám, annál feljebb kerül.



# ABSZOLÚT POZÍCIONÁLÁS (3.)

```
<style type="text/css">
div#doboz1 {
    position: absolute;
    width: 250px;
    left: 10px;
    top: 10px;
    z-index:2;
}

div#doboz2 {
    position: absolute;
    width: 250px;
    left: 200px;
    top: 40px;
    z-index:1;
}
</style>

<div id="doboz1">Lorem ipsum dolor sit amet...
<div id="doboz2">Lorem ipsum dolor sit amet...
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

orem ipsum dolor sit amet, er adipiscing elit. Phasellus placerat vel, rutrum vitae, quis, eros. Morbi elementum lit. Aliquam erat volutpat. varius ultrices neque. Fusce sis, eleifend vel, placerat mentum ac, augue. ue erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

A z-index paraméterrel megadhatjuk a mélységi sorrendet. A nagyobb szám feljebb elhelyezést eredményez a rétegeket tekintve.

# ABSZOLÚT POZÍCIONÁLÁS (4.)

```
<style type="text/css">
body {
    margin-left: 180px;
    margin-right: 10px;
    margin-top: 10px;
    margin-bottom: 10px;
}

div#doboz1 {
    position: absolute;
    width: 150px;
    left: 10px;
    top: 10px;
}

</style>
```

```
<div id="doboz1">Lorem ipsum dolor sit amet...
<div id="doboz2">Lorem ipsum dolor sit amet...
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2:

Lore ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lore ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lore ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt

Itt a body 180px-es bal margója miatt kerül a 2. doboz jobbra, és az első doboz a margó miatt előálló üres helyre van bepozícionálva abszolút módon.

# ABSZOLÚT POZÍCIONÁLÁS (5.)

```
<style type="text/css">
div#doboz1 {
    position: absolute;
    left: 1%;
    width: 18%;
    top: 10px;
}

div#doboz2 {
    position: absolute;
    left: 22%;
    width: 55%;
    top: 10px;
}

div#doboz3 {
    position: absolute;
    left: 80%;
    width: 18%;
    top: 10px;
}
</style>
```

doboz1: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz3: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

```
<div id="doboz1">Lorem ipsum dolor sit amet...</div>
<div id="doboz2">Lorem ipsum dolor sit amet...</div>
<div id="doboz3">Lorem ipsum dolor sit amet...</div>
```

Három oszlopos, folyékony arculat, melyben a pozícionálás abszolút módon, %-os értékekkel történik.

# LEBEGTETÉS

```
...  
<style type="text/css">  
div#doboz2 {  
    background-color: #C0D8C8;  
    float: left;  
    width: 300px;  
    margin: 5px;  
}  
</style>  
...
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula. Lorem ipsum dolor sit amet, consectetuer adipiscing elit.

Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque.

```
<div id="doboz1">Lorem ipsum dolor sit amet  
    <div id="doboz2">Lorem ipsum dolor sit  
amet...</div>  
        Lorem ipsum dolor sit amet...  
</div>  
<div id="doboz3">Lorem ipsum dolor sit amet</div>
```

A lebegtetés miatt a többi elem körbefolyja az adott elemet.

# LEBEGTETÉS (2.)

```
<style type="text/css">
div#doboz2 {
    float: right;
    width: 100px;
    margin: 5px;
}

div#doboz3 {
    float: right;
    width: 100px;
    margin: 5px;
    clear: right;
}
</style>
```

```
<div id="doboz1">Lorem ipsum dolor sit amet
<div id="doboz2">Lorem ipsum dolor sit amet...</div>
<div id="doboz3">Lorem ipsum dolor sit amet</div>
Lorem ipsum dolor sit amet
</div>
```

doboz1:

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit. Aliquam erat volutpat. Praesent varius ultrices neque. Fusce lorem turpis, eleifend vel, placerat vitae, elementum ac, augue. Pellentesque erat tellus, facilisis vitae, tempor mattis, bibendum ut, ligula.

doboz2: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros.

doboz3: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros.

doboz3: Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus orci urna, placerat vel, rutrum vitae, tincidunt quis, eros. Morbi elementum semper velit.

A clear tulajdonság miatt a 3. doboz nem kerülhet a 2. doboz mellé.

## HELY KISZÁMÍTÁSA (CALC)

- Gondot okoz, ha a méretezésnél többféle mértékegységet (px,em,%) használtunk, és tudnunk kellene, mekkora a rendelkezésre álló hely.
- Erre jó a **calc()**
- Példa

