

CSS bevezető

- A CSS az a nyelv, amelyet a weboldal stílusának meghatározására használunk.

Mi az a CSS?

- A CSS a Cascading Style Sheets rövidítése
- A CSS leírja, hogyan kell a HTML-elemeket megjeleníteni a képernyőn, papíron vagy más adathordozón
- A CSS rengeteg munkát takarít meg. Egyszerre több weboldal elrendezését is vezérelheti
- A külső stíluslapok CSS-fájlokban tárolódnak

CSS megjegyzések / kommentek

- A megjegyzések a kód magyarázatára szolgálnak, és segíthetnek a forráskód későbbi szerkesztésénél.
- A böngészők figyelmen kívül hagyják a megjegyzéseket.
- Az <style>elemen belül egy CSS megjegyzés kerül elhelyezésre, amely a következővel kezdődik /*és ezzel végződik */

```
/* This is a single-line comment */
```

```
p {  
  color: red;  
}
```

CSS hozzáadása HTML oldalhoz

Amikor egy böngésző beolvas egy stíluslapot, a HTML-dokumentumot a stíluslap információinak megfelelően formázza.

A stíluslap beszúrásának három módja van:

- **Külső CSS (external)**

Egy külső stíluslappal egyetlen fájl módosításával megváltoztathatja egy teljes webhely megjelenését!

Minden HTML-oldalnak tartalmaznia kell egy hivatkozást a külső stíluslapfájlra a <link> elemen belül, a fejrészen belül.

Külső stíluslap bármilyen szövegszerkesztőben írható, és .css kiterjesztéssel kell menteni.

A külső .css fájl nem tartalmazhat HTML-címkéket.

A továbbiakban ajánlott és ha a feladat máshogy nem kéri, akkor kötelező a külső stíluslap használata!

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

```
"mystyle.css"

body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

Megjegyzés: Ne írjon szóközt a tulajdonság értéke és az egység közé:

Helytelen (szóköz): margin-left: 20 px;

Helyes (szóköz nélkül): margin-left: 20px;

- **Belső CSS (internal)**

Belső stíluslap akkor használható, ha egyetlen HTML-oldal egyedi stílussal rendelkezik.

A belső stílus a <style> elemen belül, a fejrészen belül van meghatározva.

A belső stílusok a <style> elemen, a HTML-oldal <head> szakaszában vannak meghatározva

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

- **Inline CSS – soron belüli**

Egy soron belüli stílus használható egyedi stílus alkalmazására egyetlen elemhez.

Soron belüli stílusok használatához adja hozzá a style attribútumot a megfelelő elemhez. A style attribútum bármilyen CSS tulajdonságot tartalmazhat.

A soron belüli stílusok a megfelelő elem "style" attribútumában vannak meghatározva:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

Több stíluslap

Ha ugyanahhoz a választóhoz (elemhez) különböző stíluslapokban definiáltak bizonyos tulajdonságokat, akkor az utoljára olvasott stíluslap értéke kerül felhasználásra.

Ha a belső stílus a külső stíluslapra mutató hivatkozás **után** van megadva, a **<h1>** elemek "narancssárgák" lesznek:

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
<style>
h1 {
  color: orange;
}
</style>
</head>
```

