



```
<a href="home.html">Home</a></li>  
<li><a href="home-events.html">Home Events</a></li>  
<li><a href="multi-col-menu.html">Multiple Column Menu on Larger Viewports</a></li>  
<li class="has-children"><a href="#" class="current">Header Options</a><ul>  
  <li><a href="tall-button-header.html">Tall Button Header</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 Slider</a></li>  
</ul></li>
```

BEVEZETÉS A STÍLUSLAPOK HASZNÁLATÁBA

Abonyi-Tóth Andor,
ELTE Informatikai Kar
2018/2019/II.

NÉHÁNY ALAPFOGALOM

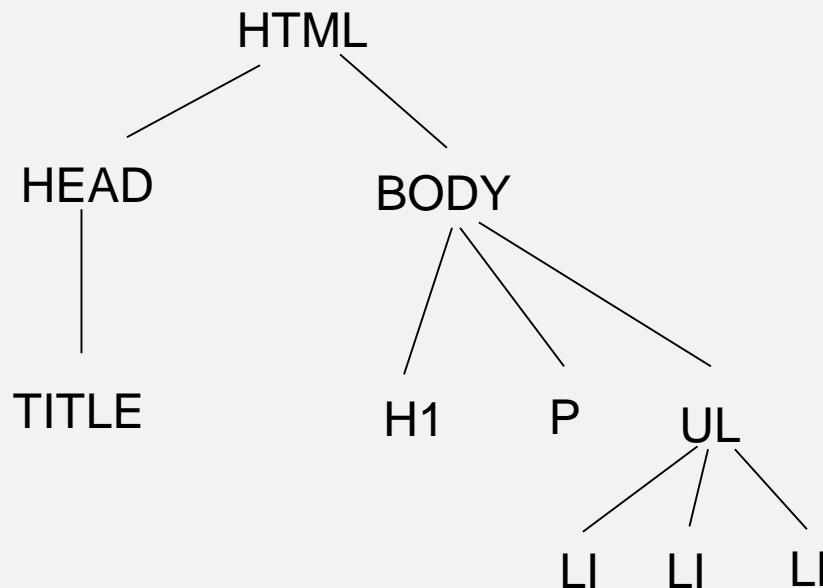
HTML DOM (DOCUMENT OBJECT MODEL)

- 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)

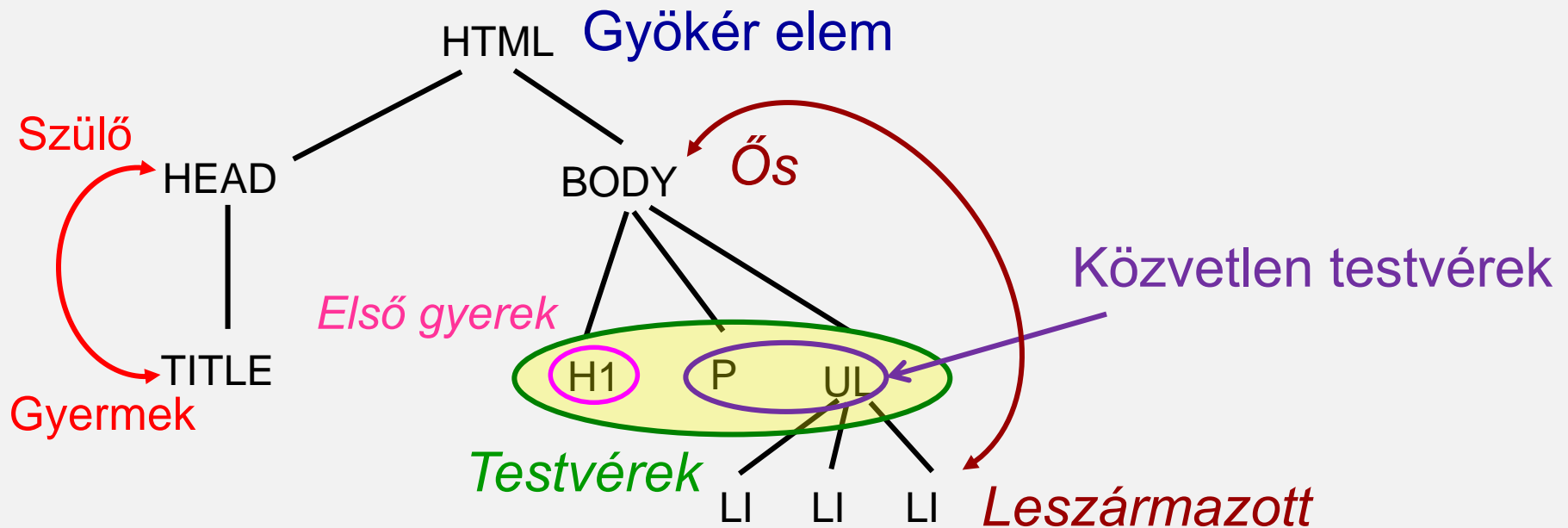
- A HTML dokumentum egy speciális elemekből álló fastruktúra, speciális többletfunkcionalitását a HTML DOM szabvány határozza meg.

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

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

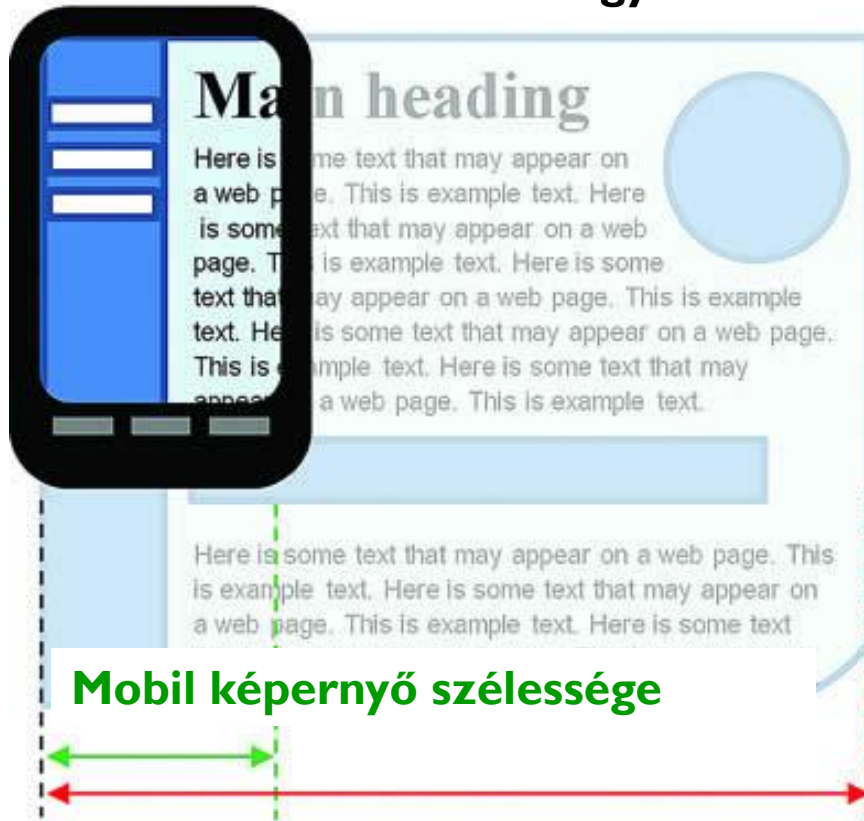
A VIEWPORT FOGALMA

- 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



Viewport szélessége (gyakran 1000px körül)

<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?

