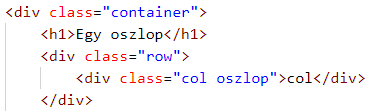
# Grid

A weblapokon gyakran egymás mellé szeretnénk helyezni szakaszokat. Erre szolgál a Bootstrap-ben a grid (rács).

Nyisd meg a grid mappát a Visual Studio Code-ban és a böngészőben!

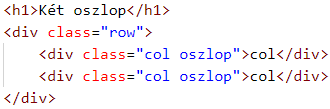
A szakaszokat egy sorba (row osztály) kell tenni, és a col osztály valamelyik változatával jelezni a szakasz szélességét. (A példákban az oszlop osztály csak a háttérszínt és a szegélyt állítja be.)

Nézzük az első példát:



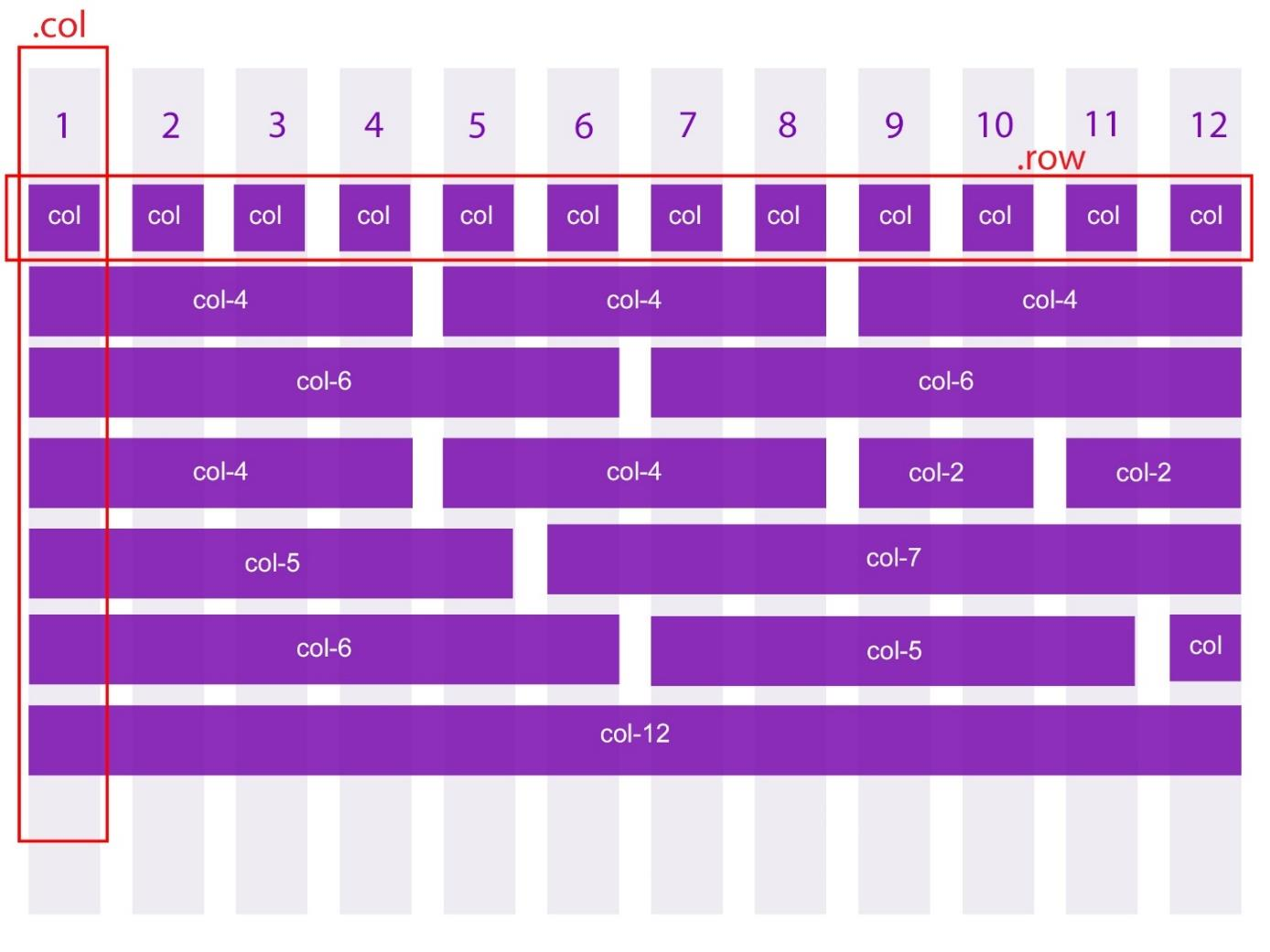
Ebben az esetben egy oszlop fog megjelenni minden szélességnél.

