



14 – FLEXBOX DEMO

Előadás demó leírása a Mobil- és webes szoftverek c.
tárgyhoz

Gincsei Gábor
2017.

Szerzői jogok

Jelen dokumentum a BME Villamosmérnöki és Informatikai Kar hallgatói számára készített elektronikus jegyzet. A dokumentumot a Mobil- és webes szoftverek c. tantárgyat felvevő hallgatók jogosultak használni, és saját céljukra 1 példányban kinyomtatni. A dokumentum módosítása, bármely eljárással részben vagy egészben történő másolása tilos, illetve csak a szerző előzetes engedélyével történhet.



BEVEZETÉS

Ebben a dokumentumban röviden áttekintjük az előadáson bemutatott demókat, hogy az a későbbiekben otthon önállóan is el lehessen végezni.

CÉLKITŰZÉS

A demók elvégzésének segítségével gyakorlatiasan meg lehet ismerkedni a Flexbox használatával, ki lehet próbálni az előadáson elhangzottakat a gyakorlatban.

A demók kifejezetten olyan hallgatóknak szólnak, akiknek a Flexbox még újdonság, abban tapasztalatuk csekély.

MIÉRT ÉRDEMES MEGISMERKEDNI EZEKEL A TECHNOLOGIÁKKAL?

A FlexBox egy modern layout rendszer megvalósításához nélkülözhetetlen CSS megoldás. A böngésző támogatottsága is egyre jobb. Mivel kényelmesebb mint a float alapú layout ezért mindenképpen érdemes az alapjait megismerni. A Bootstrap 4 már flexbox alapú layoutot fog használni.

01 – FLEXBOX

Ebben a példában megnézzük, hogy a flexbox különböző beállításával hogyan tudjuk átvariálni az oldalunk kinézetét, illetve kitérünk arra is, hogy az egyes elemek összenyomása / nyúlása hogyan befolyásolható.

Az alábbi tulajdonságokat fogjuk állítani:

- **justify-content**: elemek rendezése, tagolása a fő tengely mentén.
- **flex-wrap**: a tartalom törhet-e új sorba / oszlopba.
- **align-content**: többsoros flexbox sorainak igazítása a kereszt tengely mentén.
- **align-items**: gyerekek igazítása a fő tengelyre merőlegesen.
- **flex-shrink**: összemólyható-e a gyerek ha kevés a hely és milyen arányban.
- **flex-grow**: nyúljon-e a gyerek és a többi elemhez képest milyen arányban.

Az alábbi kiinduló HTML-t nézzük meg:

```
<div id="main">
  <div style="background-color:coral;"></div>
  <div style="background-color:lightblue;"></div>
  <div style="background-color:khaki;"></div>
  <div style="background-color:pink;"></div>
  <div style="background-color:gold;"></div>
  <div style="background-color:lightgreen;"></div>
  <div style="background-color:purple;"></div>
</div>
```

Mint látható csak különböző színű div-ek vannak egymás után téve egy main div-ben, egyebet nem tartalmaz a HTML. A színek azért a style-ban kerültek megadásra, hogy a CSS kódot csak a FlexBoxra mutasson példát.

A FLEXBOX GYEREKEK KITÖLTIK A RENDELKEZÉSRE ÁLLÓ HELYET.

A flexbox esetén az elemek automatikusan kitöltik a szülő elemet, ha kell összenyomja az elemeket (alapértelmezés szerint), vagy ha több a hely mint a gyerekek szélessége, akkor megfelelő távolságra helyezi őket a `justify-content` értéke alapján.

