

Mi a Bootstrap?

A Bootstrap egy HTML/CSS keretrendszer, segítségével gyorsan és átláthatóan hozhatunk létre responsive weboldalakat. A rendszert a Twitternél fejlesztették ki, 2011-ben tették közzé open-source licensszel, és a mai napig is ez az egyik legnépszerűbb projekt. Egy egész közösség áll mögötte, folyamatosan fejlesztik-tesztelik, így az első verziója óta már rengeteg átalakításon ment keresztül.

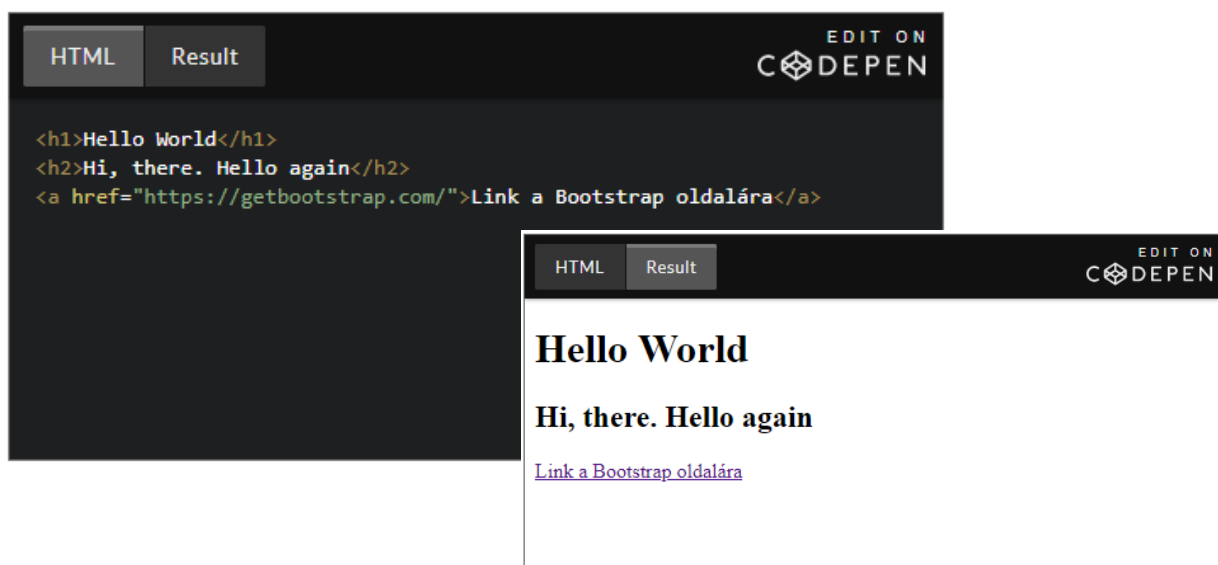
Letisztult dizájn jellemzi, számos UI elemet és JavaScript bővítményt tartalmaz. Alapvetően meglévő html és css tudásra épít, ezért könnyen elsajátítható és nagyon jól összerakott, részletes dokumentációt írtak hozzá.

A harmadik verziójában vezették be az ún. mobile-first szemléletet, és a legfrissebb 4-es változatban már erősen a Flexbox-ra épül. Hatalmas kódbázissal, és számos olyan elemmel rendelkezik, mellyel gyorsan felépíthetjük a weboldalunk vázát, amit aztán könnyen testre is szabhatunk.

Hogyan működik?