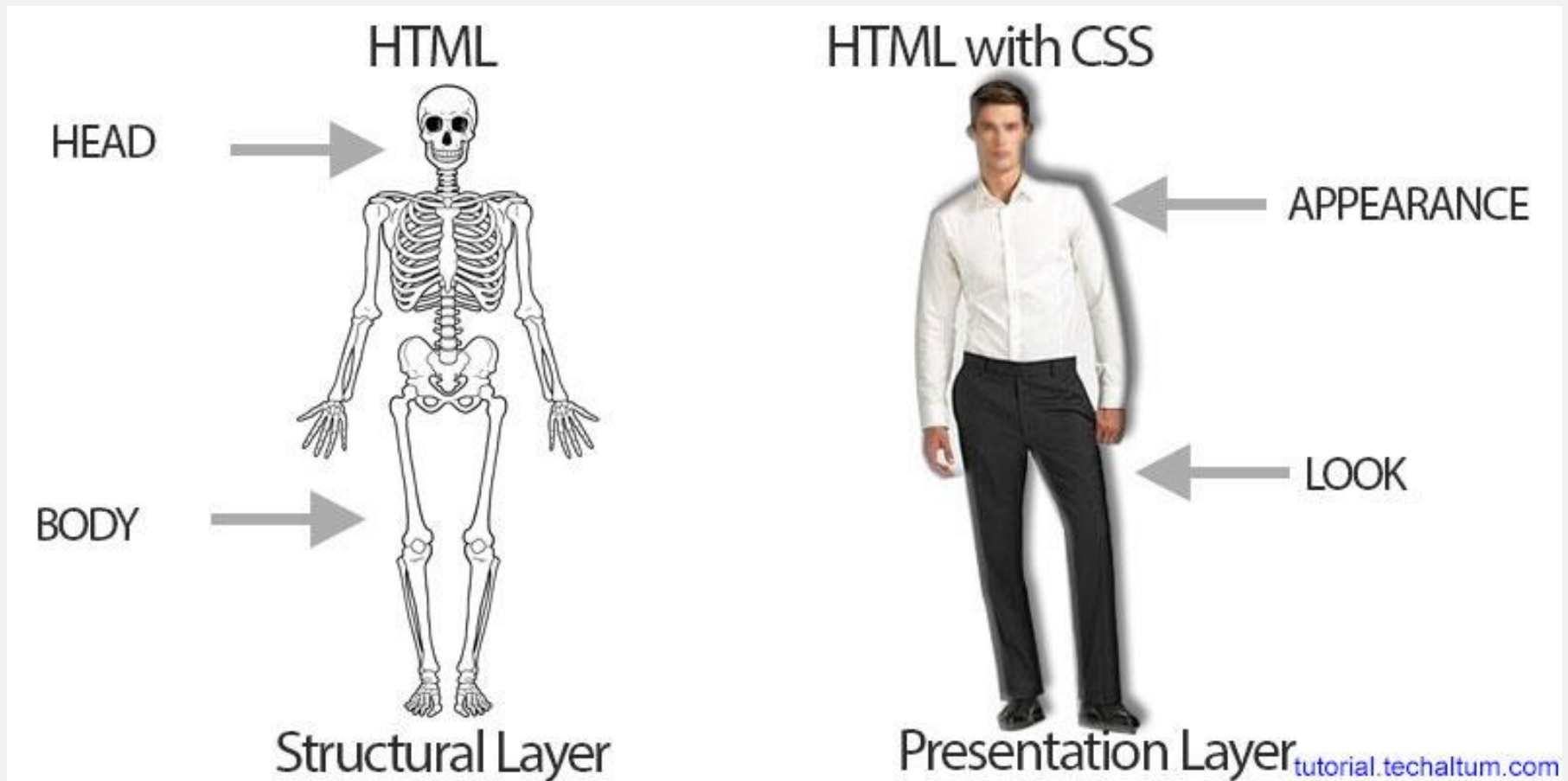
CSS



Cascading Style Sheets

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

TARTALOM ↔ MEGJELENÉS



CSS PÉLDA

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>CSS példa</title>
<style type="text/css">
    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>
```

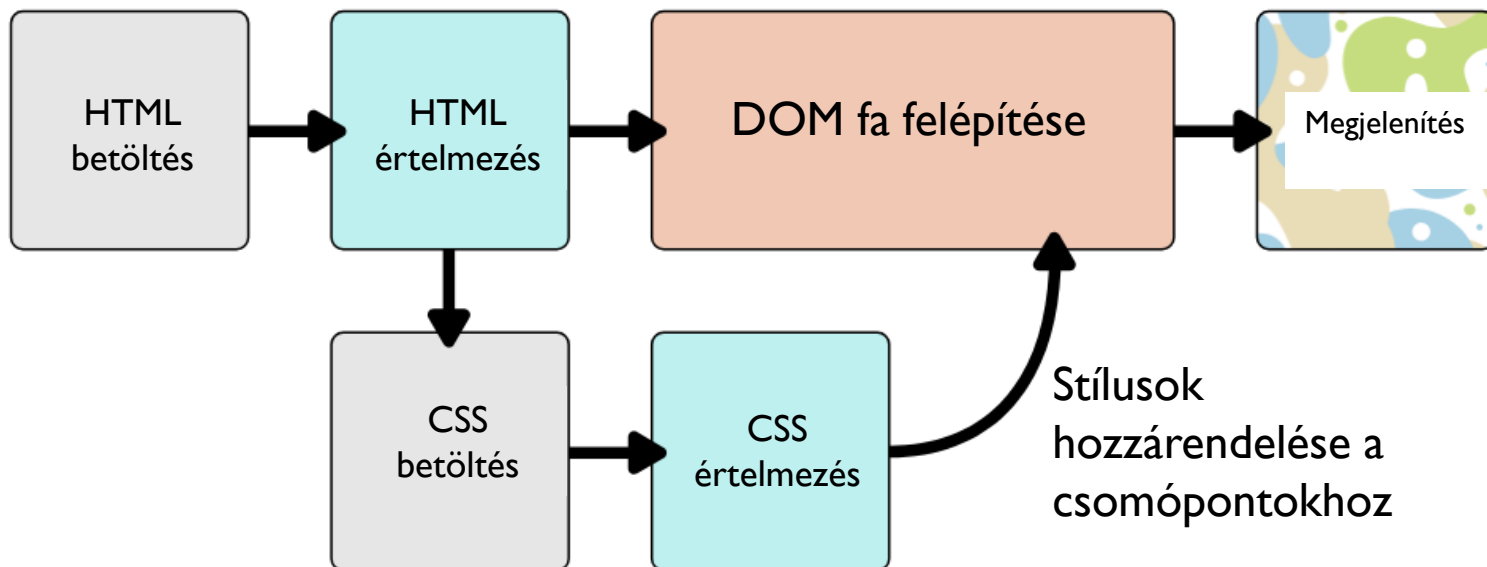
Ez a szabály azt írja le, hogy az 1-es címsorok (h1) kék színnel jelenjenek meg, és a háttérszínük világos szürke legyen.

Címsor 1

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

HOGYAN MŰKÖDIK?

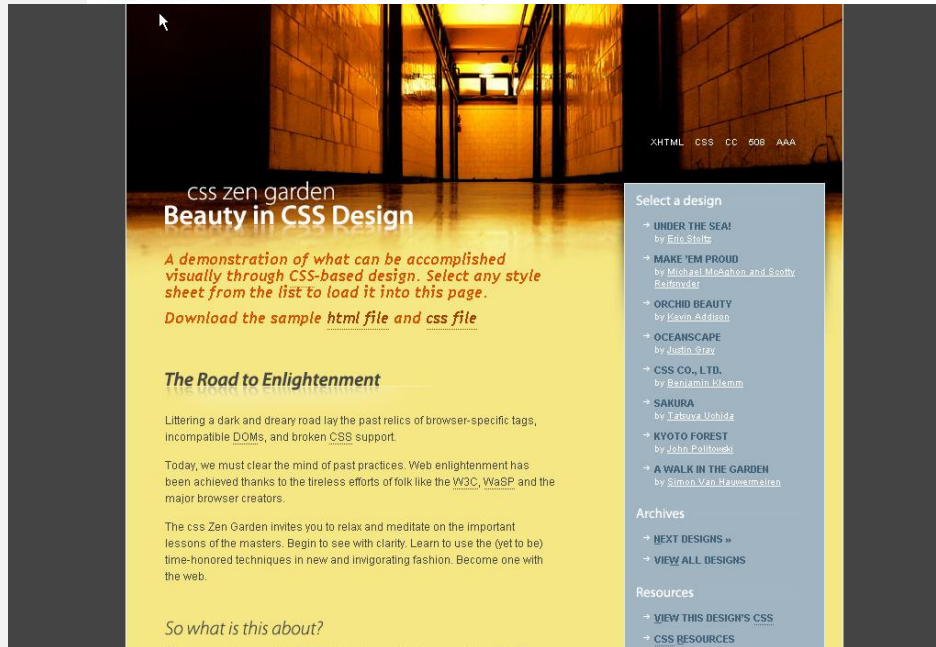
- Amikor egy böngésző megjelenít egy HTML oldalt, akkor legelőször betölti az oldalt a web-szerverről, majd értelmezi a jelöléseit (pl. HTML) és létrehozza a modellt a memóriában tárolt DOM formájában.
- A CSS betöltése után értelmezi azt, és az egyes stílusokat a DOM csomópontjaihoz rendeli, majd megjeleníti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



css zen garden
Beauty in CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.
[Download the sample html file and css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#) and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So what is this about?

Select a design

- [UNDER THE SEAL](#) by Eric Stoltz
- [MAKE 'EM PROUD](#) by Michael McAdon and Scotty Reifsnider
- [ORCHID BEAUTY](#) by Kevin Addison
- [OCEANSCAPE](#) by Justin Gray
- [CSS CO., LTD.](#) by Benjamin Klemm
- [SAKURA](#) by Tatsuya Uchida
- [KYOTO FOREST](#) by John Poltowski
- [A WALK IN THE GARDEN](#) by Simon Van Hauwermeiren

Archives

- [NEXT DESIGNS >](#)
- [VIEW ALL DESIGNS](#)

Resources

- [VIEW THIS DESIGN'S CSS](#)
- [CSS RESOURCES](#)



El Collar de Tomas
MARIA STULTZ, Colombia



OrderedZen
STEVE SMITH, United States



Bonsai
MARTIN PLAZOTTA, Austria



CSSZENGARDEN
the beauty of css design

SELECT A DESIGN

- [Under the Seal](#) by [Eric Stoltz](#)
- [Make 'em Proud](#) by [Michael McAdon](#) and [Scotty Reifsnider](#)
