

Honlapkészítés

Stíluslapok (CSS) nyelvtana

Gyakorláshoz, tanuláshoz

- <http://htmliskola.uw.hu/>
- <https://www.w3schools.com/>
 - HTML
 - CSS
 - Kvíz (tudásod teszteléséhez)
- Nagy Gusztáv: Webprogramozás
 - [http://nagygusztav.hu/sites/default/files/csatol/web_programozas - szines.pdf](http://nagygusztav.hu/sites/default/files/csatol/web_programozas_-_szines.pdf)
- <http://www.csszengarden.com/>
 - Minták → HTML és CSS kódjukkal együtt

Mi az a CSS?



index	2007.12.13....	Firefox HTML Document
karnak	2007.11.29....	JPEG-kép
nilus	2007.11.29....	JPEG-kép
stilusok	2007.12.15....	CSS fájl

- A CSS a *Cascading Style Sheets* rövidítése
- A stílusok a HTML megjelenítési elemei és attribútumai helyett használhatók, azoknál jóval több lehetőséget biztosítva
- A stílusok meghatározzák, hogy **hogyan jelenjenek meg vizuálisan** a HTML elemek
- A stíluslapok segítségével könnyen **szét lehet választani az oldal tartalmát annak kinézetétől** (a dizájntól)
- A stílusokat általában **külön állományban tároljuk** (.css kiterjesztéssel)
- Külső stíluslapokkal gyorsítani tudjuk a munkavégzést
- Több stílus is hatással lehet egy elem megjelenésére

Stíluslapok helye

- A stílusinformációkat háromféle módon **rendelhetjük** hozzá a HTML dokumentumunkhoz:
 - Külső stíluslapok
 - Beágyazott stíluslap
 - Soron belüli stílus (inline)
- Célszerű azonban éles környezetben a külső stíluslapokat alkalmazni.

Beágyazott stíluslap

```
<head>
  <style type="text/css">
    body {background-color: red}
    p {margin-left: 20px}
  </style>
  ...
</head>
```

- Kész oldalnál akkor szokás alkalmazni, ha az oldal egyedi (a külső stíluslapot nem lehetne másik oldalhoz felhasználni), vagy nagyon egyszerű stílusról van szó.
- A stílusleírást a **HTML oldal fejlécében** kell elhelyezni, **style** elembe ágyazva.
 - Az így meghatározott stílus az egész dokumentumra érvényes lesz.

Külső stíluslapok

```
<head>
  <link rel="stylesheet" type="text/css" href="stilus.css">
  ...
</head>
```

- Így **több HTML állományra is érvényes lesz**, s központilag szerkeszthetővé válik *(ne kelljen minden egyes HTML állományt megváltoztatnunk, ha a honlapunk ezentúl másképp jelenne meg)*.
- A HTML oldalhoz a **head** elemben elhelyezett **<link>** tag segítségével csatolhatunk külső CSS állományt
- Külső stíluslap fájlt nem HTML oldalanként külön-külön, hanem akár egy egész portálhoz egyetlen (vagy néhány) fájlként szokás készíteni. Így a további oldalak látogatása esetén nincs szükség a CSS fájl újbóli letöltésére.

- Ezzel a szintaxissal akár a CSS állományba is **importálhatunk** további CSS állományokat.

A CSS egy újabb megközelítése az @import használata:

```
<head>
  <style type="text/css">
    @import url("stilus.css");
  </style>
  ...
</head>
```

Soron belüli stílus

- Akkor szokás alkalmazni, ha olyan egyedi stíusról van szó, amelyik sehol máshol nem fordul elő.
 - Bár kevés alkalommal van rá szükség, de akkor jól jön, hogy az egyes HTML elemekhez helyben is tudunk stílust meghatározni.
- Itt csak **magát a megjelenést kellett leírnunk**, nem kellett meghatározni, hogy mire vonatkozik: az így definiált stílus csak az adott elemre, illetve az adott elemen belül lesz érvényes.
- Soron belüli stílus alkalmazásához a kiválasztott elemet kell **style** tulajdonsággal ellátni.

A következő példa csak e bekezdés megjelenítését változtatja meg:

```
<p style="color: red; margin-left: 20px">Ez egy bekezdés</p>
```


CSS nyelvtana - Megjegyzések

- **HTML** megjegyzést vagy kommentet a `<!--` és a `-->` jelek közé írhatunk.

```
p { <!-- kisebb betűk a bekezdésekben -->
  font-size: 0.8em;
}
```

- **CSS**-ben pedig a `/*` és a `*/` jelek közé.
- Egy megjegyzés több sort is átfoghat.

```
/* Ez itt egy megjegyzés */
p {
  text-align: center;
  /* Ez itt egy másik megjegyzés */
  color: black;
  font-family: arial
}
```

CSS nyelvtana

- Egy-egy meghatározás két részből áll, egy **kiválasztó** (kijelölő) és egy **tulajdonság** (meghatározás) részből:
 - A *kiválasztó* rész azt határozza meg, hogy mely HTML elemekre vonatkozzon a definíció, míg
 - a *tulajdonság* rész a megjelenést befolyásolja.
- Amikor elemhez rendelünk stílust, akkor csak a tulajdonság részt kell meghatároznunk.

kiválasztó { tulajdonság }

kiválasztó {tulajdonság: érték}

CSS parancsok csoportosítása

Eredeti

```
H1 {color:blue}
H2 {color:blue}
H3 {color:blue}
```

Csoportosított

```
H1, H2, H3 {color:blue}
```

```
P {color:red}
P {font-size: 12pt}
```

```
P {
    color: red;
    font-size: 12pt;
}
```

```
H2 {font-size: 12pt}
H2 {font-weight: bold}
H2 {font-family: verdana}
```

```
H2 {font: 12pt bold verdana}
```

- **Azonosító kijelölő (ID selector):** csak a HTML dokumentumban egyedi *id* jellemzővel ellátott elemre vonatkozik.
 - Az egyes tárolóelemek az **id** jellemzőjük (identifier=azonosító) tulajdonságával (célszerűen a szakaszra jellemző névvel) **azonosíthatók**.
 - Az **id-k értékeinek egyedieknek** kell lenniük, azaz mindegyik csak egyszer szerepelhet egy weblapon, különben nem lesz egyértelmű az azonosításuk.

Ez két lépésben történik:

- A **HTML-ben** **id** címkéket kell megadni: `<p id=„menu”>.....</p>`
- **CSS-ben** a kijelölés a **#** kettőskereszttel és a jellemzővel történik, a kettő között nincs szóköz:
#menu {meghatározás} pl: `#menu {color: green}`

az azonosítóval ellátott bekezdés:

`<p id="kicsikek">szöveg</p>`

Példa azonosítókielölő használatára:

```
<style type="text/css">  
#kicsikek {  
    font-size: xx-small;  
    color: #0000FF;  
}  
</style>
```

- **Osztálykijelölő (class):**

- Osztályokba sorolt elemek azonos formázásra szolgálnak.
- Szintén *két lépésben* történik: HTML és CSS

- Nem csak azonos, hanem különböző címkékhez is rendelhetők közös osztályok, ekkor a CSS kódolás a címkéket nem tartalmazza:

| .center { text-align: center }

```
<p class="right">
```

Ez egy jobbra igazított bekezdés.

```
</p>
```

```
<p class="center">
```

Ez egy középre igazított bekezdés.

```
</p>
```

p.right {text-align: right}
p.center {text-align: center}

Példa stílusosztályra:

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

```
.nagydoltpiros {
```

```
font-family: "Times New Roman", Times, serif;
```

```
font-size: xx-large;
```

```
color: #FF0000;
```

```
font-style: italic;
```

```
font-weight: bold;
```

```
font-variant: inherit;
```

```
text-transform: inherit;
```

```
}
```

```
</style>
```

a stílus alkalmazása egy bekezdésen:

```
<p class="nagydoltpiros">szöveg</p>
```

- A `<div>.....</div>` címkék (division=szakasz) egy általános **tárolóelemet** alkotnak (általában a weblap egy logikai összetartozó részét jelölik ki).
 - A közöttük elhelyezkedő tetszőleges tartalomra (szöveg, lista, táblázat, kép, hivatkozások, stb.) hivatkozni lehet, ill. formázást lehet hozzárendelni.

```
<div id="large">
    Ez egy <b>div</b> elem.
</div>

#large{
    color: #00FF00;
    font-family: arial;
    font-size: 4pt;
}
```

Szelektorok

- Szelektorok közötti kapcsolatok
 - leszármazott választó (szóköz)
 - adott elem leszármazottja.
 - gyermekválasztó (>)
 - szomszédos testvérválasztó (+)
 - A testvér-elemeknek ugyanaz a szülőeleme, és a "szomszédos" kifejezés azt jelenti, hogy "közvetlenül utánuk".
 - általános testvérválasztó (~)
 - egy meghatározott elem testvére.

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

```
div > p {  
    background-color: yellow;  
}
```

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

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

```
<!DOCTYPE html>
<html>
<head>
<style>
div > p {
    background-color: yellow;
}
</style>
</head>
<body>
```

```
<div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <span><p>Paragraph 3 in the div.</p></span> <!-- not Child but Descendant -->
</div>
```

```
<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
```

```
</body>
</html>
```

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

Kapcsoljuk össze a **html** kódokat és a **CSS** beállításokat!