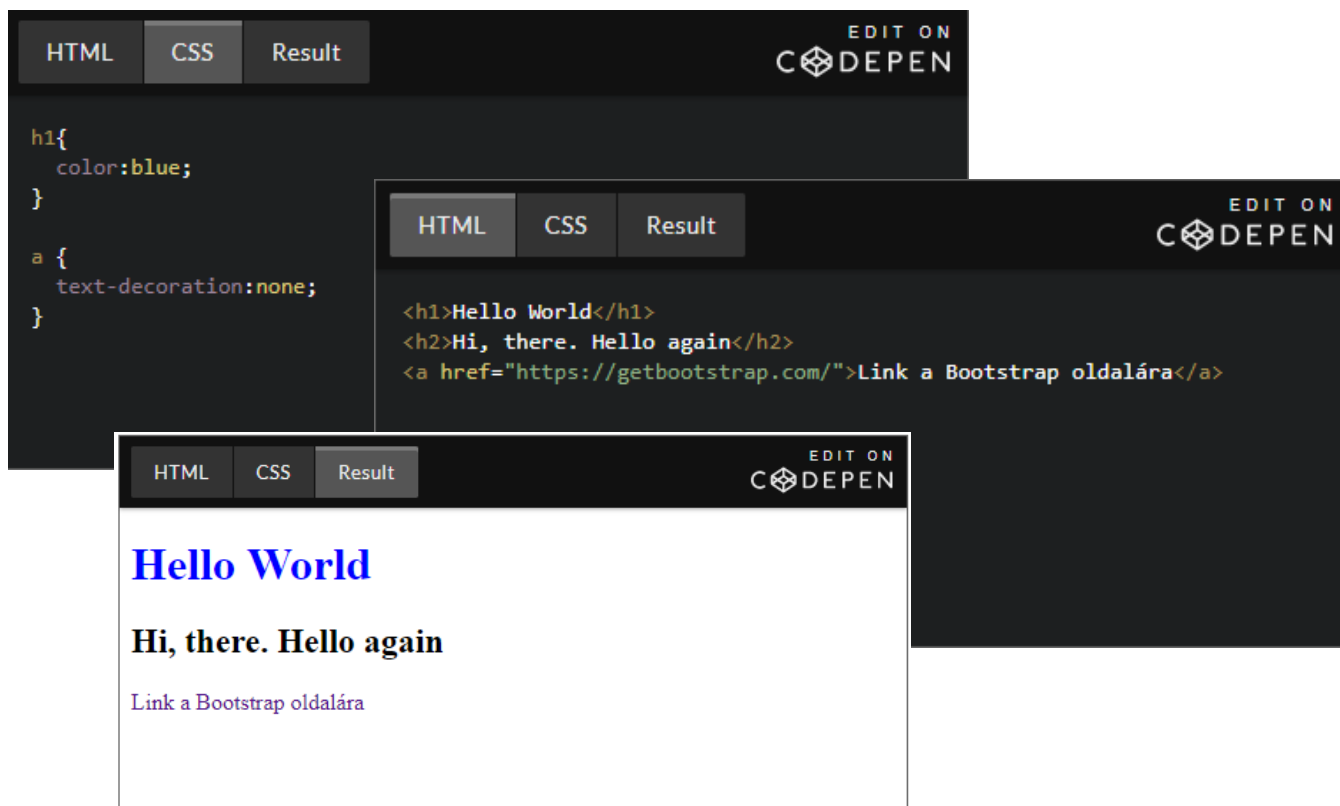
Egy alap weboldalt úgy hozunk létre, hogy írunk egy html dokumentumot, majd ehhez az oldalhoz külső vagy beágyazott (css) stíluslapokat csatolunk.

A böngészők is rendelkeznek alapértelmezett stílussal, melyek akkor is befolyásolják a weboldalak megjelenítését, ha mi semmilyen stílust nem adunk meg.



Az alábbi példában láthatjuk, hogy css használata nélkül hogyan jeleníti meg a böngésző a képen szereplő kódot:

a h1 címsor vastagon és nagyobb betűmérettel jelenik meg, a link aláhúzott és kék, kattintás után lila.

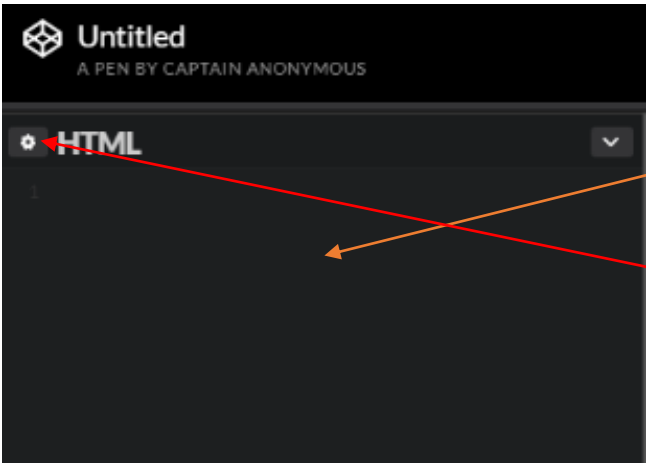


Ezeket a megjelenítéseket írhatjuk felül a css használatával. A html dokumentumunkban az egyes címkéknél be kell írni a class jellemzőt vagy jellemzőket, melyekkel az elemek azonos megjelenítését érhetjük el. Egy címkéhez bármennyi osztálykijelölőt írhatunk, és különböző címkékhez is rendelhetők közös osztályok. A css-ben az osztálykijelölők előtt pont áll (betűköz nélkül).

