

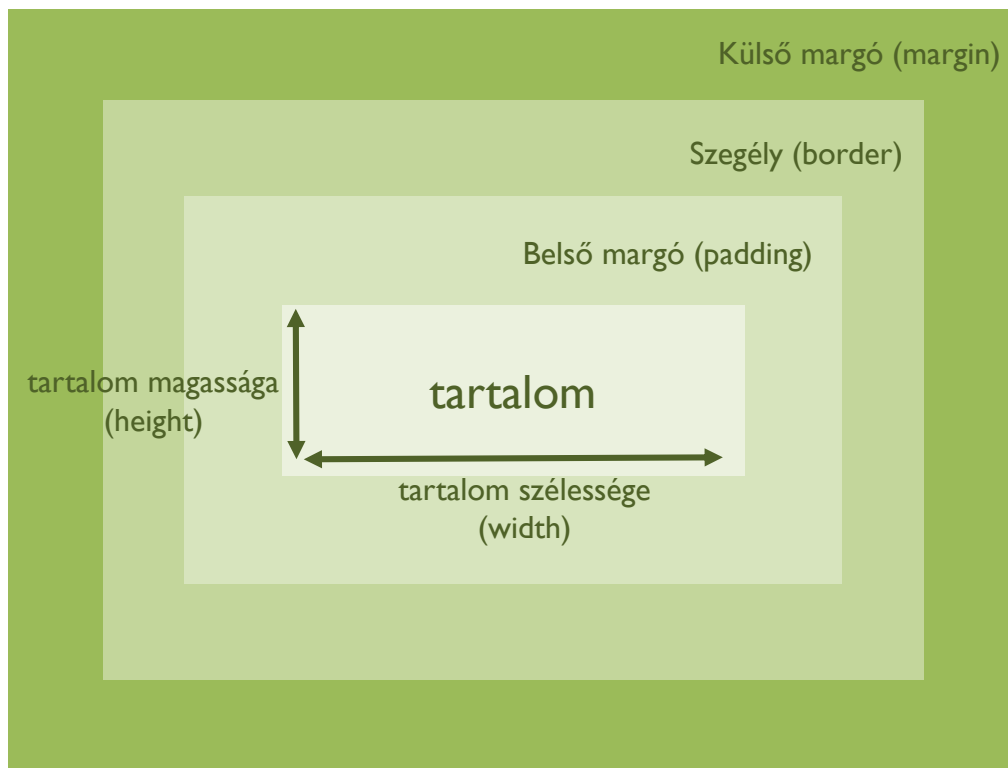


CSS - DOBOZMODELL

KÉSZÍTETTE: OLÁH KATALIN

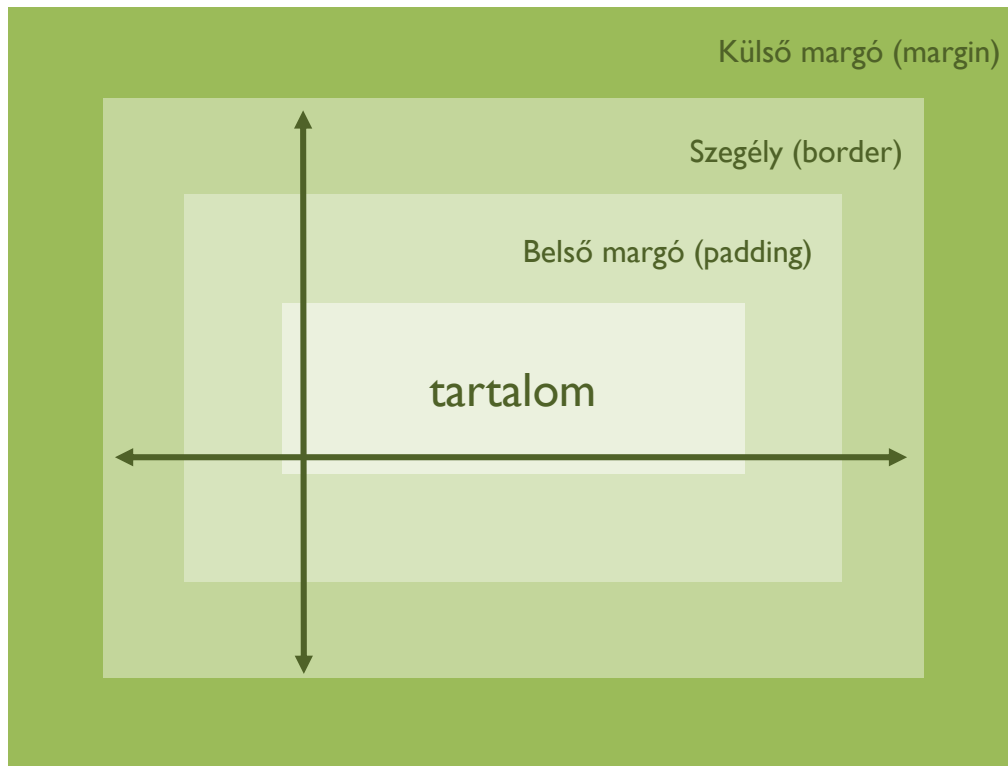


CSS DOBOZMODELL



- A böngészőben megjelenő elemek dobozokban helyezkednek el.
- **CSS Dobozmodell** (*belülről kifelé haladva*):
 1. **Tartalom:** ez tetszőleges elem lehet, pl.: bekezdés, kép, táblázat, link, stb.
 2. **Belső margó:** kitöltésnek is szokták hívni. A szegély és a tartalom közötti távolságot értjük alatta.
 3. **Szegély:** a belső margó határán helyezkedik el.
 4. **Külső margó:** szegélyen kívül helyezkedik el, az elemek eltartását, távolságot jelenti.

CSS DOBOZMODELL – AZ ELEM TÉNYLEGES MÉRETE



- **Szélesség** = doboz szélessége + bal és jobb oldali belső margó + bal és jobb oldali szegély vastagsága.
- **Magasság** = doboz magassága + alsó és felső belső margó + alsó és felső szegély vastagsága.
- Ezt a számítási módot CSS-ben lehet módosítani (ld. [box-sizing](#)).
- Ha az elem elhelyezkedését vizsgáljuk, azaz, hogy mekkora helyet foglal el, akkor a külső margók is számítanak.



ELEMEK MÉRETEZÉSE

SZÉLESSÉG - WIDTH

`width: érték;`

- Az adott elem szélességét a **width** CSS tulajdonsággal állíthatod be.
- Az érték lehet:
 - Számérték: **20px**
 - Százalék: **50%**
A szülő elemhez képest változik.
 - Arányok: **1.2em**, vagy **2.5rem**

```
img{  
  width: 200px;  
}
```

Eredmény:



200px

A magasság méretarányosan változik a szélesség változásával.

MAGASSÁG - HEIGHT

```
height: érték;
```

- Az adott elem magasságát a **height** CSS tulajdonsággal állíthatod be.
- Az érték lehet:
 - Számérték: **20px**
 - Százalék: **50%**
A szülő elemhez képest változik.
 - Arányok: **1.2em**, vagy **2.5rem**

```
img{  
    height: 200px;  
}
```