Ha azonban a belső stílus a külső stíluslapra mutató hivatkozás **előtt** van megadva, a **<h1>** elemek "navy" lesznek:

```
<head>
<style>
h1 {
  color: orange;
}
</style>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Lépcsőzetes sorrend

Milyen stílust használ a rendszer, ha egy HTML-elemhez egynél több stílus van megadva?

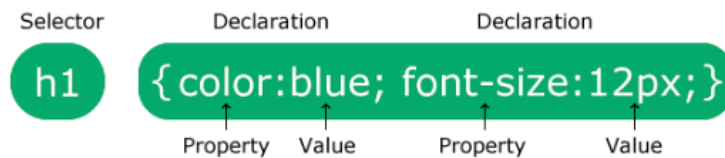
Az oldalon lévő összes stílus egy új "virtuális" stíluslapba "lépcsősödik" a következő szabályok szerint, ahol az elsőnek van a legmagasabb prioritása:

1. Soron belüli stílus (HTML elemen belül)
2. Külső és belső stíluslapok (a fejrészben)
3. Alapértelmezett böngésző

Tehát a soron belüli stílusnak van a legmagasabb prioritása, és felülírja a külső és belső stílusokat, valamint a böngésző alapértelmezett beállításait.

CSS szintaxis

A CSS-szabály egy választóból és egy deklarációs blokkból áll.



- A választó (selector) a stílusozni kívánt HTML-elemre mutat.
- A deklarációs blokk egy vagy több deklarációt (declaration) tartalmaz pontosvesszővel elválasztva.
- Minden deklaráció tartalmaz egy CSS-tulajdonságnevet(property) és egy értéket(value).
- Több CSS-deklaráció pontosvesszővel van elválasztva, a deklarációs blokkokat pedig kapcsos zárójelek veszik körül.

CSS választók (selectors)

A CSS-szelektorok a stílusozni kívánt HTML-elemek „megkeresésére” (vagy kijelölésére) szolgálnak.

A CSS-szelektorokat öt kategóriába sorolhatjuk:

- **Egyszerű kiválasztók - elemválasztók (elemek kiválasztása név, azonosító, osztály alapján)**

1. CSS elemválasztók

Az elemválasztó az elem neve alapján választja ki a HTML elemeket.

Itt az oldal összes <p> eleme középre igazítva lesz, piros szövegszínnel:

```
p {  
  text-align: center;  
  color: red;  
}
```

2. A CSS azonosító választó

Az azonosító választó egy HTML-elem id attribútuma alapján választ ki egy adott elemet.

Egy elem azonosítója egyedi az oldalon belül, így az id választó egy egyedi elem kiválasztására szolgál!

Egy adott azonosítójú elem kiválasztásához írjon be egy hash (#) karaktert, amelyet az elem azonosítója követ.

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

Megjegyzés: Az azonosító név nem kezdődhet számmal!

3. A CSS osztályválasztó

Az osztályválasztó kiválasztja a HTML elemeket egy adott osztályattribútummal.

Egy adott osztályhoz tartozó elemek kiválasztásához írjon be egy pontot (.), majd az osztály nevét.

```
.center {  
  text-align: center;  
  color: red;  
}
```

Azt is megadhatja, hogy egy osztály csak meghatározott HTML-elemekre legyen hatással.

```
p.center {  
  text-align: center;  
  color: red;  
}
```

A HTML elemek egynél több osztályra is hivatkozhatnak.

```
<p class="center large">This paragraph refers to two classes.</p>
```

Megjegyzés: Az osztálynév nem kezdődhet számmal!

4. A CSS univerzális választó

Az univerzális választó (*) kijelöli az összes HTML elemet az oldalon.

```
* {  
  text-align: center;  
  color: blue;  
}
```

5. A CSS csoportosítási választó

A csoportosítási választó kiválasztja az összes HTML-elemet azonos stílusdefiníciókkal.

Nézze meg a következő CSS-kódot (a h1, h2 és p elemek stílusdefiníciói megegyeznek):

```
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

Jobb lesz csoportosítani a kiválasztókat, minimalizálni a kódot. A kijelölők csoportosításához vesszővel válassza el az egyes választókat.

```
h1, h2, p {
  text-align: center;
  color: red;
}
```

- Kombinátorválasztók (az elemek kiválasztása a köztük lévő konkrét kapcsolat alapján)
A kombinátor olyan dolog, amely megmagyarázza a szelektorok közötti kapcsolatot. Egy CSS-választó több egyszerű választót is tartalmazhat. Az egyszerű szelektorok közé beépíthetünk egy kombinátort.

A CSS-ben négy különböző kombinátor található:

1. leszármazott választó (szóköz)

A leszármazott választó minden olyan elemhez illeszkedik, amely egy adott elem leszármazottja.

A következő példa az összes <p> elemet kiválasztja a <div> elemeken belül:

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

2. gyermekválasztó (>)

A gyermekválasztó kiválasztja az összes olyan elemet, amely egy adott elem gyermeke.