A Bootstrap is ehhez hasonlóan működik. A könyvtára tartalmaz egy óriási stíluslapot, melyekre az osztálykijelölők használatával hivatkozunk. Az olyan keretrendszerek, mint a Bootstrap anélkül változtatják meg a weboldalunk megjelenését, hogy nekünk bármilyen jelentős beavatkozásokat kellene tennünk.

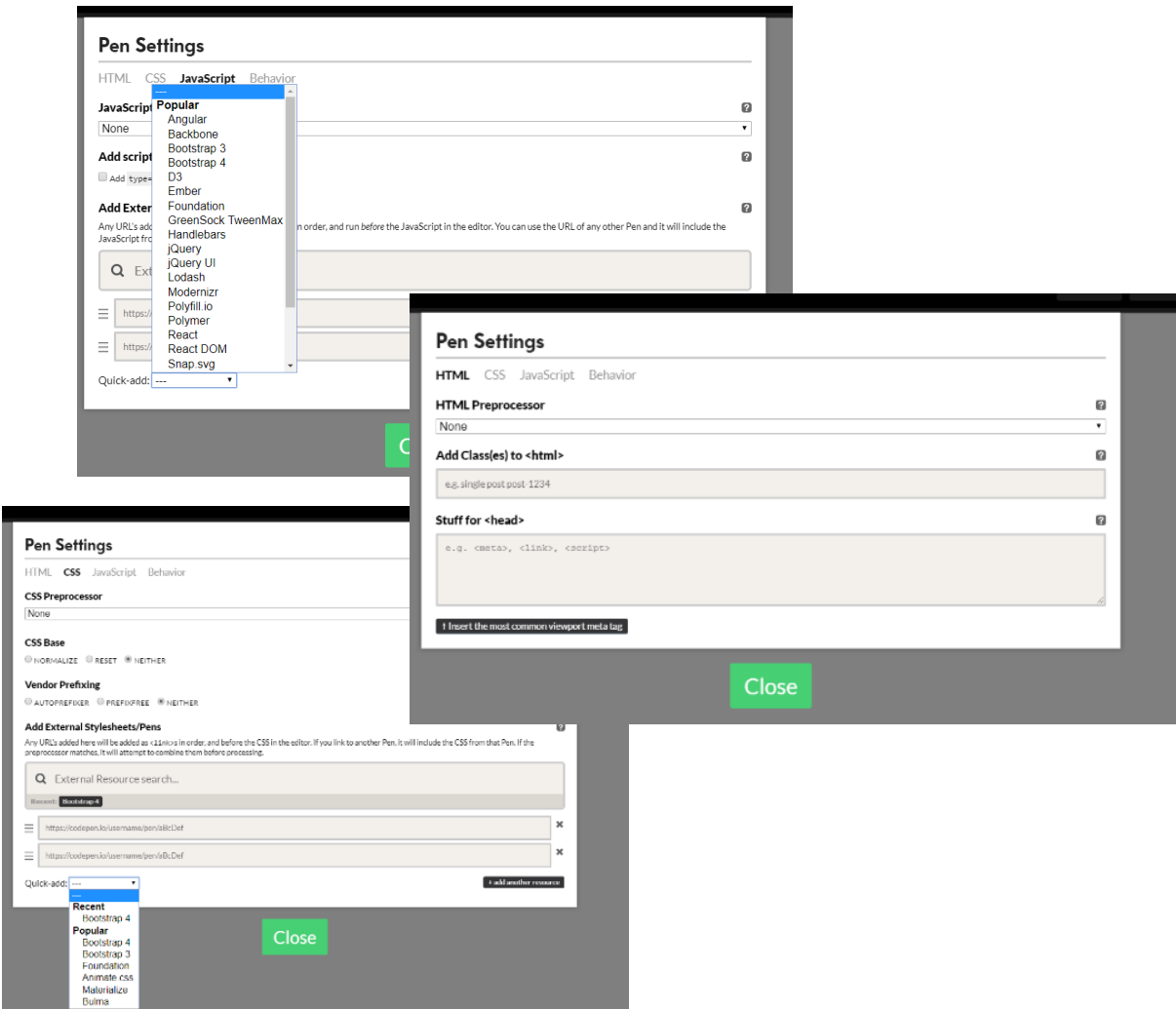
Néhány szó a CodePen-ről

A CodePen lehetőséget kínál arra, hogy egy időben szerkesszük és megjelenítsük a html, css, js kódjainkat. A „pen-eket” menthetjük a profilunkba (teljesen ingyen) és bármikor módosíthatjuk, illetve weboldalakra is beágyazhatjuk. Az itt létrehozott kódok teljesen nyilvánosak, bárki számára megtekinthetők és másolhatók. (Előfizetéssel igényelhető a „kódok titkosítása”).



A CodePen HTML ablaka a `<body>...</body>` közé eső rész.

Ha a `<head>`-be szeretnél beszúrni valamit (pl. betűtípus betöltését, keretrendszerek CDN elérését...), azt a HTML felirat melletti kis fogaskerékre kattintva tudod megtenni.



