



1

# CSS alapok

Répásné Babucs Hajnalka

# CSS

- **C**ascading **S**tyle **S**heets (rangsorolt stíluslapok)
- Vizuális megjelenést határozza meg weblap esetén
- Szöveges fájl, .css a kiterjesztés
- Weblapba beágyazható, de csatolható is
- Formázási stílusokkal tudja módosítani a weblap formázható elemeit

# Külső CSS és a HTML fájl összekapcsolása

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

- A HTML fájl <head> részébe a <title> alá kell beilleszteni
- **href** attribútum mutatja meg a külső CSS fájl elérési útvonalát
- **rel** (relation) attribútum a kapcsolat/viszony típusát határozza meg, stíluslap
- **type** attribútum a CSS fájl típusát határozza meg, szöveges fájl

# Belső CSS és a HTML fájl kapcsolata

- `<style type="text/css"> stílusok </style>`
- A HTML fájl `<head>` részébe kell berakni a `<style>` HTML kódot.
- A nyitó, valamint a záró tag-ek közé kell elhelyezni a stílusokat egymás után.

# Megjegyzés

- CSS fájlba megjegyzéseket el lehet helyezni a `/*` és `*/` jelek közé.
- `/*megjegyzés szövege*/`

# Kijelölők (stílusok) csoportosítása

- A stílusokat a kijelölők (selector) segítségével lehet megadni.
- Típusai:
  - Elem kijelölő
  - Osztály kijelölő
  - Azonosító kijelölő
  - Csoport kijelölő

## Egy stílus/kijelölő felépítése (általánosan)

```
kijelölő_neve{  
    tulajdonság: érték;  
    ...  
}
```

# Elem kijelölő (element selector)

- A legegyszerűbb kijelölő
- A kijelölő neve mutatja meg, hogy a honlapon melyik elemre vonatkozik a stílus.
- A kijelölő neve mindig egy HTML elem neve lesz



# Elem kijelölő példa



- Az oldalon található **összes bekezdést** megformázza
  - (betűstílus: Arial, betűméret kicsi, betűszín: kék)

# Osztály kijelölő (class selector)

- Az így definiált stílus csak azokra a HTML elemekre lesz érvényes, amelyekhez hozzárendeljük a stílust a HTML kódban.
- Egy weblapon akárhány példányban alkalmazhatjuk a stílust, különböző elemekre is.
- A stílus neve . (ponttal) kezdődik

# Osztály kijelölő felépítése

```
.stílus_neve{  
    tulajdonság: érték;  
}
```

- HTML kódot is módosítani kell.
- Amelyik HTML elemet szeretnénk osztály kijelölővel formázni, ott a nyitó tag-be be kell írni a `class="stílus_neve"` kódot. Ezzel rendeljük hozzá az osztály kijelölőnket a HTML elemhez.

## Osztály kijelölő példa

- Egy 3 pixel vastag folytonos piros szegéllyel keretezünk egy bekezdést és egy képet.

CSS	HTML
<pre><b>.kiemel</b>{     border: 3px solid     red; }</pre>	<pre>&lt;p <b>class="kiemel"</b>&gt;...&lt;/p&gt; ... &lt;img <b>class="kiemel"</b>     href="kep.png"&gt;</pre>

## Azonosító kijelölő (id selector)

- Az így definiált stílus csak arra az egy HTML elemre lesz érvényes, amelyhez hozzárendeljük a stílust a HTML kódban.
- Egy weblapon egy példányban alkalmazhatjuk a stílust, mert az azonosítót (id) egy weblapon csak egyszer lehet használni.
- A stílus neve # (kettős kereszttel) kezdődik

# Azonosító kijelölő felépítése

```
#stílus_neve{
```

```
    tulajdonság: érték;
```

```
}
```

- HTML kódot is módosítani kell.
- Amelyik HTML elemet szeretnénk osztály kijelölővel formázni, ott a nyitó tag-be be kell írni az `id="stílus_neve"` kódot. Ezzel rendeljük hozzá az azonosító kijelölőnket a HTML elemhez.

# Azonosító kijelölő példa

A fejléc háttérszíne legyen sötétkék, magassága 150 pixel és a betűszíne fehér.