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
</head>
```

```
  <body>
```

```
    <h2>Az osztály (class) tulajdonság</h2>
```

```
    <p>A CSS elemének class neve "city":</p>
```

```
    <h2 class="city">London</h2>
```

```
    <p>Anglia fővárosa London.</p>
```

```
    <h2 class="city">Paris</h2>
```

```
    <p>Franciaország fővárosa Párizs</p>
```

```
    <h2 class="city">Tokyo</h2>
```

```
    <p>Japán fővárosa Tokyo</p>
```

```
  </body>
```

```
</html>
```

1. Class azonosító

Az osztály (class) tulajdonság

A CSS elemének class neve "city":

London

Anglia fővárosa London.

Paris

Franciaország fővárosa Párizs

Tokyo

Japán fővárosa Tokyo

Az osztály (class) tulajdonság

A CSS elemének class neve "city":

London

Anglia fővárosa London.

Párizs

Franciaország fővárosa Párizs

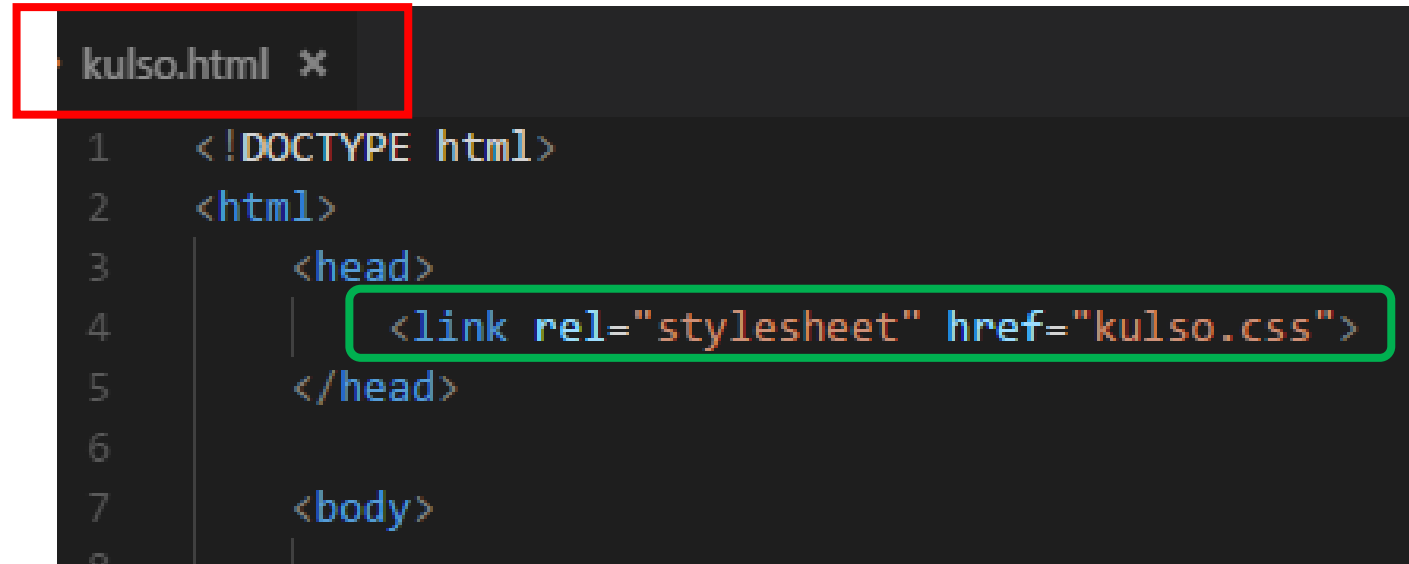
Tokyo

Japán fővárosa Tokyo

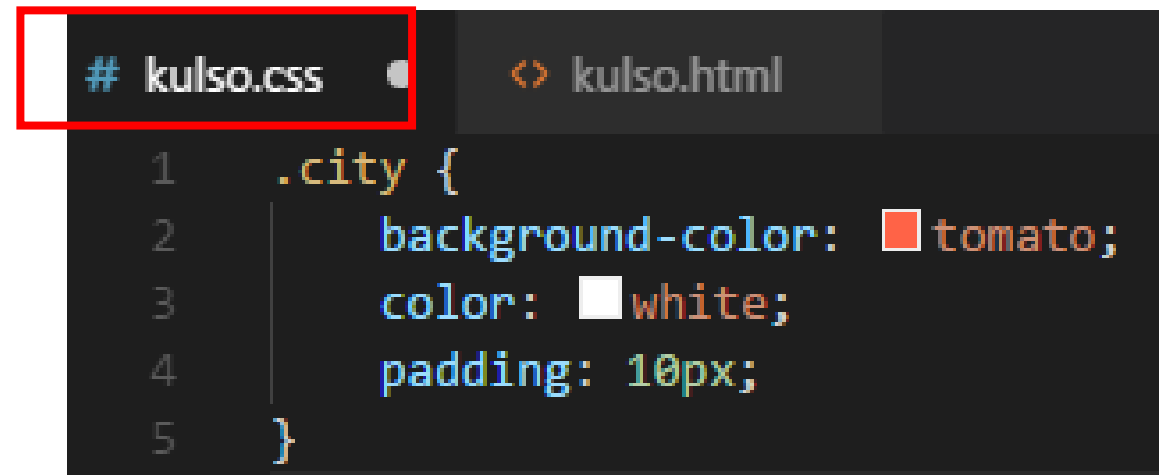
Beágyazott stíluslappal

```
<!DOCTYPE html>
<html>
<head>
<style>
.city {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>
</head>
```

Külső stíluslappal



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link rel="stylesheet" href="kulso.css">
5   </head>
6
7   <body>
8
```



```
1 .city {
2     background-color: tomato;
3     color: white;
4     padding: 10px;
5 }
```

2. ID azonosító

A stílusazonosító (id) tulajdonság

A CSS elemének id neve "myHeader":

Fejlécem



A stílusazonosító (id) tulajdonság

A CSS elemének id neve "myHeader":

Fejlécem

```
#myHeader {  
  background-color: lightblue;  
  color: black;  
  padding: 40px;  
  text-align: center;  
}
```

```
<!DOCTYPE html>  
<html>  
  <head>  
    <link rel="stylesheet" href="kulso.css">  
  </head>  
  
  <body>  
  
    <h2>A stílusazonosító (id) tulajdonság</h2>  
    <p>A CSS elemének id neve "myHeader":</p>  
  
    <h1 id="myHeader">Fejlécem</h1>  
  
  </body>  
</html>
```

Beágyazott stíluslappal

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Style the element with the id "myHeader" */
#myHeader {
    background-color: lightblue;
    color: black;
    padding: 40px;
    text-align: center;
}

/* Style all elements with the class name "city" */
.city {
    background-color: tomato;
    color: white;
    padding: 10px;
}
</style>
</head>
```

Szekciók

- A `<section>` elem egy dokumentum egy szakaszát határozza meg.
- A W3C HTML5 dokumentációja szerint:
 - "A szakasz a tartalom tematikus csoportosítása, jellemzően egy fejléccel.,,
- A *kezdőlapot* rendszerint szakaszokra lehet osztani a bevezetéshez, a tartalomhoz és az elérhetőségi adatokhoz.

WWF

The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of World Wildlife Fund. WWF was founded in 1961.

WWF's Panda symbol

The Panda has become the symbol of WWF. The well-known panda named Chi Chi that was transferred from the same year of the establishment of WWF.

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (
organization working on issues regard
restoration of the environment, forme
WWF was founded in 1961.</p>
</section>
```

```
<section>
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol
WWF originated from a panda named Chi
Beijing Zoo to the London Zoo in the
WWF.</p>
</section>
```

Média típusok

- Ma már egy weboldalt **többféle környezetben** használhatunk. Egy weboldalt megnézhetünk képernyőn, kézi-számítógépen, mobiltelefonon, vagy akár kinyomtatva is.
 - Célszerű minden médiához a hozzá legjobban illő formázást elkészíteni.
 - Pl. nyomtatásban a talpas, képernyőn a talp nélküli betűket tartjuk olvashatóbbnak.
 - Bizonyos elemeket (pl. menüket) felesleges kinyomtatni, míg weblapról kifelé mutató linkek href értékét nagyon hasznos megoldás.

Nézzünk példát a betűtípusok eltérő kezelésére. → **Nyomtatásban és képernyőn** más-más méretet és betűtípust alkalmazhatunk a következő stílusokkal:

```
@media screen {  
  p.test {font-family: verdana, sans-serif; font-size: 14px}  
}
```

```
@media print {  
  p.test {font-family: times, serif; font-size: 10px}  
}
```

```
@media screen, print {  
  p.test {font-weight: bold}  
}
```

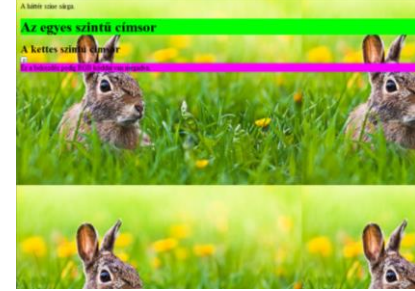
CSS betűtípusok

- A CSS **color** tulajdonság határozza meg a szöveg színét kell használni.
- A CSS **font-family** tulajdonság határozza meg a használni kívánt betűtípus.
- A CSS **font-size** tulajdonság határozza meg a szöveg méretét.
- Szöveg igazítása: **text-align**
- Szöveg dekorálása: **text-decoration**

```
h1 {  
    color: blue;  
    font-family: verdana;  
    font-size: 300%;  
}  
p {  
    color: red;  
    font-family: courier;  
    font-size: 160%;  
}
```

```
body {background-color: yellow}  
h1 {background-color: #00ff00}  
h2 {background-color: transparent}  
p {background-color: rgb(250,0,255)}
```

Háttérkép



- Az egyszínű háttér helyett akár **látványos háttérképet** is elhelyezhetünk az elemek háttereként a **background-image** tulajdonság segítségével:
background-image: url('nyuszi.jpg')
- A háttérkép alapértelmezés szerint **kitapétázva** jelenik meg, ha az elem mérete ezt szükségessé teszi. A mindkét irányú ismétlődés kikapcsolásához a **no-repeat** értéket kell adnunk.
- Elsősorban hosszabb görgethető tartalom esetén van jelentősége annak, hogy a háttérkép nem csak a görgetősávval együtt mozogva, hanem **fixen** pozícionálva is kérhető.

```
body {  
  background-color: yellow;  
  background-image: url('nyuszi.jpg');  
}  
h1 {background-color: #00ff00}  
h2 {background-color: transparent}  
p {background-color: rgb(250,0,255)}
```

```
body {  
  background-color: yellow;  
  background-image: url('nyuszi.jpg');  
  background-repeat: repeat-y;  
}
```

```
|background-attachment: fixed;
```


Pozíció

- Az alapértelmezett bal felső sarok **helyett máshol is lehet a háttérkép**. Ennek még akkor is van jelentősége, ha a teljes hátteret kitapétázva látjuk, ugyanis a kezdő tapétát van értelme máshol és máshol megadni.
- Az érték megadásnál először a **függőleges**, majd a **vízszintes pozíciót** kell megadnunk.
- A **3-3 konstans** (top, center, bottom és left, center, right) mellett **százalékos és pixeles** pozícionálás is lehetséges.