A Bootstrap használata

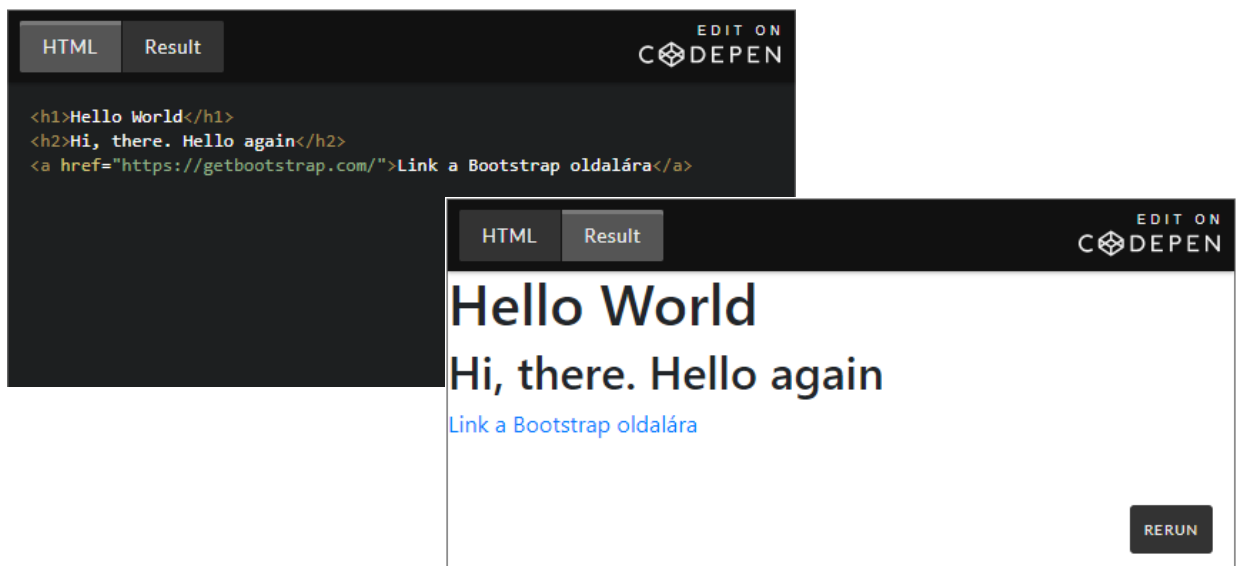
A Bootstrap nem tud dolgozni, ha nincs beillesztve a dokumentumodba. Több megoldást is kínálnak számunkra: letölthetjük vagy használhatjuk cdn-ről.

Mi az a CDN?

A CDN (Content Delivery Network) (magyarra fordítva tartalom elosztó hálózat) segítségével állítólag világszerte gyorsabb lesz a weboldalad betöltődési sebessége. A működési elve elméletben egészen egyszerű. Adott egy hálózat, amely a világ minden pontján elhelyezett szerverparkokból áll, és ezek mindegyikén ott vannak az adott statikus fájlok. A látogatóid mindig a hozzá legközelebbi szerverparkról fogják letölteni a tartalmat, és ha voltak már olyan oldalon, ahol ugyanaz a Bootstrap volt betöltve CDN-ről, mint amit mi is használunk, akkor ezt a mi oldalunkról már nem kell megtennie. Szóval hosszabb távon engedélyezi a fájl cachelését, így a böngészőnek nem kell folyamatosan lekérnie a fájlt a szerverről.