- [Orchid Beauty](#) by [Kevin Addison](#)
- [Oceanscape](#) by [Justin Gray](#)
- [CSS Co., Ltd.](#) by [Benjamin Klemm](#)
- [Sakura](#) by [Tatsuya Uchida](#)
- [Kyoto Forest](#) by [John Poltowski](#)
- [A Walk in the Garden](#) by [Simon Van Hauwermeiren](#)

ARCHIVES

- [next designs >](#)
- [View All Designs](#)

A demonstration of what can be accomplished visually through [CSS](#)-based design. Select any style sheet from the list to load it into this page.
Download the sample [html file](#) and [css file](#).

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken [CSS](#) support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the [W3C](#), [WaSP](#) and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is clearly a need for [CSS](#) to be taken seriously by graphic artists. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The code remains the same, the only thing that has changed is the external [css](#) file. Yes, really.

[CSS](#) allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. To date, most examples of neat tricks and hacks have been demonstrated by structurists and coders. Designers have yet to make their mark. This needs to change.

Participation

Graphic artists only please. You are modifying this page, so strong [CSS](#) skills are necessary, but the example files are commented well enough that even [CSS](#) novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with [CSS](#).

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

MI AZ A „CASCADING”?



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

Håkon Wium Lie (CSS fejlesztő)

MI AZ A
„CASCADING”?



A cascade szó lépcsőzetes vízesést 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.

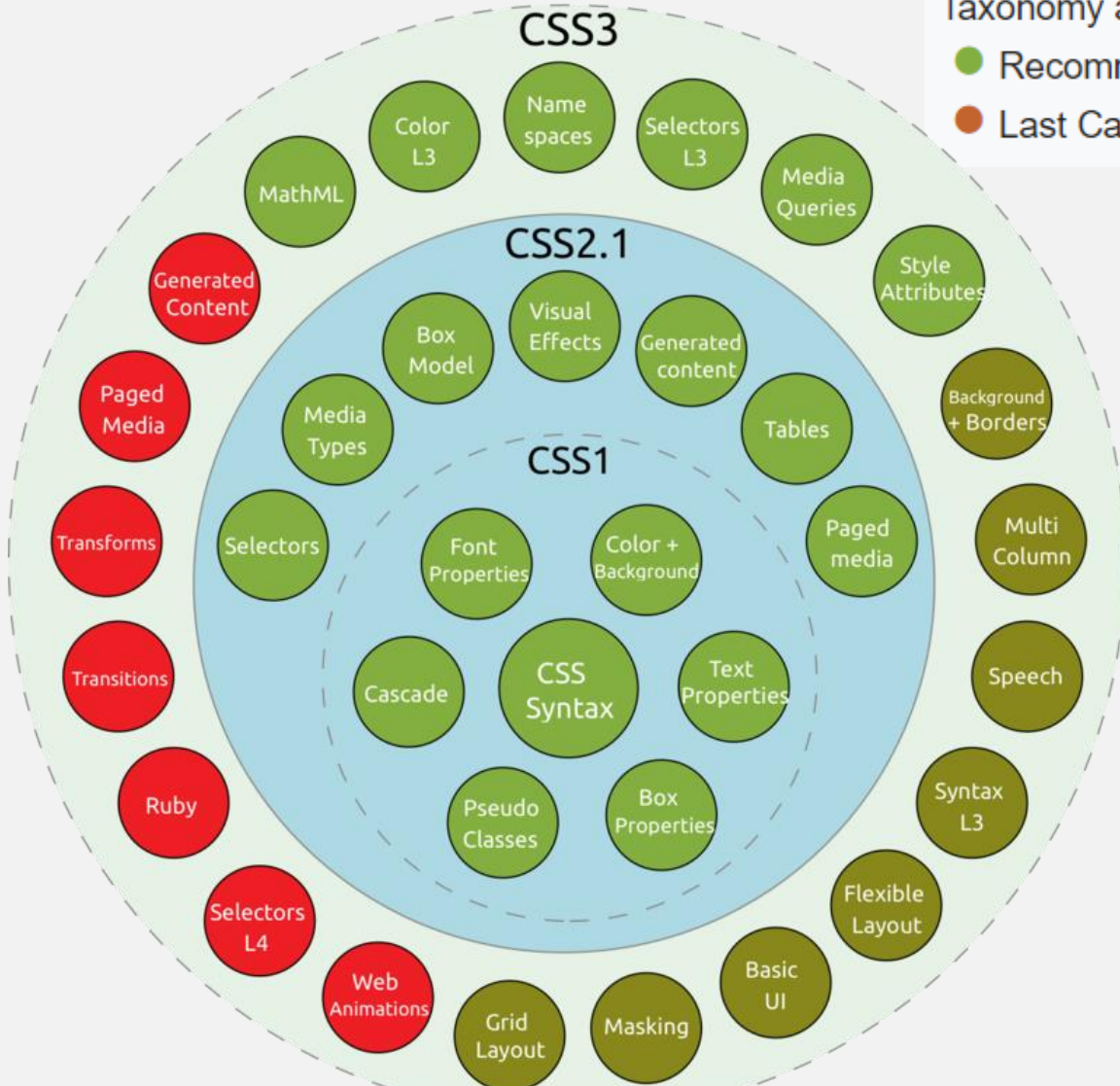
CSS VERZIÓK

- 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

Taxonomy and status of CSS3 modules.

- Recommendation
- Candidate Recommendation
- Last Call
- Working Draft.



CSS MODULOK

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