A main div 300px széles, az egyes elemek pedig 70px szélesek. Összesen 7db div van, tehát a gyerek elemek nem férnek el a rendelkezésre álló helyen, viszont a flexbox ebben az esetben összenyomja az elemeket, ha az nincs letiltva a `flex-shrink` tulajdonsággal az egyes gyerek elemeken. A `justify-content: space-between` van állítva.

```
#main {
  width: 300px;
  height: 200px;
  border: 1px solid #c3c3c3;
  display: flex;
  justify-content: space-between;
}
```

```
#main div {
  width: 70px;
  min-height: 70px;
}
```

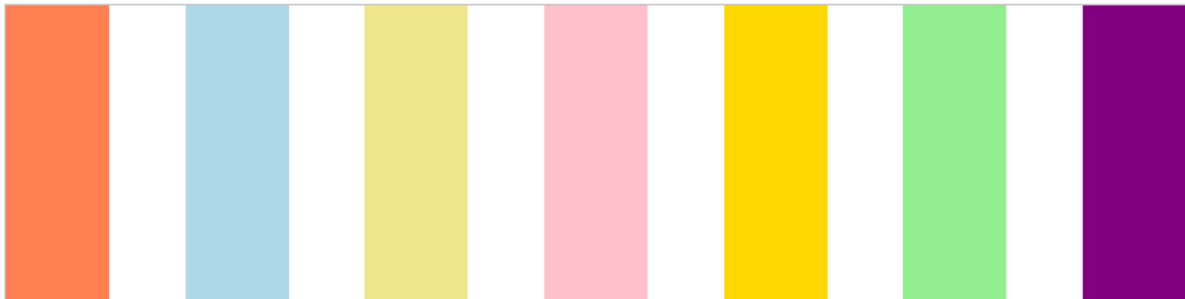
A kódot ha lefuttatjuk azt látjuk, hogy bár az egyes az egyes div-ek összenyomódtak, hiszen a `flex-shrink` alapértelmezés szerint 1-es azaz össze lehet nyomni az elemeket.

Mivel a div-ekre csak `min-height`-ot adtunk meg, ezért a magassága ki fogja tölteni a rendelkezésre álló helyet, hiszen a flexbox alapértelmezés szerint függőlegesen nyújtja az elemeket.



Vegyük szélesebbre a teljes div-et 800px-re, Ekkor már elférnek a div-ek és a fölösleges helyet a `space-between` egyenletesen elosztja az elemek között.

```
#main {
  width: 800px;
```

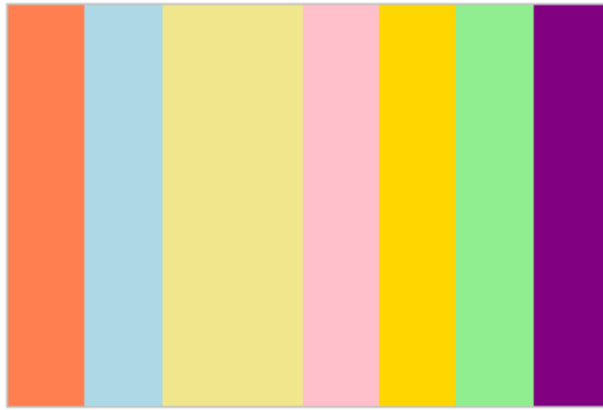


AZ ÖSSZENYOMÁS SZABÁLYOZÁSA (FLEX-SHRINK)

Ha visszaállítjuk a szélességet 300px-re és mondjuk a 3. elemet letiltjuk az összenyomást, akkor a 3. div mérete megmaradt a megadott 70px, a többi elem pedig ennyivel keskenyebb lesz, hogy elférjenek a rendelkezésre álló helyen.

```
#main {
  width: 300px;
}

#main div:nth-child(3){
  flex-shrink: 0;
}
```