[A starter template itt találjátok](#), másoljátok ki, majd egyszerűen csak illesszétek a html dokumentumotokba.

Most nézzük meg a legelső (css nélküli) példánkat hogyan jeleníti meg a Bootstrap. A különbség nem nagy, viszont látható, hogy az alapértelmezett stílust miként írta felül. Megváltozott a betűcsalád, nagyobbak a sorközök, és a link is eltérően néz ki.



Mi került a html dokumentumba?

<!doctype>

Ez a DTD (Document Type Definition), azaz dokumentum típus meghatározás. Itt adjuk meg a weboldalunk nyelvtanát a böngésző számára.

<html> </html>

A html páros címkék közötti rész alkotja a böngésző által értelmezett weblapot.

<head> </head>

A fej, az adott html oldalra vonatkozó meta-adatokat tartalmazza, melyek a fájlal kapcsolatos jellemzők és utasítások tárolására szolgálnak. Néhány meta-adatnak saját címkéje van, a többi a meta-címke jellemzőjeként adható meg.

<title> </title>

Ide írjuk a weboldalunk címét. Ez a címsorban, illetve a böngészőben a kedvencek és könyvjelzők listaelemeinél lesz látható.

<meta charset=„utf-8”>

Az univerális betűkészletet jelöli.

<body> </body>

Ezek közé a címkék közé fog kerülni az a tartalom, amit a böngésző megjelenít, és a css-sel való formázásunk is csak erre a részre fog vonatkozni.

<script> </script>

A script címkék közé a futtatni kívánt Bootstrap JavaScript forráskódjai kerültek, az src itt is a hivatkozási helyre utal és a body aljára kerülnek.

Responsive weboldal

Az okostelefonok és tablet-ek elterjedésével felmerült az igény arra, hogy a weboldalak minden kijelzőméreten az optimális élményt nyújtsák.

Ezt a problémát reszponzív weboldalak készítésével kerülhetjük el, amelynek nagyon hatékony eszköze a Bootstrap.

Kijelzőmérettől függően kerülnek elhelyezésre a weboldal egyes elemei: például míg desktop nézetben a fejléc alatti hasábok egymás mellé kerülnek, addig mobilon már egymás alá rendeződnek.

Konténerek

A containerek képezik az oldalunk fő tárolóelemeit. Magukba foglalják a tartalmat, margókat adnak és segítenek a középre igazításban, tehát rendezettséget biztosítanak. Két fajtája van: a container és a container-fluid.

„Folyékony” konténer (`div class="container-fluid"`)

A fluid container 100%-ig kitölti az oldal szélességét és mindig a böngésző ablak méretéhez igazodik

`div class="container-fluid"`

Fix szélességű container (`div class="container"`)

A fix szélességű konténer mérete mindig az aktuális böngésző méretétől függ. Ez a szélesség töréspontonként változó, de két töréspont között mindig fix szélességű. Ha f12-vel előhívjuk a fejlesztői eszköztárat, és elkezdünk játszani a szélességi értékekkel, megfigyelhetjük, hogy a töréspontok közötti értékeken kizárólag az oldalsó külső margók mérete változik. Amikor töréspontot ér a konténerünk, akkor kisebb vagy nagyobb lesz, majd ismételten csak a külső margók mérete fog változni, egészen a következő törésig.

`div class="container"`

oldalsó margók

Töréspontok/Media query-k

töréspontok: azok a szélességi értékek, amelyeken azt szeretnénk, hogy más stílust kapjon az oldalunk.

a Bootstrap 5 féle kijelzőméretet különböztet meg:

xs: <576px (portrait phones, ez az alapértelmezett méret)

sm: ≥576px (landscape phones)

md: ≥768px (tablet)

lg: ≥992px (desktop)

xl: ≥1200px (large desktop)

A width tulajdonsággal adjuk meg az aktuális megjelenítési felület szélességi értékét. A lenti feltétel akkor fog életbe lépni, ha a kijelző mérete minimum 576px.

ha ez IGAZ

Akkor ez a css lép életbe

```
@media (min-width: 576px){ ... }
```

Megadhatunk minimum, maximum és minimum és maximum szélességi értékeket.

```
@media (max-width: 576px){ ... }
```

```
@media (min-width: 576px) and (max-width: 576px) { ... }
```

