Bootstrap

# Mi az a Bootsrap?

A Bootstrap egy elterjedt HTML, CSS, JS keretrendszer, melyben reszponzív weboldalakat fejleszthetünk könnyen és gyorsan. <https://www.w3schools.com/bootstrap/default.asp>

HTML és CSS sablonokat tartalmaz, melyben meghatározza a tipográfiát, az űrlapok, táblázatok, navigáció kinézetét, látványos képi és design elemeket helyezhetünk el az oldalunkon nagyon egyszerűen.

A Bootstrap könyvtárat le is tölthetjük (<https://getbootstrap.com/docs/4.5/getting-started/introduction/> ) és bemásolhatjuk saját weboldalunk könyvtárrendszerébe, vagy használhatjuk az online CDN-t.

Az online CDN könyvtárt a HTML head részében kell az oldalunkhoz csatolni:

### Bootstrap 3 esetén:

<!-- Latest compiled and minified CSS -->  
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">  
  
<!-- jQuery library -->  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
  
<!-- Latest compiled JavaScript -->  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

### Bootstrap 4 esetén:

<!-- Latest compiled and minified CSS -->  
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
  
<!-- jQuery library -->  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>  
  
<!-- Popper JS -->  
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>  
  
<!-- Latest compiled JavaScript -->  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

### Bootstrap4 vagy Bootstrap3?

A BOOTSTRAP4 és BOOTSTRAP3 közötti legnagyobb különbség, hogy a 4 flexbox technikát használ az elemek úsztatásához, ezért fix szélességű elemekkel nem tud dolgozni.

## Hogy működik?

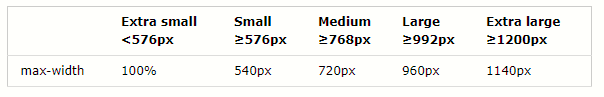
Helyezzük el a dokumentumunk fejlécében a Bootstrap css-re való hivatkozást.

<!DOCTYPE html>  
<html lang="en">  
<head>  
  <title>Bootstrap Example</title>  
  <meta charset="utf-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1">  
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">  
  
</head>  
<body>  
  
<div class="container-fluid">  
  <h1>My First Bootstrap Page</h1>  
  <p>This is some text.</p>  
</div>  
  
</body>  
</html>

Építsük fel a HTML oldalt az elképzelésünk szerint, majd használjuk a Bootstrap osztályhivatkozásokat.

A fő tároló elemhez rendeljük a .container vagy a .container-fluid osztályt.

.container használata esetén reszponzív, fix szélességű tárolót kapunk:



.container-fluid esetén a teljes böngésző szélességet kitölti a tárolónk.

Az oldal formázásához további alapbeállításokat használhatunk (<https://www.w3schools.com/bootstrap4/bootstrap_containers.asp> ):

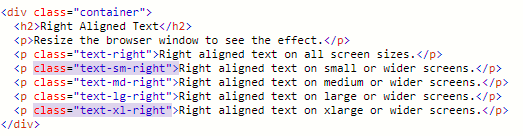
<div class="container p-3 my-3 bg-dark text-white"></div>

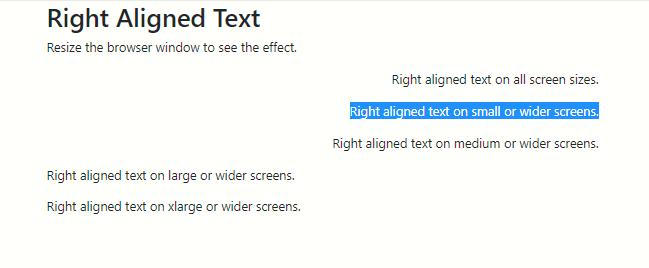
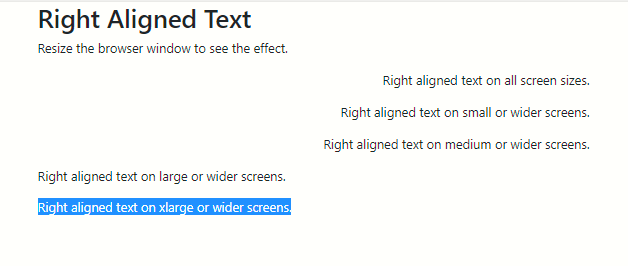
#### Typográfia