CSS	HTML
<pre>#fejlec{   background-color: darkblue;   height: 150px;   color: white; }</pre>	<pre>&lt;div id="fejlec"&gt; ... &lt;/div&gt;</pre>

## Csoport kijelölő (grouping selector)

- Ezt a kijelölőt akkor használjuk, amikor különböző elemekre ugyanazok a tulajdonságok érvényesek.
- Ennek a szelektor használatának egyik célja, hogy minimalizálja a kódot.
- A csoport szelektor neve az úgy épül fel, hogy az egyes stílusok neve vesszővel elválasztva, stíluson belül pedig azok a tulajdonságok, amelyek a felsorolt stílusokra jellemző.



# Csoport kijelölő felépítése

```
nev1, nev2, nev3 {  
    tulajdonság: érték;  
}
```

## Csoport kijelölő példa

- A bekezdésnek és az 1. és 2. szintű címsor betűszíne legyen kék, és középre igazított.

CSS	HTML
<b>p, h1, h2</b> { color: blue; text-align: center; }	HTML fájlt csak akkor kell módosítani, ha felsorolt stílusnevek között van osztály-, vagy azonosító kijelölő.

# Szín megadási módok (Hexadecimális szám és szöveges mód)

## ■ Hexadecimális számmal (#RRGGBB vagy #RGB)

- `background-color: #0000FF;`
- `background-color: #00F;`

## ■ Szöveges módon

- `background-color: blue;`

# Szín megadási módok (RGB/RGBA)

- `background-color: rgb(0,0,255);`
- `background-color: rgba(0,0,255,0.5);`
  - A 4. paraméter az átlátszóság
  - 0: teljesen átlátszó, 1: teljesen látható, a kettő közötti valós értéket veheti fel

# Szín megadási módok (HSL/HS�A)

- `background-color: hsl(0,0,255);`
- `background-color: hsla(0,0,255,0.5);`
  - 1. paramétere (hue): 0 és 360 fok közötti érték lehet. 0, vagy 360 piros, 120 a zöld és 240 a kék színt jelenti.
  - 2. paramétere (saturation): 0 és 100% (full color) közötti százalékos értéket vehet fel.
  - 3. paramétere (lightness): 0 (black) és 100% (white) közötti százalékos értéket vehet fel.
  - hsla függvény esetén a 4. paraméter az opacity: 0 és 1 közötti valós értéket vehet fel, amivel a szín láthatóságát határozzuk meg

# Leggyakrabban használt mértékegységek

- ▶ pt – betűméret
- ▶ px – képpont
- ▶ em – arány (1 em = 100%)
- ▶ in – inch
- ▶ % – százalék

# Háttérformázás

CSS tulajdonság	Lehetséges értékek	Leírás
<b>background</b>	szín háttérkép ismétlődés pozíció	háttér megadása egy utasításban

► Pl.:

```
background: #fff url(kep.png) no-repeat right top;
```

# Háttérszín meghatározás

CSS tulajdonság	Lehetséges értékek	Leírás
<b>background-color</b>	szín	háttérszín meghatározása

► Pl.:

```
background-color: #fff;
```



# Háttérkép beállítás

CSS tulajdonság	Lehetséges értékek	Leírás
<b>background-image</b>	url("kep.png")	A háttérkép elérési útvonala
<b>background-repeat</b>	<ul style="list-style-type: none"><li>• repeat (ismétlődés)</li><li>• no-repeat (ne ismétlődjön)</li><li>• repeat-x (vízszintesen ismétlődjön)</li><li>• repeat-y (függőlegesen ismétlődjön)</li></ul>	A háttérkép ismétlődését határozhatjuk meg

# Háttérkép rögzítése

CSS tulajdonság	Lehetséges értékek	Leírás
<b>background-position</b>	<ul style="list-style-type: none"><li>• top (felülre)</li><li>• center (középre)</li><li>• bottom (alulra)</li><li>• left (balra)</li><li>• right (jobbra)</li></ul>	A háttérkép pozícióját adja meg. Konkrét számérték is használható (px)
<b>background-attachment</b>	<ul style="list-style-type: none"><li>• scroll (görgethető)</li><li>• fixed (rögzített)</li></ul>	A háttérkép csatolásának módja

# Háttérkép példák

- background-position: left top;
  - A háttérkép a bal felső sarokba kerül.
  - Az első a vízszintes pozíció, a második pedig a függőleges pozíciót jelenti.
- background-position: 25px 30px;
  - A háttérkép pontos pozíciója
  - x koordinátája: 25px
  - y koordinátája: 30px
  - Százalékban is megadható

# Szöveg, betűk formázása

CSS tulajdonság	Lehetséges értékek	Leírás
<b>color</b>	szín	betűszín
<b>font</b>	font-style font-variant font-weight font-size/line-height font-family	gyűjtő tulajdonságként adjuk meg

