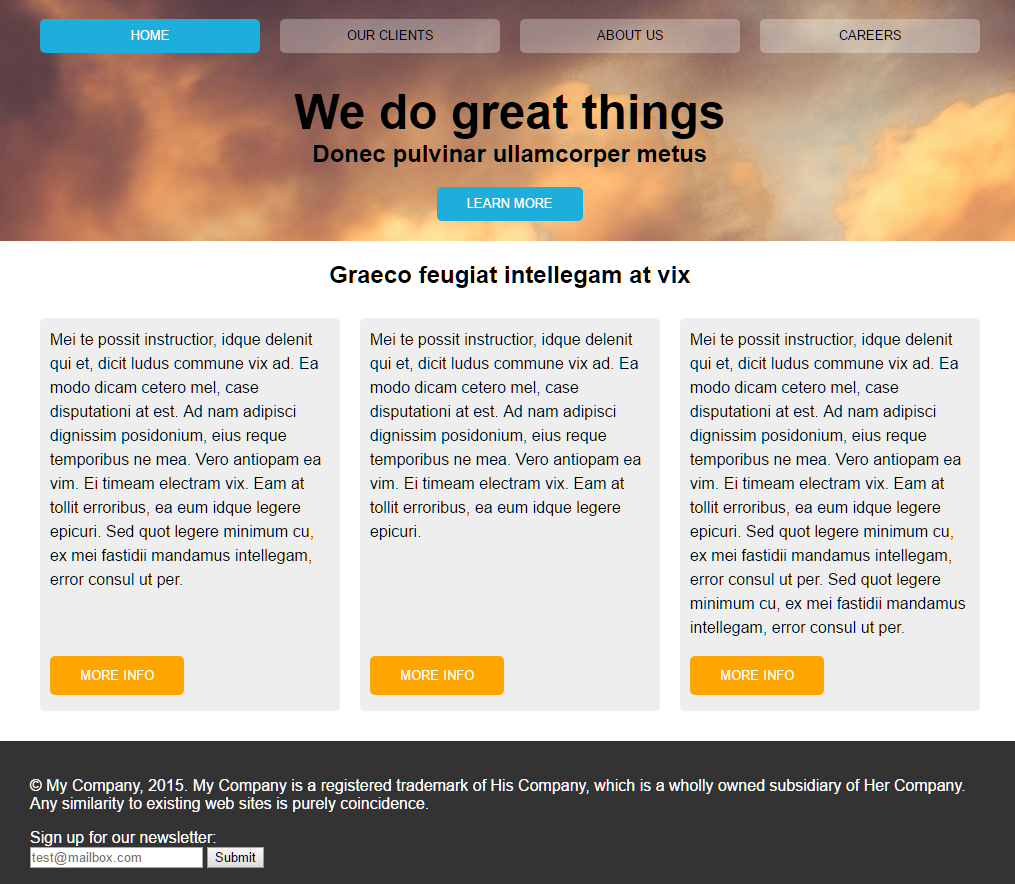
# Flexbox 5.

## Bevezetés

Ebben a leckében egy reszponzív weblap sablont készítünk a flexbox segítségével. Ez nem értelmes szövegeket tartalmaz, de a kép és a szövegek cseréjével tetszőleges weblaphoz felhasználható.



## Előkészületek

1. Nyisd meg a *flexbox-layout* mappát, majd abból az *index.html és a styles.css* fájlokat a Visual Studio Code-ban!
2. Nyisd meg az oldalt a böngészőben is!
3. Nézd át az eddig elkészített HTML és CSS kódokat!

## Normalizálás

Minden böngészőnek vannak alapbeállításai, amelyeket alkalmaz a weblap elemeire. Ezek sajnos nem teljesen egységesek, így a weblapok nem ugyanúgy jelennek meg minden böngészőben.

Ezt a problémát kétféleképpen próbálják megoldani:

1. Az egyik módszer a *resetelés*, vagyis az összes alapértelmezett stílus törlése. Ez azzal jár, hogy minden elemet nekünk kell megformázni, ami rengeteg plusz munkát jelent.
2. A másik módszer a *normalizálás*, amely az alapbeállítások egységesítését jelenti. Ha valamelyik böngésző eltér a többitől, azt javítja.

Mi ez utóbbi módszert alkalmazzuk. Mindössze arra van szükség, hogy a weblaphoz hozzákapcsoljuk a *normalize.css* fájlt, amelyet a <https://necolas.github.io/normalize.css/> oldalról lehet letölteni. Ezt a HTML fájlban már el is végeztük. A fájl mérete mindössze 7 kB, így oldalunk betöltődését nem fogja nagyon lelassítani.

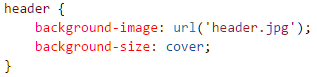
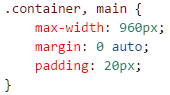
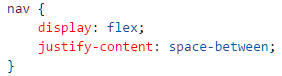
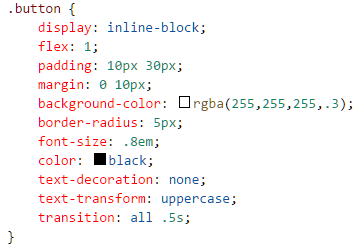
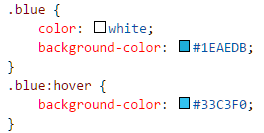
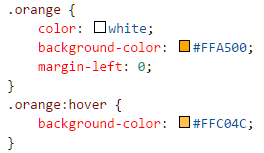
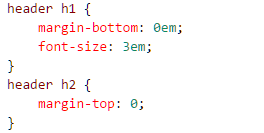
Nyisd meg a mappában lévő normalize.css fájlt, és nézd meg! Bőven el van látva megjegyzésekkel, de nem baj, ha nem értesz mindent belőle.