További információ a W3Scool oldalán: <https://www.w3schools.com/bootstrap4/bootstrap_typography.asp>

Például nézzük meg a szöveg igazításának lehetőségeit! Hasonlítsuk össze a két osztály működését:

class="text-sm-right" 🡨 🡪 class="text-xl-right"



#### Színek használata:

A szöveg funkciója, mondanivalója alapján használhatunk különböző színű szövegeket.   
<https://www.w3schools.com/bootstrap4/bootstrap_colors.asp>

#### Különböző stílusú táblázatok

<https://www.w3schools.com/bootstrap4/bootstrap_tables.asp>

#### Képek

<https://www.w3schools.com/bootstrap4/bootstrap_images.asp>

#### Navigáció

<https://www.w3schools.com/bootstrap4/bootstrap_navbar.asp>

#### Caroussel

<https://www.w3schools.com/bootstrap4/bootstrap_carousel.asp>

#### Media

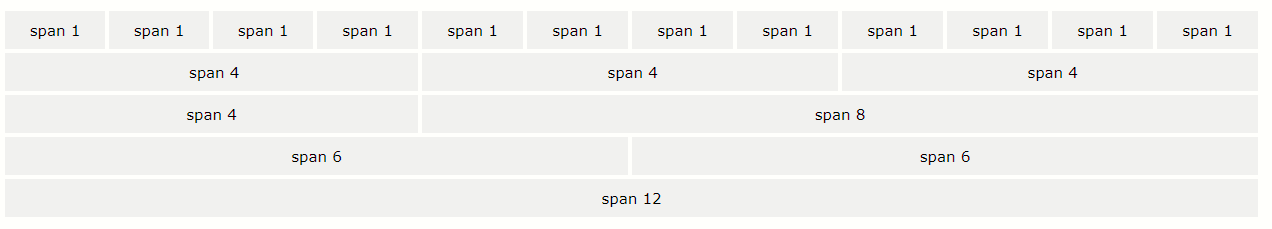
<https://www.w3schools.com/bootstrap4/bootstrap_media_objects.asp>

#### FORM

<https://www.w3schools.com/bootstrap4/bootstrap_forms.asp>

## Bootstrap grid használatának szabályai

A bootstrap keretrendszer 12 egyforma oszlopra osztja a rendelkezésre álló teret. Azt kell megondolnunk, hogy az általunk elképzelt oldalszerkezet egyes elemei hány oszlop szélesek legyenek.



Az alábbi példában látható, hogy a fejléc szélessége 12 oszlop, a menüé 2 oszlop, a tartalomé 6 oszlop, és a szélső elemé 4 oszlop, míg a lábléc szélessége 12 oszlop.

|  |  |  |
| --- | --- | --- |
| fejléc | | |
| menü | tartalom | oldalsó elem (aside) |
| lábléc | | |

A Bootstrap grid rendszere reszponzív, azaz a képernyő méretétől függően változik az elrendezés.

Bootstrap3 szabvány kimenetek méretei:

* .col-XS-(telefon - a képernyő mérete kevesebb, mint 768px)
* .col-sm- (tablet - smal devices - a képernyő mérete nagyobb, mint 768)
* .col-md- (laptop - medium devices - a képernyő mérete nagyobb, mint 992px)
* .col-lg- (asztali gép - large devices - a képernyő mérete nagyobb, mint 1200px)

Bootstrap4 szabvány kimenetek méretei:

* .col- (a képernyő mérete kevesebb, mint 576px)
* .col-sm- (smal devices - a képernyő mérete nagyobb, mint 576px)
* .col-md- (medium devices - a képernyő mérete nagyobb, mint 768px)
* .col-lg- (large devices - a képernyő mérete nagyobb, mint 992px)
* .col-xl- (xlarge devices - a képernyő mérete nagyobb, mint 1200px)

### A szabályok:

1. Az oszlopokat mindig egy .container vagy egy .container-fluid osztálykijelölővel ellátott tárolóban kell lehelyezni.
2. Új oszlopelrendezés esetén használjuk újabb tárolót, melyhez a row osztálykijelölőt rendeljük.
3. Az oszlopok szélességét százalékosan határozza meg a Bootstrap, azaz a szülő elemhez képest mindig relatív szélességet használunk.