```
#blokk { width: calc(50% - 100px); }
```

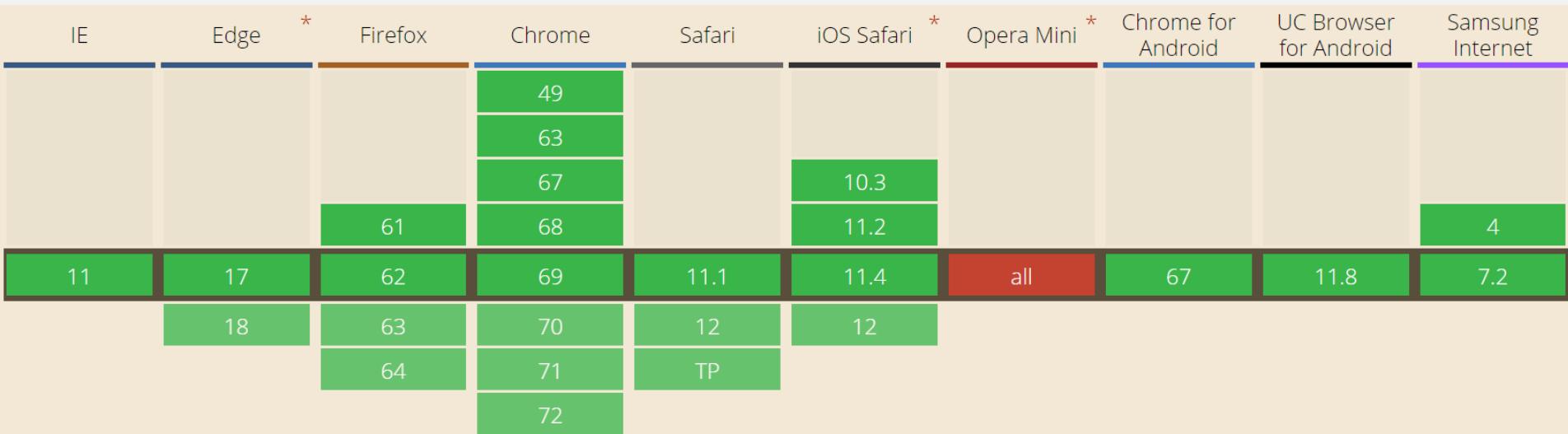
- A blokk 100 képponttal keskenyebb, mint a rendelkezésre álló hely 50%-a

# HELY KISZÁMÍTÁSA (CALC)

- Ahhoz, hogy régebbi böngészőkben is működjön böngészőre vonatkozó prefixeket is használni kell..
  - -webkit: safari/chrome
  - -moz: mozilla firefox
  - -o: opera
  - -ms: microsoft

```
#blokk {  
    width: -webkit-calc(50% - 100px);  
    width: -moz-calc(50% - 100px);  
    width: calc(50% - 100px);  
}
```

# CALC BÖNGÉSZŐTÁMOGATOTTSSÁGA



<http://caniuse.com/#feat=calc>

# LÁTHATÓSÁG

- **Visibility:** visible | hidden | collapse
  - Visible: látható
  - Hidden: rejtett
  - Collapse: táblázat elemeknél használatos, egy oszlop vagy sor elrejtésére használható
- Attól, hogy az elem nem jelenik meg, attól még a helyet fenntartja a böngésző. Ha nem ezt akarjuk, akkor a display: none; tulajdonságot használjuk!

# LÁTHATÓSÁG PÉLDA

```
<!DOCTYPE html>
<html>
<head>
  <title>Visibility</title>
  <style type="text/css">
    .rejtett
      {visibility:hidden; float:left; }
  </style>
</head>
<body>

<p>Lorem ipsum ...</p>
</body>
</html>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut tempus vitae massa a tincidunt. Proin accumsan aliquam quam sit amet tristique. Proin convallis ex felis, quis viverra turpis commodo eu. Phasellus at lacus nisi. Quisque vel sem sodales, varius felis a, egestas tellus. Nullam scelerisque venenatis maximus. Phasellus blandit vulputate tempus.