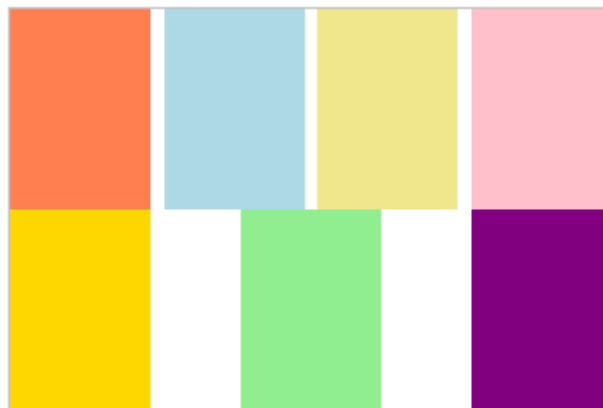


TARTALOM TÖRÉSE TÖBB SORRA (FLEX-WRAP)

Állítsuk be, hogy ha kevés a hely, akkor törhessen több sorba a tartalom. Ezt a `flex-wrap: wrap`-el tehetjük meg.

```
#main {  
  ...  
  flex-wrap: wrap;  
}
```

Ahogy az alábbi ábrán is látható, akkor az egyes elemek úgy oszlanak meg a sorok között, hogy egyiket se kelljen összenyomni. A jelen beállítások mellett 4 div jut a felső sorba és 3 az alsóba és az egyes elemek közötti hely egyenletesen oszlik meg, de csak soron belül.

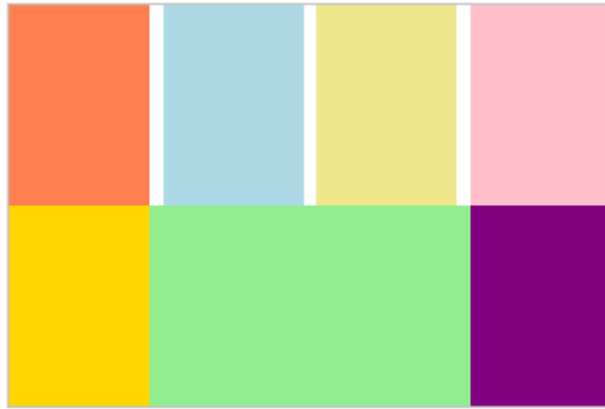


ELEM NYÚJTÁSA (FLEX-GROW)

A következő példában engedjük meg, hogy a zöld div a 6. nőjön meg ha van rendelkezésre álló hely.

```
#main div:nth-child(6){  
  flex-grow: 1;  
}
```

Ekkor az alábbi elrendezést kapjuk. A zöld elem kitölti a rendelkezésre álló helyet.



TARTALOM FÜGGŐLEGES IGAZÍTÁSÁNAK MEGADÁSA (ALIGN-CONTENT)

Viszont az továbbra is kérdés, hogy függőlegesen miért töltik ki a div-et az összes rendelkezésre álló helyet. Állítsuk be az `align-content` tulajdonságot `center`-re.

```
#main {
  ...
  align-content: center;
}
```

Ekkor már nem az alapértelmezett `stretch` fog érvényre jutni, hanem középre fogja igazítani az egyes elemeket a flexbox az alábbiak szerint:



ELEMENK FÜGGŐLEGES IGAZÍTÁSÁNAK MEGADÁSA (ALIGN-ITEMS)

Azonban, ha nem az `align-content`-et állítjuk `center`-re hanem az `align-items`-et, akkor az alábbi elrendezést kapjuk

```
#main {
  ...
  align-content: center;
  align-items: center;
}
```



Jó látható, hogy az `align-items` a soron belül igazította az elemeket középre, addig az `align-content` a teljes tartalmaz igazította középre (ezért nincs üres hely a két sor között.)

EGY ELEM FÜGGŐLEGES IGAZÍTÁSÁNAK MÓDOSÍTÁSA (ALIGN-SELF)

Ha egy-egy elemet másképpen szeretnénk elrendezni, mint amit az `align-items`-ben megadtunk, arra az `align-self` beállítás való. A 3. divet (sárga) igazítsuk a sor tetejére úgy, hogy a többi elem közére van igazítva.

```
#main div:nth-child(3){  
  flex-shrink: 0;  
  flex-self: flex-start;  
}
```