► Pl.:

font: italic small-caps bold 12px/30px Georgia, serif

# Betűtípus, betűstílus, betűméret, betűvastagság

CSS tulajdonság	Lehetséges értékek	Leírás
<b>font-family</b>	betűtípus pl.: <i>Arial</i> , vagy <i>"Times New Roman"</i>	a szöveg betűtípusát határozzuk meg
<b>font-style</b>	<ul style="list-style-type: none"><li>• normal (normál)</li><li>• italic (dőlt)</li><li>• oblique (dőlt)</li></ul>	a betű stílusát határozza meg
<b>font-size</b>	szám, százalék, arány	betűméretét határozza meg arány esetén van egy használható képlet: <i>pixels/16=em</i>
<b>font-weight</b>	<ul style="list-style-type: none"><li>• normal (normál)</li><li>• bold (félkövér)</li></ul>	betű stílusát, vastagságát határozza meg

# Szöveg stílus, a betűk vízszintes nyújtása, sor magasság

CSS tulajdonság	Lehetséges értékek	Leírás
<b>font-variant</b>	normal (normál) small-caps (kiskapitális)	szöveg stílust lehet megadni vele
<b>font-strech</b>	normal ( <i>normál</i> ) condensed ( <i>sűrített</i> ) ultra-condensed extra-condensed semi-condensed expanded ( <i>nyújtott</i> ) semi-expanded extra-expanded ultra-expanded	a betűk vízszintes nyújtása
<b>line-height</b>	szám, százalék	a sor magassága

# Szöveg igazítása, díszítése

CSS tulajdonság	Lehetséges értékek	Leírás
<b>text-align</b>	center (középre) left (balra) right (jobbra) justify (sorkizárt)	szöveg igazítása
<b>text-justify</b>	auto ( <i>automatikus</i> ) inter-word ( <i>növeli/csökkenti a szavak közti távolságot</i> )	ha a szövegigazítás sorkizárt (text-align: justify), akkor ez a tulajdonság határozza meg a sorkizárt igazítás módszerét.
<b>text-decoration</b>	none (nincs) overline (felül húzott) line-through (áthúzott) underline (aláhúzott)	szöveg díszítése, aláhúzása

# Betűk átalakítása, első sor behúzása, betűköz, szóköz

CSS tulajdonság	Lehetséges értékek	Leírás
<b>text-transform</b>	<ul style="list-style-type: none"><li>• uppercase (nagybetűs)</li><li>• lowercase (kisbetűs)</li><li>• capitalize (kezdőbetűk nagybetűsek)</li></ul>	a betűk formázása, átalakítása
<b>text-indent</b>	szám	első sor behúzás
<b>letter-spacing</b>	szám	betűköz
<b>word-spacing</b>	szám	szóköz



# Kurzor

CSS tulajdonság	Lehetséges értékek	Leírás
<b>cursor</b>	<ul style="list-style-type: none"><li>• auto</li><li>• default</li><li>• help</li><li>• move</li><li>• pointer</li><li>• wait</li><li>• text</li><li>• zoom-in</li><li>• zoom-out</li></ul>	a kurzor alakjának beállítása adott elem esetén

# Felsorolások

CSS tulajdonság	Lehetséges értékek	Leírás
<b>list-style</b>	listajel_típusa pozíció kép_elérési_útvonal	listák, felsorolások beállítása

► Pl.:

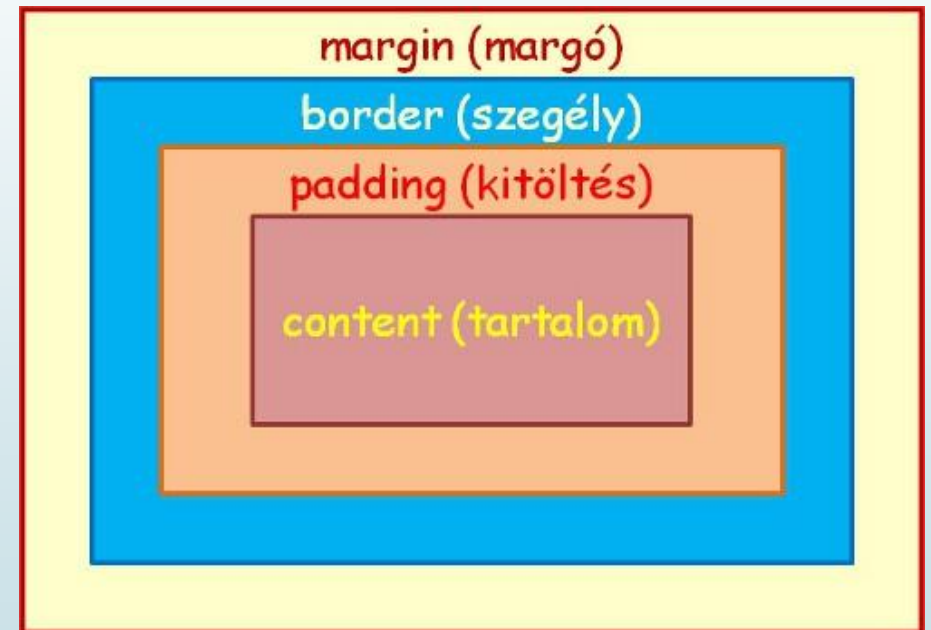
```
list-style: square inside url('listajel.png');
```