Cras sit amet tempor metus, in pharetra velit. Maecenas non ipsum id tortor vestibulum rutrum. Integer massa lacus, gravida in condimentum fringilla, placerat ut metus. Donec porttitor gravida libero, vel luctus eros gravida at.

# LAYOUTOK (ELRENDEZÉSEK)

# LEARN CSS LAYOUT

- <http://learnlayout.com/>



## Learn CSS Layout

This site teaches the CSS fundamentals that are used in any website's layout.

I assume you already know what selectors, properties, and values are. And you probably know a thing or two about layout, though it may still be a rage-provoking activity for you. If you want to learn HTML and CSS from the beginning, you should check out [this tutorial](#). Otherwise, let's see if we can save you some fury on your next project.

[Get Started](#)

# CSS LAYOUT GENERATOR

- <http://csslayoutgenerator.com/>

The screenshot shows the CSSLayoutGenerator.com interface. On the left, there are several configuration sections:

- DOCTYPE:** HTML 5
- CSS Reset:** CSS reset from Eric ...
- Layout width:** Fixed width: 1000 px
- Header:** Height
- Footer:** Height
- Sidebars:** 2 sidebars (selected), both left, width: 250 px
- Additional options:** Keep the footer at the bottom of browser window (selected)

In the center, a large yellow button says "GENERATECSS". Below it, a banner reads "30+ CSS Generators now all in one place. Click here! NEW". To the right, there is a "Preview" section showing a wireframe of a three-column layout with a header, two sidebars, and a footer.

# FREE CSS LAYOUTS

- <http://www.free-css.com/free-css-layouts/page1>

**FREE CSS LAYOUTS PAGE 1**

« Previous | 1 | 2 | 3 | ... | 20 | 21 | Next »

**Free CSS Layouts**

**Navigation Bar:** Free CSS Templates  
Free CSS Layouts

**Main Content Area:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Text:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Free CSS Layouts**

**Navigation Bar:** Free CSS Templates  
Free CSS Layouts

**Main Content Area:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Text:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Free CSS Layouts**

**Navigation Bar:** Free CSS Templates  
Free CSS Layouts

**Main Content Area:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Text:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Free CSS Layouts**

**Navigation Bar:** Free CSS Templates  
Free CSS Layouts

**Main Content Area:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Text:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Free CSS Layouts**

**Navigation Bar:** Free CSS Templates  
Free CSS Layouts

**Main Content Area:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

**Text:** Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusamus et iusto odio dignis sit aspernatur vero. Etiam velit.

# RESPONSIFY - A RESPONSIVE TEMPLATE GENERATOR

- <http://app.responsify.it/>

responsify	nav item	nav item	nav item
<b>section h1</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec.</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.</p>	<b>aside h2</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor.</p>	<b>aside h2</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor.</p>	
<b>section h2</b> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam sodales urna non odio egestas tempor. Nunc vel vehicula ante. Etiam bibendum iaculis libero, eget molestie nisl pharetra in. In semper consequat est, eu porta velit mollis nec. Curabitur posuere enim eget turpis feugiat tempor. Etiam ullamcorper lorem dapibus velit suscipit ultrices. Proin in est sed erat facilisis pharetra.</p>			