A következő példában az ablak felső részén, középen jelenik meg a kép.

```
body {  
  background-color: yellow;  
  background-image: url('nyuszi.jpg');  
  background-repeat: no-repeat;  
  background-position: top center;  
}
```



A teljes böngésző ablakot betöltő háttérkép

- Töltse ki az egész oldalt, ne ismétlődjön
- Szükség szerint méretezze a képet
- A kép az oldalon középen legyen
- Ne görgessen a görgetősávok között (rögzített)

```
html {  
    background: url(img_flower.jpg) no-repeat center fixed;  
    background-size: cover;  
}
```

Háttér méretezése az oldalhoz

background-size

- **Contain**

- Tartalmazza a háttérképet úgy, hogy a **lehető legnagyobb legyen** (de annak szélessége és magassága is illeszkedjen a tartalom területére).
- A háttérkép és a háttér pozicionálási terület arányától függően előfordulhatnak olyan háttérterületek, amelyekre a háttérkép nem terjed ki.

- **Cover**

- A háttérképet úgy méretezi, hogy a háttérterület **teljesen lefedje** a tartalomterületet (mind szélessége, mind magassága megegyezik vagy meghaladja a tartalomterületet).
- Mint ilyen, előfordulhat, hogy a háttérkép néhány része nem látható a háttér pozicionálási területen.

Original image:



background-size: contain:



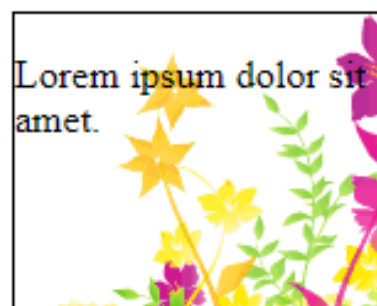
```
.div1 {  
  border: 1px solid black;  
  height: 120px;  
  width: 150px;  
  background: url(img_flwr.gif);  
  background-repeat: no-repeat;  
  background-size: contain;  
}
```

background-size: cover:



```
.div2 {  
  border: 1px solid black;  
  height: 120px;  
  width: 150px;  
  background: url(img_flwr.gif);  
  background-repeat: no-repeat;  
  background-size: cover;  
}
```

No background-size defined:



```
.div3 {  
  border: 1px solid black;  
  height: 120px;  
  width: 150px;  
  background: url(img_flwr.gif);  
  background-repeat: no-repeat;  
}
```

Szegélyek

- A CSS **border** tulajdonságaival szegélyeket rajzolhatunk az elemek köré.
- A szegélyek színét beállíthatjuk a **border-color** tulajdonsággal
 - Minden oldal egyforma színű:
`border-color: #0000ff;`
 - A fenti és lenti szegély `#ff0000` színű , míg a bal és jobb oldali `#0000ff` színű:
`border-color: #ff0000 #0000ff;`
 - A fenti szegély `#ff0000`, az oldalsók `#00ff00` és a lenti `#0000ff` színűek:
`border-color: #ff0000 #00ff00 #0000ff;`
 - A négy oldal az óramutató járásának megfelelően (kezdve a fenti szegéllyel) a felsorolt 4 értéket veszi fel:
`border-color: #ff0000 #00ff00 #0000ff rgb(250,0,255));`
- A szegély vastagságát a **border-width** tulajdonsággal állíthatjuk be.

A háttér színe sárga.

Az egyes szintű címsor

A kettes szintű címsor

Ez a bekezdés pedig RGB kóddal van megadva.

```
h1 {background-color: #00ff00;  
border-style: dotted;  
}
```

```
h2 {background-color: transparent;  
border-style: double;  
}
```

```
p {background-color: rgb(250,0,255);  
border-style: solid;  
}
```

Az egyes szintű címsor

border-color: #ff0000 #0000ff;

Ez a bekezdés pedig RGB kóddal van megadva.

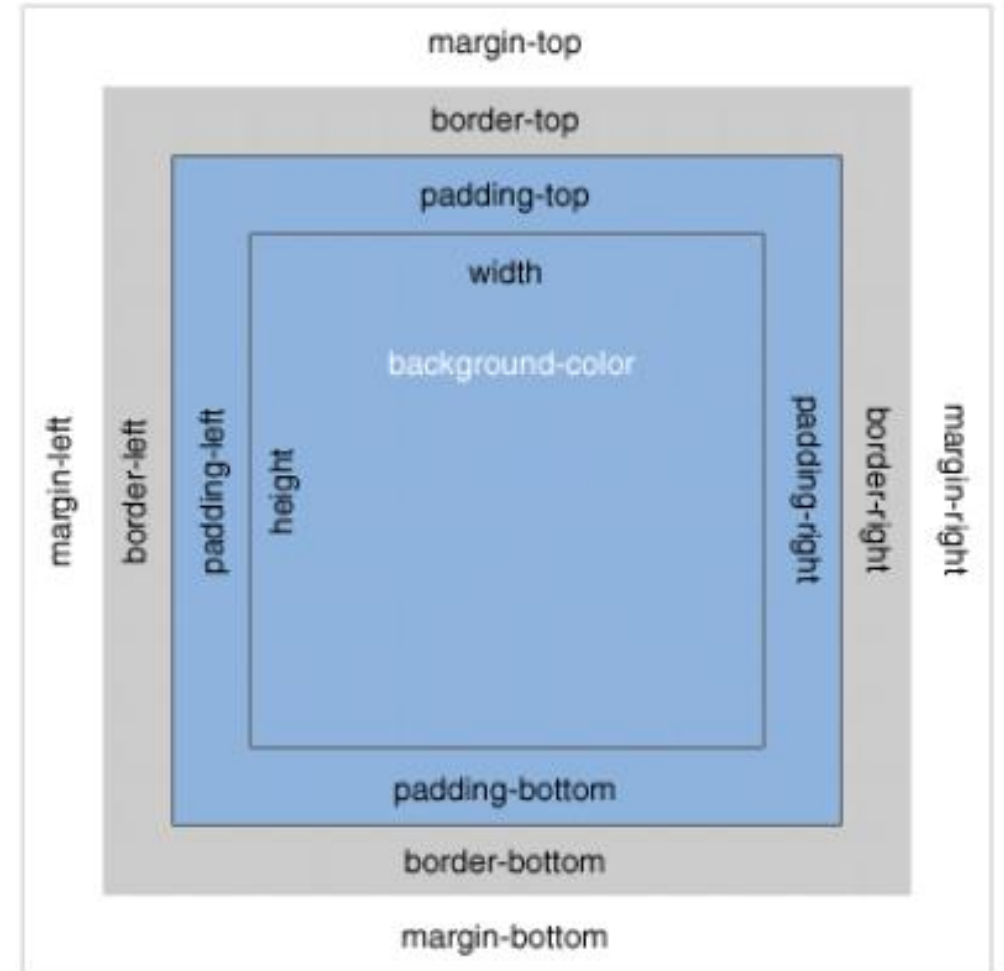
border-width: 15px;

Ez a bekezdés pedig RGB kóddal van megadva.

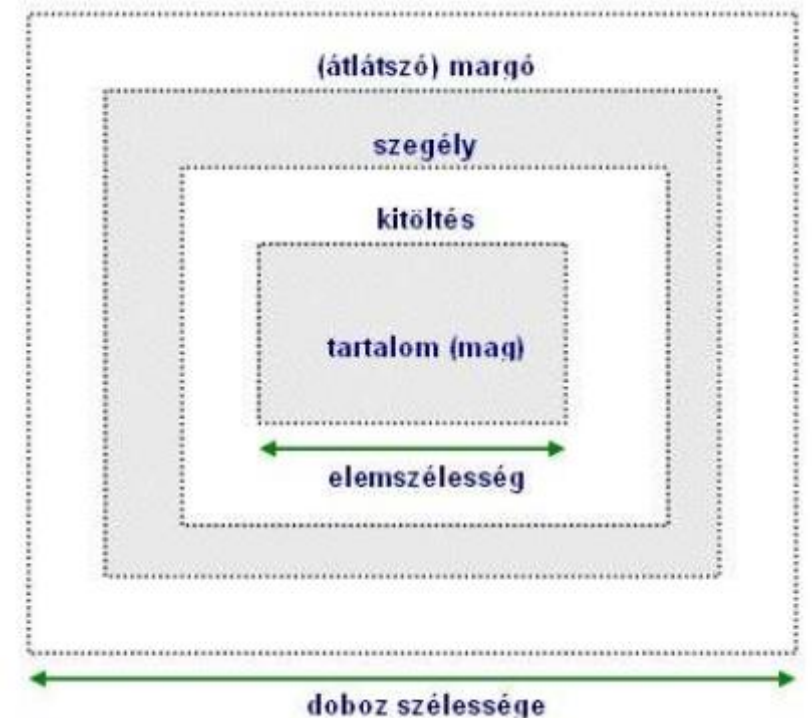
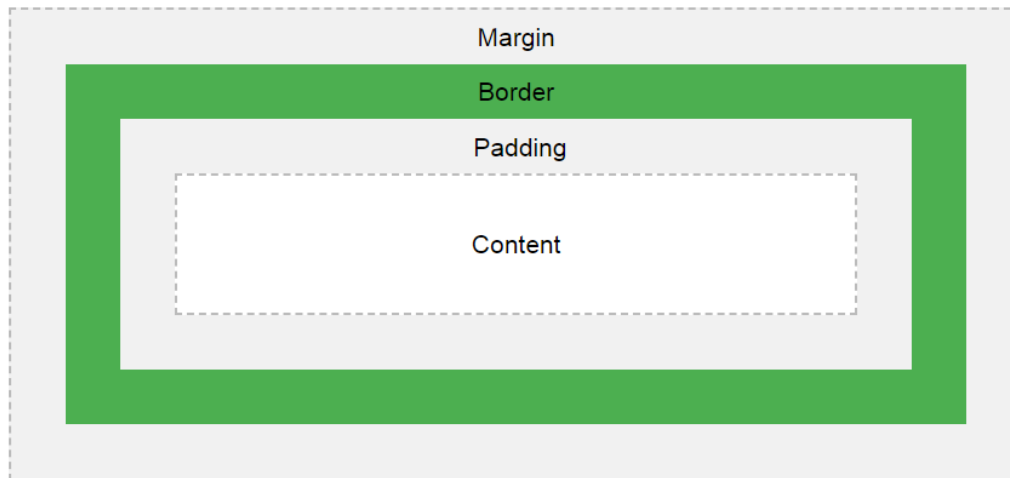
border-left-width: 15px;