A szélesség méretarányosan változik a magasság változásával.

Eredmény:



MINIMÁLIS/MAXIMÁLIS SZÉLESSÉG

`min-width: érték;`

és

`max-width: érték;`

- Az elemeknek a szélességen kívül meghatározhatjuk a minimális (**min-width**) és maximális szélességét (**max-width**) is.

- Az érték lehet:

- Számérték: **20px**
- Százalék: **50%**
A szülő elemhez képest változik.
- Arányok: **1.2em**, vagy **2.5rem**

```
img{  
  width: 80%;  
  min-width: 200px;  
}
```

A kép szélessége a szülő elem 80%-a lesz, de nem lehet kisebb 200 pixelnél.

A mintán a div tárolóelem szélessége 210 pixel. Annak a 80%-a 168 pixel, de a kép 200 pixel nagyságú maradt, mert be van állítva a minimum szélessége.

Eredmény:

A dobozban van egy cica.



MAGASSÁG - HEIGHT

```
height: érték;
```

- Az adott elem magasságát a **height** CSS tulajdonsággal állíthatod be.
- Az érték lehet:
 - Számérték: **20px**
 - Százalék: **50%**
A szülő elemhez képest változik.
 - Arányok: **1.2em**, vagy **2.5rem**

```
img{  
    height: 200px;  
}
```

A szélesség méretarányosan változik a magasság változásával.

Eredmény:



MINIMÁLIS/MAXIMÁLIS MAGASSÁG

`min-height: érték;`

és

`max-height: érték;`

- Az elemeknek a magasságán kívül meghatározhatjuk a minimális (**min-height**) és maximális magasságát (**max-height**) is.
- Az érték lehet:
 - Számérték: **20px**
 - Százalék: **50%**
A szülő elemhez képest változik.
 - Arányok: **1.2em**, vagy **2.5rem**

```
img{  
  height: 70%;  
  max-height: 150px;  
}
```

A kép magassága a szülő elem 70%-a lesz, de nem lehet magasabb 150 pixelnél.

A mintán a div tárolóelem magassága 300 pixel. Annak a 70%-a 210 pixel, de a kép 150 pixel magas maradt, mert be van állítva a maximális magassága.

Eredmény:

A dobozban van egy cica.

