# Bevezetés

Ebben a leckében megismerkedünk a Bootstrappel. A legújabb 4-es (pontosabban 4.3-as) változatot fogjuk tanulni.

A Bootstrap egy CSS framework (keretrendszer). Ez azt jelenti, hogy:

* Beállítja az alap stílusokat (betűméret, …).
* Rendelkezésünkre bocsát rengeteg előre definiált CSS osztályt, amelyet felhasználhatunk az oldalainkon. Természetesen ezek mellett használhatunk saját stílusokat is, sőt a kapottakat is felülírhatjuk velük.
* Rendelkezésünkre bocsát egy grid (rács) rendszert, amelynek segítségével könnyen és reszponzívan elhelyezhetjük az oldalak elemeit. (A Bootstrap grid a Flexboxon alapul:-)
* Kapunk még egy sor komponenst is (pl. navigáció, fejléc, …), amelyeket az osztályok segítségével alakíthatunk.

Néhány komponens (pl. menü) működéséhez Javascript is szükséges. Ezt szintén tartalmazza a Bootstrap.

A Bootstrap előnyei:

* Alap HTML és CSS ismeretekkel is könnyen használható.
* Kevés munkával is látványos oldalakat hozhatunk létre.
* Reszponzív oldalakat készíthetünk vele.
* Minden modern böngészővel kompatibilis.

# Reboot

Korábban tanultuk, hogy a *normalize.css* egységesre állítja az alapértelmezett stílusokat a különböző böngészőkben, így oldalaink mindegyikben ugyanúgy jelennek meg.

A Bootstrapben továbbfejlesztették a *normalize.css*-t *Reboot* néven. Ebben van néhány lényeges újdonság, amelyekre a stílusok írásánál figyelnünk kell:

* Minden elemre beállítja a box-size: border-box; stílust. Ez azt jelenti, hogy a dobozok szélességébe és magasságába beleszámítanak a külső és belső margók és a szegélyek is.
* Mértékegységként a rem-et (=root em) használja. Ez a body vagy a HTML elemben megadott betűmérethez képest adja meg a méretet. Ha például a body-ban 100%-os (általában 16px) méretet adtunk meg, akkor 2rem = 32px.   
  Ha a böngészőben átállítják az alapértelmezett betűméretet, akkor ez is változik.   
  (Abban különbözik az em-től, hogy az az adott elemre beállított betűmérethez viszonyít.)
* Annak az operációs rendszernek a betűtípusait használja, amelyen futtatjuk.
* A szöveges elemeknek (címsorok, bekezdések, felsorolások) nincs felső margójuk.

# Grid

Az elemeket mindig egy tárolóba (container) tesszük.

A tárolóban lehetnek sorok (row).

A sorokban megadhatunk tetszőleges számú oszlopot (column).



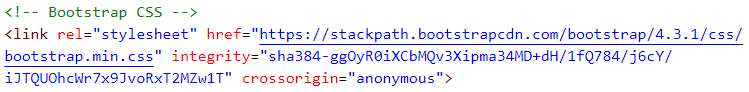
Az egyes sorokban különböző oszlopszámok lehetnek.

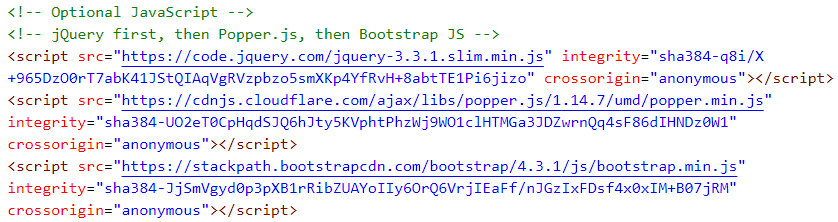
(Részletesen lásd a harmadik leckében.)

# Első Bootstrap projekt

1. Készíts egy új mappát *proba* néven!
2. Nyisd meg a Visual Studio Code-ban!
3. Hozz létre benne egy új fájlt *index.html* néven!
4. Nyisd meg a <https://getbootstrap.com/> oldalt a böngészőben, nyomd meg a Get started gombot, és a megjelenő oldalról másold a starter template-et az *index.html* fájlba!
5. Írd át az oldal nyelvét magyarra a html tag-ben!
6. Írd át az oldal címét „Próba”-ra!
7. Nézd meg az oldalt a böngészőben!

