



szit

[< CSS](#)

## CSS3

- **Szerző:** Sallai András
- Copyright © Sallai András, 2011-2020
- Licenc: GNU Free Documentation License 1.3
- Web: <https://szit.hu> [<https://szit.hu>]

### Szegély

```
div {  
  border-radius: 15px;  
  box-shadow: 10px 10px 5px #888888;  
}
```

Egy doboz, aminek van egy kis radius meg egy kis árnyék beállítva

```
div {  
  border-radius: 15px;  
  box-shadow: inset 10px 10px 5px #888888;  
}
```

Egy doboz, aminek van egy kis radius meg egy kis árnyék beállítva

### Szöveg árnyékolása

```
div {  
  text-shadow: vízszintes-eltolás függőleges-eltolás foltosítás szín;  
}
```

Példa:

```
text-shadow: 5px 5px 5px #888888;
```

```
<div>  
Szöveg  
</div>
```

# Szöveg

## Átméretezhető

Szintaxis:

```
resize: none|both|horizontal|vertical:  
  
div {  
    resize:both;  
    overflow:auto;  
}
```

### Átméretezhető

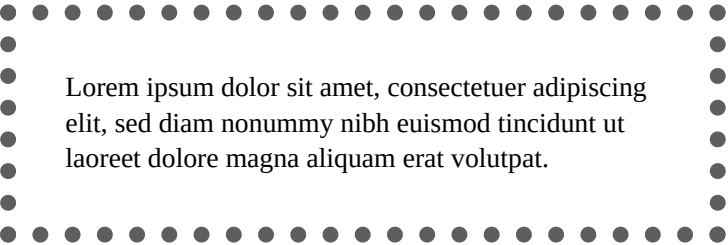
## Háttér nyírása

background-clip tulajdonsággal állítjuk. Lehetséges értékei:

- border-box
- padding-box
- content-box

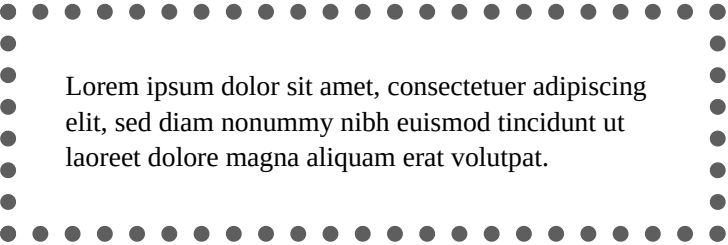
### Border

```
background-color: yellow;  
background-clip: border-box;
```



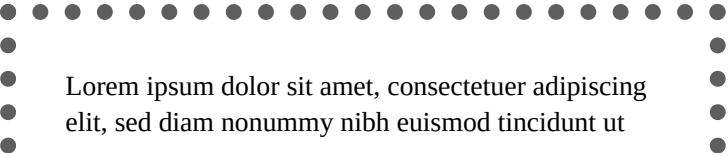
### Padding

```
background-color: yellow;  
background-clip: padding-box;
```



### Content

```
background-color: yellow;  
background-clip: content-box;
```



laoreet dolore magna aliquam erat volutpat.

## Külső font

```
@font-face {
  font-family: linliber;
  src: url('http://szit.hu/download/fontok/LinLibertine_RI_G.ttf');
}
p {
  font-family: linliber;
}
```

Valami

Valami

Lorem ipsum dolor est amet

## Színátmenet

```
background-image: linear-gradient(bottom, rgb(0,30,140) 15%, rgb(72,111, 253) 58%);
background-image: -o-linear-gradient(bottom, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -moz-linear-gradient(bottom, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -webkit-linear-gradient(bottom, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -ms-linear-gradient(bottom, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
```

Lorem ipsum

dolores amet

```
background-image: linear-gradient(left, rgb(0,30,140) 15%, rgb(72,111, 253) 58%);
background-image: -o-linear-gradient(left, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -moz-linear-gradient(left, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -webkit-linear-gradient(left, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
background-image: -ms-linear-gradient(left, rgb(0,30,140) 15%, rgb(72,111,253) 58%);
```

Lorem ipsum

dolores amet

Megadhatunk fokokban kifejezett értéket is, de 0° máshol van liner-gradient és a többi között. A linear-gradient esetén 0deg lentől felfele. A -webkit-linear-gradient és a többi esetén 0deg balról jobbra.