# CSS TULAJDONSÁGOK

<https://www.w3.org/Style/CSS/all-properties.en.html>

# CSS I,2,3 TULAJDONSÁG LISTA

- <http://meiert.com/en/indices/css-properties/>

Property	CSS 1	CSS 2	CSS 2.1	CSS 3	Initial Value
<a href="#">alignment-adjust</a>	X	X	X	✓	auto
<a href="#">alignment-baseline</a>	X	X	X	✓	baseline
<a href="#">animation</a>	X	X	X	✓	depending on individual properties
<a href="#">animation-delay</a>	X	X	X	✓	0
<a href="#">animation-direction</a>	X	X	X	✓	normal
<a href="#">animation-duration</a>	X	X	X	✓	0
<a href="#">animation-iteration-count</a>	X	X	X	✓	1
<a href="#">animation-name</a>	X	X	X	✓	none
<a href="#">animation-play-state</a>	X	X	X	✓	running
<a href="#">animation-timing-function</a>	X	X	X	✓	ease
<a href="#">appearance</a>	X	X	X	✓	normal

# CSS TULAJDONSÁGOK (W3C)

- <https://www.w3.org/Style/CSS/all-properties.en.html>

Show editors' drafts

Pub. Property	Specification	St.
✓ --*	CSS Custom Properties for Cascading Variables Module Level 1	CR
✓ align-content	CSS Flexible Box Layout Module Level 1	CR
✓ align-content	CSS Box Alignment Module Level 3	WD
✓ align-items	CSS Flexible Box Layout Module Level 1	CR
✓ align-items	CSS Box Alignment Module Level 3	WD
✓ align-self	CSS Flexible Box Layout Module Level 1	CR
✓ align-self	CSS Box Alignment Module Level 3	WD
✓ alignment-baseline	CSS Inline Layout Module Level 3	WD
✓ all	CSS Cascading and Inheritance Level 3	CR
✓ all	CSS Cascading and Inheritance Level 4	CR
✓ animation	CSS Animations Level 1	WD
✓ animation-delay	CSS Animations Level 1	WD

v	Overview-x	CSS Overview Module Level 3	WD
✓	overflow-y	CSS Overflow Module Level 3	WD
✓	padding	CSS 2.1	REC
✓	padding	CSS Box Model Module Level 3	WD
✓	padding-block	CSS Logical Properties and Values Level 1	WD
✓	padding-block-end	CSS Logical Properties and Values Level 1	WD
✓	padding-block-start	CSS Logical Properties and Values Level 1	WD
✓	padding-bottom	CSS 2.1	REC
✓	padding-bottom	CSS Box Model Module Level 3	WD
✓	padding-inline	CSS Logical Properties and Values Level 1	WD
✓	padding-inline-end	CSS Logical Properties and Values Level 1	WD
✓	padding-inline-start	CSS Logical Properties and Values Level 1	WD
✓	padding-left	CSS 2.1	REC

# VICCES CSS PÉLDÁK

```
.rich-people {          .ninja {  
    top: 1%;           color: black;  
}  
.  
working-class {        visibility: hidden;  
    bottom: 99%;       animation-duration: 0.00001s;  
}  
  
.ghost {              .autobots {  
    color: white;      transform: translate3d();  
    opacity: 0.1;       }  
}
```

# VICCES CSS PÉLDÁK

```
#chucknorris {  
    color: #BADA55;  
}  
  
.hobbit {  
    height: 50%;  
}  
.hobbit #foot {  
    width: 200%;  
}  
  
#eminem {  
    word-spacing: 0;  
}  
  
#lego {  
    display: block;  

```

<http://www.hongkiat.com/blog/funny-css-puns/>

# VICCES CSS PÉLDÁK

```
#tower-of-pisa {  
    font-style: italic;  
}
```

```
#usa + #mexico {  
    border: 1px dashed;  
}
```

```
#berlin-wall-1989 {  
    border-collapse: collapse;  
    height: 0;  
}
```

```
#titanic {  
    float: none;  
    bottom: 0;  
}
```