A következő példa az összes <p> elemet kijelöli, amelyek egy <div> elem gyermekei:

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

3. szomszédos testvérválasztó (+)

A szomszédos testvérválasztó egy olyan elem kiválasztására szolgál, amely közvetlenül egy másik meghatározott elem után van.

A testvérelemeknek ugyanazzal a szülő elemmel kell rendelkezniük, a „szomszédos” pedig „azonnal következőt” jelent.

A következő példa az első <p> elemet választja ki, amely közvetlenül a <div> elemek után kerül elhelyezésre:

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

4. általános testvérválasztó (~)

Az általános testvérválasztó kiválasztja az összes olyan elemet, amely egy adott elem következő testvére.

A következő példa az összes <p> elemet kijelöli, amelyek a <div> elemek következő testvérei:

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

- Pszeudoosztály szelektorok (elemek kiválasztása egy bizonyos állapot alapján)
Egy pszeudoosztályt használnak egy elem speciális állapotának meghatározására. Például a következőkre használható:
 1. Stílusozzon egy elemet, amikor a felhasználó rámutat
 2. A meglátogatott és nem látogatott linkek stílusa eltérő
 3. Stílusozzon egy elemet, amikor az fókuszba kerül

Szintaxis:

```
selector:pseudo-class {  
    property: value;  
}
```

```
/* unvisited link */  
a:link {  
    color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
    color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {  
    color: #FF00FF;  
}  
  
/* selected link */  
a:active {  
    color: #0000FF;  
}
```

- Pszeudoelem-választók (egy elem egy részének kijelölése és stílusa)

A CSS pszeudoelemet az elem meghatározott részeinek stílusára használjuk.

Például a következőkre használható:

1. Stíluszon egy elem első betűjét vagy sorát
2. Tartalom beszúrása egy elem tartalma elé vagy után

Szintaxis

A pszeudoelemek szintaxisa:

```
selector::pseudo-element {
  property: value;
}
```

- Attribútumválasztók (elemek kiválasztása attribútum vagy attribútumérték alapján)
Lehetőség van olyan HTML elemek stílusozására, amelyek meghatározott attribútumokkal vagy attribútumértékekkel rendelkeznek.

1. CSS [attribútum] választó

A [attribute]szelektor egy megadott attribútummal rendelkező elemek kiválasztására szolgál.

```
a[target] {
  background-color: yellow;
}
```

Bővebben: https://www.w3schools.com/css/css_attribute_selectors.asp

CSS színek

A színek megadása előre meghatározott színnevek vagy RGB, HEX, HSL, RGBA, HSLA értékek használatával történik.

1. CSS háttérszín (background-color: value;)
2. CSS szöveg színe (color: value;)
3. CSS szegélyszín (border: value;)

- **CSS RGB színek**

Az RGB színérték a PIROS, ZÖLD és KÉK fényforrásokat jelöli.

Minden paraméter (piros, zöld és kék) határozza meg a szín intenzitását 0 és 255 között.

Például az rgb(255, 0, 0) pirosként jelenik meg, mert a piros a legmagasabb értékre (255), a többi pedig 0-ra van állítva.

A fekete megjelenítéséhez állítsa az összes színparamétert 0-ra, így: rgb(0, 0, 0).

A fehér szín megjelenítéséhez állítsa az összes színparamétert 255-re, így: `rgb(255, 255, 255)`.



- **RGBA érték**

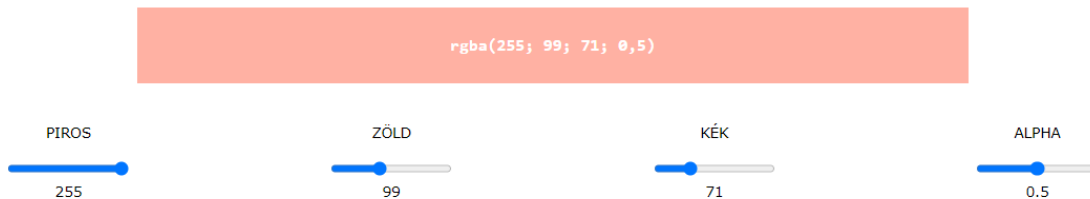
Az RGBA színértékek az RGB színértékek kiterjesztése egy alfa-csatornával – amely a szín átlátszatlanságát határozza meg.