Egyező beállítások:

```
background-image: linear-gradient(0deg, rgb(0,30,140) 15%, rgb(72,111, 253) 58%);
background-image: -webkit-linear-gradient(275deg, rgb(0,30,140) 15%, rgb(72,111, 253) 58%);
```

## Betű szélesség

A választott betűtípusnak támogatnia kell.

```
font-stretch: extra-condensed;
```

- normal
- ultra-condensed
- extra-condensed
- condensed
- semi-condensed
- semi-expanded
- expanded
- extra-expanded
- ultra-expanded
- initial
- inherit

Ha itt van különbség:

Lorem ipsum dolor amet (normal)

Lorem ipsum dolor amet (ultra-expanded)

Lorem ipsum dolor amet (condensed)

Lorem ipsum dolor amet (20px)

Lásd még: [http://www.w3schools.com/cssref/css3\\_pr\\_font-stretch.asp](http://www.w3schools.com/cssref/css3_pr_font-stretch.asp) [[http://www.w3schools.com/cssref/css3\\_pr\\_font-stretch.asp](http://www.w3schools.com/cssref/css3_pr_font-stretch.asp)]

## Dobozméretek

A dobozok szélességébe alapértelmezetten nem számít bele a szegély. Ha egymás mellé szeretnénk helyezni két dobozt 50%-50% kitöltéssel, a dobozok nem férnek el a szegély miatt.

### inline-block megoldás

```
<div class="container">
  <div class="box egy">Első div</div>
  <div class="box ketto">Második div</div>
</div>

.container {
  width:400px;
  border:2px solid navy;
  font-size: 0;
}
.container .box {
  box-sizing: border-box;
  display: inline-block;
  width: 50%;
  font-size: 1rem;
}
.container .egy {
  border: 10px solid red;
}
.container .ketto {
  border: 10px solid green;
}
```

Fontos a **külső doboz fontméretének nullázása**, mivel a forráskódban lesz egy white-space karakter, aminek a szélességével számolnunk kell.



A box-sizing: border-box; beállítás segít számunkra. A border-box értékkel megmondhatjuk, hogy a doboz szélességébe a szegélyt is számoljuk bele.

### float: left megoldás

Az alábbi példában 3 dobozt szeretnénk egymás mellét tenni:

index.html

```

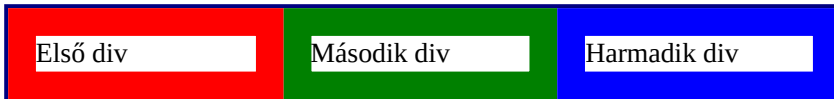
<div class="container">
  <div class="box">Első div.</div>
  <div class="box">Második div</div>
  <div class="box">Harmadik div</div>
</div>

```

```

container {
  width: 30em;
  border: 2px solid navy;
}
.box {
  box-sizing: border-box;
  float: left;
}
.container .box:nth-child(1) {
  width: 33%;
  border: 1em solid red;
}
.container .box:nth-child(2) {
  width: 33%;
  border: 1em solid green;
}
.container .box:nth-child(3) {
  width: 34%;
  border: 1em solid blue;
}

```



| A box-sizing lehetséges értékei |   |
|---------------------------------|---|
| Érték                           | Leírás                                  |
| content-box                     | Max szélesség és magasság a tartalomnál |
| border-box                      | Max szélesség és magasság margónál      |
| inherit                         | Az értékek a szülőktől örökölték        |