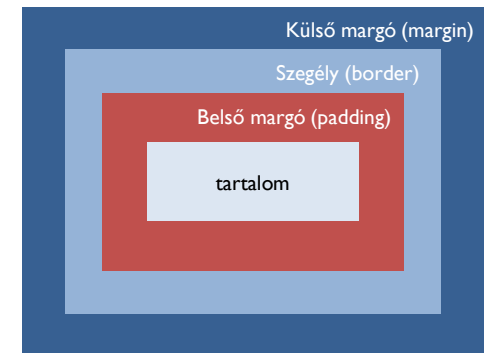




BELSŐ MARGÓ BEÁLLÍTÁSAI

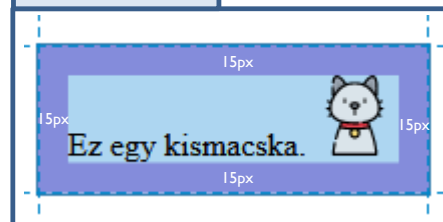
BELSŐ MARGÓ - PADDING

- A belső margó a szegély és a tartalom közötti távolságot jelenti.
- A **padding** CSS tulajdonsággal tudod beállítani.
- Az értéke lehet abszolút (**px**)/relatív (**em**, **rem**, **%**) mértékegységű **számérték**.



```
p{  
  padding: 15px;  
}
```

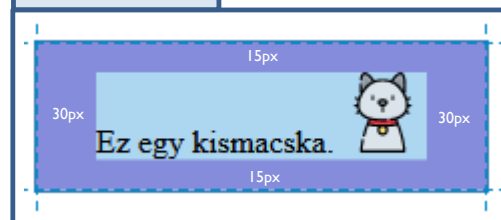
Eredmény:



Egy paraméter esetén mind a négy oldalon ugyanakkora belső margót állíthatsz be.

```
p{  
  padding: 15px 30px;  
}
```

Eredmény:



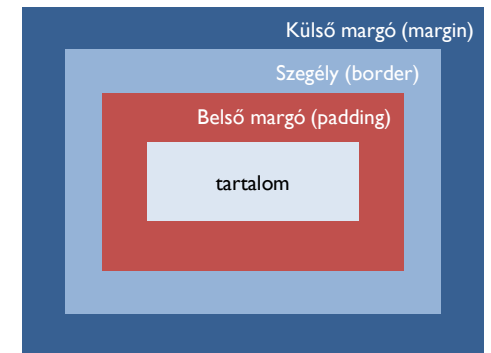
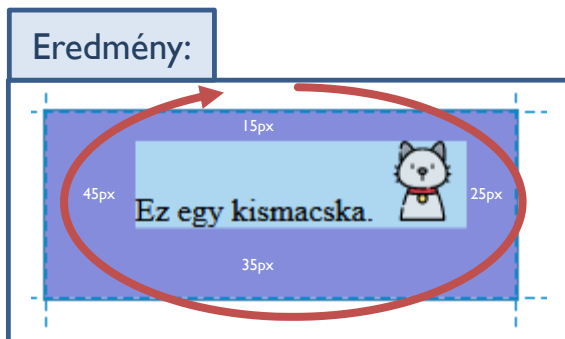
Két paraméter esetén:

1. paraméterrel a függőleges irányú belső margót tudod beállítani.
2. paraméterrel a vízszintes irányú belső margót tudod beállítani.

BELSŐ MARGÓ - PADDING

- Négy paraméter esetén a felső belső margótól kezdve az óramutató járásával megegyező irányba állíthatod be az egyes oldalak belső margóját.

```
p{  
  padding:15px 25px 35px 45px;  
}
```

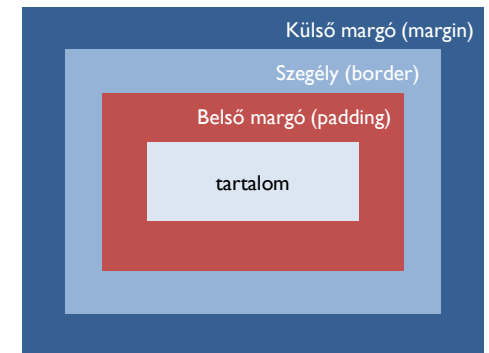


Négy paraméter esetén:

1. paraméter: felső belső margó
2. paraméter: jobb oldali belső margó
3. paraméter: alsó belső margó
4. paraméter: bal oldali belső margó

BELSŐ MARGÓ – IRÁNY SZERINTI BONTÁSA

- A **padding** CSS tulajdonságot a négy irány szerint tovább bontható.