Az RGBA színérték a következőkkel van megadva:

`rgba (piros, zöld , kék, alfa)`

Az alfa paraméter egy 0,0 (teljesen átlátszó) és 1,0 (egyáltalán nem átlátszó) közötti szám:

Kísérletezzen az alábbi RGBA-értékek keverésével:



- **CSS HEX színek**

A CSS-ben egy szín a következő formában adható meg hexadecimális értékkel:

`#rrggbb`

Ahol rr (piros), gg (zöld) és bb (kék) hexadecimális értékek 00 és ff között (ugyanaz, mint a decimális 0-255).

Például az `#ff0000` pirosan jelenik meg, mert a piros a legmagasabb értékre (ff), a többi pedig a legalacsonyabb értékre (00) van beállítva.

A fekete megjelenítéséhez állítsa az összes értéket 00-ra, így: `#000000`.

A fehér megjelenítéséhez állítsa az összes értéket ff-re, így: `#ffffff`.



- **CSS HSL színek**

A CSS-ben a szín a színárnyalat, telítettség és világosság (HSL) segítségével a következő formában adható meg:

hsl (színárnyalat , telítettség , világosság)

A színárnyalat egy fok a színekörön 0 és 360 között. A 0 a vörös, a 120 a zöld és a 240 a kék.

A telítettség százalékos érték. A 0% a szürke árnyalatát jelenti, a 100% pedig a teljes színt.

A könnyedség is százalék. 0% fekete, 50% nem világos vagy sötét, 100% fehér

Példa

hsl(0, 100%, 50%)	hsl(240, 100%, 50%)
hsl(147, 50%, 47%)	hsl(300, 76%, 72%)
hsl(39, 100%, 50%)	hsl(248, 53%, 58%)

CSS hátterek

A CSS-háttér tulajdonságai az elemek háttéreffektusainak hozzáadására szolgálnak. Ezekben a fejezetekben a következő CSS-háttértulajdonságokat ismerheti meg:

- background-color (CSS háttérszín)

A background-color tulajdonság egy elem háttérszínét adja meg.

```
body {  
  background-color: lightblue;  
}
```

Átlátszatlanság / Átlátszóság

A opacitytulajdonság egy elem átlátszatlanságát/átlátszóságát határozza meg. 0,0 és 1,0 közötti értéket vehet fel. Minél alacsonyabb az érték, annál átláthatóbb:

átlátszatlanság 1	átlátszatlanság 0,6	átlátszatlanság 0,3	átlátszatlanság 0,1
-------------------	---------------------	---------------------	---------------------

Példa

```
div {  
  background-color: green;  
  opacity: 0.3;  
}
```

Próbáld ki magad "

Megjegyzés: Ha a `opacity` tulajdonságot arra használja, hogy átlátszóságot adjon egy elem háttéréhez, akkor az összes alárendelt elem ugyanazt az átlátszóságot örökli. Ez nehezen olvashatóvá teheti a teljesen átlátszó elemen belüli szöveget.

- **Átlátszóság RGBA használatával**

Ha nem szeretne átlátszatlanságot alkalmazni a gyermekelemekre, mint a fenti példánkban, használja az **RGBA** színértékeket. A következő példa a háttérszín átlátszatlanságát állítja be, nem a szöveget:



Az RGBA színérték a következőkkel van megadva: `rgba(piros, zöld, kék, alfa)`. Az *alfa* paraméter egy 0,0 (teljesen átlátszó) és 1,0 (teljesen átlátszatlan) közötti szám.

```
div {  
  background: rgba(0, 128, 0, 0.3) /* Green background with 30% opacity */  
}
```

- **background-image(CSS háttérkép)**

A `background-image` tulajdonság egy elem háttéréként használandó képet határoz meg. Alapértelmezés szerint a kép ismétlődik, így a teljes elemet lefedi.

```
body {  
  background-image: url("paper.gif");  
}
```

Megjegyzés: Ha háttérképet használ, olyan képet használjon, amely nem zavarja a szöveget. Az így beillesztett kép szintén a megfelelő mappában kell, hogy legyen.