A normalizáláson kívül a normalize.css javítja a böngészők apróbb hibáit is.

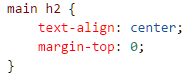
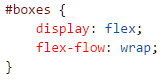
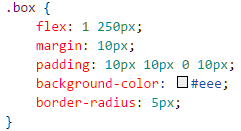
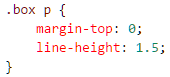
Ha optimalizálni szeretnéd az oldalat, e helyett a fájl helyett belinkelheted a tömörített változatot erről a címről (kisebb, mint 3 kB):

<https://cdnjs.cloudflare.com/ajax/libs/normalize/8.0.0/normalize.min.css>

## Fejléc (header) formázása

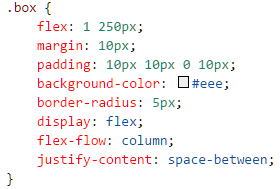
1. Állítsd be a header szakasz hátterét a következőképpen:  
     
   Figyeld meg a háttérképet különböző ablakméreteknél! Mit jelent a background-size: cover; beállítás?
2. A header és a footer szakaszokban van egy container osztályú szakasz, amely a tartalom szélességét állítja be. A main szakasz is hasonló, ezért ezeket most egyszerre állítjuk be:  
   
3. A header szakaszban lévő container-ben középre kell igazítani a szöveget (a többiben nem):  
   
4. A navigáció elemeit egymás mellé kell helyezni:  
   
5. A button osztályú hivatkozásokból gombokat készítünk:  
     
   Értelmezd a beállításokat, majd próbáld ki!
6. Amikor a gombok fölé visszük az egérmutatót, azok világosabb színűek (kevésbé átlátszóak) lesznek:  
     
   Próbáld ki!
7. A kék gomboknál más a betű- és a háttérszín:  
     
   Próbáld ki!
8. A narancsszínű gomboknál is mások a színek, és a bal oldali margót nullázni kell:  
     
   Próbáld ki!
9. Állítsd be a fejléc címsorait így:  
     
   Próbáld ki!

## A középső rész (main) formázása

1. Formázd meg a main szakaszban lévő címsort:  
   
2. A boxes azonosítójú rész legyen flex container! A benne lévő dobozok egymás mellé kerüljenek, sortöréssel! (kék vonalak a lenti ábrán)  
     
   Próbáld ki!
3. Állítsd be a dobozok tulajdonságait:  
   
4. Formázd meg a dobozokban lévő bekezdéseket:  
     
   Próbáld ki!

Azt szeretnénk, hogy a dobozokban lévő gombok alulra, egy vonalba kerüljenek. Ezért a dobozokat is flex tárolóvá alakítjuk, és a bennük levő elemeket (bekezdés és gomb) egymás alá helyezzük. (piros vonalak a lenti ábrán)



1. Egészítsd ki a box osztályt így:  
     
   Próbáld ki!

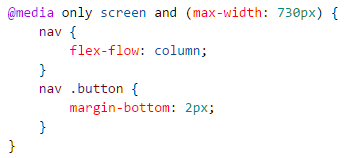
## A lábléc (footer) formázása

1. Formázd meg a footer szakaszt önállóan úgy, hogy a háttere sötétszürke (#333) színű, a betűk pedig fehérek legyenek!

## A navigáció reszponzívvá tétele

Bizonyos szélesség alatt a navigáció gombjai olyan keskenyek lesznek, hogy nem fér ki bennük a szöveg. Ilyenkor egymás alá szeretnénk tenni a gombokat.



1. Nyisd meg a böngészőben a fejlesztő eszközöket (F12 vagy Ctrl+Shift+I), és helyezd őket a megnyitott oldal mellé! Az elválasztó vonal húzásával nézd meg, milyen szélességénl kell módosítani a navigációt!
2. Készíts egy médialekérdezést a CSS fájl végére:  
     
   Ez a gombok egymás alá helyezésén kívül egy két pontos margót hagy mindegyik alatt. Próbáld ki!

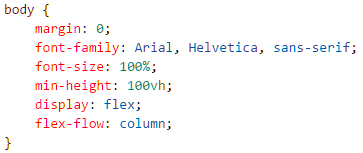
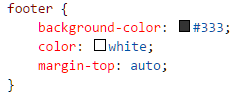
## Sticky footer

Ha elég széles az ablak, a lábléc alatt még marad üres hely. Ez elég csúnya, ezért ilyen esetekben a footert az oldal aljára helyezzük. (Ezt nevezzük sticky footer-nek.)

Ehhez először be kell állítani, hogy a body elem magassága legalább a viewport magasságával megegyező legyen. Ebben segítségünkre van egy új CSS mértékegység, a vh.

1vh = a viewport magasságának 1/100-ad része. A teljes viewport magassága 100vh. (Ehhez hasonlóan használható a vw is, amely a viewport szélességének 1/100-ad része.)

Ezen kívül a body elemet is flex tárolóvá kell tennünk, és a benne lévő elemeket (header, main, footer) egymás alá kell helyeznünk. Ezután a footer felső margóját automatikusra állítjuk.

1. Egészítsd ki a body stílust így:  
   
2. Állítsd a footer felső margóját automatikusra:  
   

## Önálló feladatok 1.

Helyezd el a szükséges prefixeket a CSS kódban az Autoprefixer segítségével!

## Önálló feladatok 2.

A Chrome fejlesztőeszközök segítségével nézd végig az oldal elemeinek formázását!

## Önálló feladatok 3.

Töltsd fel egy online tárolóhelyre ezt a feladatsort és az órán készített fájlokat! Otthon is nézd át őket!