TABLE OF SPECIFICATIONS

Ordered from most to least stable:

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

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



The image shows the standard login interface of the NEPTUN system. At the top, it says 'Eötvös Loránd Tudományegyetem' and features the 'NEPTUN' logo with the tagline 'Egységes Tanulmányi Rendszer'. Below this, the user is identified as 'Hallgatói ELTE_HW2(341)'. There are three language selection buttons (Hungarian, English, German) and two input fields for 'Azonosító' (ID) and 'Jelszó' (Password). A blue 'Bejelentkezés' (Login) button with a right arrow is at the bottom. The background includes a photo of a smiling woman holding a green folder and a 'Microsoft In' logo at the bottom right.

FELHASZNÁLÓI SZINTEN FELÜLÍRT MEGJELENÉS



The image shows the login interface with a user level override. The background is solid black. The text is white. At the top, it says 'Hallgatói ELTE_HW2(352)'. Below this, there are three language selection buttons (Hungarian, English, German) and two input fields for 'Azonosító' (ID) and 'Jelszó' (Password). To the right of the language buttons, the text 'Build: 455 (2018.07.19.) P20180912' is displayed. Below the input fields, the text 'Támogatott böngészők: Microsoft Internet Explorer 10.0+ ;' is shown. The 'Bejelentkezés' button is not visible in this version.

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

CSS SZINTAXIS

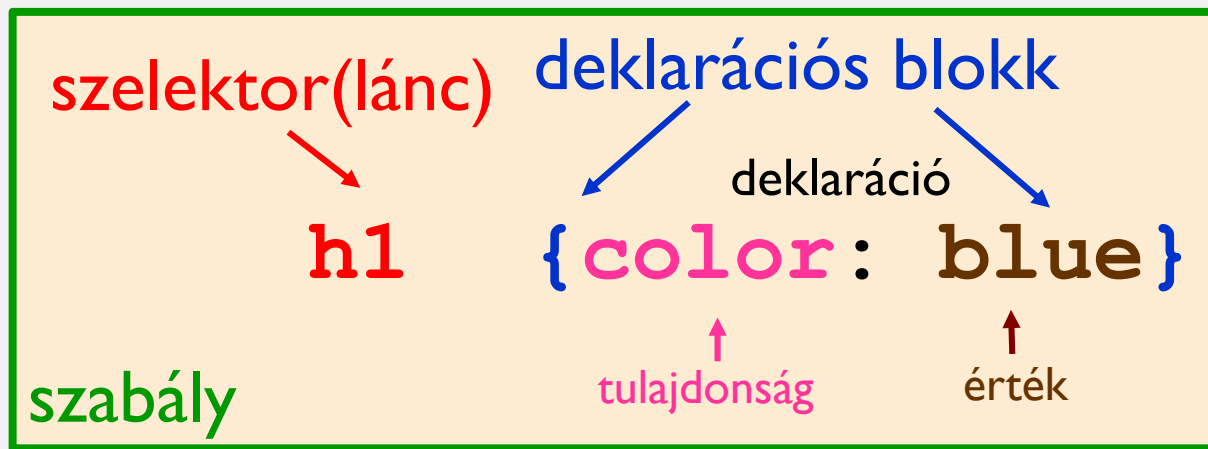
CSS SZABÁLY (RULE)

A CSS szabály két részből áll

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

Több elemet is fel lehet sorolni, ekkor már szelektorláncról 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 type="text/css">
      @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 type="text/css">
  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

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

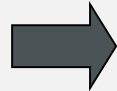
```
h1 {color:blue}
```

