

# Számítógép-hálózatok tárgy

---

Reszponzív weboldal tervezés

CSS Flexbox

(részben Dömsödi Balázs munkája)

# Reszponzív weboldal tervezés

---

*„Responsive Web Design is about using HTML and CSS to automatically resize, hide, shrink, or enlarge a website to make it look good on all devices (desktops, tablets, and phones).”*

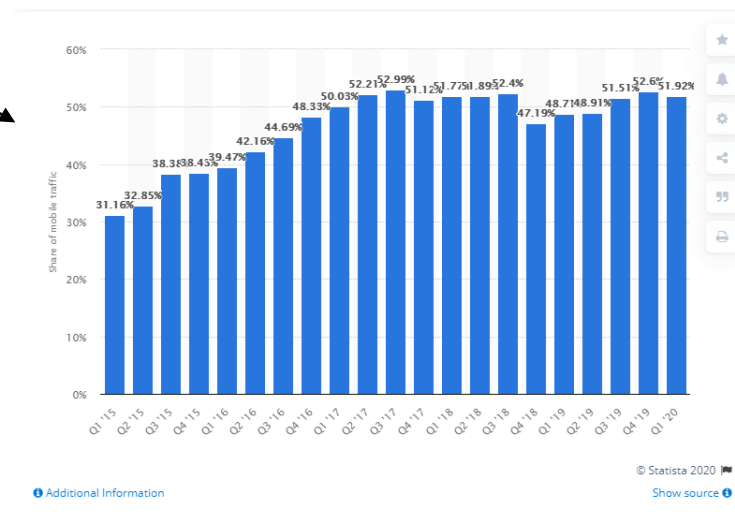
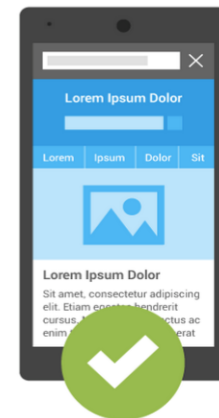
*w3schools.com*

*The technique has developed to the point that most designers no longer say, 'responsive web design,' we say, 'web design,' and mean exactly the same thing.*

Source: <https://www.webdesignerdepot.com/2015/03/the-future-of-responsive-web-design/>

# Bevezetés

- ❑ Google változtatott a rangsorolási algoritmusán 2015 április végén
  - mobil barát weboldalak előbbre: „mobilgeddon”
  - a felhasználók részéről is egyre nagyobb az igény mobilbarát weboldalra
- ❑ 2019/20-ban a webes forgalom kb. 52%-a mobil eszközről történik
  - <https://www.statista.com/statistics/277125/share-of-website-traffic-coming-from-mobile-devices>
- ❑ Megjegyzendő, hogy sokan laptopon folytatják a mobilon megkezdett böngészést
- ❑ Megoldás:
  - **reszponzív weboldal**
  - adaptív weboldal



# Reszponzív oldal



- A weboldalnak egy olyan kialakítása, amelyik rugalmasan alkalmazkodik a különböző böngészők képernyőjének méretéhez azért, hogy
  - ugyanolyan optimális megjelenést
  - ugyanolyan felhasználói élményt
  - ugyanolyan könnyű kezelhetőségetbiztosítson a felhasználónak minden böngészésre alkalmas eszközön, például desktop, laptop, mobil, tablet, tévé, stb..
- Eközben a navigáció és az olvasás nem válik nehézkessé
  - az elemek szélessége megváltozik
  - a navigáció változik
  - az oldalsó oszlopok fokozatosan, a képernyőméret függvényében kerülnek egymás alá
- Két irányzat:
  - Bootstrap grid irányzat: webfejlesztés tárgyban jelenleg
  - **CSS flexbox és Grid: mi most ezzel foglalkozunk**