Írjatok Ti is hasonlókat! <http://padlet.com/abonyita/8wglh5stnteo>

# CSS ELŐFELDOLGOZÓK (PREPROCESSORS)

<http://www.sitepoint.com/6-current-options-css-preprocessors/>

# CSS ELŐFELDOLGOZÓK

Segítségükkel jobban karbantartható kódot készíthetünk

Leggyakrabban használt előfeldolgozók

1. SaSS
2. Less.js
3. Stylus
4. CSS-Crush
5. Myth
6. Rework

# SASS

- <http://sass-lang.com/>

```
$serif-font-stack: "Georgia", "Times New Roman", serif  
$monospace-font-stack: "Cousin", "Courier"
```

```
body
```

```
    font: normal 18px/22px $serif-font-stack
```

```
pre, code
```

```
    font: 600 bold 18px/22px $monospace-font-stack
```

# LESS

- <http://lesscss.org/>

```
<head>
  <!-- Reference Less JavaScript file -->
  <script src="path/to/your/less.js"></script>
  <link rel="stylesheet/less" type="text/css" href="your/styles.less" />
</head>
```

```
.button {
  display: inline-block;
  width: 80%;
  max-width: 200px;
  border-radius: 5px;
  background-color: black;
  color: white;
  font-size: 14px;
  margin: 5px;
  padding: 8px;
}
```

```
.button-checkout-process {
  .button(); /* Mixin */
  background-color: silver;
}
```

A button szabályt tartalmát itt  
is felhasználjuk.

# STYLUS

- <http://stylus-lang.com/>
- Több, mint 60 egyedi funkció

```
.button  
  display inline-block  
  border-color saturation(#000, 50%)  
  border-radius 5px
```



```
.button {  
  display: inline-block;  
  border-color: #959595;  
  -webkit-border-radius: 5px;  
  -moz-border-radius: 5px;  
  border-radius: 5px;  
}
```

# MODULÁRIS, ÚJRAFELHASZNÁLHATÓ CSS

<http://www.creativebloq.com/css3/create-modular-and-scalable-css-9134351>

<http://blog.rlnd.hu/ujrafelhasznalhato-css/>

<http://adamlaki.com/bem-modszertan/>

# ÚJRAFELHASZNÁLHATÓ CSS

A statikus CSS újragondolásával, különböző konvenciók bevezetésével elérhetjük, hogy a kódunk újrafelhasználható, átlátható, karbantartható legyen.

Ilyen konvenciók:

- **Object Oriented CSS (OOCSS)**
- **Scalable and Modular Architecture for CSS (SMACSS)**
- **Don't Repeat Yourself CSS (DRY CSS)**
- **Block, Element, Modifier (BEM)**

# DON'T REPEAT YOURSELF CSS (DRY CSS)

- Csoportosítsuk az újrafelhasználható CSS deklarációkat
- Nevezzük el ezeket a csoportokat logikusan
- Adjunk szelektorokat a különböző csoportokhoz

## DRY CSS REVIEW

Group re-usable properties

```
.{  
background-color: #fff;  
border-color: #ccc;  
}  
.{  
background-color: #fff;  
border-color: #bbb;  
}
```

Name the groups logically

```
#LIGHT-WHITE-BACKGROUND,  
.light-white-background {  
background-color: #fff;  
border-color: #ccc;  
}  
#MEDIUM-WHITE-BACKGROUND,  
.medium-white-background {  
background-color: #fff;  
border-color: #bbb;  
}  
  
#LIGHT-RED-TEXT,  
#MEDIUM-RED-TEXT,  
#PALE-RED-BACKGROUND,  
#LIGHT-RED-BACKGROUND,  
#GREY-BUTTON,  
#GREY-BUTTON-HOVER,  
#GREY-BUTTON-CLICK,
```

Add selectors to various groups

```
#LIGHT-WHITE-BACKGROUND,  
.translation,  
.entry .wp-caption,  
#full-article .entry img,  
.recent-comment .comment-text,  
.roundup h3,  
.post-header-sharing,  
#post-categories td.label,  

```

# OBJECT ORIENTED CSS (OOCSS)

- A struktúrát válasszuk el a designtól
  - A prezentációra vonatkozó deklarációkat válasszuk külön a struktúrára, pozícionálásra szolgáló deklarációktól!
- A konténert válasszuk el tartalomtól
  - A komponensek megjelenésének tartalmazó elemtől való függését meg kell szüntetni. Bármelyik objektumot tudni kell áttenni másik tartalmazó elembe úgy, hogy ne változzon meg a megjelenése!