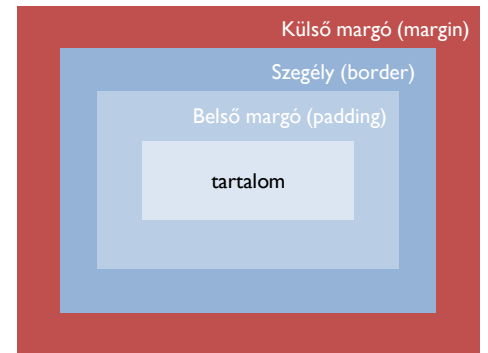
Lehetséges értékei	Leírás
<code>padding-top: 10px;</code>	Felső belső margó
<code>padding-right: 10px;</code>	Jobb oldali belső margó
<code>padding-bottom: 10px;</code>	Alsó belső margó
<code>padding-left: 10px;</code>	Bal oldali belső margó



KÜLSŐ MARGÓ BEÁLLÍTÁSAI

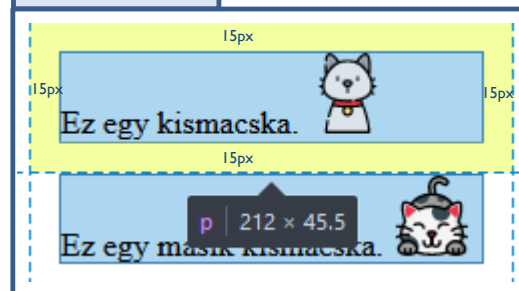
KÜLSŐ MARGÓ - MARGIN

- A külső margó a szegélyen kívül helyezkedik el, az elemek eltartását, távolságát jelenti.
- A **margin** CSS tulajdonsággal tudod beállítani.
- Az értéke lehet abszolút (**px**)/relatív (**em**, **rem**, **%**) mértékegységű **számérték**.



```
p{
  margin:15px;
}
```

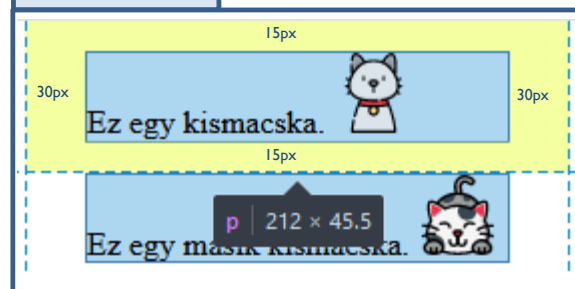
Eredmény:



Egy paraméter esetén mind a négy oldalon ugyanakkora külső margót állíthatsz be.

```
p{
  margin:15px 30px;
}
```

Eredmény:



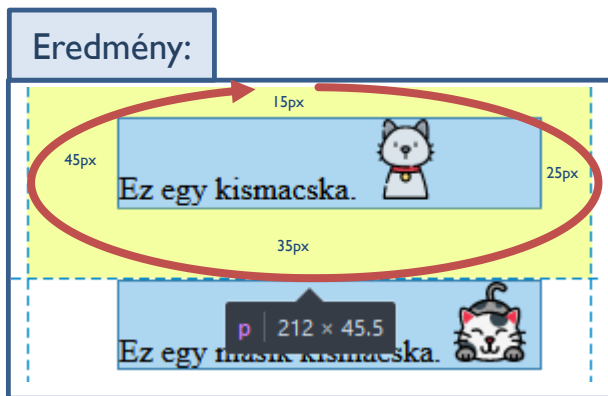
Két paraméter esetén:

1. paraméterrel a függőleges irányú külső margót tudod beállítani.
2. paraméterrel a vízszintes irányú külső margót tudod beállítani.

KÜLSŐ MARGÓ - MARGIN

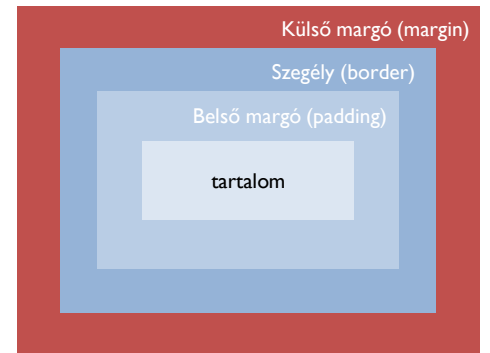
- Négy paraméter esetén a felső külső margótól kezdve az óramutató járásával megegyező irányba állíthatod be az egyes oldalak külső margóját.

```
p{  
  margin:15px 25px 35px 45px;  
}
```



Négy paraméter esetén:

1. paraméter: felső külső margó
2. paraméter: jobb oldali külső margó
3. paraméter: alsó külső margó
4. paraméter: bal oldali külső margó