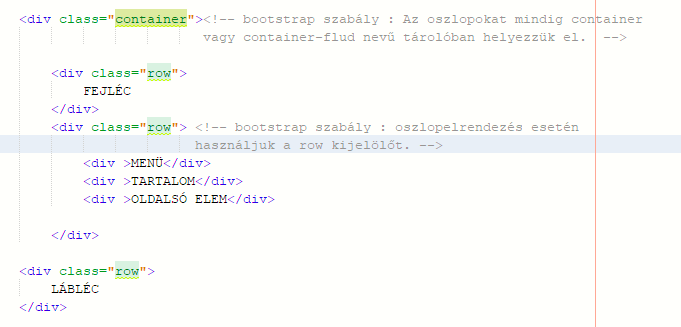
### Mit jelent ez? Nézzük az alábbi példát:

Készítsük el az alábbi oldalszerkezetet:

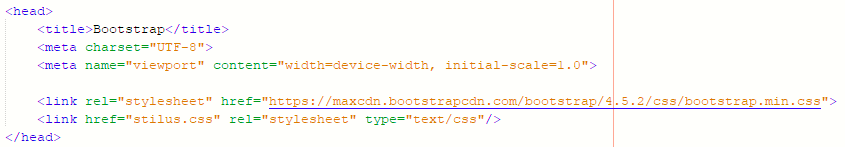
|  |  |  |
| --- | --- | --- |
| fejléc | | |
| menü | tartalom | oldalsó elem (aside) |
| lábléc | | |

Mint fentebb láthattuk a fejléc 12 oszlop szélességű, a menü két oszlopos, a tartalom szélessége 6 oszlop, a szélső elemé 4. A láblécé újra 12.

Ehhez alakítsuk ki a szokásos oldalszerkezetet (div-ekkel, vagy html 5 szemantikus tároló elemekkel).

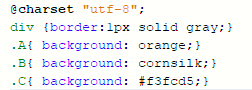


A HTML dokumentum head elemébe helyezzük el a bootstrap css-re való hivatkozást.

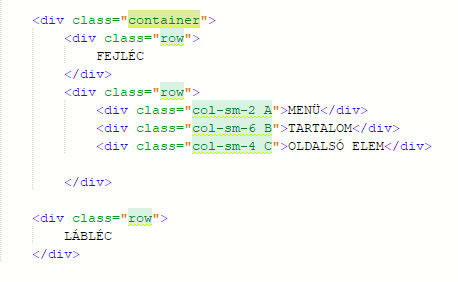


Valamint saját stíluslapot is csatolhatunk az oldalhoz.

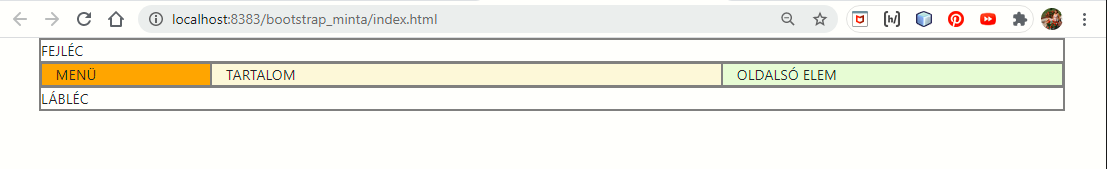
A saját stíluslapon egyenlőre állítsunk szegélyt a tárolóknak, hogy lássuk, mi történik. Valamint hozzunk létre A, B, C, D osztályjelölővel ellátott stílusosztályokat, ahol a háttérszíneket fogjuk beállítani, szintén a jobb láthatóság kedvéért.



Ezután adjuk hozzá az osztályokat a HTML tárolókhoz! Figyeljük meg, hogy ha több osztályt is hozzá akarunk rendelni egyetlen elemhez, akkor a class attribútumot csak egyszer használjuk, és az idézőjelbe szóközzel elválasztva írjuk az osztályok neveit.



Az eredmény a böngészőben megtekinthető:



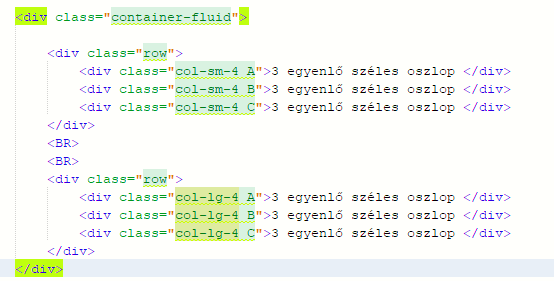
A .container osztály miatt az oldal adott szélességű, nem fut végig a böngésző teljes széleségén, mint azt a .container-fluid osztálynál tenné.