Doboz modell

- A CSS stíluslapnyelv egy HTML dokumentum megjelenítéskor az egyes elemekhez **láthatatlan dobozokat** rendel.
- Minden doboznak lehetnek **külön formázási tulajdonságai**, mint pl. méret, előtérszín, betűtípus, szegély, háttér, stb.
- **Hierarchikusan** egymásba ágyazhatók:
 - Blokkszintű (bekezdés, lista, táblázat, div szakasz)
 - Sorközi (inline) elemek
- Minden doboznak van egy „magja”,
 - az őt körülvevő kitöltéssel „bélelés” (padding),
 - szegéllyel (border) és
 - margóval (margin).



- A CSS szintaktikailag két nagyon hasonló tulajdonság-csoportot tartalmaz:
 - a **margin** tulajdonsággal a szegélyen (border) kívüli = külső margó,
 - a **padding** tulajdonsággal pedig a tartalom és a szegély közötti belső margót lehet beállítani.
- **Méretezés, pozícionálás**
 - A **doboz szélessége** az elem, a kitöltés, a szegély és a margó területének összege.
- **Méret megadásának iránya:**
 - **Felül-jobbra-alul-balra**
 - Pl.: padding: 10px 11px 12px 13px



Margók

```
p {  
  margin-left: 3px;  
  margin-top: 0px;  
  margin-right: 1px;  
  margin-bottom: 2px;  
}
```

- Az egyes elemek a kereten (legyen az akár 0 képpontnyi) kívüli helyfoglalását szabályozhatjuk a margók segítségével.
- Margók esetén is van lehetőségünk, hogy egyszerre mind a négy oldal értékét, akár különbözőre is beállíthassuk: **margin: 0px 1px 2px 3px;**
- A négy paraméter helyett használhatunk **egyet** is, ilyenkor mind a négy oldalon azonos margót kapunk: **margin: 0px;**
- Ha csak **két értéket** adunk meg paraméterként, akkor az első paraméter a függőleges, a második paraméter pedig a vízszintes irányú margókra fog vonatkozni: **margin: 10px 0px;**
- Csak a bal oldali margót definiálja: **margin-left: 2cm**
- Minden oldalét definiálja, de más-más értékkel: **margin: 2cm 5px 2em 5%;**

Szegély

- A szegélyek (**border**) a margón belül helyezkednek el.

```
h1 {  
  border-top-style: solid;  
  border-right-style: double;  
  border-bottom-style: dotted;  
  border-left-style: dashed;  
}
```

Szélesség:

Tulajdonság	Leírás	Lehetséges értékek
border-width border-top-width border-right-width border-bottom-width border-left-width	Keret vastagságának megadása.	thin medium thick length

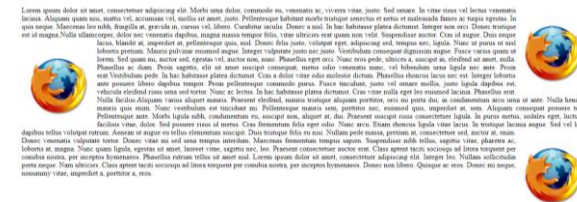
Stílus:

Tulajdonság	Leírás	Lehetséges értékek
border-style border-top-style border-right-style border-bottom-style border-left-style	Keret stílusának megadása. Ha nem állítjuk be, nem fog látszani a keret!	none hidden dotted dashed solid double groove ridge
		inset outset

Szín:

Tulajdonság	Leírás	Lehetséges értékek
border-color border-top-color border-right-color border-bottom-color border-left-color	Keret színének megadása.	color-rgb color-hex color-name

Lebegtetés (float)



- A **float** tulajdonság segítségével különböző elemeket az oldal bal vagy jobb szélére *úsztathatunk*. Ha egy elem úsztatva van, a tartalom többi része (szöveg) körbe veszi.
- A float és a *körbefutó tartalom* közötti hely kihagyást a **float margin** tulajdonságával lehet változtatni.

Float értékei:

- **left**: az elem balra tolódik, a szöveg az elem jobb oldalán folyik (fentről lefele)
- **right**: jobbra tolódik, a szöveg bal oldalt folyik le
- **none**: nem történik körülfolyás

Ha a float tulajdonság hatását meg akarjuk szüntetni akkor használjuk a **clear**: beállítást.

- Ha viszont két képet szeretnénk eltérő beállításokkal, akkor azonosítókat kell használni.

```
<body>
  <p>A Lorem Ipsum egy egyszerű szövegrészlete, szövegutá
  Nem csak 5 évszázadot élt túl, de az elektronikus betűk
  <img src=pic.jpg width=94 height=84 class="kep1"> Az 19
  A Lorem Ipsum egy egyszerű szövegrészlete, szövegutánza
  Nem csak 5 évszázadot élt túl, de az elektronikus betűk
  Az 1960-as években népszerűsítették a Lorem Ipsum részl
  Az 1960-as években népszerűsítették a Lorem Ipsum részl
  <img src=pic2.jpg width=94 height=84 class="kep2">A Lor
  Nem csak 5 évszázadot élt túl, de az elektronikus betűk
  Az 1960-as években népszerűsítették a Lorem Ipsum részl
```

```
img.kep1{
  float: right;
  margin: 1em;
}
img.kep2{
  float: left;
  margin: 1em;
}
p{
  text-align: justify;
}
```

clear

- Ezzel az utasítással azt adhatjuk meg, hogy egy elem a körülötte lévő elemeket mely irányból zárja ki mindenképp.

Értékei:

- none - Lehetővé teszi a lebegő elemeket mindkét oldalán.

Ez az alapértelmezett beállítás.

- left - Nincs lebegő elem a bal oldalon
- right- Nincs lebegő elem a jobb oldalon
- both - Nincs lebegő elem a bal oldalon vagy a jobb oldalon
- inherit - Az elem örökli a szülő egyértelmű értékét

```
.div1 {  
    float: left;  
    width: 100px;  
    height: 50px;  
    margin: 10px;  
    border: 3px solid #73AD21;  
}  
  
.div2 {  
    border: 1px solid red;  
}  
  
.div3 {  
    float: left;  
    width: 100px;  
    height: 50px;  
    margin: 10px;  
    border: 3px solid #73AD21;  
}  
  
.div4 {  
    border: 1px solid red;  
    clear: left;  
}
```

Without clear

div1

div2 - Notice that div2 is after div1 in the HTML code. However, since div1 floats to the left, the text in div2 flows around div1.

With clear

div3

div4 - Here, clear: left; moves div4 down below the floating div3. The value "left" clears elements floated to the left. You can also clear "right" and "both".

Az oldalszerkezet kialakítása



A div és span tagok

- A div és span tagok célja az, hogy rá lehessen akasztani valamilyen CSS formázást, ha nincs más alkalmas tag (vagy az túl erőltetett lenne) a HTML kódban.
- A két tag között egyetlen különbség:
 - **div** blokk szintű (mint pl. a p vagy a table),
 - **span** soron belüli (mint pl. a strong vagy az a) elem.
- Az elemek szélesség (width) és magasság (height) tulajdonsága segítségével az elem mérete befolyásolható.
- Elsősorban doboz-jellegű elemeknél van értelme használni:
 - például egy kép méretét, vagy a navigációs sáv szélességét gyakran állítjuk be ilyen módon

```
<div id="doboz"> Ez egy doboz </div>
```

```
div#doboz {  
  width: 200px;  
  padding: 10px;  
}
```

Linkek formázása

- Linkek különböző állapotaihoz eltérő színek:

Álosztály neve	Leírás
<code>:link</code>	A még meg nem látogatott linkeket választja ki.
<code>:visited</code>	A már meglátogatott linkeket választja ki.
<code>:hover</code>	Azt a linket választja ki, amelyik felett éppen áll az egér.
<code>:active</code>	Azt linket választja ki, amelyiken állva az egérrel a felhasználó lenyomta az egérgombot, de még nem engedte fel.

```
<body>
<p><b><a href="default.asp" target="_blank">Ez egy link</a></b></p>
    A Lorem Ipsum az 1500-as évek óta standard szövegrészletként szol
```