Nézzük meg alaposabban a bemásolt kódban, hogyan került a Bootstrap a projektbe!

A fejlécben egy CSS fájlt importáltunk. Ez tartalmazza a teljes CSS kódot.  


A body végén pedig három Javascript fájlt adtunk meg. ezek tartalmazzák a Javascript kódot. Azért van három fájl, mert a Bootstrap használja a jQuery és a Popper Javascript könyvtárakat is.  


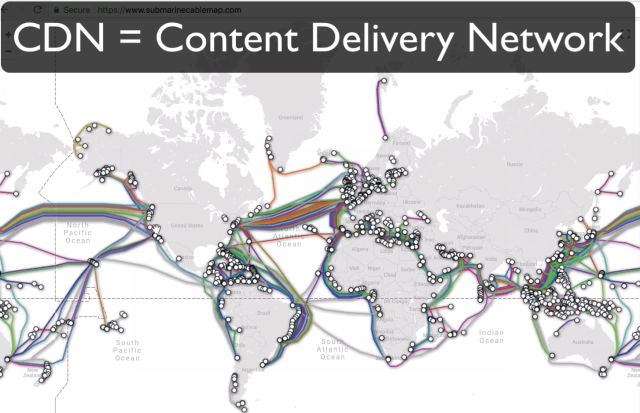
A CSS és a JavaScript részt külön-külön is bemásolhattuk volna a dokumentációból.

Figyeld meg, hogy a CSS és a Javascript fájlokat nem töltöttük le, hanem egy CDN szolgáltató szerveréről tölti be az oldalunk.

A CDN szolgáltatóknak a világ minden pontján vannak gyors szervereik, és így oldalunkat gyorsabban be lehet tölteni. Ha például Amerikában szeretnék megnyitni az oldalunkat, a Bootstrap fájljait egy közelben lévő gyors szerverről tölthetik be.

Ráadásul sokan használják ezeket a fájlokat, és ha a felhasználó gépére már le vannak töltve, akkor nem kell őket újra letölteni.

Az alábbi térkép a CDN szolgáltató szervereit mutatja:



Van más módszer is. Ha akarjuk, letölthetjük a fájlokat a dokumentáció Download oldalán az első a Download gombra kattintva. Ekkor egy ZIP fájlt kapunk, amit ki kell csomagolni a projekt mappájába, és a HTML fájlban el kell készíteni a megfelelő link és script sorokat. Ez sokkal több munkával jár, és az oldal betöltése is lassabb lesz, ezért inkább az első módszert alkalmazzuk.

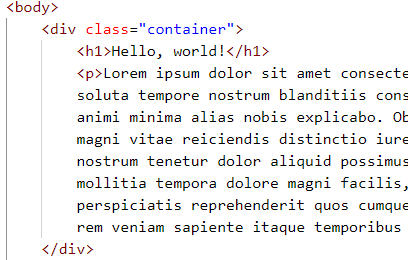
# Container

A Bootstrapben az egyes elemeket konténerekbe kell tenni.

Kétféle konténer van:

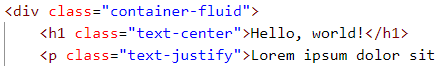
* container: nem tölti ki a teljes szélességet, de bizonyos méreteknél változik a szélessége.
* container-fluid: mindig kitölti a teljes szélességet.

Próbáld ki!

1. Készíts egy 100 szóból álló bekezdést a címsor után! (A bekezdésbe beírod, hogy lorem100, és megnyomod az Entert.)
2. A címsort és a bekezdést tedd egy div elembe, és ahhoz rendeld hozzá a container osztályt!  
   
3. Próbáld ki a böngészőben az oldalt különböző szélességekkel! Kapcsold be a vizsgálatot, és nézd meg a div, a h1 és a p elemeket! Melyiknek milyen margója és paddingje van? Mekkora méreteknél változik a szöveg szélessége?
4. Írd át a container osztályt container-fluid-ra, és nézd meg a különbséget a böngészőben!

# Szöveg igazítása

A szöveg igazítását a text-left, text-center, text-right és text-justify osztályokkal adhatjuk meg.

Ha például a címet középre szeretnénk igazítani, a szöveget pedig sorkizártra, akkor ezt kell írnunk:  


Próbáld ki!