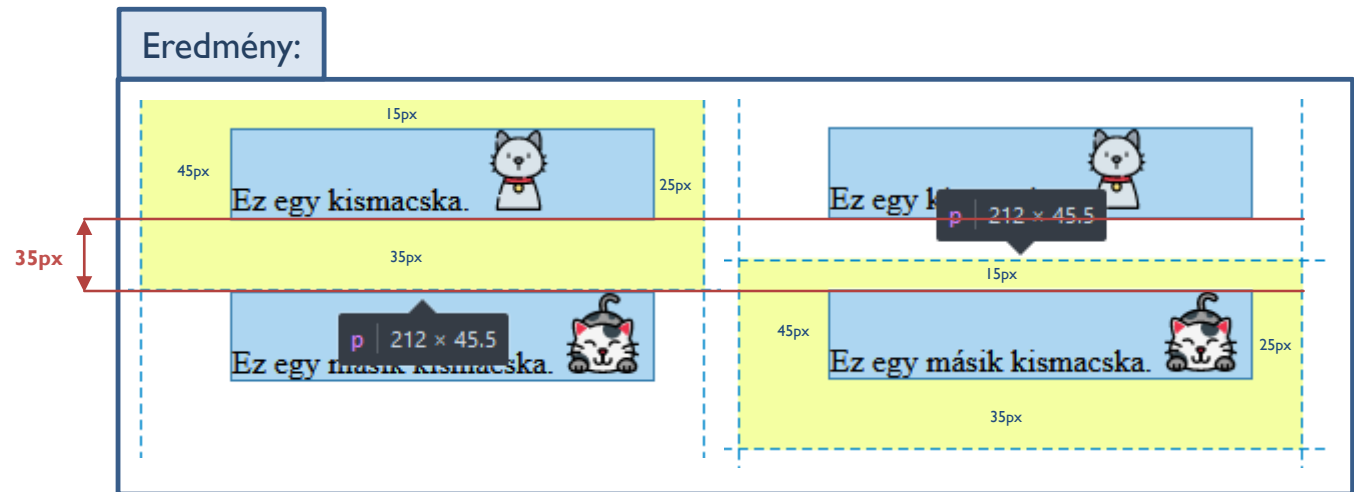


ALSÓ ÉS FELSŐ KÜLSŐ MARGÓK MÉRTÉKÉNEK ELTÉRÉSE

- Szabvány szerinti szabály:
 - Ha az elemek alsó és felső külső margójának mértéke eltér, akkor az értékek közül a **nagyobbik érték fog érvényesülni** a böngészőben.
 - Azaz a böngésző csak a nagyobb külső margó beállítást fogja érvényesíteni.

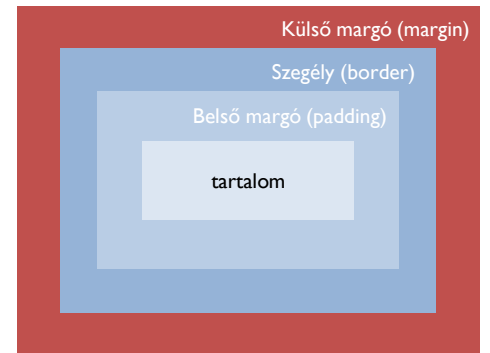


```
p{
  margin: 15px 25px 35px 45px;
}
```



KÜLSŐ MARGÓ – IRÁNY SZERINTI BONTÁSA

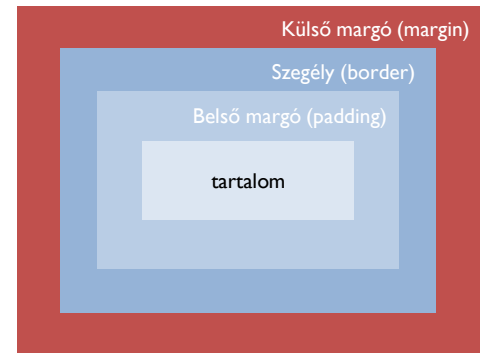
- A **margin** CSS tulajdonságot a négy irány szerint tovább bontható.



Lehetséges értékei	Leírás
<code>margin-top: 10px;</code>	Felső külső margó
<code>margin-right: 10px;</code>	Jobb oldali külső margó
<code>margin-bottom: 10px;</code>	Alsó külső margó
<code>margin-left: 10px;</code>	Bal oldali külső margó

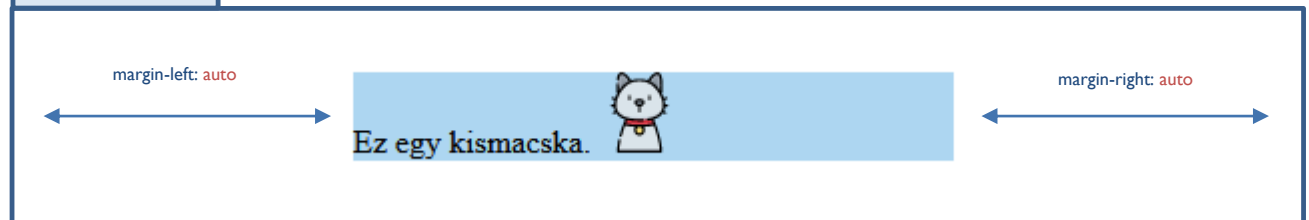
KÖZÉPRE IGAZÍTÁS KÜLSŐ MARGÓ SEGÍTSÉGÉVEL

- A blokk szintű elemeket automatikus külső margó segítségével lehet vízszintesen középre igazítani.
 - Ebben az esetben a bal és jobb oldali külső margó értékét **auto** –ra kell állítanod.



```
p{  
  background-color: rgb(173, 214, 241);  
  width: 300px;  
  margin-left:auto;  
  margin-right:auto;  
}
```

Eredmény:



- A blokk szintű elem középre igazítása nem egyezik meg a szöveg igazítással (text-align)!
- Ahogy mintán is láthatod a bekezdés, mint blokk szintű elem került középre az oldalon, de a szöveg a dobozon belül balra igazított maradt.