- **background-repeat(CSS háttér-ismétlés)**

Alapértelmezés szerint a `background-image` tulajdonság vízszintesen és függőlegesen is megismétli a képet.

A `background-repeat` értékei:

`background-repeat: repeat-x;` vízszintesen ismétlődik a kép

`background-repeat: repeat-y;` függőlegesen ismétlődik a kép

`background-repeat: no-repeat;` nem ismétlődik a kép

Bővebben: https://www.w3schools.com/css/css_background_repeat.asp

- **background-attachment(CSS háttérmelléklet)**

A `background-attachment` tulajdonság megadja, hogy a háttérképet görgetni kell-e vagy rögzíteni kell (nem gördül az oldal többi részével):

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}
```

Adja meg, hogy a háttérképnek az oldal többi részével együtt kell gördülnie:

```
body {
  background-image: url("img_tree.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: scroll;
}
```

- **background-position (CSS háttérpozíció)**

A background-position tulajdonság a háttérkép pozíciójának megadására szolgál.

Megjegyzés: Alapértelmezés szerint a háttérkép az elem bal felső sarkába kerül, és függőlegesen és vízszintesen is megismétlődik.

CSS szintaxis

```
background-position: value;
```

Value

```
left top
left center
left bottom
right top
right center
right bottom
center top
center center
center bottom
```

- **background**

A kód lerövidítéséhez lehetőség van az összes háttértulajdonság megadására egyetlen tulajdonságban. Ezt gyorsírási tulajdonságnak nevezik.

```
body {
  background: #ffffff url("img_tree.png") no-repeat right top;
}
```

A gyorsírási tulajdonság használatakor a tulajdonságértékek sorrendje a következő:

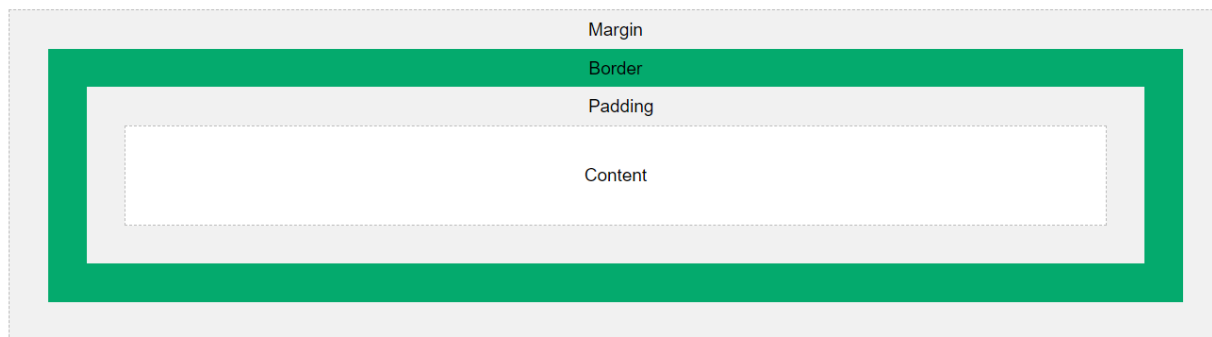
- background-color
- background-image
- background-repeat
- background-attachment
- background-position

Nem számít, ha valamelyik érték hiányzik, ha a többi ebben a sorrendben van. Vegye figyelembe, hogy a fenti példákban nem használjuk a background-attachement tulajdonságot, mivel nincs értéke.

CSS Box Model

Minden HTML-elem doboznak tekinthető.

A CSS-dobozmodell lényegében egy olyan doboz, amely minden HTML-elemet körülölel. A következőkből áll: margók, szegélyek, kitöltés és a tényleges tartalom. Az alábbi képen látható a doboz modellje:



A különböző részek magyarázata:

- **Tartalom(content)** – A doboz tartalma, ahol szöveg és képek jelennek meg
- **Kitöltés(padding)** – Megtisztítja a tartalom körüli területet. A párnázás átlátszó
- **Szegély(border)** – A kitöltés és a tartalom körül húzódó szegély
- **Margó(margin)** – Megtisztítja a határon kívüli területet. A margó átlátszó

A dobozmodell lehetővé teszi, hogy szegélyt adjunk az elemek köré, és teret definiáljunk az elemek között.