A következő példában két ilyen szakaszt teszünk egymás után. Ezek mindig egymás mellett fognak megjelenni, és ugyanolyan szélesek lesznek.

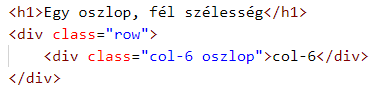


A col osztálynak vannak olyan változatai, ahol a col után kötőjellel egy 1 és 12 közötti számot adunk meg (például col-6).

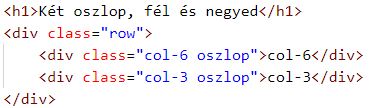
A Bootstrap a képernyő teljes szélességét 12 részre osztja fel. A kötőjel utáni szám azt jelzi, hogy a szakasz hány részt foglaljon el.



A következő példában a szakasz az első hat részt foglalja el (mellette nem lesz semmi):



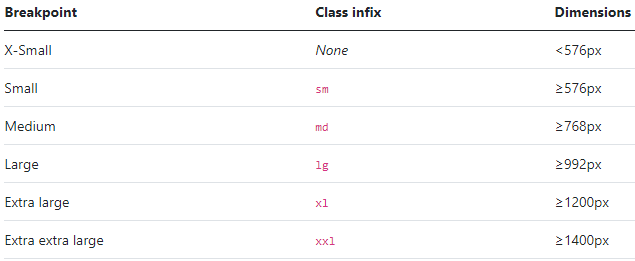
Ezután megadhatunk további szakaszokat is. Mennyit foglal el a következő példában a második szakasz?



Ami nem fér ki egy sorba, az a következő sorba kerül.

Az eddigiek minden kijelzőméretre vonatkoznak, de a Bootstrapben a különböző kijelzőkhöz különböző oszlopszélességeket is megadhatunk.

A lehetséges kijelzőméretek:



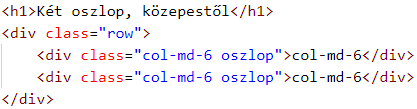
A beállításoknál a kijelzőméretet rövidítve adjuk meg (pl. -md-), és a megadott mérettől felfelé is érvényes, amíg nem adunk meg mást.

Például a "col-md-3" 768 pixeltől felfelé adja meg, hogy három részt foglaljon el az oszlop.

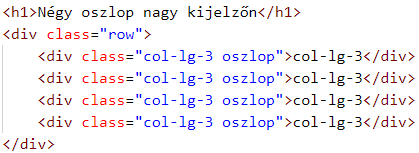
Az extra small kijelzőt nem jelezzük. Eddig erre adtuk meg a szélességeket, vagyis az összes kijelzőre érvényesek voltak.

Az alapértelmezett érték "col-12", ezt nem kell megadnunk. Ha csak a közepes vagy a nagy kijelzőre adunk meg méretet, akkor a kisebbekre az alapértelmezett "col-12" lesz érvényes (vagyis a teljes szélességet kitölti).

Mit jelent a "col-md-6" osztály? Próbáld ki a következő példában különböző ablakszélességeknél!



A következő példában mekkora kijelzőn lesz négy oszlopos a megjelenítés?

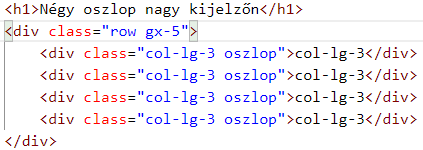


Ugyanarra az oszlopra több szélességet is megadhatunk. Például a   
"col-md-4 col-lg-3" osztály 768 képpont alatt 12 résznyi szélességet ad meg, 768‑tól 992 px-ig 4 résznyit, afölött pedig 3 résznyit. Vagyis telefonon egy oszlop lesz, tableten 3, laptopon és asztali gépen 4.

Írd át ilyenre az előző példát, és próbáld ki!

Az oszlopok távolságát a gx-? osztály segítségével tudjuk módosítani. Az alapértelmezett érték 1.5rem, ami, ha nem állították a böngésző betűméretét, akkor 24px. A kérdőjel helyére 0 és 5 közötti számot adhatunk meg. (1: 0.25rem, : 2: 0.5rem, 3: 1rem, 4: 1.5rem, 5: 3rem)

Próbaképpen állíts be az utolsó sorban gx-5 értéket:



Figyeld meg, hogy a sor teljes hossza megváltozik, kilóg a containerből!



# Margók

A külső és belső margók beállításához használhatunk rövidítés osztályokat.

Például az "m-3" osztály 3-as külső margót állít be az elem minden oldalán. A méret 0 és 5 között lehet, és itt sem pixelben értendő, hanem úgy, mint az oszlopok távolságánál, de itt lehet auto is.