# mobile first megoldás

---

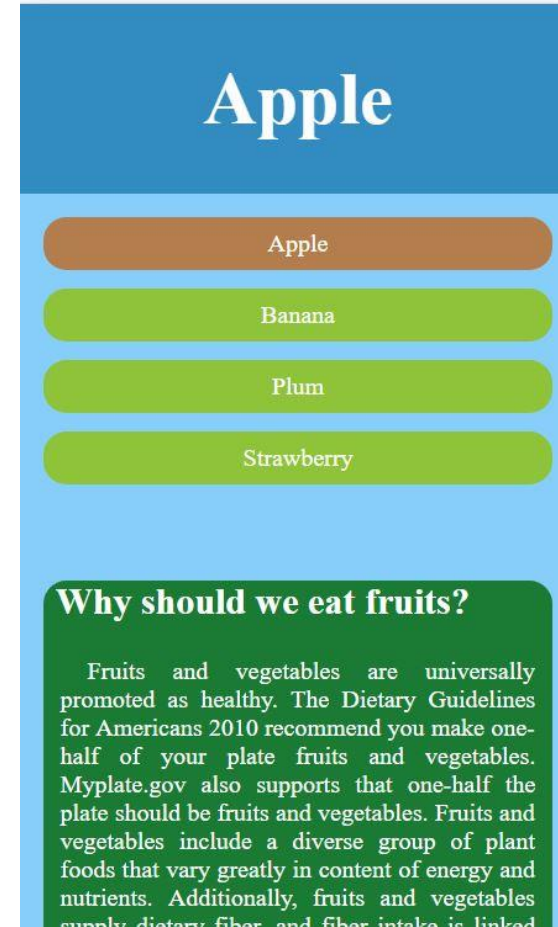
- ❑ Először a mobil nézetet készítjük el, aztán a desktopot, és a többi
- ❑ Oka: a mobilos verzió esetében kerülünk szembe a kis képernyőméret miatt leginkább korlátozásokkal, előnyösebb ha innen indulnunk ki
- ❑ Szükség van pl:
  - mutatóujjal használható menü
  - átméreteződő képek
  - megfelelően olvasható betűméret
  - nem telezsúfolt tartalom
- ❑ Érdemes a legkisebb (pl 320 pixel) töréspontból kiindulni, és onnan haladni „felfelé” (pl 2000 pixel)
  - 2 töréspontot fogunk használni
  - a megoldás tetszőlegesen bővíthető további töréspontokkal

# Viewport beállítása

---

- ❑ viewport: a weboldal felhasználó által látható területe. Különböző kijelző méreteknél változik.
- ❑ HTML5-ben bevezetett meta elem.
- ❑ **`<meta name="viewport" content="width=device-width, initial-scale=1.0">`**
- ❑ közli a mobil böngészővel hogy mobilképes az oldal, és ne a teljes verziót nyomja össze
- ❑ `width=device-width` -> az oldal width tulajdonsága kövesse a kijelző szélességét.
- ❑ `initial-scale=1.0` -> meghatározza, hogy mekkora nagyítást (zoom) használjon az oldal, amikor először töltődik be. Például a betűk konzisztens mérete miatt fontos.

# Viewport nélkül, és viewporttal



# Relatív méretezés

---

- Ha egy elem max-width CSS tulajdonságát 100%-ra és a height tulajdonságát auto-ra állítjuk, kisebb területen arányosan kisebb méretben jelenik meg.
- **max-width:100%;height:auto;**
- Minden reszponzív oldalra kell a viewport elem!
- Példák a viewport elem + relatív méretezés használatával és nélkülük:  
(mobil nézet elérhető chrome esetén: F12)
  - [https://www.w3schools.com/html/example\\_withoutviewport.htm](https://www.w3schools.com/html/example_withoutviewport.htm)
  - [https://www.w3schools.com/html/example\\_withviewport.htm](https://www.w3schools.com/html/example_withviewport.htm)



*@media only screen and (min-width: 1200px)*  
*@media only screen and (max-width: 768px) stb.*