```
div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}
```

CSS Padding (kitöltés)

A CSS padding-tulajdonságok arra szolgálnak, hogy teret hozzanak létre egy elem tartalma körül, bármely meghatározott határon belül.

A CSS segítségével teljes ellenőrzése alatt áll a kitöltés felett. Vannak olyan tulajdonságok, amelyek segítségével be lehet állítani a kitöltést egy elem minden oldalához (felső, jobb, alsó és bal).

A CSS-ben olyan tulajdonságok vannak, amelyekkel megadhatja a kitöltést egy elem egyes oldalaihoz:

- padding-top
- padding-right

- padding-bottom
- padding-left

Az összes kitöltési tulajdonság a következő értékekkel rendelkezhet:

- *hossz* – megadja a kitöltést px, pt, cm stb.
- % - megadja a kitöltést a tartalmazó elem szélességének %-ában
- inherit – megadja, hogy a kitöltést a szülőelemtől kell örökölni

Megjegyzés: Negatív értékek nem megengedettek.

```
div {
    padding-top: 50px;
    padding-right: 30px;
    padding-bottom: 50px;
    padding-left: 80px;
}
```

Padding - Gyorsírási tulajdonság

A kód lerövidítéséhez lehetőség van az összes kitöltési tulajdonság megadására egy tulajdonságban.

A paddingtulajdonság a következő egyedi kitöltési tulajdonságok gyorsírása:

- padding-top
- padding-right
- padding-bottom
- padding-left
-

Tehát a következőképpen működik:

Ha a paddingtulajdonságnak négy értéke van:

- **padding: 25px 50px 75px 100px;**
 - a felső párnázás 25 képpont
 - a jobb oldali kitöltés 50 képpont
 - Az alsó párnázás 75 képpont
 - bal oldali kitöltés 100 képpont

```
div {
    padding: 25px 50px 75px 100px;
}
```

Ha a paddingtulajdonságnak három értéke van:

- **padding: 25px 50px 75px;**
 - a felső párnázás 25 képpont
 - A jobb és bal oldali kitöltés 50 képpontos
 - Az alsó párnázás 75 képpont

Ha a paddingtulajdonságnak két értéke van:

- **padding: 25px 50px;**
 - A felső és alsó párnázás 25 képpontos
 - A jobb és bal oldali kitöltés 50 képpontos

Ha az paddingingatlannak egy értéke van:

- **padding: 25px;**
 - mind a négy kitöltés 25 képpontos

CSS Margins (margók)

A CSS margin-tulajdonságok arra szolgálnak, hogy teret hozzanak létre az elemek körül, bármely meghatározott határon kívül.

A CSS segítségével teljes ellenőrzése alatt áll a margók felett. Vannak tulajdonságok a margók beállítására az elemek minden oldalán (felső, jobb, alsó és bal).

A CSS-ben olyan tulajdonságok vannak, amelyekkel megadhatja a margót egy elem mindkét oldalán:

- margin-top
- margin-right
- margin-bottom
- margin-left

Az összes margó tulajdonság a következő értékekkel rendelkezhet:

- auto - a böngésző kiszámítja a margót
- *hossz* – margót ad meg px, pt, cm stb.
- % - margót ad meg %-ban a tartalmazó elem szélességéhez
- inherit – megadja, hogy a margót a szülőelemtől örököljék

Megjegyzés: Negatív értékek megengedettek.