# SCALABLE AND MODULAR ARCHITECTURE FOR CSS (SMACSS)

- **Base:** az alapdefiníciók, a css reset
- **Layout:** az oldalon fellelhető sorok és oszlopok beállításai. Grid css.
- **Module:** az oldalunk kis részegységei. Cikk lista, cikk részletes oldal, tartalmi dobozok.
- **State:** Az állapotok pl. aktív, inaktív, kinyitott, becsukott.
- **Theme:** az oldal azon stílusjegyei, amitől igazán egyedi lesz az oldalunk.

# BLOCK, ELEMENT, MODIFIER (BEM)

**Block:** A *blokk* a befoglaló elemünket, az oldal egy-egy nagyobb építőegységét jelenti, mint a fejléc, tartalom, oldalsáv, vagy a lábléc! Ez a rész lesz a kijelölőnk alapja!

**Element:** Az *elem* a *block* egy alkotó része. Az elemet a blokk után írjuk két darab alul vonással összekötve.

**Modifiers:** A módosítók segítségével a meglévő blokkok, elemek megjelenésén módosíthatunk. Például, ha van egy üzenet blokkunk, aminek több állapota van, egy semleges, egy figyelmeztetés/hiba visszajelzés, valamint egy sikeres visszaigazolás. Ebben az esetben ez 3 db külön osztály kijelölőt jelent úgy, hogy az utóbbi két esetben módosítanunk kell az eredeti elem megjelenésén. A módosítókat két kötőjellel jelöljük!

# BLOCK, ELEMENT, MODIFIER (BEM)

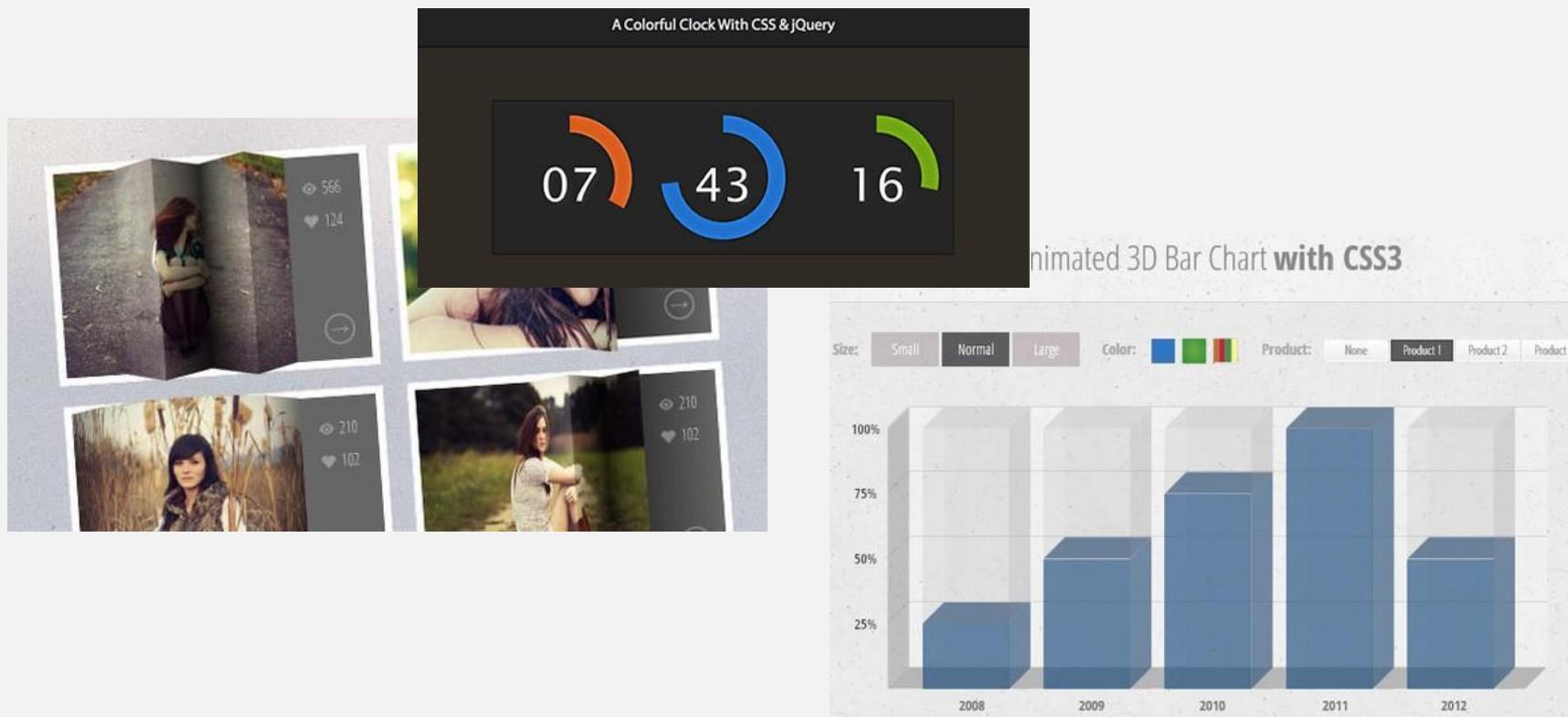


```
.message {  
  background: #eee;  
  border-radius: 5px;  
  color: #333;  
  font-size: 16px;  
  padding: 10px;  
}  
  
.message--alert {  
  @extend .message;  
  background: #da4531;  
  color: #fff;  
}  
  
.message--success {  
  @extend .message;  
  background: #49DA31;  
  color: #fff;  
}
```