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

a:visited {
  color: green;
}

a:hover {
  color: hotpink;
}

a:active {
  color: blue;
}
```


Pozícionálás

- CSS-ben az elemek helyzetét a **position** tulajdonsággal tudjuk megadni.
- Alapértelmezett értéke a **static**, amikor az elemek a normál helyzetükben jelennek meg.

Relatív

- Először be kell állítani a **position** tulajdonság értékét **relative**-ra. Ezután a top (fent), right (jobb), bottom (lent), left (bal) tulajdonságokkal lehet megadni, hogy az *elem mennyivel legyen eltolva a normális helyzetéhez képest*.

Abszolút

- Abszolút helyzetmegadáskor a **position** értéke **absolute** legyen, és ezután is a fenti négy tulajdonsággal tudunk pozícionálni.
- Az abszolút pozícionálású elemek a (dokumentumfában) legközelebb eső *olyan szülő elemhez képest lesznek pozícionálva*, amelyekre szintén meg van adva a position tulajdonság MÁS értékkel, mint a static.

Fix

- Ha az **absolute** tulajdonság értékét **fix**-re állítjuk, akkor *az elem mindig a helyén marad*, nem lesz a többi elemmel együtt elgörgetve a gördítőssáv használatakor. Ekkor is lehet pozícionálni a top, right, bottom, left tulajdonságokkal.

Relatív

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="ch1104.css" rel="stylesheet" type="text/css">
<title>HTML fájl</title>
</head>
<body>
<div id="block_1">div id="block_1"</div>
<div id="block_2">div id="block_2"</div>
<div id="block_3">div id="block_3"</div>
<div id="block_4">div id="block_4"</div>
</body>
</html>
```

```
@charset "utf-8";

#block_1, #block_2, #block_3, #block_4 {
    position: relative;
    width: 400px;
}

#block_1 {
    background-color: green;
    left: 400px;
}

#block_2 {
    background-color: blue;
    right: 50px;
}

#block_3 {
    background-color: red;
    top: 100px;
    left: 200px;
}

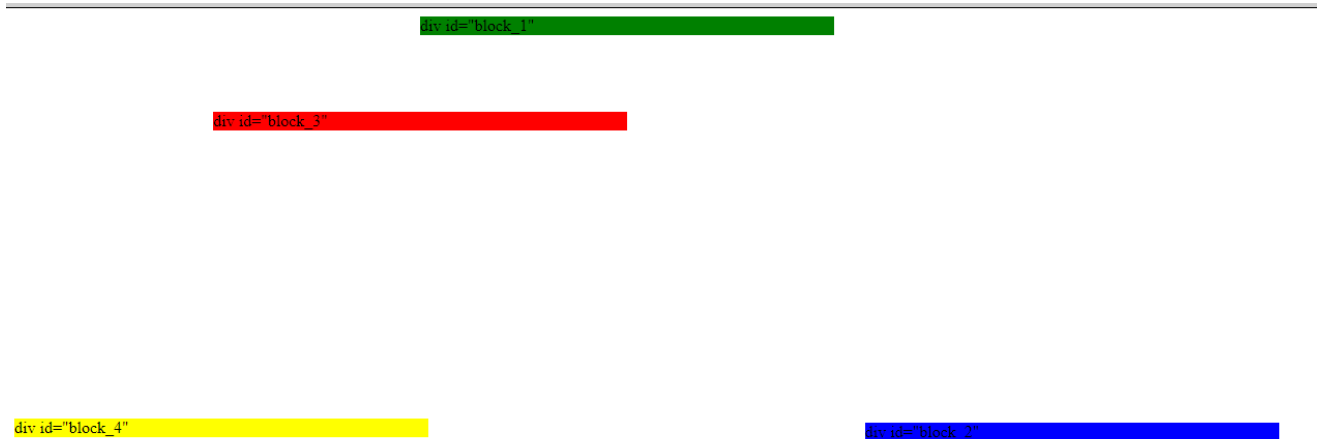
#block_4 {
    background-color: yellow;
    bottom: 30px;
}
```

div id="block_4"

div id="block_1"

div id="block_3"

Abszolút



```
#block_1, #block_2, #block_3, #block_4 {  
    position: absolute;  
    width: 400px;  
}
```

```
#block_1 {  
    background-color: green;  
    left: 400px;  
}
```

```
#block_2 {  
    background-color: blue;  
    right: 50px;  
    top: 400px;  
}
```

```
#block_3 {  
    background-color: red;  
    top: 100px;  
    left: 200px;  
}
```

```
#block_4 {  
    background-color: yellow;  
    bottom: 30px;  
}
```

Fix

Görgetéskor a kép
mindig ugyanott marad

```
</head>
<body>
<div id="fix"></div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus neque.
Mauris porta quam vitae diam. Integer sed dui non lectus iaculis vestibulum. In
velit. Ut luctus, diam sed sollicitudin malesuada, nibh urna venenatis enim, nec
viverra urna leo eget tellus. In pharetra. eros ut ultrices viverra. lacus felis
```

```
#fix {
  position: fixed;
  right: 0;
  bottom: 0;
}
```

sectetur adipiscing elit. Phasellus neque. Mauris porta quam vitae diam. Integer sed dui non lectus iaculis vestibulum. In velit. Ut luctus, diam sed sollicitudin enim, nec viverra urna leo eget tellus. In pharetra, eros ut ultrices viverra, lacus felis posuere turpis, et nonummy ante lacus eu nibh. Morbi eu mi in nibh idunt est. Integer vitae nisl. In ut est. In hac habitasse platea dictumst. Sed vulputate semper lacus. Vivamus ante enim, tempor varius, ultricies quis, viverra sed, euismod lobortis. Integer accumsan, lorem in pharetra ultrices, erat sem porttitor purus, in tincidunt augue enim vel arcu. Mauris vitae velit fermentum felis Praesent lorem nisi, euismod aliquam, nonummy id, molestie quis, mi. Morbi eu magna. Donec et nisi. Ut ut nunc. Vivamus eu purus. Mauris eget sem vel risus c orci. Etiam magna augue, ultricies sit amet, tincidunt sed, porta non, lectus. Fusce quam sapien, adipiscing nec, gravida sit amet, hendrerit non, augue. Nulla raesent viverra placerat justo. Aliquam augue. Duis mattis interdum erat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. ipit facilisis, tempus sit amet, risus. Nulla tempus laoreet lectus. Sed nec lacus et leo ultrices commodo. Quisque commodo mattis mauris. Nam lacinia orci id leo. esent iaculis orci tincidunt nisl. Etiam sed ipsum. Pellentesque eget tortor. Maecenas faucibus eros sodales risus. Aenean ut felis ac enim lacinia egestas. Nullam as, lacinia vitae, dapibus non, felis. Morbi erat. Fusce vitae pede. Duis nisl. Sed ac nulla eget nisi feugiat mattis. Nullam neque diam, vehicula quis, bibendum in, a. Vivamus et elit. Proin et augue. Curabitur facilisis. Praesent auctor turpis id velit. Nunc malesuada adipiscing dolor. In hac habitasse platea dictumst. Donec vel t velit. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Pellentesque vestibulum porta nisi. Vestibulum ante ipsum ultrices posuere cubilia Curae; Integer eu lacus eu mauris rhoncus aliquam. Nam sed ipsum. Praesent sed turpis rhoncus odio varius tristique. In sagittis purus metus, placerat ac, cursus nec, egestas ut, ligula. Nam sollicitudin erat id neque. Suspendisse potenti. Ut vitae augue. Praesent et arcu mollis dui rhoncus hendrerit. ulla aliquet. Ut ac dolor nec arcu rutrum pulvinar. In ullamcorper elit sed lectus. Etiam bibendum mollis velit. Nullam semper odio mollis mauris. Cras nibh, , adipiscing sit amet, accumsan et, urna. Proin interdum erat vel odio. In fringilla arcu sit amet mauris. Vivamus urna nisi, sagittis at, consequat sed, volutpat eu, ci. Sed blandit ullamcorper ante. Sed in augue ut nunc pharetra mollis. Phasellus eleifend porta orci. Fusce semper, risus ut blandit pellentesque, odio nulla accumsan orci, et mollis massa pede id libero. Vestibulum turpis ligula, hendrerit ut, gravida ut, facilisis in, augue. Ut tristique adipiscing justo. Maecenas dapibus odio ut nibh. Ut libero. Sed quam. Mauris bibendum mollis tortor. Etiam luctus vestibulum enim. Suspendisse egestas ultrices dolor. Sed tellus velit, consectetur et, sollicitudin ut, ullamcorper in, purus. Vivamus sit amet leo sit amet erat tempor blandit. Vivamus elit magna, fringilla quis, gravida sit amet, congue at, orci. In rutrum dui vel magna. Duis lacus augue, imperdiet eu, molestie et, ullamcorper nec, dolor. Sed tincidunt, purus sit amet tempor tincidunt, erat purus sodales tortor, non ullamcorper nunc dui non ante. Quisque vehicula laoreet felis. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Curabitur dui. Proin sit amet massa. Proin adipiscing. Maecenas quis dui ut nisl suscipit lacinia. Sed scelerisque tellus id arcu. Integer dapibus sodales arcu. Proin in lorem. Nullam id diam eget tellus vestibulum accumsan. Proin ullamcorper, sapien scelerisque ultrices feugiat, nisi ante viverra metus, a varius turpis quam et lacus. Vestibulum ut risus et enim viverra scelerisque. Phasellus pellentesque lectus eu ante. Sed volutpat, libero quis sagittis pellentesque, nibh felis malesuada est, nec semper elit risus pharetra sapien. Proin bibendum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vehicula sollicitudin eros. Integer dolor urna, dignissim nec, volutpat non, vestibulum ut, lacus. Duis in leo sollicitudin. Phasellus a libero ac sem feugiat molestie. Maecenas sit amet arcu. Cras diam ligula, pretium sed, tristique ut, malesuada sed, justo. Etiam commodo orci ferme ipsum dolor sit amet, consectetur adipiscing elit. Morbi dignissim semper nisi. Curabitur bibendum. Aliquam eleifend urna sed tortor. Morbi lorem. Aliquam bibendum. Nu sollicitudin pharetra, nunc mi pellentesque dui, vitae sollicitudin leo dui in mauris. Suspendisse potenti. Etiam euismod orci id felis. Vivamus in urna. Cras sollicitudin dolor feugiat. Ut in massa in dolor gravida ultricies. Nunc orci. Integer sodales aliquet neque. Nullam luctus interdum dui. Morbi molestie faucibus leo. Vestibulum ante ipsum pri luctus et ultrices posuere cubilia Curae; Integer sagittis ultrices lectus. Sed sapien. Fusce dictum justo ut enim. Vestibulum id nisl. Etiam feugiat turpis volutpat tortor. Vivam vitae risus. Nunc non felis. Integer nibh nisl, luctus eget, egestas vitae, vulputate nec, neque. Sed pede libero, suscipit sit amet, elementum nec, aliquet varius, lacus. Fusce fe lacus turpis, posuere sed, dignissim id, ultrices vel, diam. Duis venenatis ante eu leo. Sed massa diam, malesuada nec, eleifend vel, congue quis, turpis. Mauris accumsan ar Morbi in nisl. Nam eget risus non ante pharetra pulvinar. Morbi sagittis sem in tortor. Donec non massa a metus vulputate porta. Suspendisse ac massa in est bibendum temp