```
<!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>
```

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.

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

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 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", "File feltöltéssel", and "Közvetlen bevitellel". The "URI segítségével" tab is selected. Below the tabs, there is a section titled "Érvényesítés URI segítségével" with the instruction "Adja meg a dokumentum (HTML CSS-sel vagy csak CSS) URI-jét, melyet érvényesíteni szeretne:". There is a text input field labeled "Cím:". Below the input field, there is a link "További lehetőségek". At the bottom, there is a button labeled "Ellenőrzés".

W3C[®] CSS Validation Service
Cascading Style Sheets (CSS) és stíluslapokkal rendelkező (X)HTML dokumentumok ellenőrzése

URI segítségével File feltöltéssel Közvetlen bevitellel

Érvényesítés URI segítségével

Adja meg a dokumentum (HTML CSS-sel vagy csak CSS) URI-jét, melyet érvényesíteni szeretne:

Cím:

► **További lehetőségek**

Ellenőrzés

MÉRTÉKEGYSÉGEK

RELATÍV MÉRTÉKEGYSÉGEK

Egy másik tulajdonság értékétől függenek, 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

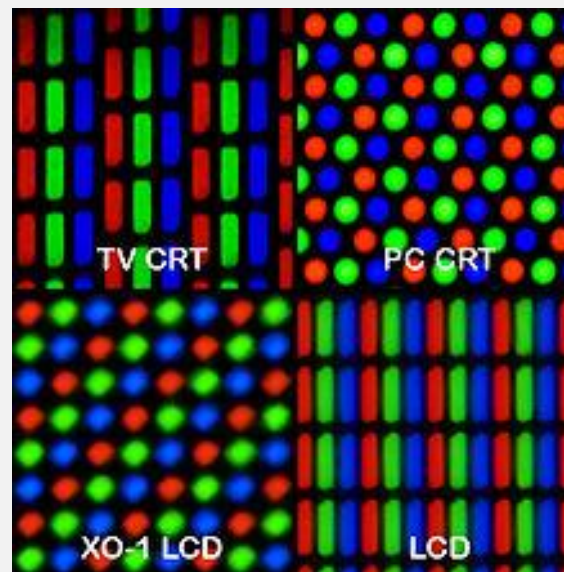
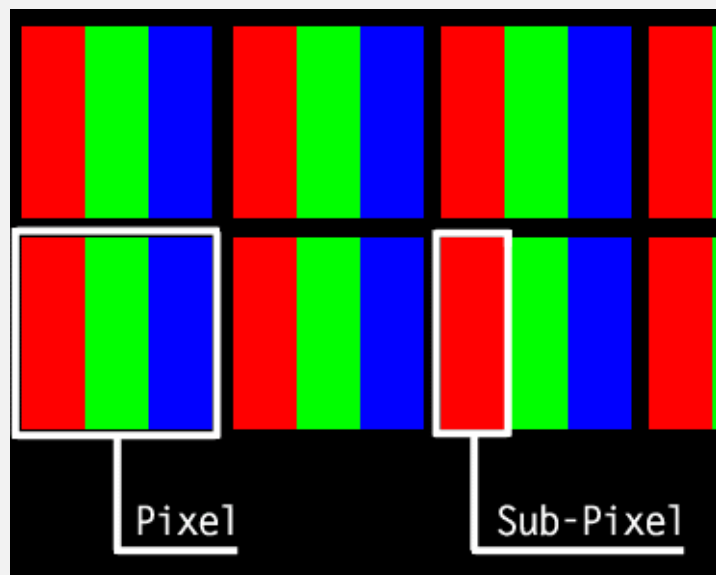
MÉRTÉKEGYSÉGEK (CSS3)

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

Abszolút	Relatív
<ul style="list-style-type: none">• in (=2.54 cm)• cm• mm• pt (=1/72in)• pc (=12pt)• px (képpont)	<ul style="list-style-type: none">• em (adott környezet betűméretéhez képest történő megadás. pl. 1.2em)• % (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%)• ex (x-magasság - többnyire az adott font x betűjének magasságához képest történő megadás. pl. 2ex)• rem (a gyökér elem betűméretéhez képest történő relatív méretmegadás pl. 1.5rem)• ch (Az adott betűcsalád 0 karakterének szélességéhez történő relatív megadás. pl. 2ch)• vw (A viewport szélességéhez igazodik. 1vw a viewport szélesség 1%-ának felel meg.• vh (A viewport magasságához igazodik. 1vh a viewport magasság 1%-ának felel meg.

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.



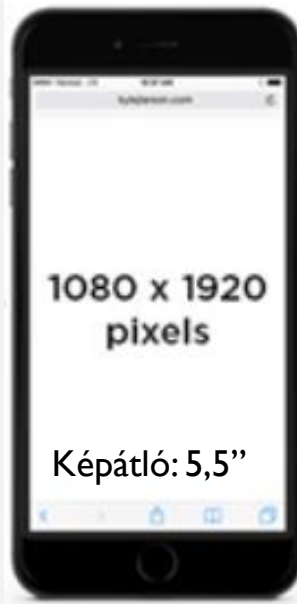
Alpixelek különböző elrendezése

NE A FIZIKAI FELBONTÁSBÓL INDULJUNK KI!

IPHONE 6 PLUS



Physical Pixels
the number of actual