Ehhez hasonlóan a "p-2” osztály 2-es méretű paddinget állít be az elem körül.

Ha nem minden oldalon szeretnénk, a következő betűket tegyük az m vagy a p után:  
t – top, b – bottom, s – start (bal), e – end (jobb), x – start and end, y – top and bottom.

Ha egy elem egy oldalán van auto margó, akkor azon az oldalon maximális lesz a margó mérete. Ha két oldalán (mx-auto), akkor vízszintesen lesz középre igazítva, ha alul és felül (my-auto), akkor pedig függőlegesen.

Mit jelentenek a következő osztályok?

mt-0, ms-1, px-2, py-3, mx-auto, my-auto

A kijelzőméreteket nem csak az oszlopok szélességénél lehet alkalmazni, hanem a margónál és a paddingnél is.

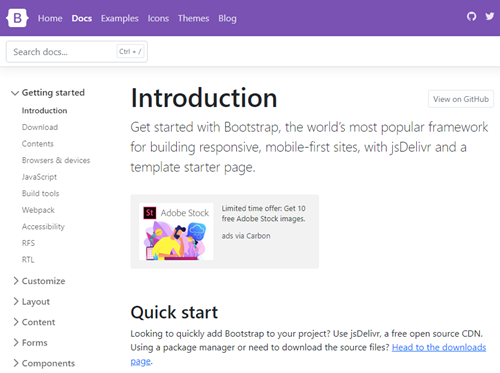
Például "my-5 my-md-0" azt jelenti, hogy a közepes (768px alatti) szélességeknél alul és felül 5 egységnyi margó lesz, nagyobb kijelzőknél viszont 0 (vagyis nem lesz).

## Dokumentáció

Ha utána szeretnél nézni bárminek a Bootstrap-ben, a dokumentációban megteheted:  
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>

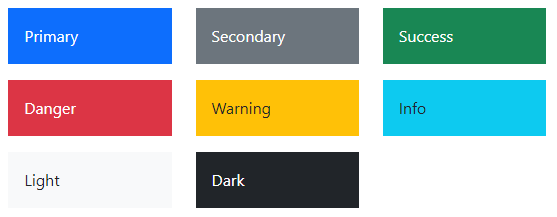
Például a margókról itt találsz információt:   
<https://getbootstrap.com/docs/5.0/utilities/spacing/>

A dokumentáció bal oldalán választhatsz témát, vagy használhatod a keresőt is.



# Színek

A Bootstrap-ben alapértelmezetten használható színeket az alábbi ábra mutatja (a *color* szóra keresve megtalálod a dokumentációban):

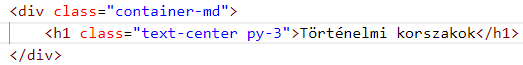
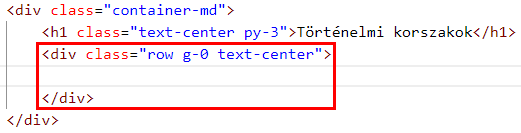
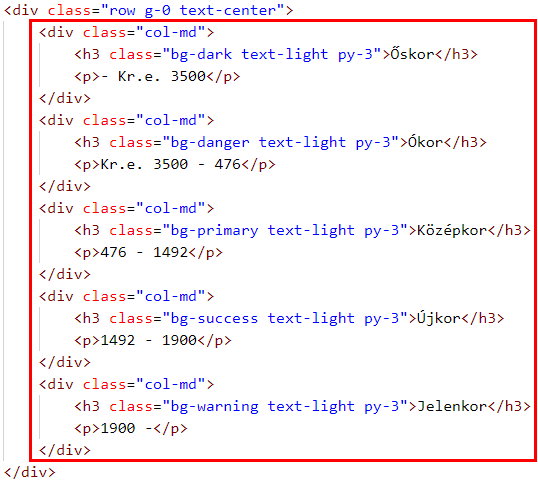
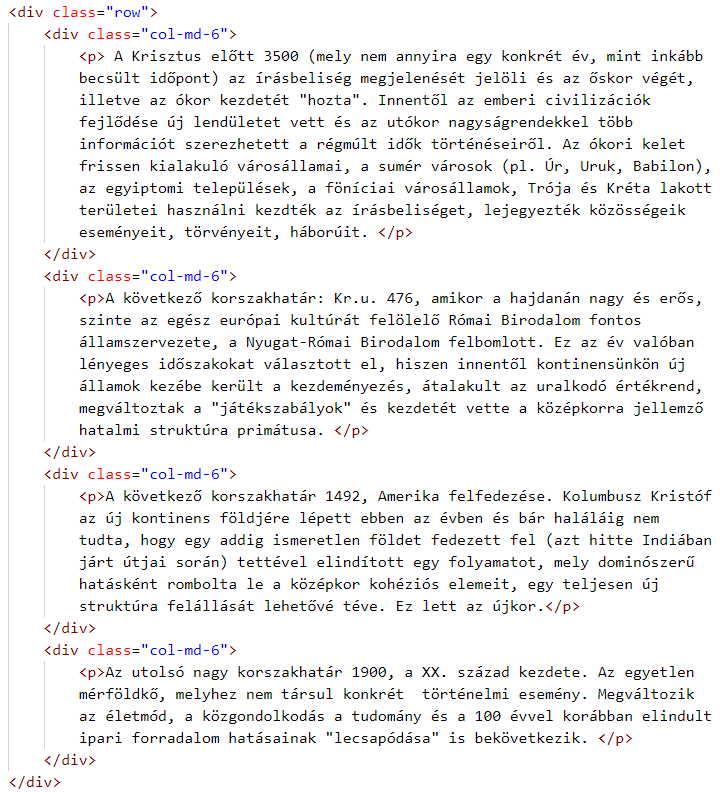
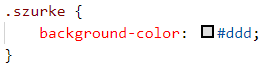
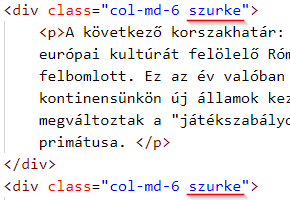
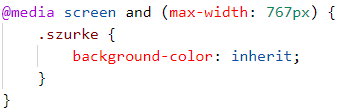