# Listajel típusa, pozíciója, kép

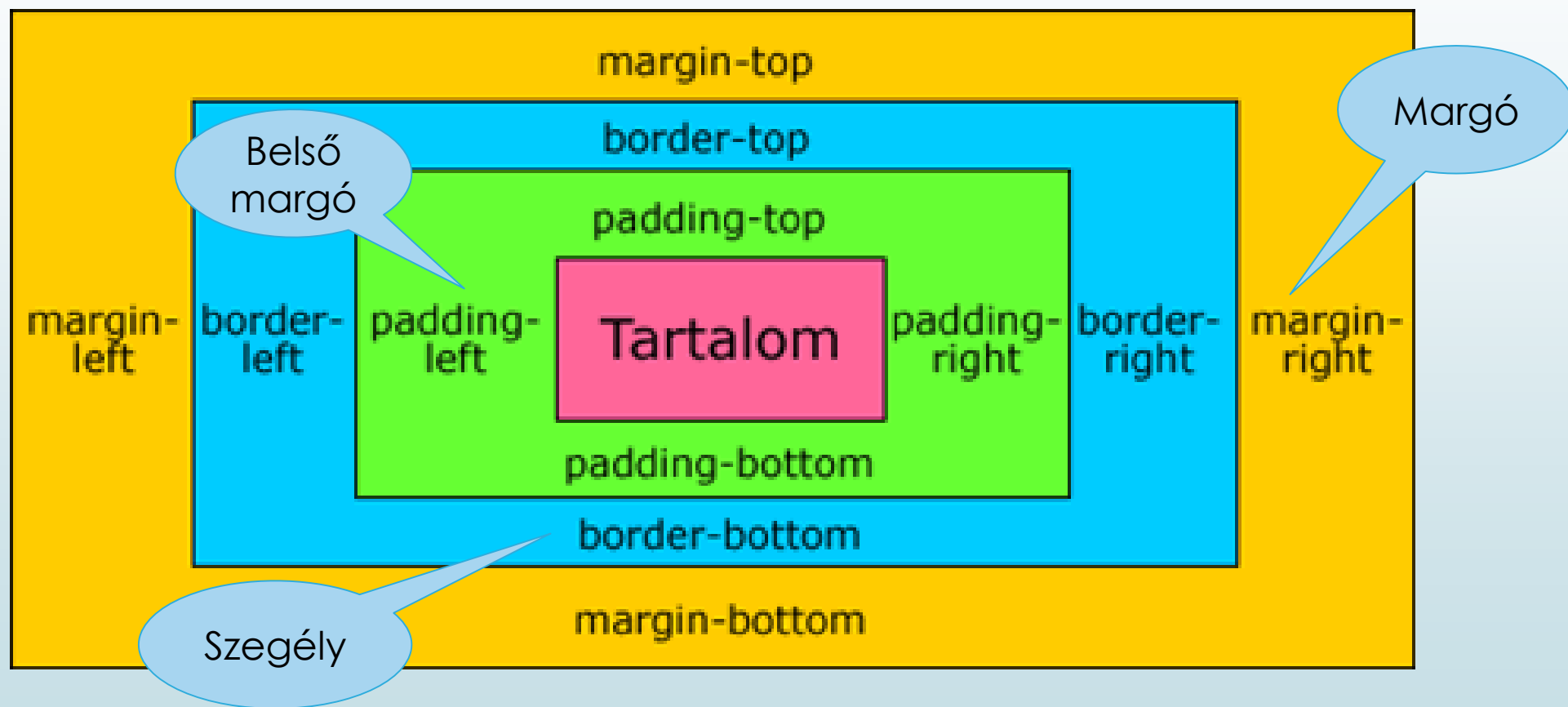
CSS tulajdonság	Lehetséges értékek	Leírás
<b>list-style-type</b>	<ul style="list-style-type: none"><li>• disc (telített kör)</li><li>• circle (üres kör)</li><li>• square (négyzet)</li><li>• lower-alpha (a,b,c...)</li><li>• upper-alpha (A,B,C...)</li><li>• lower-roman (i, ii, iii,...)</li><li>• upper-roman (I, II, III, ...)</li></ul>	listajel típusa
<b>list-style-position</b>	<ul style="list-style-type: none"><li>• inside (a listaelem területén belül van)</li><li>• outside (a listaelem területén kívül helyezkedik el)</li></ul>	a listajel pozíciója
<b>list-style-image</b>	url("kep.png")	a listajel kép és annak az elérési útvonala