pixels in the display



1080 x 1920
pixels

Képtárló: 5,5"

Pixelsűrűség (pixel per inch)
400ppi

SAMSUNG SYNCMASTER S24D330H



Képtárló

24"

Felbontás

1920x1080

92ppi

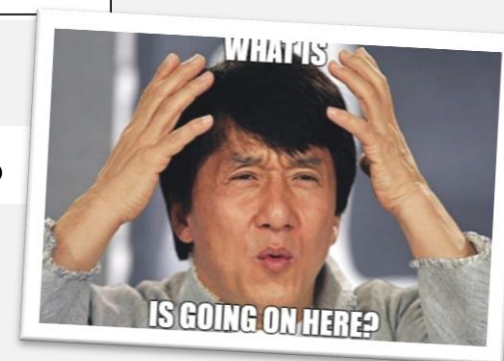
IPHONE FELBONTÁSOK ÉS VIEWPORT MÉRETEK

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

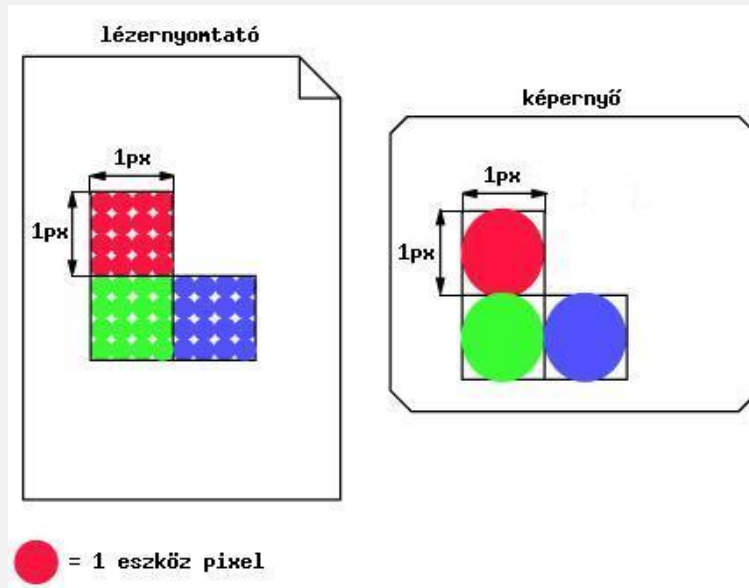
320×480	320×480	320×568	375×667	414×736
---------	---------	---------	---------	---------



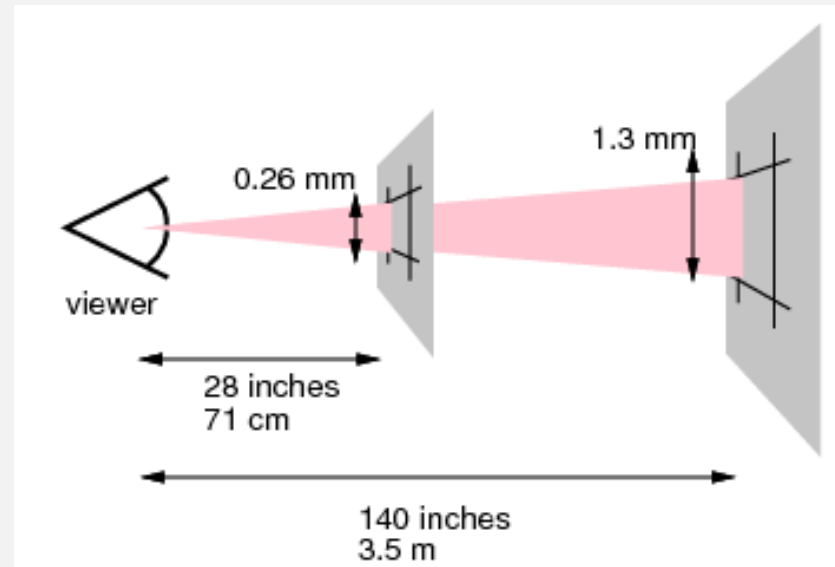
REFERENCIA PIXEL

- A CSS szabvány **nem a hardveres** pixellel számol, hanem úgynevezett **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 képpontsűrűségű eszköz egy képpontjának szokásos olvasási távolságához (kb. 71 cm) tartozó látószöggel definiálják.

REFERENCIA PIXEL



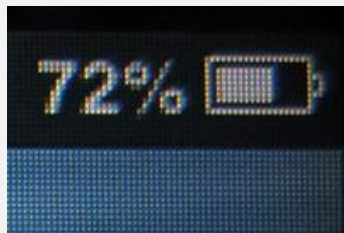
Egy lézernyomtató a képernyő pixelt több pontból fogja nyomtatni, attól függően, hogy mekkora a pontsűrűsége.



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

RETINA MEGJELENÍTŐ

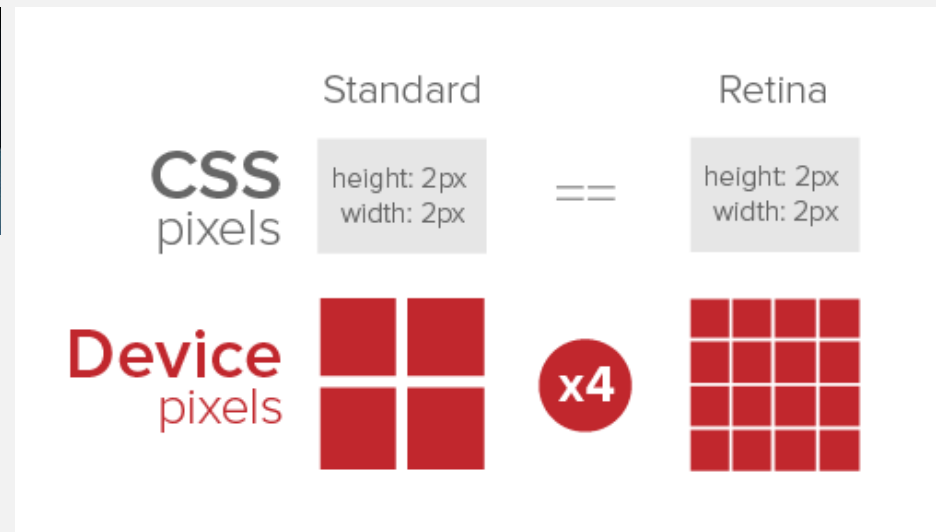
A retina megjelenítő képpontsűrűsége jóval nagyobb, mint elődjeié. Itt már a pixeleket (tipikus olvasási távolságból) az emberi szem nem képes megkülönböztetni.



Nem retina
megjelenítő

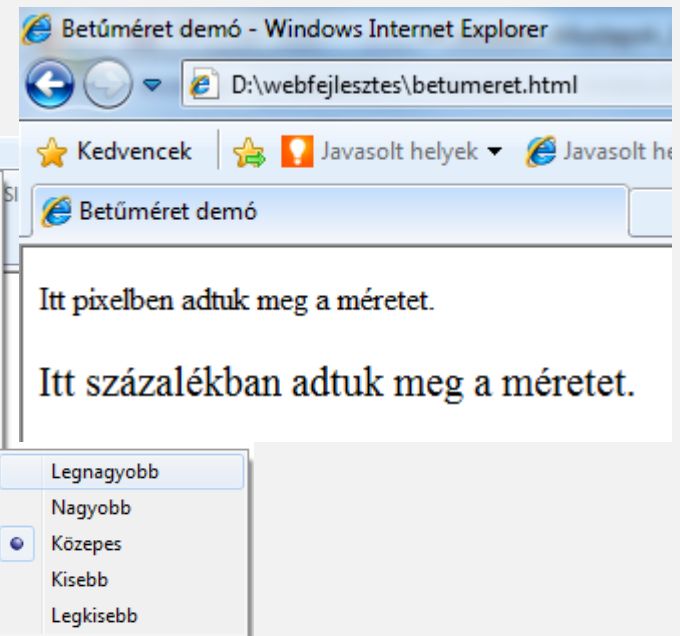
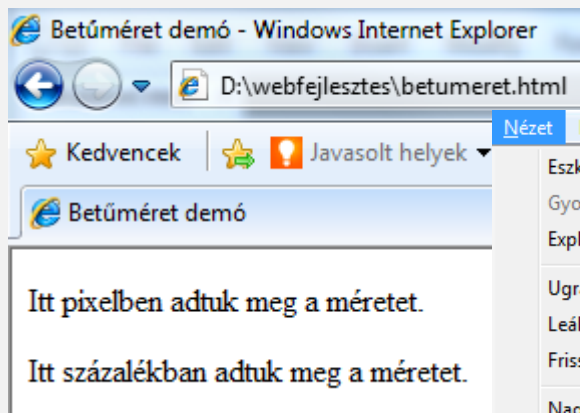


Retina
megjelenítő




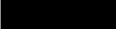






























BETŰMÉRET MEGADÁS PIXELBEN





























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



SZÍNJELÖLÉSEK, SZÍN NEVEK

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

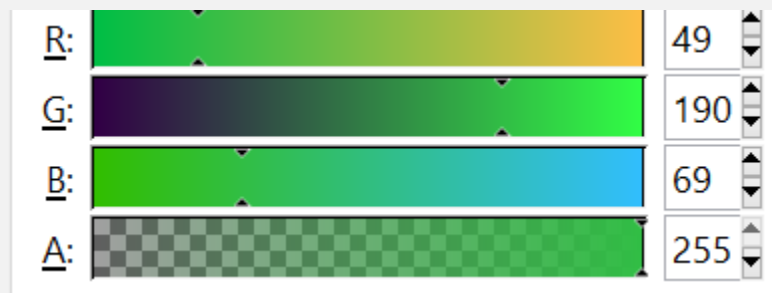
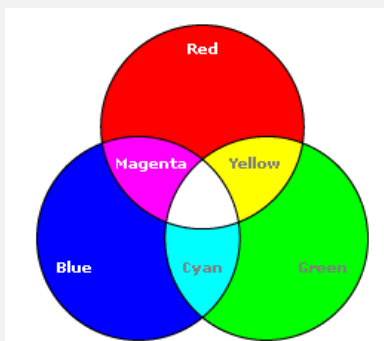
Named	Numeric	Color name	Hex rgb	Decimal
		black	#000000	0,0,0
		silver	#C0C0C0	192,192,192
		gray	#808080	128,128,128
		white	#FFFFFF	255,255,255
		maroon	#800000	128,0,0
		red	#FF0000	255,0,0
		purple	#800080	128,0,128
		fuchsia	#FF00FF	255,0,255
		green	#008000	0,128,0
		lime	#00FF00	0,255,0
		olive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

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

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


RGB SZÍN-KOORDINÁTARENDSZER

- A műszaki életben leggyakrabban használt színekoordináta-rendszer, 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/>