Színek

<https://getbootstrap.com/docs/4.0/utilities/colors/>

.bg-primary

.bg-secondary

.bg-success

.bg-danger

.bg-warning

.bg-info

.bg-light

.bg-dark

.bg-white

.text-primary

.text-secondary

.text-success

.text-danger

.text-warning

.text-info

.text-light

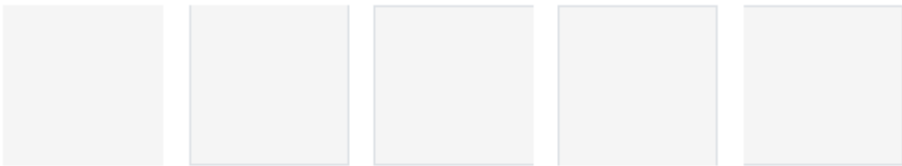
.text-dark

.text-muted

.text-white

Szegélyek

<http://getbootstrap.com/docs/4.0/utilities/borders/>



```
<span class="border-0"></span>
<span class="border-top-0"></span>
<span class="border-right-0"></span>
<span class="border-bottom-0"></span>
<span class="border-left-0"></span>
```


Bootstrap Grid

	1	2	3	4	5	6	7	8	9	10	11	12
.col	col	col	col	col	col	col	col	col	col	col	col	col
.row		col-4			col-4				col-4			
	col-6					col-6						
	col-4			col-4				col-2		col-2		
	col-5				col-7							
	col-6					col-5						col
	col-12											

Rácsrendszer a könnyű pozícionáláshoz

- a sorok (.row) a függőleges elhelyezésben segítenek (vertikális blokk)
- az oszlopok (.col) a soron (row) belül helyezkednek el, és a vízszintes elhelyezésben segítenek (horizontális blokk)
- minden sor 12 oszlopból áll
- az oszlopok szélességét úgy határozhatjuk meg, hogy a "col-" után azt a számot írjuk, ahány oszlopnyi helyet szeretnénk elfoglalni
- az alkalmazott oszlopszélességek összege nem haladhatja meg a 12-t (ha meghaladja, akkor a fennmaradó automatikusan az elemek alá kerül)
- a html kódunkban minden sor tartalmát egy .row osztállyal ellátott div elembe kell ágyazni:

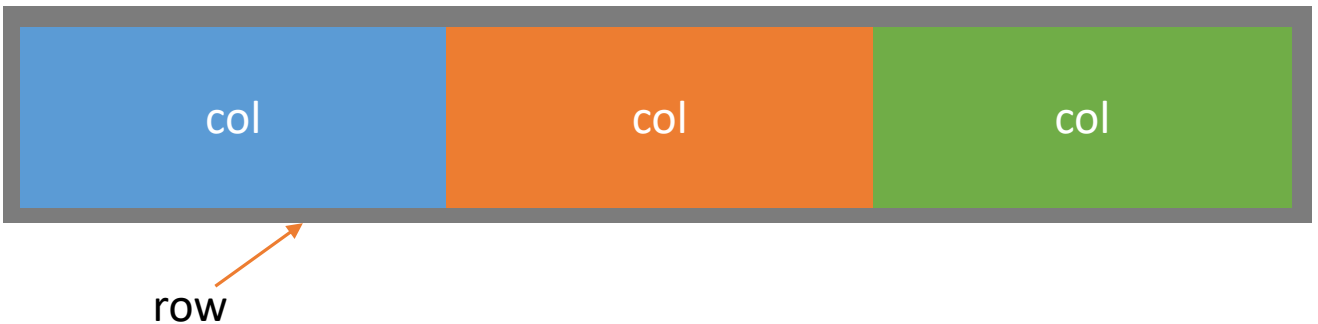
```
<div class="row"> <div class="col"></div> </div>
```

Egyenlő szélességű oszlopok

Ha a soron belüli "div" elemeinket a "col" osztállyal látjuk el, automatikusan azonos szélességeket kapnak (<div class="col"></div>).

Megfigyelhetjük, ahogy az elemek nagyobb kijelzőméreteken kiszélesednek, kisebbeken pedig szűkülnek.

```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```



Különböző szélességű oszlopok

Az oszlopok szélességét úgy határozhatjuk meg, hogy a "col-" után azt a számot írjuk, ahány oszlopnyi helyet szeretnénk elfoglalni.