```
p {  
    margin-top: 100px;  
    margin-bottom: 100px;  
    margin-right: 150px;  
    margin-left: 80px;  
}
```

A kód lerövidítéséhez lehetőség van az összes margó tulajdonság megadására egy tulajdonságban.

A margintulajdonság a következő egyedi margótulajdonságok gyorsírása:

- margin-top
- margin-right
- margin-bottom
- margin-left

Tehát a következőképpen működik:

Ha a margintulajdonságnak négy értéke van:

- **margó: 25 képpont 50 képpont 75 képpont 100 képpont;**
 - a felső margó 25 képpont
 - a jobb margó 50 képpont
 - az alsó margó 75 képpont
 - a bal margó 100 képpont

Ha a margintulajdonságnak három értéke van:

- **margó: 25 képpont 50 képpont 75 képpont;**
 - a felső margó 25 képpont
 - a jobb és a bal margó 50 képpont
 - az alsó margó 75 képpont

Ha a margintulajdonságnak két értéke van:

- **margó: 25 képpont 50 képpont;**
 - a felső és az alsó margó 25 képpont
 - a jobb és a bal margó 50 képpont

Ha az marginatlnak egy értéke van:

- **margó: 25 képpont;**
 - mind a négy margó 25 képpont

Az automatikus érték

Beállíthatja a margó tulajdonságot, autohogyan vízszintesen középre állítsa az elemet a tárolójában. Az elem ekkor felveszi a megadott szélességet, a fennmaradó terület pedig egyenlő arányban lesz felosztva a bal és a jobb margó között.

CSS borders (határok)

A CSS-szegély tulajdonságai lehetővé teszik egy elem szegélyének stílusának, szélességének és színének megadását.

A border-styletulajdonság megadja, hogy milyen szegélyt kell megjeleníteni.

A következő értékek megengedettek:

- dotted- Pontozott szegélyt határoz meg
- dashed- Szaggatott szegélyt határoz meg
- solid- Szilárd határt határoz meg
- double- Kettős szegélyt határoz meg
- groove- 3D barázdált szegélyt határoz meg. A hatás a szegélyszín értékétől függ
- ridge- 3D bordázott szegélyt határoz meg. A hatás a szegélyszín értékétől függ
- inset- 3D-s beszúrt szegélyt határoz meg. A hatás a szegélyszín értékétől függ

- outset- Meghatároz egy 3D kiindulási szegélyt. A hatás a szegélyszín értékétől függ
- none- Nem határoz meg határt, tehát nincs szegély
- hidden- Rejtett határt határoz meg

A border-style tulajdonság 1-4 értéket tartalmazhat (a felső szegély, a jobb oldali, az alsó és a bal szegély).

CSS határ szélessége

A border-width tulajdonság a négy szegély szélességét adja meg.

A szélesség beállítható meghatározott méretként (px, pt, cm, em stb.), vagy a három előre meghatározott érték valamelyikével: vékony, közepes vagy vastag

CSS szegélyszín

A border-colortulajdonság a négy szegély színének beállítására szolgál.

A szín a következőképpen állítható be:

- név - adjon meg egy színnevet, például "piros"
- HEX - adjon meg egy HEX értéket, például "#ff0000"
- RGB – adjon meg egy RGB értéket, például "rgb(255,0,0)"
- HSL - adjon meg egy HSL értéket, például "hsl(0, 100%, 50%)"
- Átlátszó
-

Megjegyzés: Ha border-colornincs beállítva, akkor örökli az elem színét.

Különleges oldalszínek

A border-color tulajdonság 1-4 értéket tartalmazhat (a felső szegély, a jobb oldali, az alsó és a bal szegély).

CSS Border – Gyorsírási tulajdonság

Ahogy az előző oldalon is láttuk, számos tulajdonságot figyelembe kell venni a szegélyek kezelésekor.

A kód lerövidítéséhez lehetőség van az összes egyedi határtulajdonság megadására egy tulajdonságban.

Az borderingatlan a következő egyedi határterületi ingatlanok gyorsírása:

- border-width
- border-style(kívánt)
- border-color

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

CSS lekerekített határok

A border-radiustulajdonság lekerekített szegélyek hozzáadására szolgál egy elemhez