```
.tshirt{  
  color:#000;  
  background-color:#FFF;  
  height:100%;  
  width:auto;  
  overflow:hidden;  
}
```

ÁTLÁTSZATLANSÁG (ALFA CSATORNA)

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

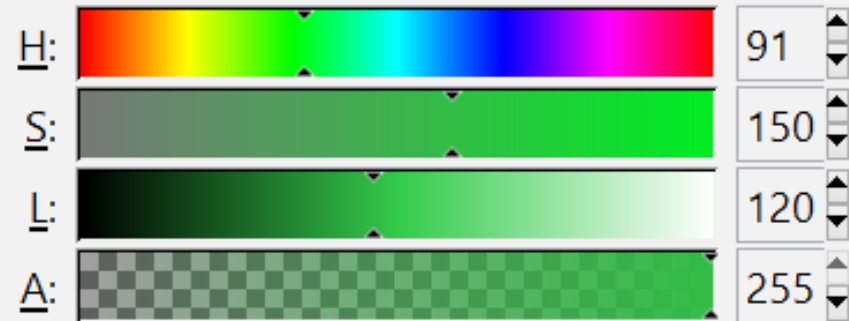


HEADING 1

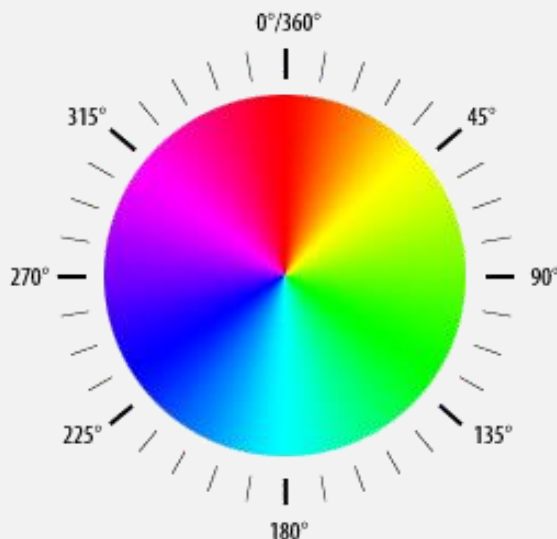
```
h1 {  
    color: rgb(0, 0, 0);  
    background-color: rgba(255, 255, 255, 0.5);  
}
```

HSL SZÍN-KOORDINÁTARENDSZER

- 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.



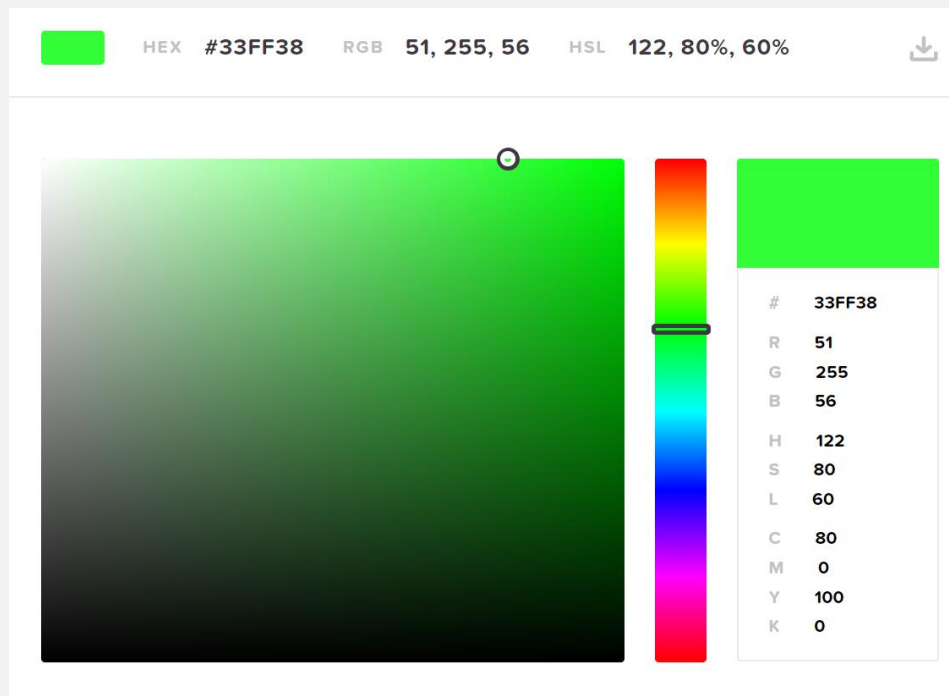
A színkör ellentétes oldalain a kiegészítő (komplementer) színek találhatóak.

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%)">`

SZÍNKÓDOK MEGHATÁROZÁSA

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



FALLBACK

- Érdeemes 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);  
}
```