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

## Nagyítás

```

div {
  zoom: 200%
}

```

|          |   |
|----------|---|
| normal   | Nincs nagyítás  |
| szám     | Lebegőpontos szám, nagyítás mértékét megadva. Ha szám < 1.0 kicsinyítés történik. Másként kicsinyítés |
| százalék | Pozitív lebegőpontos egész szám. Nagyítás mértéke   |

teszt szöveg eredeti

teszt szöveg nagyítva

## Rugalmas dobozok

Ha rugalmas dobozt szeretnénk, a **display: flex** tulajdonság értékpárost, a **szülő doboz számára** kell beállítanunk.

```

<div id="az">
  <div id="egy">Első</div>

```

```
<div id="ketto">Második</div>
</div>
```

```
div {
  border: 1px solid #ecd858;
  padding: 10px;
}
#az {
  display: flex;
}
#egy {
  flex: 1.0;
}
#ketto {
  flex: 2.0;
}
```

Működés:



Régebbi típusú böngészők esetén:

|             |          |             |              |
|-------------|----------|-------------|--------------|
| -webkit-box | -moz-box | -ms-flexbox | -webkit-flex |
|-------------|----------|-------------|--------------|

```
display: flex;           /* Opera, Firefox, Chrome */
display: -moz-box;       /* Firefox */
display: -webkit-box;    /* iOS, Safari, Chrome */
display: -webkit-flex;   /* Későbbi Chrome */
display: -ms-flexbox;    /* IE 10 */
```

```
flex: 1.0;
-moz-box: 1.0;
-webkit-box: 1.0;
-webkit-flex: 1.0;
-ms-flexbox: 1.0;
```

Több információt a flex használatáról [CSS flexbox](#) fejezet ad.

## Hasábok

```
#az {
  column-count: 3;
  -webkit-column-count: 3;
  -moz-column-count: 3;

  column-gap: 1em;
  -webkit-column-gap: 0.5em;
  -moz-column-gap: 1em;

  column-rule: 2px solid #ff00ff;
  -webkit-column-rule: 2px solid #ff00ff;
  -moz-column-rule: 2px solid #ff00ff;

  margin: 5%;
  padding: 10px;
  border: 1px solid #dcb028;
  border-radius: 10px;
}

<div id="az">
  Lorem ipsum dolor sit amet, consectetur ...
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam condimentum blandit sapien non euismod libero varius et mollis velit volutpat. Vivamus posuere massa aliquam. Donec ultrices placerat

consequat. Phasellus sit amet est non  
lorem vestibulum placerat iaculis vitae  
justo. Lorem ipsum dolor sit amet,  
consectetur adipiscing elit. Vivamus  
bibendum feugiat eleifend. Duis

tincidunt tempus eros, ac laoreet magna  
suscipit eu. Praesent quis facilisis  
lacus. Quisque imperdiet egestas lectus  
dapibus iaculis. Morbi egestas nulla eu  
magna sagittis vitae eleifend lacus

sodales. Nullam vel nunc nisl.  
Vestibulum eget sapien eros. Donec ac  
pellentesque odio. In hac habitasse  
platea dictumst.

## Több háttérkép

```
body {  
  background-image: url(http://szit.hu/p/images/kep001.jpg),  
  url(http://szit.hu/p/images/kep003.jpg);  
  
  background-size: 100px, 200px;  
  background-repeat: no-repeat;  
}
```



## Kép vágása alakzat alapján

```
img {  
  width: 200px;  
  height: 200px;  
  clip-path: circle(50%);  
}
```