```
p {  
  border: 2px solid red;  
  border-radius: 5px;  
}
```

CSS betűtípusok

A betűtípus kiválasztása fontos

A megfelelő betűtípus kiválasztása nagy hatással van arra, hogy az olvasók hogyan élnek meg egy webhelyet.

A megfelelő betűtípus erős identitást teremthet márkájának.

Fontos, hogy könnyen olvasható betűtípust használjunk. A betűtípus értéket ad a szöveghez. Fontos a betűtípus megfelelő színének és szövegméretének kiválasztása is.

Általános betűtípuscsaládok

A CSS-ben öt általános betűtípuscsalád létezik:

1. **A Serif** betűtípusok minden betű szélén egy kis körvonal található. A formalitás és az elegancia érzetét keltik.
2. **A Sans-serif** betűtípusok tiszta vonalakkal rendelkeznek (nincs mellékelve kis vonás). Modern és minimalista megjelenést kölcsönöznek.
3. **Monospace** betűtípusok - itt minden betű azonos fix szélességű. Mechanikus megjelenést kölcsönöznek.
4. **A kurzív** betűtípusok az emberi kézírást utánozzák.
5. **A Fantasy** betűtípusok dekoratív/játékos betűtípusok.
- 6.

Az összes különböző betűtípusnév valamelyik általános betűtípuscsaládhoz tartozik.

A CSS betűtípus-család tulajdonsága

A CSS-ben a font-familytulajdonságot használjuk a szöveg betűtípusának megadására.

Megjegyzés : Ha a betűtípus neve egynél több szóból áll, akkor azt idézőjelbe kell tenni, például: "Times New Roman".

Tipp: A font-familyböngészők/operációs rendszerek közötti maximális kompatibilitás biztosítása érdekében a tulajdonságnak több betűtípusnevet is tartalmaznia kell tartalékrendszerként. Kezdje a kívánt betűtípussal, és fejezze be egy általános családdal (hogy a böngésző hasonló betűtípust válasszon az általános családból, ha más betűtípus nem áll rendelkezésre). A betűtípusneveket vesszővel kell elválasztani.

Betű stílus

A font-style tulajdonságot többnyire dőlt szöveg megadására használják.

Ennek a tulajdonságnak három értéke van:

- normál - A szöveg normál módon jelenik meg

- dőlt – A szöveg dőlt betűvel jelenik meg
- ferde – A szöveg „ferde” (a ferde nagyon hasonlít a dőlt betűhöz, de kevésbé támogatott)

Betűsúly

A font-weight tulajdonság megadja a betűtípus súlyát (normal, bold)

Betűtípus változat

A font-varianttulajdonság megadja, hogy a szöveg kisbetűs betűtípussal jelenjen-e meg vagy sem. Kisbetűs betűtípus esetén minden kisbetűt nagybetűvé alakít a rendszer. A konvertált nagybetűk azonban kisebb betűmérettel jelennek meg a szövegben, mint az eredeti nagybetűk (normal, small-caps)

Betűméret

A font-sizet tulajdonság beállítja a szöveg méretét.

A szövegméret kezelésének képessége fontos a webdesignban. Azonban ne használjon betűméret-beállításokat, hogy a bekezdések címsornak, vagy a címsorok bekezdésnek tűnjenek. Mindig a megfelelő HTML-címkéket használja, például a <h1> - <h6>-t a címsorokhoz és a <p>-t a bekezdésekhez.

A betűméret értéke lehet abszolút vagy relatív méret.

Abszolút méret:

- A szöveget meghatározott méretre állítja
- Nem teszi lehetővé a felhasználó számára, hogy minden böngészőben módosítsa a szöveg méretét (kisegítő lehetőségek miatt rossz)
- Az abszolút méret akkor hasznos, ha a kimenet fizikai mérete ismert

Relatív méret:

- Beállítja a méretet a környező elemekhez viszonyítva
- Lehetővé teszi a felhasználó számára, hogy módosítsa a szöveg méretét a böngészőben

Megjegyzés: Ha nem ad meg betűméretet, a normál szöveg, például a bekezdések alapértelmezett mérete 16 képpont (16px=1em).

Reszponzív betűméret

A szöveg mérete mértékegységgel állítható be vw, ami a "nézetablak szélességét" jelenti.

CSS magasság, szélesség és maximális szélesség

A height és widthtulajdonságoknak a következő értékei lehetnek:

- auto- Ez az alapértelmezett. A böngésző kiszámítja a magasságot és a szélességet
- length- Meghatározza a magasságot/szélességet px-ben, cm-ben stb.
- %- Meghatározza a tartalmazó blokk magasságát/szélességét százalékban
- initial- Beállítja a magasságot/szélességet az alapértelmezett értékre
- inherit- A magasság/szélesség a szülőértéktől öröklődik