URL MEGADÁS

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

Példa

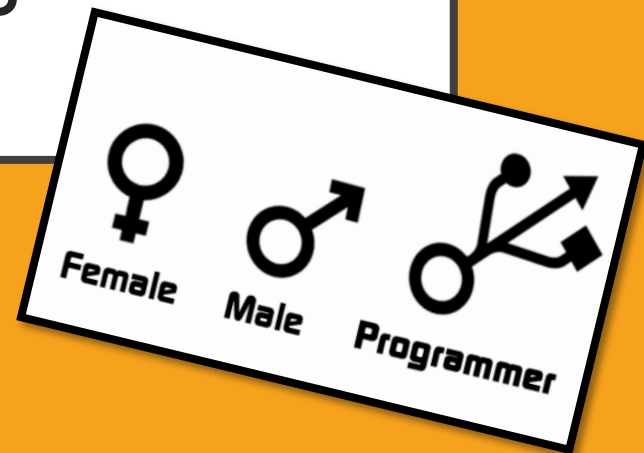
```
body {  
  background-image: url('kepek/hatter.jpg');  
}  
  
li {  
  list-style: url('http://v.hu/kep.png') disc  
}
```

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 szülőikben 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 öröklí 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.

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

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

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 öröklí a szülő elem szegély beállításait.

SZELEKTOROK (KIJELÖLŐK)

P

$P + P$

$P \ B$

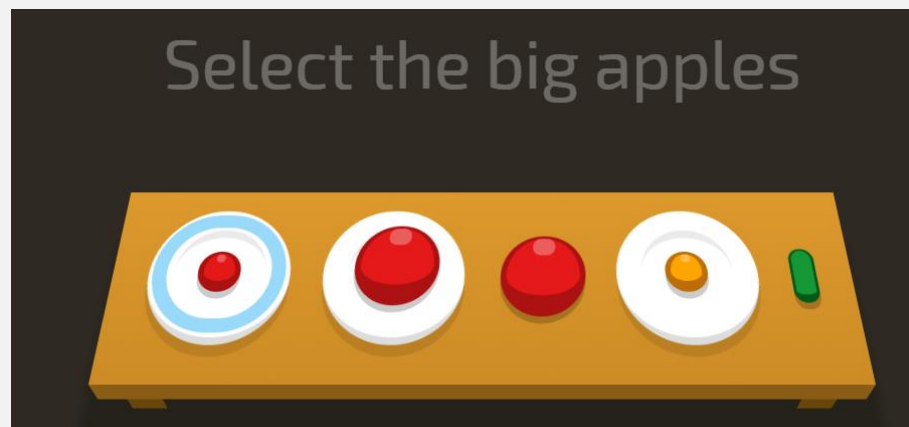
$P > B$

$P \sim B$

P, B

SZELEKTOROK GYAKORLÁSA

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



CSS Editor	style.css	HTML Viewer	table.html
<pre>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</pre>		<pre>1 <div class="table"> 2 <plate id="fancy"> 3 <apple class="small" /> 4 </plate> 5 <plate> 6 <apple /> 7 </plate> 8 <apple /> 9 <plate> 10 <orange class="small" /> 11 </plate> 12 <pickle class="small" /> 13 </div></pre>	

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.

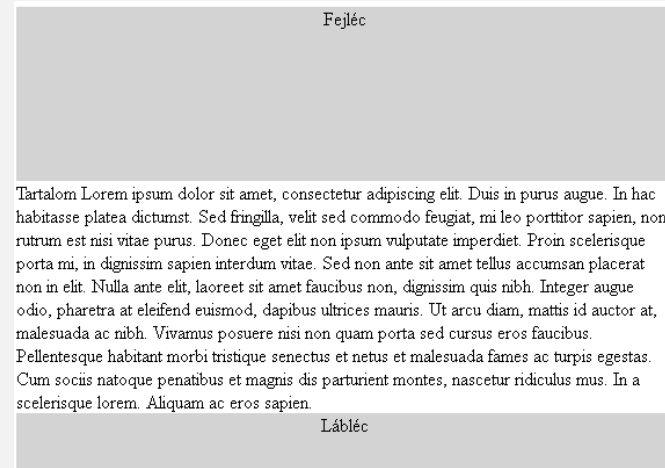
ID ÉS CLASS ALKALMAZÁSA UGYANAZON ELEMEN

```
<style type="text/css">
  #oldalfejec    {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="fejec" class="kozep">Fejléc</header>
<main id="tartalom">Tartalom Lorem ipsum... </main>
<footer id="oldallablec" class="kozep">Lábléc</div>
```