```

```



## Háttér

```
body {  
  background-color: #269;
```

```
background-size: 100px 100px, 100px 100px, 20px 20px, 20px 20px;
background-position: -2px -2px, -2px -2px, -1px -1px, -1px -1px;
background-image: -webkit-linear-gradient(white 2px, transparent 2px),
                  -webkit-linear-gradient(0, white 2px, transparent 2px),
                  -webkit-linear-gradient(rgba(255,255,255,.3) 1px, transparent 1px),
                  -webkit-linear-gradient(0, rgba(255,255,255,.3) 1px, transparent 1px);
background-image: -moz-linear-gradient(white 2px, transparent 2px),
                  -moz-linear-gradient(0, white 2px, transparent 2px),
                  -moz-linear-gradient(rgba(255,255,255,.3) 1px, transparent 1px),
                  -moz-linear-gradient(0, rgba(255,255,255,.3) 1px, transparent 1px);
background-image: linear-gradient(white 2px, transparent 2px),
                  linear-gradient(90deg, white 2px, transparent 2px),
                  linear-gradient(rgba(255,255,255,.3) 1px, transparent 1px),
                  linear-gradient(90deg, rgba(255,255,255,.3) 1px, transparent 1px);

}
```

## Másik háttér

```
body {
background:
radial-gradient(black 20%, transparent 20%) 0 0,
radial-gradient(black 20%, transparent 20%) 20px 20px,
radial-gradient(rgba(255, 255, 255, 0.1) 20%, transparent 20%) 0 0,
radial-gradient(rgba(255, 255, 255, 0.1) 20%, transparent 20%) 20px 20px;
background-color:#282828;
background-size:40px 40px;
}
```



## Transzformációk

### A transform tulajdonság

Szintaxis:

```
transform: none|transform-functions|initial|inherit;
```

Használható transzformációs függvények:

- `translate(x,y)`
- `translate3d(x,y,z)`
- `translateX(x)`
- `translateY(y)`
- `translateZ(z)`
- `scale(x,y)`
- `scale3d(x,y,z)`
- `scaleX(x)`
- `scaleY(y)`
- `scaleZ(z)`
- `rotate(szög)`
- `rotate3d(x,y,z, szög)`
- `rotateX(szög)`
- `rotateY(szög)`
- `rotateZ(szög)`
- `skew(x-szög,y-szög)`
- `skewX(szög)`
- `skewY(szög)`
- `perspective(n)`
- `matrix(n, n, n, n, n, n)`
  - 6 darab 3×2 mátrix
- `matrix3d(n, n, n, n, n, n, n, n, n, n, n, n, n, n, n, n)`
  - 16 darab 4×4 mátrix

### Forgatás

```
#egy {  
  background-color: navy;  
  color: white;  
  padding: 5px;  
  position: absolute;  
  top: 50px;  
  transform: rotate(-90deg);  
}
```

teszt

## 3D

```
.belso {  
  position: relative;  
  top: 100px;  
  width: 200px;  
  height: 200px;  
  background-color: green;  
  -webkit-transform:  
    perspective(1000px)  
    rotateY(50deg)  
    translateZ(100px)  
  ;  
}
```

Lorem ipsum dolor sit amet

## Skew

```
transform: skew(30deg,20deg);  
-ms-transform: skew(30deg,20deg); /* IE 9 */  
-webkit-transform: skew(30deg,20deg); /* Safari and Chrome */  
-o-transform: skew(30deg,20deg); /* Opera */  
-moz-transform: skew(30deg,20deg); /* Firefox */
```

Lorem ipsum dolores amet

## Tranzit

```
div#transit001Kulso {
    position: relative;
    height: 60px;
    background-color: green;
    width: 120px;
}

div#transit001 {
    position: absolute;
    display: inline-block;
    background-color: navy;
    color: white;
    left: -80%;
    transition: all 0.6s ease;
}

div#transit001Kulso:hover div#transit001 {
    left: 0;
}
```

Mozgassuk az egeret a zöld doboz felé, a tranzit beindításához:

```
<style>
div#transit002 {
    width: 100px;
    height: 100px;
    background-color: navy;
    transition: width 2s;
}
div#transit002:hover {
    width: 300px;
}
</style>
<div id="transit002"></div>
```

## Transzformáció példa

```
<div class="doboz">
    
</div>

.doboz {
    width: 200px;
    height: 200px;
    border: 0;
    border-radius: 50%;
    overflow: hidden;
    transition-duration: 2s; /* vissza is lassan */
}

.doboz:hover {
    transform: rotate(45deg);
    transition-duration: 2s; /* forgatás lassan */
}

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



## Media Query

A CSS korábbi verzióinál lehetőségünk volt külön CSS megvalósítására nyomtatott felület és képernyőre szánt felületre, külön-külön. A CSS3-mal a lehetőségek nőttek. Beállíthatunk külön CSS-t a fekvőtájolású és az állótájolású megjelenítők esetén. A monitor persze az ember nem szokta forgatni, de a mai mobiltelefonokat már szoktuk.