# Dobozmodell

- Ezt a modellt használva a weblap minden eleme egy téglalap alakú dobozt formáz.



# Dobozmodell részei



# Dobozmodell szabályok

- A doboz szélességét a jobb és bal oldali margó, szegély és belső margó, valamint a tartalom szélességének összege adja.
- A doboz magasságát a felső és alsó margó, szegély és belső margó, valamint a tartalom magasságának összege adja.
- Tartalom lehet szöveg, kép, link, stb.

# Külső margó

CSS tulajdonság	Lehetséges értékek	Leírás
<b>margin</b>	szám, százalék	margó

- A margin egy gyűjtő tulajdonság, mert egyszerre meg lehet adni az az adott elemre vonatkozó margó beállításokat:

margin: felső jobb alsó bal; → margin: 10px 20px 10px 20px;

margin: alsó-felső bal-jobb; → margin: 10px 20px;

## Külső margó részei

CSS tulajdonság	Lehetséges értékek	Leírás
<b>margin-top</b>	szám, százalék	felső margó
<b>margin-right</b>	szám, százalék	jobb oldali margó
<b>margin-bottom</b>	szám, százalék	alsó margó
<b>margin-left</b>	szám, százalék	bal oldali margó



# Belső margó

CSS tulajdonság	Lehetséges értékek	Leírás
<b>padding</b>	szám, százalék	belső margó

- A padding egy gyűjtő tulajdonság, mert egyszerre meg lehet adni az az adott elemre vonatkozó belső margó beállításokat:

padding: felső jobb alsó bal; → padding: 10px 20px 10px 20px;

padding: alsó-felső bal-jobb; → padding: 10px 20px;

## Belső margó részei

CSS tulajdonság	Lehetséges értékek	Leírás
<b>padding-top</b>	szám, százalék	felső belső margó
<b>padding-right</b>	szám, százalék	jobb oldali belső margó
<b>padding-bottom</b>	szám, százalék	alsó belső margó
<b>padding-left</b>	szám, százalék	bal oldali belső margó

# Szegély

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border</b>	vastagság stílus szín	szegély

- A border egy gyűjtő tulajdonság, mert egyszerre meg lehet adni az az adott elemre vonatkozó szegély beállításokat:

border: vastagság stílus szín; → border: 2px solid blue;

## Szegély részei

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border-top</b>	vastagság stílus szín	felső szegély
<b>border-right</b>	vastagság stílus szín	jobb oldali szegély
<b>border-bottom</b>	vastagság stílus szín	alsó szegély
<b>border-left</b>	vastagság stílus szín	bal oldali szegély

# Szegély vastagsága

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border-width</b>	<ul style="list-style-type: none"><li>• thin (vékony)</li><li>• medium (közepes)</li><li>• thick (vastag)</li><li>• szám</li><li>• százalék</li></ul>	szegélyvastagság
<b>border-top-width</b>		felső szegély vastagsága
<b>border-right-width</b>		jobb oldali szegély vastagsága
<b>border-bottom-width</b>		alsó szegély vastagsága
<b>border-left-width</b>		bal oldali szegély vastagsága