A menü és a tartalom és az oldalsó elem aránya a fentebb leírtak szerinti.

Méretezzük a böngészőnket! Amikor az ablak szélessége kisebb lesz, mint 576px, az elemek automatikusan egymás alatt helyezkednek el.

### Mi a különbség az sm, md, lg, xl osztályok között?

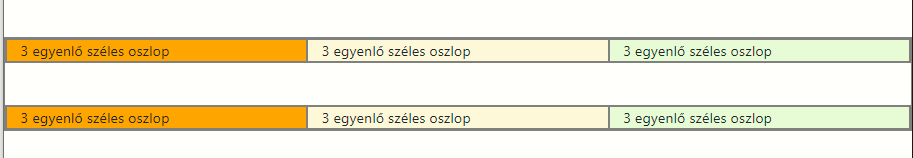
Nézzük az alábbi elrendezést:



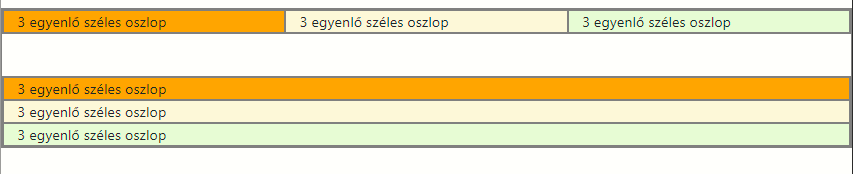
Most a container fluid osztályt használtuk, ezért a tartalom kitölti a böngésző teljes szélességét.

A két .row osztályhoz rendelt tároló tartalma csak abban különbözik egymástól, hogy az egyikben a div-ekhez a col-sm-4 osztályt rendeltük, a másikban pedig a col-lg-4-et.

Első pillantásra a böngészőben sem látszik a különbség:

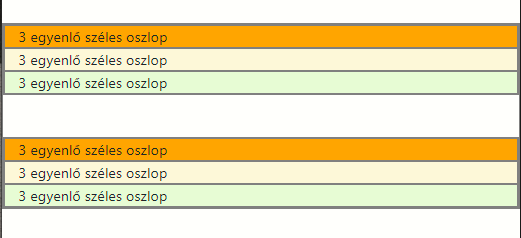


De méretezzük csak a böngészőt! Mi történik, ha a böngésző szélessége 992px alá csökken?



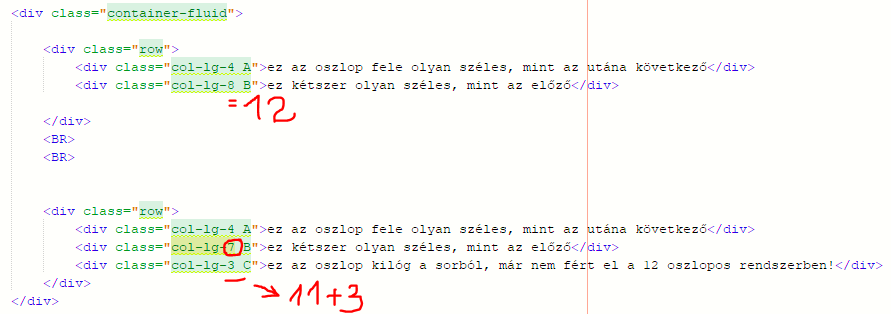
A felső tárolóban lévő div-ek még megtartják a 3 oszlopos elrendezésüket, az alsók azonban egymás alá kerülnek.

Amennyiben a böngésző méretét tovább csökkentjük 576px alá, az első blokk div elemei is egymás alá kerülnek.