display:none vagy visibility:hidden

display:none

- Az elem rejtve lesz, és az oldal megjelenik, mintha az elem nem lenne ott

visibility: hidden

- Az elem rejtve lesz, de még mindig hatással van az elrendezésre. Az elem még mindig ugyanazt a helyet foglalja el, mint korábban.

Link gomb

Ez egy link

A Lorem Ipsum az 1500
papírra, ezt használta.

```
a:link, a:visited {  
    background-color: #36f446;  
    color: white;  
    padding: 14px 25px;  
    text-align: center;  
    text-decoration: none;  
    display: inline-block;  
}  
  
a:hover, a:active {  
    background-color: red;  
}
```

Horizontális menü

- A következő eredményt szeretnénk látni:



- *Kiindulópont* az ismert felsorolás szerkezetének megadása:

```
<ul>
  <li><a href="#">Menü 1</a></li>
  <li><a href="#">Menü 2</a></li>
  <li><a href="#">Menü 3</a></li>
</ul>
```

- Menü 1
- Menü 2
- Menü 3

1. A menü az alatta levő szövegtől elválik, birtokolja a teljes vízszintes sávját.
2. Csak maga a szöveg felesleges térközök és jelölő nélkül.
3. Fix szélességű gombok érhetők el. Érdeemes azonban vigyázni ezzel a módszerrel, mert ha a szöveg nem fér ki, a dizájn szétesik.
4. Ne legyen aláhúzva a link
5. A menün kívüli háttérszín és a jobb oldali szegély színe itt meg kell, hogy egyezzen.
6. Az egérkurzorra színváltással reagál a menüpont. Érdeemes megfigyelni, hogy a link a teljes li területét elfoglalja, ezért a teljes li elem linkként működik.
7. Ez a sor oldja meg, hogy a menüpontok egymás mellé kerüljenek.

```
ul {  
  float:left; } 1.  
width:100%;  
padding:0;  
margin:0; } 2.  
list-style-type:none;  
}  
a {  
  float:left; } 3.  
width:6em;  
text-decoration:none; 4.  
color:white;  
background-color:purple;  
padding:0.2em 0.6em; } 5.  
border-right:1px solid white;  
}  
a:hover { background-color:#ff3300 } 6.  
  
li { display:inline } 7.
```

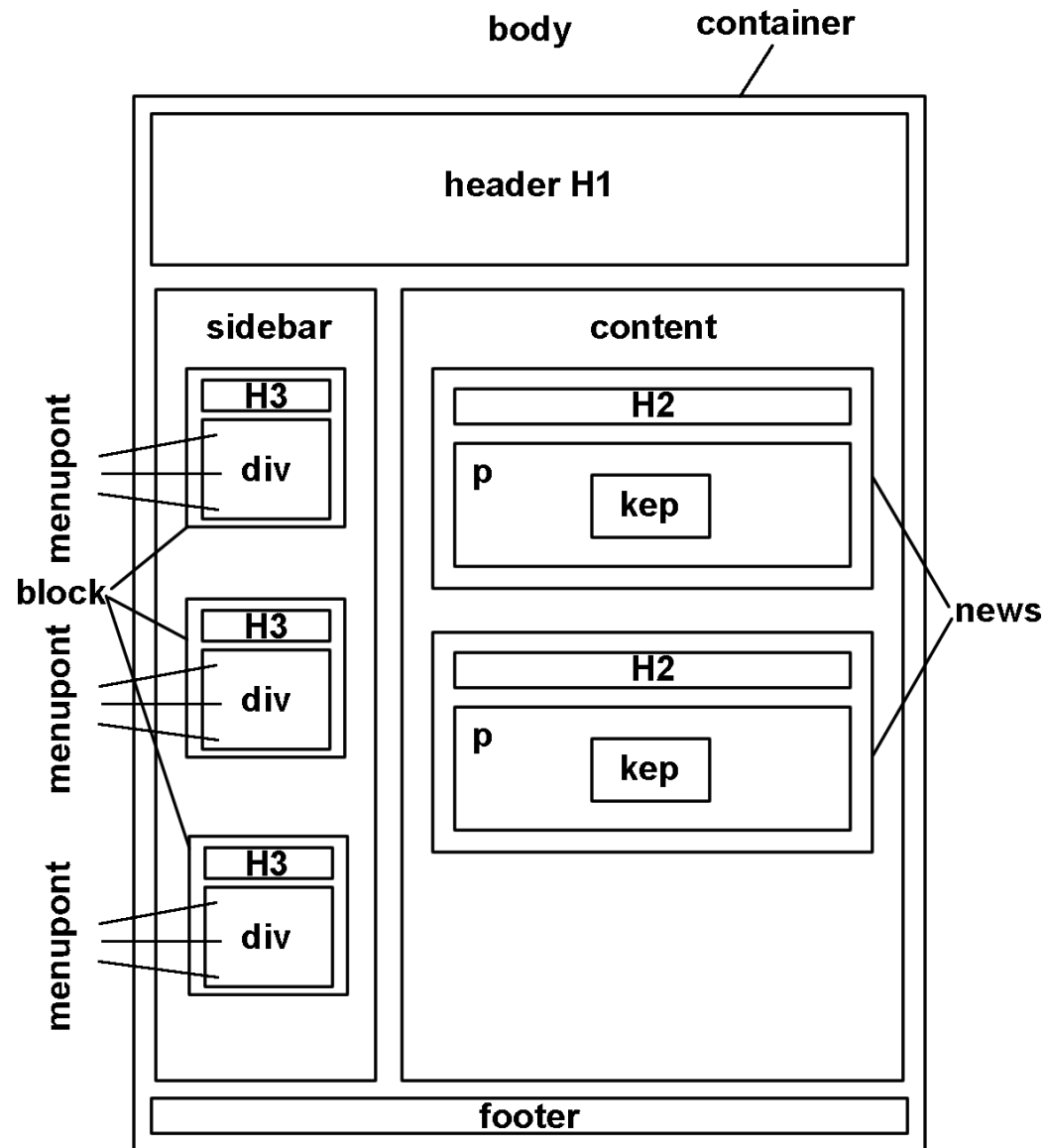

Oldalkialakítás táblázat nélkül

- A HTML szerkezet kialakításánál alapvetően a **fentről lefelé, azon belül balról jobbra** haladó tervezést érdemes követni.

Lorem Ipsum	
<p>A Lorem Ipsum egy egyszerű szövegrészlete, szövegutánczata a betűszedő és nyomdaiparnak. A Lorem Ipsum az 1500-as évek óta standard szövegrészletként szolgált az iparban;</p>	<h2>Miért használjuk?</h2> <p>Ez egy régóta elfogadott tény, miszerint egy olvasót zavarja az olvasható szöveg miközben a szöveg elrendezését nézi. A Lorem Ipsum használatának lényege, hogy többé-kevésbé rendezettebb betűket tartalmaz, ellentétben a Tartalom helye, Tartalom helye-féle megoldással.</p> <p>Sok <u>desktop</u> szerkesztő és weboldal szerkesztő használja a Lorem <u>Ipsum</u>-ot mint alapbeállítású szövegmodellt, és egy keresés a <u>lorem ipsum</u>-ra sok félkész weboldalt fog eredményezni.</p>
Lorem Ipsum	