# Szegély stílus

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border-style</b>	<ul style="list-style-type: none"><li>• none (<i>nincs</i>)</li><li>• hidden (<i>rejtett</i>)</li><li>• dotted (<i>pontozott vonal</i>)</li><li>• dashed (<i>szaggatott vonal</i>)</li><li>• solid (<i>folytonos vonal</i>)</li><li>• double (<i>dupla vonal</i>)</li><li>• groove (<i>bemélyített</i>)</li><li>• ridge (<i>kidomborodó</i>)</li><li>• inset (<i>süllyesztett</i>)</li><li>• outset (<i>kiemelkedő</i>)</li></ul>	szegély stílus
<b>border-top-style</b>		felső szegély stílusa
<b>border-right-style</b>		jobb oldali szegély stílusa
<b>border-bottom-style</b>		alsó szegély stílusa
<b>border-left-style</b>		bal oldali szegély stílusa

## Szegély színe

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border-color</b>	szín	szegély színe
<b>border-top-color</b>		felső szegély színe
<b>border-right-color</b>		jobb oldali szegély színe
<b>border-bottom-color</b>		alsó szegély színe
<b>border-left-color</b>		bal oldali szegély színe

# Táblázathoz kapcsolódó plusz tulajdonságok 1.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>caption-side</b>	<ul style="list-style-type: none"><li>• top (<i>felülre</i>)</li><li>• bottom (<i>alulra</i>)</li></ul>	a tábla felirat elhelyezése
<b>vertical-align</b>	<ul style="list-style-type: none"><li>• baseline (<i>alapvonal</i>)</li><li>• top (<i>felülre</i>)</li><li>• middle (<i>középre</i>)</li><li>• bottom (<i>alulra</i>)</li><li>• ...</li></ul>	cella tartalom függőleges igazítása



## Táblázathoz kapcsolódó plusz tulajdonságok 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>border-collapse</b>	<ul style="list-style-type: none"><li>• <i>separate</i> (a szomszédos cellák keretei elkülönülnek)</li><li>• <i>collapse</i> (a szomszédos cellák keretei összevonásra kerülnek)</li></ul>	keretösszevonás
<b>border-spacing</b>	vszám fszám	cella távolságát határozza meg

# Pszeudo osztályok 1.

CSS tulajdonság	Használata	Leírás
<b>:link</b>	a:link	Alapértelmezett linkre vonatkozó beállításokat adhatjuk meg
<b>:visited</b> (felkeresve)	a:visited	A már meglátogatott (felkeresett) linkre vonatkozó beállításokat adhatjuk meg
<b>:hover</b> (fölötte állva)	a:hover	Amikor a kurzorral a link fölé megyünk, erre az állapotra vonatkozó tulajdonságokat állíthatjuk be
<b>:active</b> (használatban)	a:active	Használatban lévő (amikor éppen rá kattintunk) linknek a tulajdonságait állíthatjuk be.

## Pszeudo osztályok 2.

- Más tag-eknél is használhatóak
  - Pl.: **tr:hover{}**
  - A táblázat sorára vonatkozó beállításokat határozhatom meg, amikor fölötte állok
- A linkre vonatkozó stílusok sorrendje: **LVHA**, azaz
  - a:link{}
  - a:visited{}
  - a:hover{}
  - a:active{}

# Egy elem átlátszósága/láthatósága

CSS tulajdonság	Lehetséges értékek	Leírás
<b>opacity</b>	0.0 és 1.0 közötti értéket vehet fel	Az adott elem láthatóságának meghatározása. 0.0 átlátszó 1.0 teljesen látható az elem
<b>visibility</b>	<ul style="list-style-type: none"><li>• visible (<i>látható</i>)</li><li>• hidden (<i>rejtett</i>)</li><li>• collapse (<i>csak tábla elemeknél használható eltávolítja az adott sort, vagy oszlopot</i>)</li></ul>	Az adott elem láthatóságát határozza meg.

# Túlcsordulás 1.

- Előfordulhat, hogy az adott elem/doboz tartalma nagyobb, mint amekkora helyet szánunk neki
  - A tároló elem méretét mi határoztuk meg például, így a tartalom „túlcsordulhat”.
- Ezt a problémát az overflow tulajdonsággal küszöbölhetjük ki.

## Túlcsordulás 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>overflow</b>	<ul style="list-style-type: none"><li>• visible (<i>látható</i>)</li><li>• hidden (<i>rejtett</i>)</li><li>• scroll (<i>görgethető</i>)</li><li>• auto (<i>ha nincs elég hely, akkor automatikusan scroll állapotú lesz</i>)</li><li>• initial (<i>alapértelmezett értéket hajtja végre</i>)</li></ul>	Az adott tároló elem mérete kisebb, mint az elem tartalma.