SZEGÉLY BEÁLLÍTÁSAI

SZEGÉLY BEÁLLÍTÁSA – BORDER – SHORTHAND PARANCS

```
border: border-width border-style border-color;
```

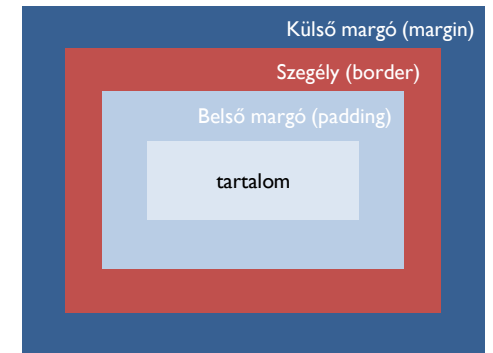
- A szegély a belső margó határán helyezkedik el.
- A szegély beállítását a **border** rövid paranccsal adhatod meg.
- A megadási sorrend számít:
 1. paraméter a **border-width**, a szegély vastagsága
 2. paraméter a **border-style**, a szegély stílusa

A leggyakrabban használt szegély stílusok:

- *none* – nincs szegély
 - *solid* – folytonos szegély
 - *dotted* – pontozott szegély
 - *dashed* – szaggatott szegély
 - *double* – dupla szegély
3. paraméter a **border-color**, a szegély színe

3D hatású szegélyek:

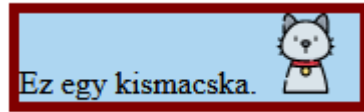
- *groove*
- *ridge*
- *inset*
- *outset*



SZEGÉLY BEÁLLÍTÁSA – BORDER - PÉLDA

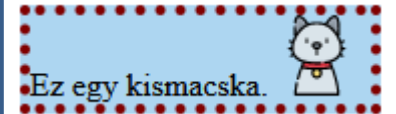
```
p{  
  border: 5px solid maroon;  
}
```

Eredmény:



```
p{  
  border: 5px dotted maroon;  
}
```

Eredmény:



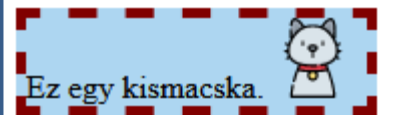
```
p{  
  border: 5px double maroon;  
}
```

Eredmény:



```
p{  
  border: 5px dashed maroon;  
}
```

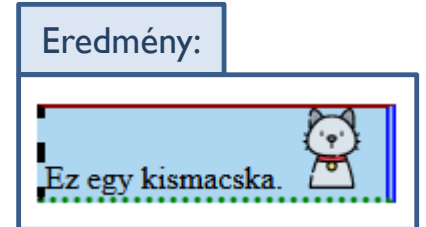
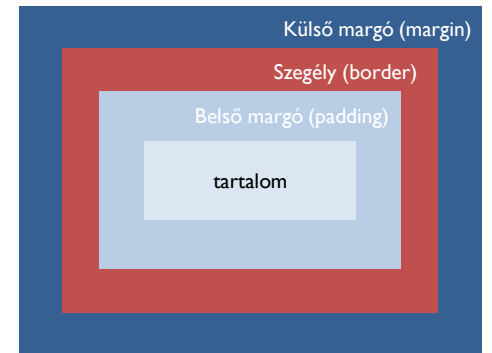
Eredmény:



SZEGÉLY – IRÁNY SZERINTI BONTÁSA

- A **border** CSS tulajdonságot a négy irány szerint tovább bontható.

Lehetséges értékei		Leírás
<code>border-top:</code>	<code>2px solid darkred;</code>	Felső szegély
<code>border-right:</code>	<code>5px double blue;</code>	Jobb oldali szegély
<code>border-bottom:</code>	<code>3px dotted green;</code>	Alsó szegély
<code>border-left:</code>	<code>4px dashed black;</code>	Bal oldali szegély

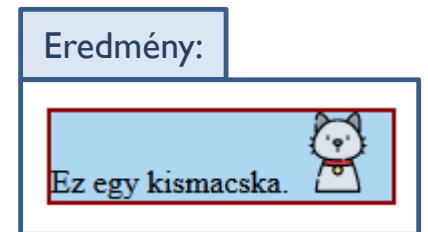
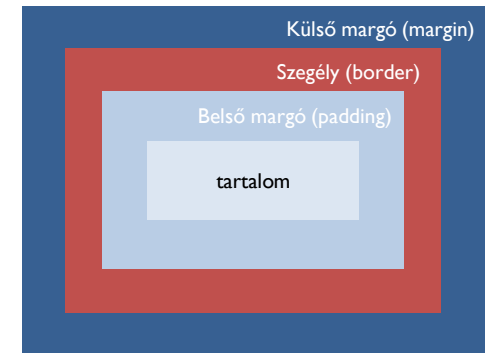


SZEGÉLY – TULAJDONSÁG SZERINTI BONTÁSA

- A **border** CSS tulajdonságot három tulajdonság szerint tovább lehet bontani.