# MENŐ CSS PÉLDÁK

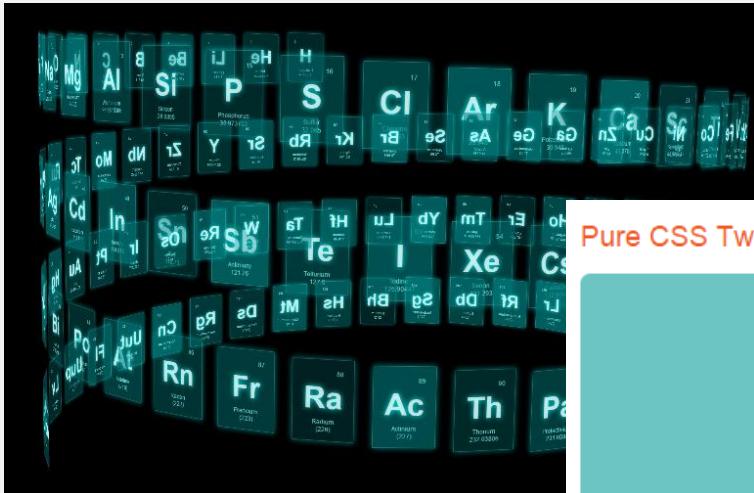
# 150 COOL CSS EXAMPLES AND TUTORIALS

- <http://1stwebdesigner.com/css-effects/>

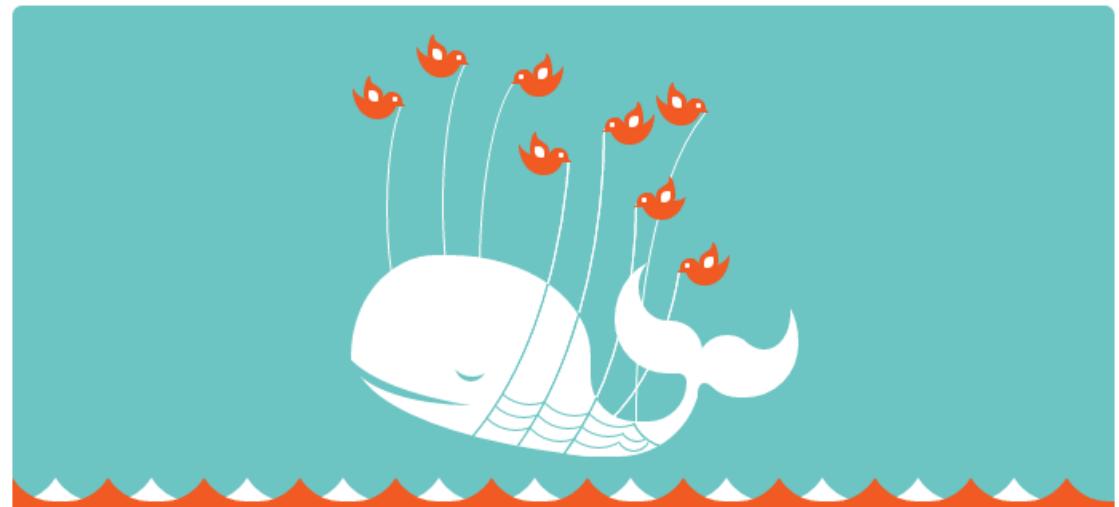


# 22 STUNNING EXAMPLES OF CSS3 ANIMATION

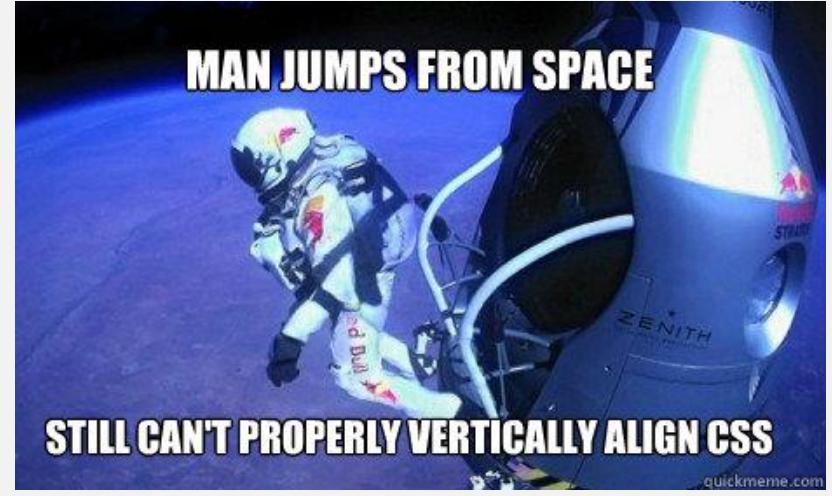