Az orientáció beállításával lekérdezhetjük egy eszköz vagy megjelenítő milyen tájolással van beállítva. A fekvő és az állótájolású eszközökhöz is külön CSS-t hozhatunk létre. A böngésző persze a megjelenítés szélességét és magasságát kérdezi le. Ha a magasság nagyobb mint a szélesség, akkor állótájolásként (portrait) értelmezi, ha szélesség a nagyobb a magasságnál, akkor a fekvő tájolásról (Landscape) beszélünk.

Az álló és fekvőtájoláshoz a CSS-t így kezdhetjük:

```
/* Álló tájolás */  
  
@media screen and (orientation:portrait) {  
    /* Ide jönnek álló tájolás esetén a CSS meghatározások */  
}  
  
/* Fekvő tájolás */  
  
@media screen and (orientation:landscape) {  
    /* Ide jönnek fekvő tájolás esetén a CSS meghatározások */  
}
```

## Konkrét példa

Illesszünk egy weboldalba egy képet. Például:

```

```

Majd határozzuk meg a következő CSS-t:

```
body {  
    margin: 0;  
}  
@media screen and (orientation:portrait) {  
    img {  
        opacity: 0.3;  
    }  
}  
  
@media screen and (orientation:landscape) {  
    img {  
        opacity: 0.6;  
    }  
}
```

Méretezzük át a böngészőt, majd figyeljük mit történik.

## Átmenettel

Látványosabb lehet a változás, ha teszünk hozzá egy kis átmenetet.

```
body {  
    margin: 0;  
}  
@media screen and (orientation:portrait) {  
    img {  
        opacity: 1;  
        -webkit-transition: opacity 2s ease;  
        -moz-transition: opacity 2s ease;  
        -o-transition: opacity 2s ease;  
        transition: opacity 2s ease;  
    }  
}  
  
@media screen and (orientation:landscape) {  
    img {  
        opacity: 0.2;  
        -webkit-transition: opacity 2s ease;  
        -moz-transition: opacity 2s ease;  
        -o-transition: opacity 2s ease;  
        transition: opacity 2s ease;  
    }  
}
```

## iPhone

Az iPhone nem támogatja a mediai query lekérdezéseket. Ha valaki iPhone-ra is szeretne optimalizálni, akkor a következő sorok oldják meg a problémát:

```
/* Álló tájolás */  
@media screen and (max-width: 320px) {  
    body { opacity: 1; }  
}  
  
/* Fekvő tájolás */  
@media screen and (min-width: 321px) and (max-width: 480px) {  
    body { opacity: 0.2; }  
}
```

Vegyük észre a max-width: 320px és a min-width: 321px feltételeket.

Az iPhone esetén most túl nagy képet kaphatunk. A background-size segítségével megoldhatjuk a problémát:

```
@media screen and (max-device-width: 480px) {  
    html, body {  
        -moz-background-size: 80% auto;  
        -webkit-background-size: 80% auto;  
        background-size: 80% auto;  
    }  
}
```

- <http://www.w3.org/TR/css3-mediaqueries/> [<http://www.w3.org/TR/css3-mediaqueries/>]

## Szelektor tulajdonságai

```
#egy::selection {  
    background-color: yellow;  
    color: blue;  
}
```

```
#egy::-moz-selection {  
    background-color: yellow;  
    color: blue;  
}
```

Mindenhol:

```
::selection {  
    background-color: yellow;  
    color: blue;  
}
```

```
::-moz-selection {  
    background-color: yellow;  
    color: blue;  
}
```