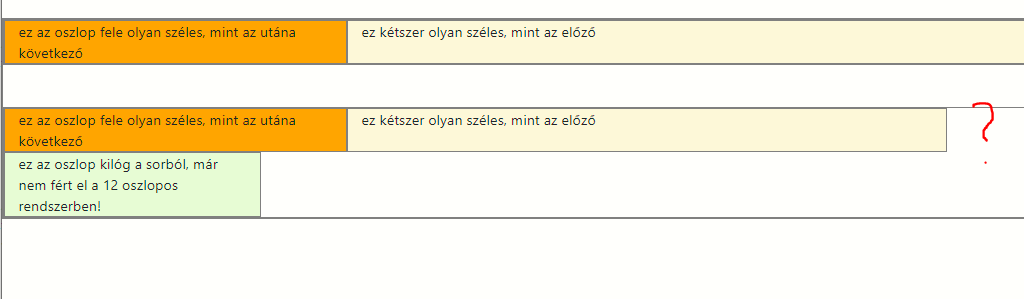


### Mi történik akkor, ha elrontjuk az oszlopok szélességét, és nem jön ki a matek?

Az alábbi példában ezt is kipróbálhatjuk:

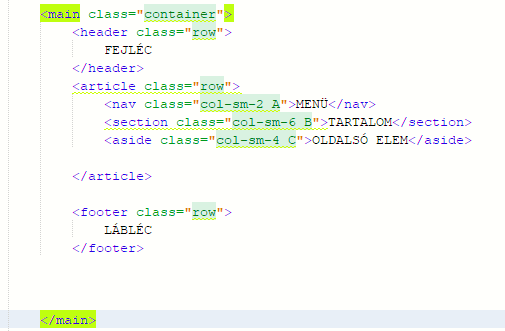


A felső blokkban szépen elrendezve jelennek meg az oszlopok, az alsó viszont több helyen is hibás:

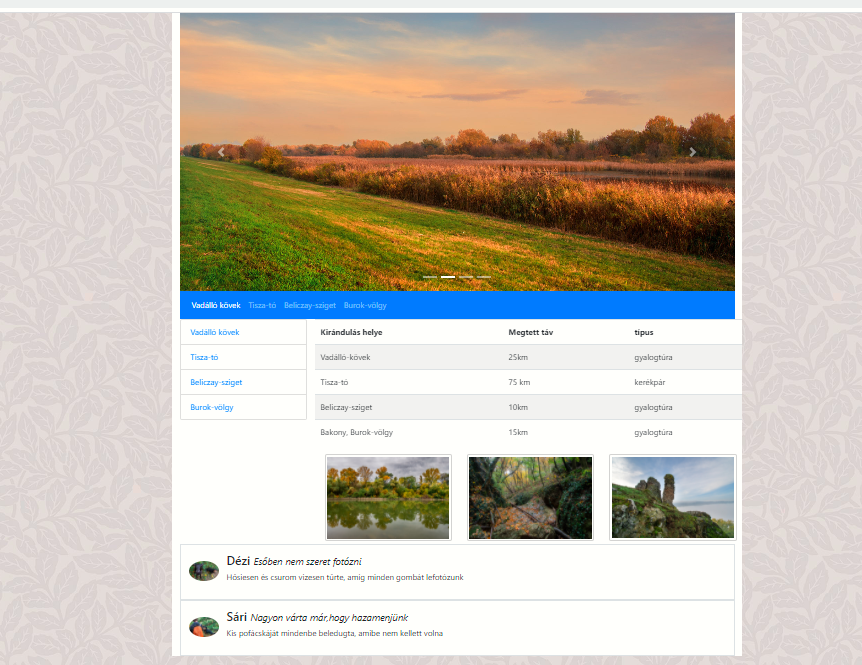


### Használhatjuk a html 5 szemantikus tárolót? header, nav, section, aside, stb?

Természetesen!



# Feladat: Készítsd el az alábbi oldalt Bootstrap4 segítségével



1. A felső képsor egy caroussel. <https://www.w3schools.com/bootstrap4/bootstrap_carousel.asp>
2. Alatta navigációs menü (kék). <https://www.w3schools.com/bootstrap4/bootstrap_navbar.asp>
3. A bal oldalon felsorolások <https://www.w3schools.com/bootstrap4/bootstrap_list_groups.asp> , benne linkek.
4. A tartalom egy táblázat <https://www.w3schools.com/bootstrap4/bootstrap_tables.asp> .
5. Alatta képek egymás mellé rendezve bootstrap row/col-sm-4 osztályokkal. <https://www.w3schools.com/bootstrap4/bootstrap_grid_system.asp>
6. A láblécben média típusú bootstrap elemek vannak. <https://www.w3schools.com/bootstrap4/bootstrap_media_objects.asp>