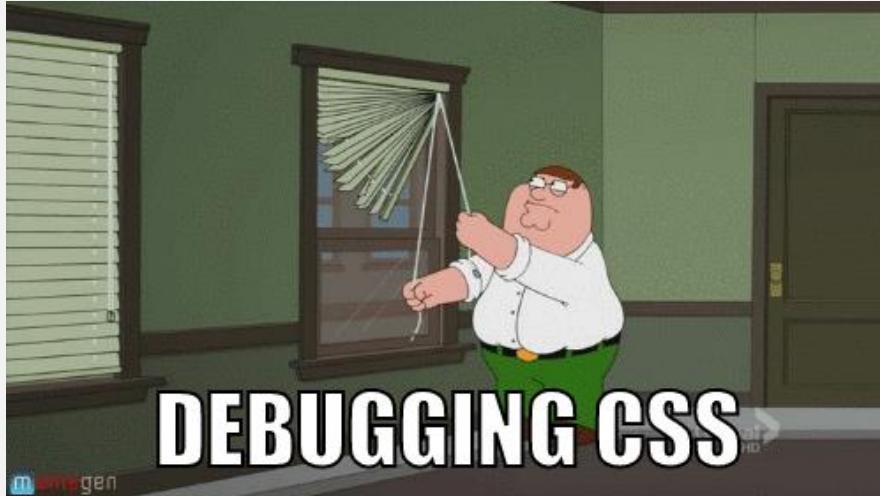
- <http://www.creativebloq.com/css3/animation-with-css3-712437>



Pure CSS Twitter 'Fail Whale'



# VÉGE



Amikor egy programozó belenyúl  
az oldal css-ébe

<http://bit.ly/2dm0Fu4>