# Megjelenítés 1.

- Minden HTML elemhez tartozik egy megjelenési forma
  - A bekezdés (p) és címsorok (h1-h6) blokk szintű elemek
  - Az img sorfolytonos (inline) elemnek számít
- Ezeknek az elemeknek, ha akarjuk megváltoztathatjuk a display tulajdonsággal a megjelenítésének módját

## Megjelenítés 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>display</b>	<ul style="list-style-type: none"><li>• inline (<i>alapértelmezett, soron belüli megjelenítés, mint pl. span, vagy img</i>)</li><li>• blokk (<i>blokk szintű elemként viselkedik azaz új sorban jelenik meg, mint pl. p</i>)</li><li>• none (<i>nem jelenik meg</i>)</li><li>• list-item (<i>lista elemként viselkedik, mint pl. li</i>)</li></ul>	Az elem megjelenítési módjának meghatározása.



# Helyzetmegadás, pozícionálás

- Ötféle helyzetmegadást különböztetünk meg: abszolút, viszonyított, rögzített és statikus, sticky.
- A böngésző vízszintesen balról jobbra, függőlegesen pedig fentről lefele olvassa a dokumentumot → Ezt szövegfolyamnak hívjuk.
- Ebből a normál szövegfolyamból a **position** (helyzet) és a **float** (úsztatás) CSS tulajdonsággal lehet kivenni egy-egy HTML elemet.

# Abszolút helyzetmegadás

- Azt az elemet, amelynek a helyzetét abszolút helyzetmegadással adtuk meg, eltávolítjuk a szövegfolyamból és az adott elem tárolótömbjéhez (pl. maga a HTML elem) képest állítjuk be a helyzetét.
- `positon: absolute;`

## Viszonyított/relatív helyzetmegadás

- Azt az elemet, amelynek a helyzetét relatív helyzetmegadással adtuk meg, NEM távolítjuk el a szövegfolyamból és az elem adott pozíciójához képesti elmozdulást adhatjuk meg.
- `positon: relative;`

## Rögzített helyzetmegadás

- Ha egy elem helyzetét rögzített helyzetmegadással adtuk meg, akkor rögzítjük az adott elemet egy pontba, és ott is maradnak, még akkor is, ha a weblapot lejjebb görgetjük.
- Ebben az esetben az elem a felhasználó által látható részhez (vagy a böngésző ablakhoz) képest rögzített.
- `position: fixed;`

# Statikus helyzetmegadás

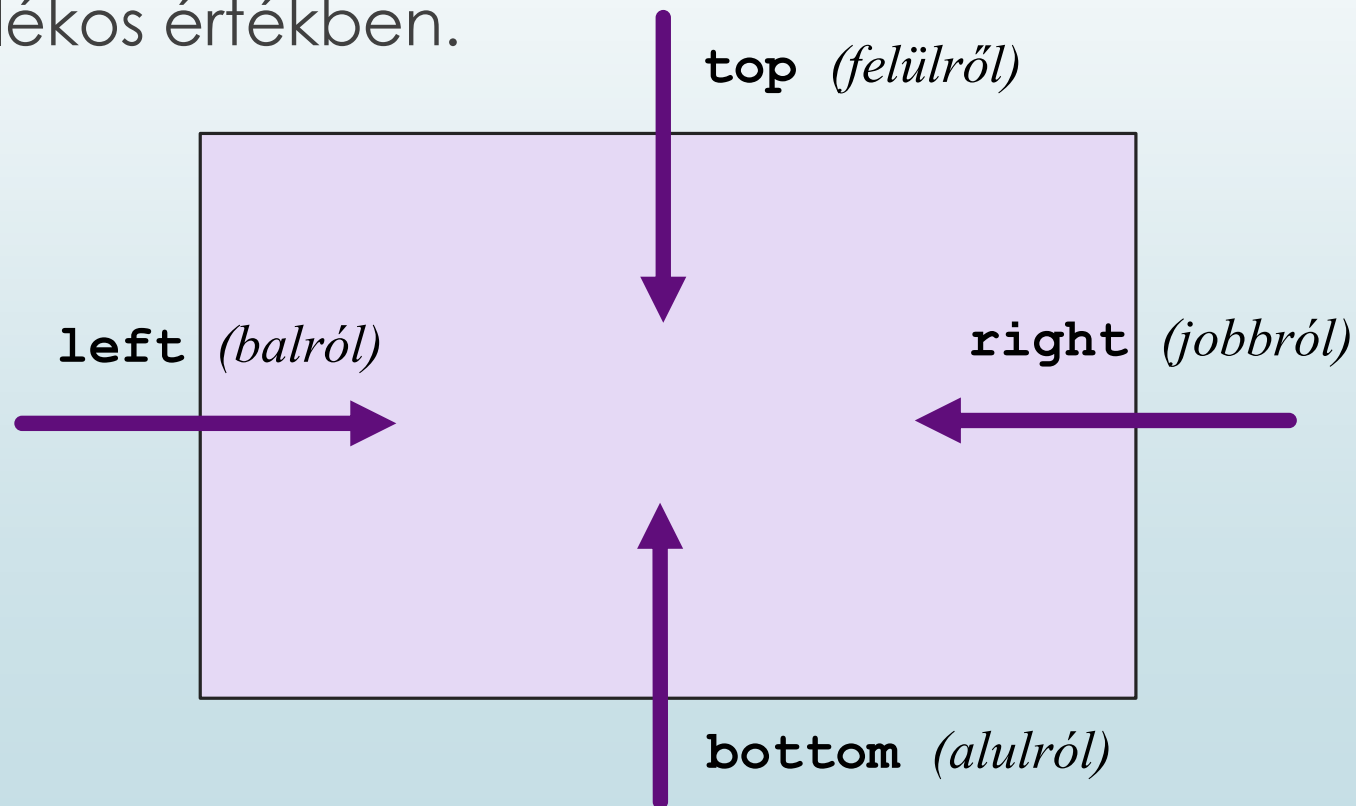
- Alapértelmezett helyzetmegadás.
- Akkor van szükségünk erre a beállításra, ha egy olyan elem, amelyet a szövegfolyamban elfoglalt helyén szeretnénk hagyni, de valamilyen másfajta helyzetmegadást örököl attól az elemtől, amely magába foglalja.
- `positon: static;`