Weblap elrendezések CSS-sel

- Container (tároló – DIV-eket befogadó)
 - A **container** nevet gyakran alkalmazzák az oldal fő tárolójának azonosításához.
 - A container szélessége alapvetően az egész oldal szélességét határozza meg.
- Header (fejrész)
- Menu (navigáció)
- Sidebar (menü)
- Content (tartalom)
- Footer (lábléc)



```

<body>
<div class="container">
  <div class="header">
    <h1 class="header">Lorem Ipsum</h1>
  </div>
  <div class="left">
    <p>A Lorem Ipsum egy egyszerű szövegré
  </div>
  <div class="content">
    <h2>Miért használjuk?</h2>
    <p>Ez egy régóta elfogadott tény, misz
    <p>Sok desktop szerkesztő és weboldal
  </div>
  <div class="footer">Lorem Ipsum</div>
</div>
</body>

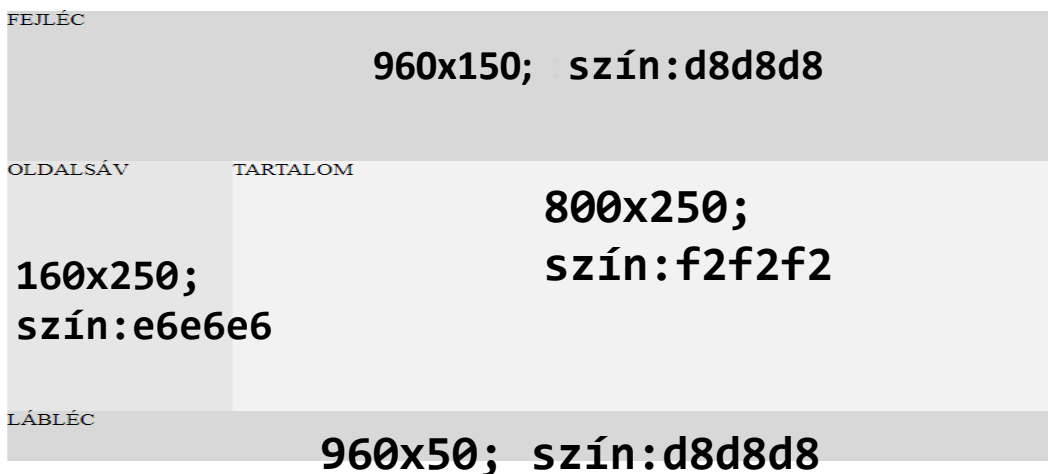
```

Lorem Ipsum	
<p>A Lorem Ipsum egy egyszerű szövegrészlete, szövegutánnata a betűszedő és nyomdaiarnak. A Lorem Ipsum az 1500-as évek óta standard szövegrészletként szolgált az iparban;</p>	<p>Miért használjuk?</p> <p>Ez egy régóta elfogadott tény, miszerint egy olvasót zavarja az olvasható szöveg miközben a szöveg elrendezését nézi. A Lorem Ipsum használatának lényege, hogy többé-kevésbé rendezettebb betűket tartalmaz, ellentétben a Tartalom helye, Tartalom helye-féle megoldással.</p> <p>Sok <u>desktop</u> szerkesztő és weboldal szerkesztő használja a Lorem <u>Ipsum</u>-ot mint alapbeállítású szövegmodellt, és egy keresés a lorem <u>ipsum</u>-ra sok félkész weboldalt fog eredményezni.</p>
Lorem Ipsum	

```

div.container {
  width: 100%;
  margin: 0px;
  border: 1px solid gray;
  line-height: 150%;
}
div.header, div.footer {
  padding: 0.5em;
  color: white;
  background-color: gray;
  clear: left;
}
h1.header {
  padding: 0;
  margin: 0;
}
div.left {
  float: left;
  width: 160px;
  margin: 0;
  padding: 1em;
}
div.content {
  margin-left: 190px;
  border-left: 1px solid gray;
  padding: 1em;
}

```



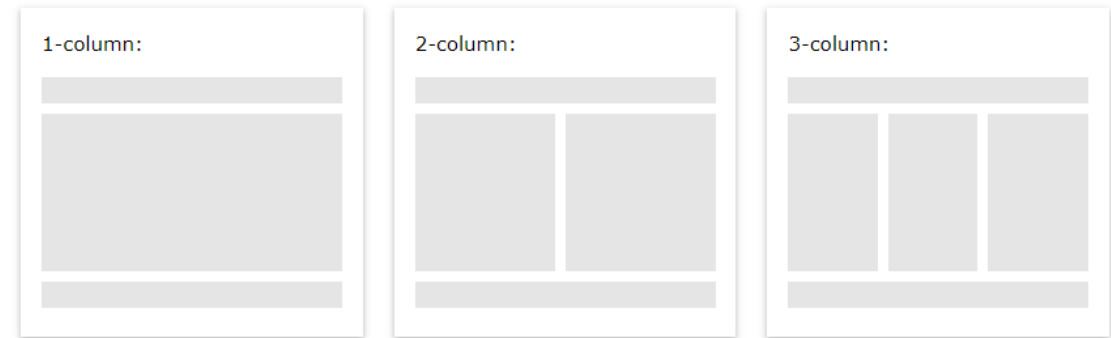
```
<body>  
<div class="header">Fejléc</div>  
<div class="sidebar">Oldalsáv</div>  
<div class="content">Tartalom</div>  
<div class="footer">Lábléc</div>  
</body>
```

CSS kód

```
.header {  
  background-color:#d8d8d8;  
  width:960px;  
  height:150px;  
}  
.sidebar {  
  background-color:#e6e6e6;  
  width:160px;  
  height:250px;  
  float:left;  
}
```

```
.content {  
  background-color:#f2f2f2;  
  width:800px;  
  height:250px;  
  float:left;  
}  
.footer {  
  background-color:#d8d8d8;  
  width:960px;  
  height:50px;  
  clear:both;  
}
```

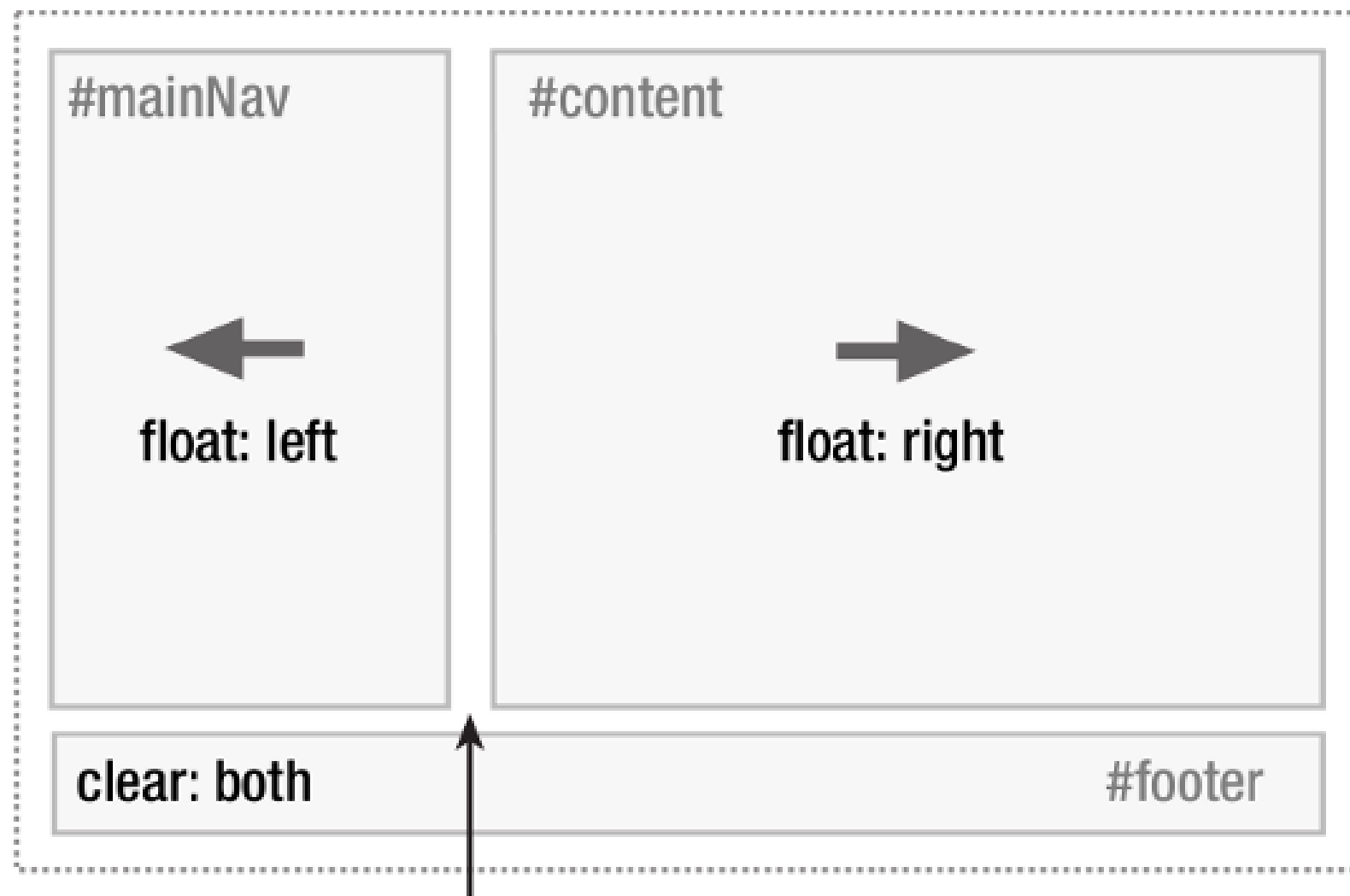
- A **fejléc** (*header*) általában a webhely tetején található (vagy közvetlenül egy felső navigációs menü alatt). Gyakran tartalmaz egy logót vagy a weboldal nevét
- A **navigációs sáv** olyan hivatkozásokat tartalmaz, amelyek segítenek a webhelyen navigáló látogatóknak
- **Content:** Az ebben a részben található elrendezés gyakran a célfelhasználóktól függ. A leggyakoribb elrendezés az alábbiak egyike (vagy azok kombinálása):
 - 1 oszlop (mobilböngészőknél)
 - 2 oszlop (tablettákhoz és laptopokhoz)
 - 3 oszlopos elrendezés (csak asztali gépekhez)



- A **lábléc** (*footer*) az oldal alján található. Gyakran olyan információkat tartalmaz, mint a szerzői jog és a kapcsolattartási adatok

Menü és tartalom

#wrapper



Virtuális margó

HTML:

```
<div id="wrapper">
  <div id="content">
    ...
  </div>
  <div id="mainNav">
    ...
  </div>
  <div id="footer">
    ...
  </div>
</div>
```

CSS:

```
#wrapper {
  width: 640px;
}
#content {
  width: 520px;
  float: right;
}
#mainNav {
  width: 180px;
  float: left;
}
#footer {
  clear: both;
}
```

Display (megjelenítés)

- A **Display** megjelenítési tulajdonság a legfontosabb CSS tulajdonság az elrendezés vezérléséhez.
- A megjelenítési tulajdonság határozza meg, hogy hogyan jelenik meg egy elem.
- Minden HTML elemnek alapértelmezett megjelenítési értéke van attól függően, hogy milyen típusú elem.
 - Az alapértelmezett megjelenítési érték a legtöbb elem számára **blokk vagy inline**.