# Media Query

- Ugyan a CSS grid, a relatív méretezés és a viewport segítségével már valóban reszponzív a tartalom, bizonyos kijelző méretek határainál érdemes lehet teljesen átrendezniünk azt.
- CSS3-ban bevezetett technika. A **@media** szabályt használja arra, hogy olyan változásokat írjon le, amelyek csak bizonyos feltételek mellett érvényesek.
- Mi egy töréspontot fogunk használni: 768px
- A Bootstrap több töréspontot is használ, pl: 600px, 992px, 1200px



# box-sizing tulajdonság

---

```
* {  
    box-sizing: border-box;  
}
```

- ❑ Állítsuk be minden HTML elem box-sizing tulajdonságára a border-box értéket.
- ❑ Ez biztosítja, hogy az elemekhez tartozó padding és border hozzáadódjon a width és height értékekhez is.
- ❑ *Tipp: próbáljuk ki majd a legvégén, hogy mi történik a box-sizing kikommentelésével*

# Tesztelés

- ❑ A legtöbb esetben a böngésző egyszerű átméretezésével szimulálni tudjuk a különböző nézeteket
- ❑ Nagyon ritkán, ez a megoldás nem működik tökéletesen, elsősorban mobilnézet esetén
  - ennek az az oka, hogy hiába méretezzük át megfelelően a böngészőt, attól még a program számítógépes verziója fut, ahhoz képest pedig működésében lehetnek eltérések a böngésző mobilos applikációjában (pl parallax háttér, stb esetén)
- ❑ Fontos, hogy a legvégén ellenőrizzük le az oldalt a legördülő menüvel is.



# CSS Flexbox

---

# A Flexbox elrendezés célja

---

- Az egyes elemek közötti **tér hatékony rendezése** abban az esetben is, ha adott elemek mérete előre ismeretlen és/vagy változó
- Az elemek **szélességének, magasságának és sorrendjének** dinamikus változtatása, hogy azok a lehető legoptimálisabban töltsék ki a rendelkezésükre álló teret
- Olyan komplex webes elrendezések kialakításának elősegítése, melyeknél **fontos szempont a reszponzivitás**

# A Flexbox működése

---

- ❑ A flex konténerben található elemek **kiszélesednek**, hogy kitöltsék a szabad területet vagy **összezsugorodnak**, hogy megelőzzék az overflow-t
- ❑ A **reszponzív beállításokat teljes mértékben a CSS-ben tartja**, ellentétben a Bootstrap Grid Systemmel, ahol az egyes elemek oszlopszélességét a HTML kódban tudtuk kezelni
- ❑ Az elrendezés **flex-konténerekből** és a bennük elhelyezett **flex-elemekből** áll

# Flex-konténer

---

- ❑ Először a flex-konténert kell meghatároznunk, ami a flex-elemeket fogja tartalmazni.
- ❑ Állítsuk a blokk szintű elem `display` tulajdonságát "flex" értékre.