## Sticky helyzetmegadás

- A relatív és a rögzített helyzetmegadás kombinációja! Az elem relatív helyzetben van, amíg el nem éri az adott pozíciót.
- Ahogy eléri a pozícióhoz ragad az elem, azaz rögzített helyzetbe kerül.
- `position: sticky;`

# Pozíciók meghatározása a helyzetmegadáshoz 1.

- Négy irányból adhatjuk meg a pozíciókat pixel, vagy százalékos értékben.



## Pozíciók meghatározása a helyzetmegadáshoz 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>top</b>	szám, százalék	függőleges pozíció megadása felülről
<b>bottom</b>	szám, százalék	függőleges pozíció megadása alulról
<b>left</b>	szám, százalék	vízszintes pozíció megadása balról
<b>right</b>	szám, százalék	vízszintes pozíció megadása jobbról

- Általában egyszerre egy vízszintes és egy függőleges pozícióval szoktuk meghatározni az elem konkrét pozícióját.



# Úsztatás 1.

- Legtöbb esetben képeket szoktunk úsztatni
  - Például amikor azt szeretnénk elérni, hogy a képet vegye körbe a szöveg.
- Természetesen más elemet is tudunk úsztatni az oldalon belül
  - Például szakaszt, listaelemeket, stb.

## Úsztatás 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>float</b>	<ul style="list-style-type: none"><li>• none (nincs)</li><li>• left (balra)</li><li>• right (jobbra)</li></ul>	elem úsztatása

# Mezőkiürítés 1.

- Megakadályozza, hogy egy elem bal, jobb, vagy mindkét oldalán legyen egy olyan elem, amire be van állítva a float tulajdonság.
- Ezzel letiltjuk, hogy adott irányba lebegő elem legyen a clear tulajdonsággal rendelkező elem körül.

## Mezőkiürítés 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>clear</b>	<ul style="list-style-type: none"><li>• left (<i>balra</i>)</li><li>• right (<i>jobbra</i>)</li><li>• both (<i>mindkét irányba</i>)</li></ul>	mezőkiürítés

# Tartalom elrendezése (z-index) 1.

- A z-index felel az elem z-tengelyen való elhelyezkedéséért, azaz hogy milyen mélységben van az adott elem a térben (3D).
- 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 van a felhasználóhoz, vagyis feljebb van a z tengelyen.

## Tartalom elrendezése (z-index) 2.

CSS tulajdonság	Lehetséges értékek	Leírás
<b>z-index</b>	szám	elem elrendezése

- Alapértelmezett értéke 0.
- Ha két elem azonos mélységgel rendelkezik, akkor a HTML-ben a később leírt fog "feljebb" megjelenni.
- Ha más értéket adunk meg, a nagyobb számmal rendelkező elem kerül közelebb a felhasználóhoz.