```
<div class="row">
  <div class="col-4">col-4</div>
  <div class="col-6">col-6</div>
  <div class="col-2">col-2</div>
</div>
```



Oszlopszélesség a tartalom függvényében

Ha a div elemünket a col-auto osztállyal látjuk el, a tárolónk mérete automatikusan a szöveg méretéhez fog igazodni.

```
<div class="row">
  <div class="col-auto">az oszlopok</div>
  <div class="col-auto">szélessége a tartalomhoz igazodik</div>
  <div class="col-auto">col-auto</div>
</div>
```

az oszlopok

szélessége a tartalomhoz igazodik

col-auto

Különböző kijelzőméretek, különböző oszlopszélességekkel

5 féle kijelzőméret:

- XS (<576px)
- SM (>=576px)
- MD (>=768px)
- LG (>=992px)
- XL (>=1200px)

Ezek a méretezések lehetőséget biztosítanak arra, hogy a különböző kijelzőkre más-más oszlopszélességeket adjunk meg.

A területek minimális szélességeken alapulnak (mobil-first!), ezért pl a col-md-6 beállítás a közepes, kis, és extra nagy kijelzőkre fog vonatkozni.

```
<div class="row">
  <div class="col-12 col-sm-8 col-md-6">col-12, col-sm-8, col-md-6</div>
  <div class="col-12 col-lg-2">col-12, col-lg-2</div>
  <div class="col-12 col-xl-1">col-12, col-xl-1</div>
</div>
```

col-12, col-sm-8, col-md-6

col-12,
col-lg-2

col-12,
col-xl-1

A sor minden elemére érvényes függőleges igazítások

Felülre igazítás: `<div class="row align-items-start">`

```
<div class="row align-items-start">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```



Középre igazítás: `<div class="row align-items-center">`

```
<div class="row align-items-center">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```



Alulra igazítás: `<div class="row align-items-end">`

```
<div class="row align-items-end">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
```



Az oszlopok függőleges igazításai

```
<div class="row">  
  <div class="col align-self-start">felülre igazítás</div>  
  <div class="col align-self-center">középre igazítás</div>  
  <div class="col align-self-end">alulra igazítás</div>  
</div>
```

felülre igazítás

középre igazítás

alulra igazítás

A sor minden elemére érvényes vízszintes igazítások

elemek igazítása balra: <div class="row justify-content-start">

```
<div class="row justify-content-start">  
  <div class="col-4">col-4</div>  
  <div class="col-4">col-4</div>  
</div>
```

col-4

col-4

elemek igazítása középre: <div class="row justify-content-center">

```
<div class="row justify-content-center">  
  <div class="col-4">col-4</div>  
  <div class="col-4">col-4</div>  
</div>
```

col-4

col-4

elemek igazítása jobbra: `<div class="row justify-content-end">`

```
<div class="row justify-content-end">
  <div class="col-4">col-4</div>
  <div class="col-4">col-4</div>
</div>
```



elemek igazítása körben: `<div class="row justify-content-around">`

```
<div class="row justify-content-around">
  <div class="col-4">col-4</div>
  <div class="col-4">col-4</div>
</div>
```



elemek igazítása a sor két oldalára: `<div class="row justify-content-between">`

```
<div class="row justify-content-between">
  <div class="col-4">col-4</div>
  <div class="col-4">col-4</div>
</div>
```



Margók

A Bootstrap4 5 szintű külső és belső margó értéket különböztet meg, méretük 0,25rem-től 3rem-ig terjed.

külső margó: m
belső margó: p

ha pl. 1 rem külső margót szeretnénk beállítani, használjuk a m-3 osztálykijelölőt

szint	méret
0	0rem
1	0,25rem
2	0,5rem
3	1rem
4	1,5rem
5	3rem

margó(k) helye	Külső margó - margin	Belső margó - padding
körben	m - margin	p - padding
fent	mt - margin-top	pt - padding-top
lent	mb - margin-bottom	pb - padding-bottom
jobb oldalon	mr - margin-right	pr - padding-right
bal oldalon	ml - margin-left	pl - padding-left
vízszintesen és szimmetrikusan	mx - margin-x	px - padding-x
vízszintes középre helyezés	mx-auto	
függőleges középre helyezés	my-auto	
töréspontonként változó margók	pl. ml-md-5	pl. pt-md-5