Lehetséges értékei	Leírás
<code>border-width: 2px;</code>	Szegély vastagsága
<code>border-style: solid;</code>	Szegély stílusa
<code>border-color: darkred;</code>	Szegély színe

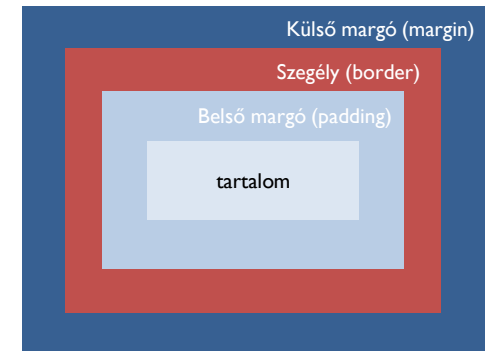
- **!!** Ha a szegély stílusát nem határozzod meg, akkor nem jelenik meg a szegély.



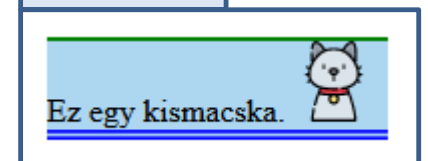
SZEGÉLY – IRÁNY ÉS TULAJDONSÁG SZERINTI BONTÁS KOMBINÁLÁSA

`border-irány-tulajdonság: érték;`

Lehetséges értékei	Leírás
<code>border-top-width: 2px;</code>	Felső szegély vastagsága 2 pixel
<code>border-top-style: solid;</code>	Felső folytonos szegély
<code>border-top-color: green;</code>	Felső szegély színe zöld
<code>border-bottom-width: 5px;</code>	Alsó szegély vastagsága 5 pixel
<code>border-bottom-style: double;</code>	Alsó szegély stílusa dupla vonalú
<code>border-bottom-color: blue;</code>	Alsó szegély színe kék
...	



Eredmény:





LEKEREKÍTÉS

LEKEREKÍTÉS – BORDER-RADIUS

```
border-radius: érték;
```

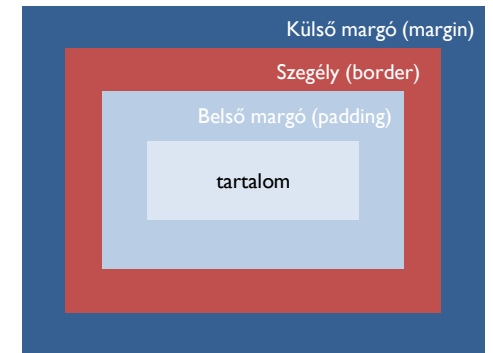
- Az elemek sarkait a **border-radius** CSS tulajdonsággal tudod lekerekíteni.
- Az érték lehet:
- Az érték lehet:
 - Számérték: **20px**
 - Százalék: **50%**
 - Arányok: **1.2em**, vagy **2.5rem**

```
img{  
    border-radius: 25px;  
}
```

Eredmény:



Egy paraméter esetén mind a négy sarok lekerekített lesz az adott mértékkel.

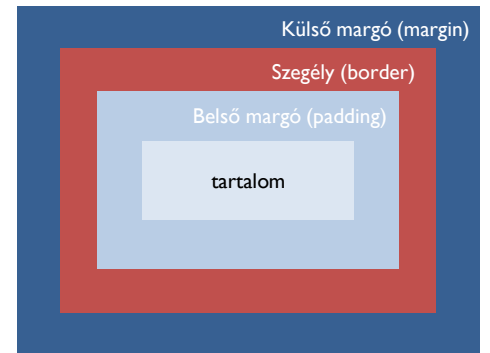


LEKEREKÍTÉS – BORDER-RADIUS

- Ha a **border-radius** két paraméteres, akkor
 1. **paraméter** a bal felső és a jobb alsó sarok lekerekítését jelenti.
 2. **paraméter** a jobb felső és a bal alsó sarok lekerekítését jelenti.

```
img{  
    border-radius:15px 50%;  
}
```

Eredmény:

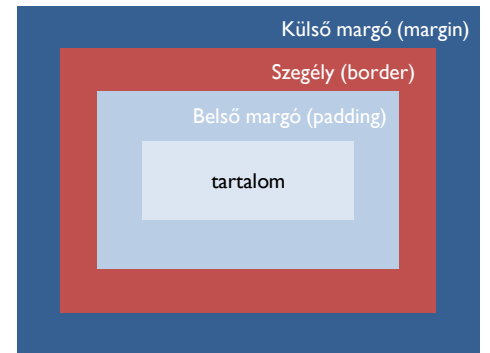


LEKEREKÍTÉS – BORDER-RADIUS

- Ha a **border-radius** négy paraméteres, akkor
 1. **paraméter**: a bal felső sarok lekerekítése
 2. **paraméter**: a jobb felső sarok lekerekítése
 3. **paraméter**: a jobb alsó sarok lekerekítése
 4. **paraméter**: a bal alsó sarok lekerekítése

```
img{  
  border-radius: 80px 20px 60px 40px;  
}
```

Eredmény:



LEKEREKÍTÉS – BORDER-RADIUS – TOVÁBBI PÉLDÁK

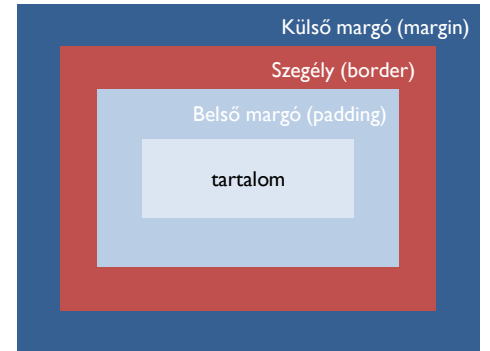
```
img{  
  border-radius: 50%;  
}
```

Eredmény:



```
img{  
  border-radius: 0px 50%;  
  border: 3px solid #68acdd;  
}
```

Eredmény:





DOBOZ MÉRETEZÉSÉNEK MÓDOSÍTÁSA

EGY KIS ISMÉTLÉS

- A mintán láthatsz egy div tárolóelemet, amibe elhelyeztünk egy bekezdést és egy képet.
- A kép szélességét 100%-osra állítottuk, így a kép kitölti teljes szélességbe a tárolóelemet (azaz a szülőelemét).
- **Kérdés:**
 - *Mi történik akkor, ha képre ráhelyezünk egy 10 pixeles vastagságú szegélyt és 5 pixel nagyságú belső margót?*

Eredmény:



HTML

```
<div>
  <p>A dobozban van egy cica.</p>
  
</div>
```

CSS

```
div{
  width: 250px;
  height: 250px;
  background-color: lightblue;
  border: 3px solid maroon;
}
img{
  width: 100%;
}
```


EGY KIS ISMÉTLÉS

- A képen azt láthatod, hogy a kép kilóg a tárolóelemből.
- Ez azért fordulhat elő, mert az alapértelmezett CSS dobozmodell szerint **az elem tényleges szélessége** az a tartalom szélessége + bal és jobboldali belső margó + bal és jobboldali szegély.
- A példában:
 - A kép szélessége 100%-os vagyis megegyezik a szülő elem szélességével.
 - Ehhez hozzáadódik még a bal és jobb oldali belső margó, valamint a bal és jobb oldali szegély. Azaz összesen 30 pixelrel lóg ki a tárolóelemből.
- **Kérdés:**
 - Van-e lehetőség rá, hogy az elem tényleges szélességébe számítsdjon bele a belső margó és a szegély mérete?

Eredmény:

A dobozban van egy cica.



HTML

```
<div>
  <p>A dobozban van egy cica.</p>
  
</div>
```

CSS

```
div{
  width: 250px;
  height: 250px;
  background-color: lightblue;
  border: 3px solid maroon;
}
img{
  width: 100%;
  background-color: rgb(129, 197, 89);
  border: 10px solid rgb(13, 100, 172);
  padding: 5px;
}
```

DOBOZ MÉRETÉNEK MÓDOSÍTÁSA – BOX-SIZING

```
box-sizing: border-box;
```

- A **box-sizing: border-box;** CSS tulajdonság értékpárral állíthatod be, hogy az elem szélességébe számítódjon bele a belső margó és a szegély mértéke.
- Így a példán is láthatod, hogy a kép szélessége az 100% maradt, de a plusz belső margó és a szegély mértéke beleszámítódott a kép szélességébe és így nem lóg ki a tároló (szülő) elemből.
- A CSS dobozmodellben alapértelmezett érték a **box-sizing: content-box;**

Eredmény:



HTML

```
<div>
  <p>A dobozban van egy cica.</p>
  
</div>
```

CSS

```
div{
  width: 250px;
  height: 250px;
  background-color: lightblue;
  border: 3px solid maroon;
}
img{
  width: 100%;
  background-color: rgb(129, 197, 89);
  border: 10px solid rgb(13, 100, 172);
  padding: 5px;
  box-sizing: border-box;
}
```