A színeket más osztályokkal együtt használhatjuk. Ha például vörösre szeretnénk állítani egy elem hátterét, akkor a bg-danger osztályt alkalmazzuk az elemre. Erre a következő feladatban láthatsz példákat.

# Alkalmazás

Ezután alkalmazzuk a tanultakat egy példán! A bal oldalon számítógép képernyőjén, a jobb oldalon telefonon látható az elkészítendő oldal:

1. Nyisd meg a kapott korok nevű mappát a Visual Studio Code-ban!
2. Nézd meg a kapott HTML fájlt! Figyeld meg, hogyan csatoltuk a szükséges Bootstrap fájlokat és a stilusok.css fájlt! Fontos, hogy a stilusok.css fájlt a Bootstrap CSS fájlja után csatoltuk, mert így felülírhatjuk a Bootstrap stílusait (és fordítva nem)!
3. A fenti két elrendezés között 768 képpont szélességnél, az lg méret határánál szeretnénk váltani. Nagyobb szélességnél kell a két oldali margó, kisebbnél nem, ezért először egy tárolót készítünk. A további elemeket ebben fogjuk elhelyezni.  
   
4. A tárolóban először elhelyezünk egy első szintű címsort, amelynek szövege középre van igazítva, és belső margója alul-felül 3:  
   
5. A tárolóban ezután készítünk egy sort a korok neveinek. Az oszlopok között nem lesz üres hely, és a sorban minden szöveg legyen középre lesz igazítva!  
   
6. Ezután jönnek az oszlopok az egyes korokhoz. A korok md mérettől kezdve egymás mellett, kisebb méretnél egymás alatt lesznek. (col-md)  
   Minden kornál van egy h3-as cím és egy bekezdés. A cím háttere minden kornál más, a betűszín világos (fehér) és alul-felül 3 egységnyi padding van.  
   Készítsd el oszlopokat így, majd próbáld ki a böngészőben különböző ablakszélességekkel: (Másold, amit lehet!)  
     
   Figyeld meg, hogy most 5 oszlopot hoztunk létre! Ezt csak úgy lehet, ha a col-md után nem adunk meg számot, mert a 12 részt nem tudjuk 5-felé osztani.
7. A többi szöveg a képernyő méretétő függően 1 vagy 2 oszlopba kerül, ezért ezeknek új sor kell az előző sor után a tárolóban.  
   Az md méret alatt egy oszlopban lesz a szöveg, fölötte kettőben (col-md-6) Készítsd el az új sorban az oszlopokat így (A szöveget másold a forras.txt fájlból!):  
     
   Próbáld ki!
8. A második és a harmadik oszlop háttérszínét szeretnénk szürkére állítani, de a Bootstrap színei között nincs megfelelő. Ezért készítünk egy saját stílus osztályt szurke néven a stilusok.css fájlban:  
   ,  
   majd ezt hozzárendeljük a második és a harmadik oszlophoz:  
   
9. A szürke háttérre azonban csak az md mérettől felfelé (768 képponttól) van szükség, alatta nem. Saját stílusok után nem írhatjuk az -md-t, ezért a stilusok.css fájlt kiegészítjük egy médialekérdezéssel:  
     
   Az inherit az örökölt értéket jelenti. Jelen esetben a body színét örökli a sor, a sor színét pedig az oszlop.