Div elemek

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

- A **blokkszintű elem** mindig egy új vonalon kezdődik, és felveszi a rendelkezésre álló teljes szélességet (amennyire csak lehet, balra és jobbra húzódik).
- Egy **inline elem** nem indul új vonalon, és csak annyi szélességet igényel, amennyire szükséges.
 - Ez egy inline elem egy bekezdésen belül.
 - Egy gyakori példa az inline elemek létrehozása a vízszintes menük számára
 - Az inline-blokk elemek olyanok, mint az inline elemek, de **szélességük és magasságuk lehet.**

inline elemek

-
- <a>
-


```
.floating-box {
  float: left;
  width: 150px;
  height: 75px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.after-box {
  clear: left;
  border: 3px solid red;
}
```

Az eredmény ugyanaz, de eltérő CSS elemet használ.

```
.floating-box {
  display: inline-block;
  width: 150px;
  height: 75px;
  margin: 10px;
  border: 3px solid #73AD21;
}

.after-box {
  border: 3px solid red;
}
```

```
<body>

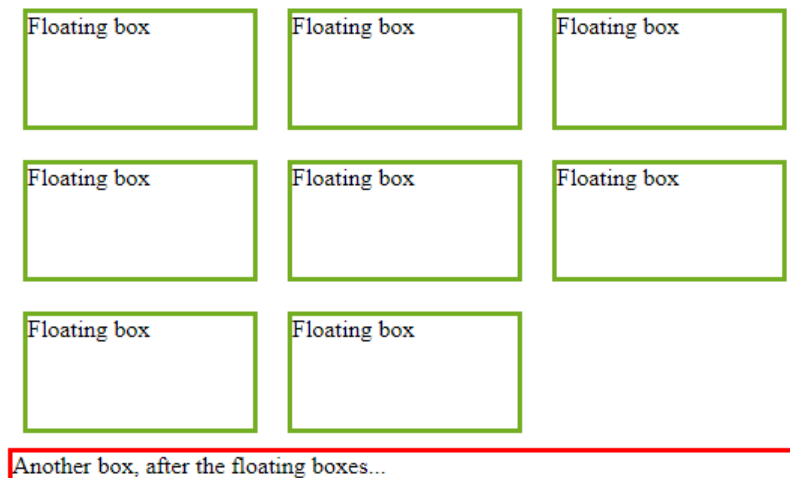
<h2>The Old Way - using float</h2>

<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>
<div class="floating-box">Floating box</div>

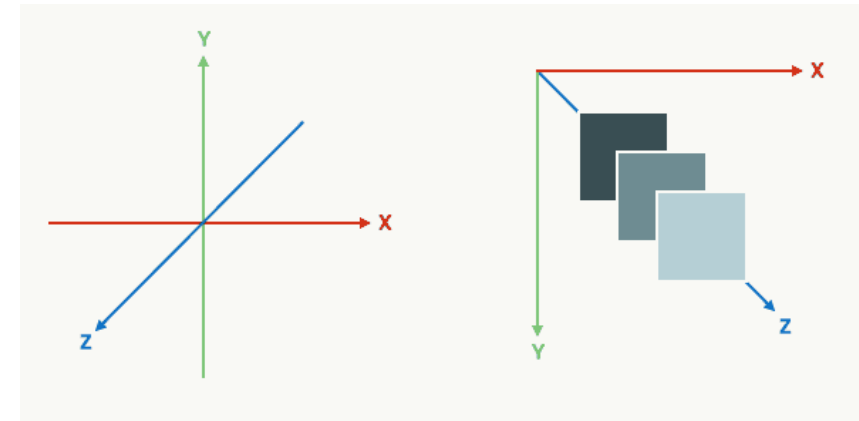
<div class="after-box">Another box, after the floating boxes...</div>

</body>
```

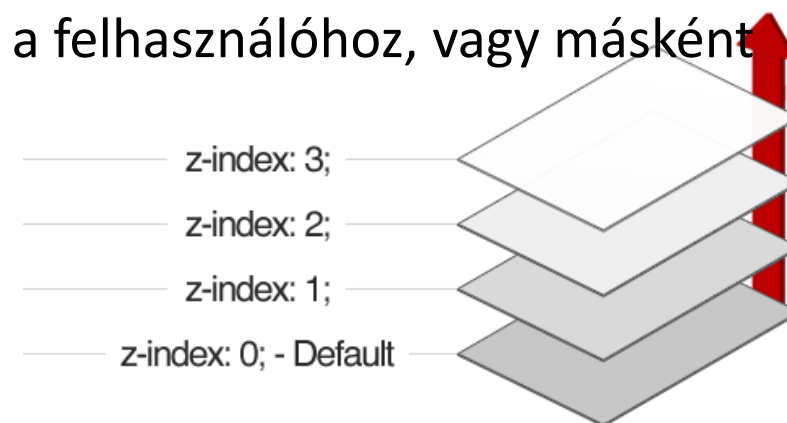
The Old Way - using float



Rétegek (z-index)



- Az elemek fedhetik egymást.
- A CSS alapú megjelenítéskor minden abszolútként pozícionált elemnek van egy pozíciója a **harmadik dimenzióban**, a "z" tengely mentén (ide érthetjük a gyöker elemet is, ami a left:0, top:0 pozícióba van pozícionálva). Az alapértelmezett érték 0.
- Az "x" tengely a **vízszintes**, az "y" tengely a **függőleges** pozíciót jelöli, a "z" tengely pedig ebben az esetben a **mélységet**.
 - A mélységet akkor tudjuk értelmezni a monitor sík kijelzőjén, ha több ilyen elem egymásra kerül, **egymást elfedi**.
 - Ekkor az fogja elfedni a másikat, mely előrébb, "közelebb" van a felhasználóhoz, vagy másként mondva, "feljebb" van a z tengelyen.



- A z tengelyen való elhelyezkedést a **z-index** tulajdonság segítségével, egy egész számot használva adhatjuk meg.
- Ha más értéket adunk meg, a nagyobb számmal bíró elem kerül közelebb a felhasználóhoz.



```
<body>

<h1>This is a heading</h1>

<p>Because the image has a z-index of -1, it wil
text.</p>

</body>
```

```
img {
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
```

További lehetőségek...

- **CSS Style Image / RWD Image:** Képek formázása → Lekerekítés, Bélyegkép, Reszponzív (automatikus) méretezés, Áttetszőség, Átfedések, Képgaléria, Egérműveletek
- **CSS Buttons:** Gombok (menügomb) → Színe, mérete, formája, animálása, árnyékolt, felirat
- **Overflow:** A CSS túlcsordulási tulajdonság szabályozza, hogy mi történik olyan tartalommal, amely túl nagy ahhoz, hogy illeszkedjen egy adott területhez.
- **Horizontal & Vertical Align**
- **Opacity:** Az átlátszatlanság tulajdonság egy elem áttetszőségét / átlátszóságát határozza meg.
 - `opacity:hover` → gyakran használják egérmutatóval az áttetszőség megváltoztatásához
- **Horizontal & Vertical Navigation Bar**
- **Dropdown:** lebegő legördülő menü
- **Image Gallery**
- **Forms:** űrlapok
- **Counters:** számlálók

Oldal vizsgálata

- A megértéshez érdemes a működő példát a **jobb gomb/Vizsgálat** menüpont nézetével is megnézni.

Lorem Ipsum

A Lorem Ipsum egy egyszerű szövegrészlete, szövegutánzata a betűszedő és nyomdaiparnak. A Lorem Ipsum az 1500-as évek óta standard szövegrészletként szolgált az iparban;

használjuk?

Ez egy régóta elfogadott tény, miszerint egy olvasót zavarja az olvasható szöveg miközben a szöveg elrendezését nézi. A Lorem Ipsum használatának lényege, hogy többé-kevésbé rendezettebb betűket tartalmaz, ellentétben a Tartalom helye, Tartalom helye-féle megoldással.

Sok desktop szerkesztő és weboldal szerkesztő használja a Lorem Ipsum-ot mint alapbeállítású szövegmodellt, és egy keresés a lorem ipsum-ra sok félkész weboldalt fog eredményezni.

Elements Console Sources Network Performance >> X

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" == $0
  <head>...</head>
  <body>
    <div class="container">
      <div class="header">
        <h1 class="header">Lorem Ipsum</h1>
      </div>
      <div class="left">
        <p>...</p>
      </div>
      <div class="content">
        <h2>Miért használjuk?</h2>
        <p>...</p>
        <p>...</p>
      </div>
      <div class="footer">
      </div>
    </body>
  </html>
```

html

Styles Event Listeners DOM Breakpoints Properties

Filter :hov .cls +

element.style {
}
html {
 display: block;
}
user agent stylesheet

margin -

border -

padding -

725 x 393.813

Filter ☐ Show all

display block

height 393.813...

width 725px

CSS sablonok

- A CSS **sablonokat** (CSS **templates**) weblapsablonoknak is nevezik.
 - Ezek valójában szép dizájnnal ellátott, publikálásra kész honlapok, amelyekbe csak a saját szöveged kell beírni.
- A szöveg módosítása után feltöltheted szerverre és már működik is a kész statikus honlapod.
- Az Interneten sok fizetős CSS-sablont találhatsz, de nagyon sok **ingyenes** sablon is van, ezeket azzal a feltétellel töltheted le, hogy az oldaladon (általában alul) megjelenik egy hivatkozás a sablon készítőjére.
- Pl.: <https://templated.co/>