## HTML:

```
<div class="flex-container">  
</div>
```

## CSS:

```
.flex-container {  
  display: flex;  
}
```

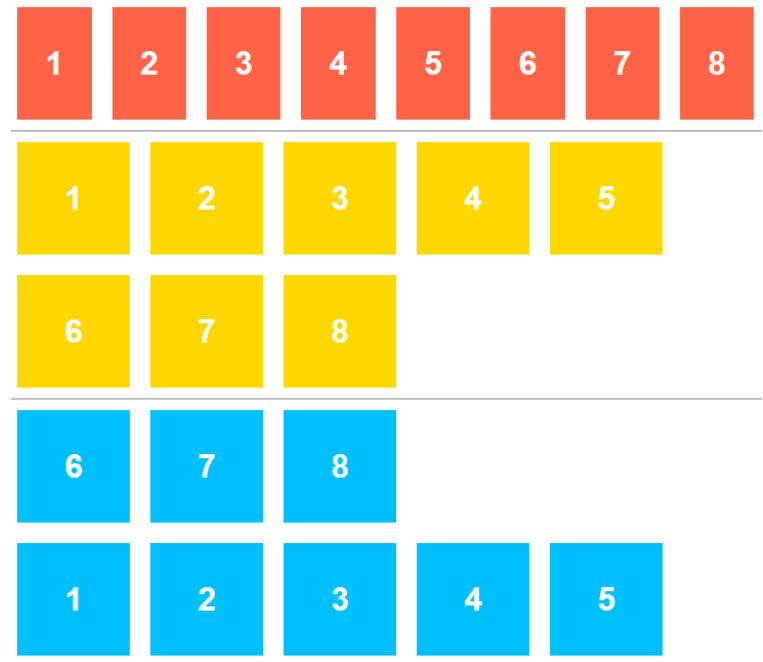
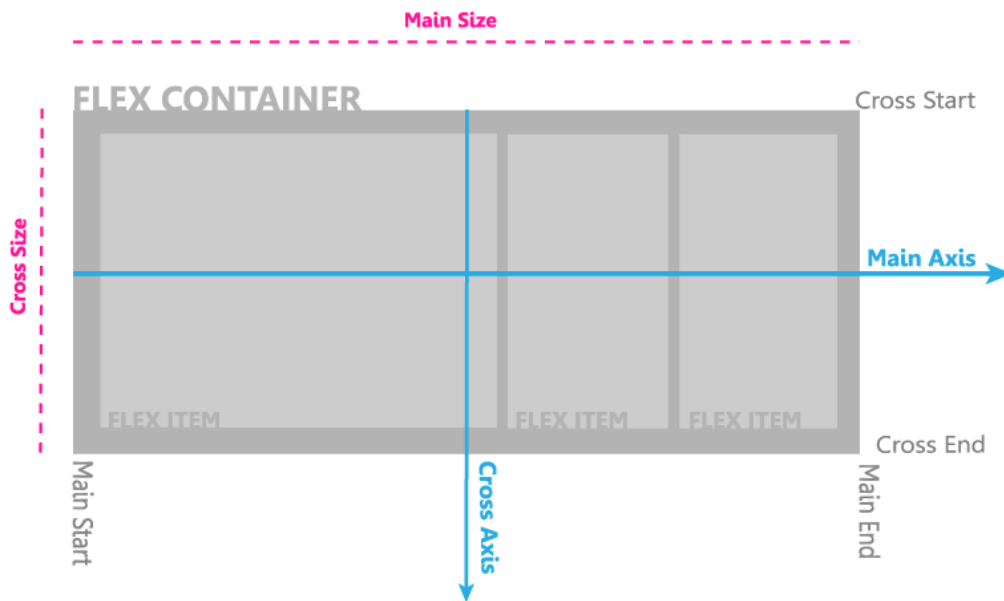
# Flex vonal

---

- A flex-elemek a flex-konténerben flex vonalakon helyezkednek el, ezek "képzeletbeli sávok", amelyekre soronként igazodnak a flex-konténer elemei. Arra használjuk őket, hogy ezek mentén határozzuk meg az egyes elemek egymáshoz viszonyított és soronként vagy oszloponként (tengelyenként) vett elrendezését.



# Tengelyek és flex vonalak



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

Forrás: <https://css-tricks.com/almanac/properties/f/flex-wrap/>

# A flex-konténer tulajdonságai

---

- ❑ flex-direction: flex-elemek rendeződési iránya (vízszintes vagy függőleges)
- ❑ flex-wrap: a flex-elemek sortörése kapcsolható ki vagy be a segítségével
- ❑ flex-flow: flex-direction és flex-wrap együttes megadására használható
- ❑ justify-content: flex-elemek főtengely szerinti térközét kezeli
- ❑ align-items: kereszt tengely szerinti térköz
- ❑ align-content: flex vonalak elrendezése

# Flex-direction

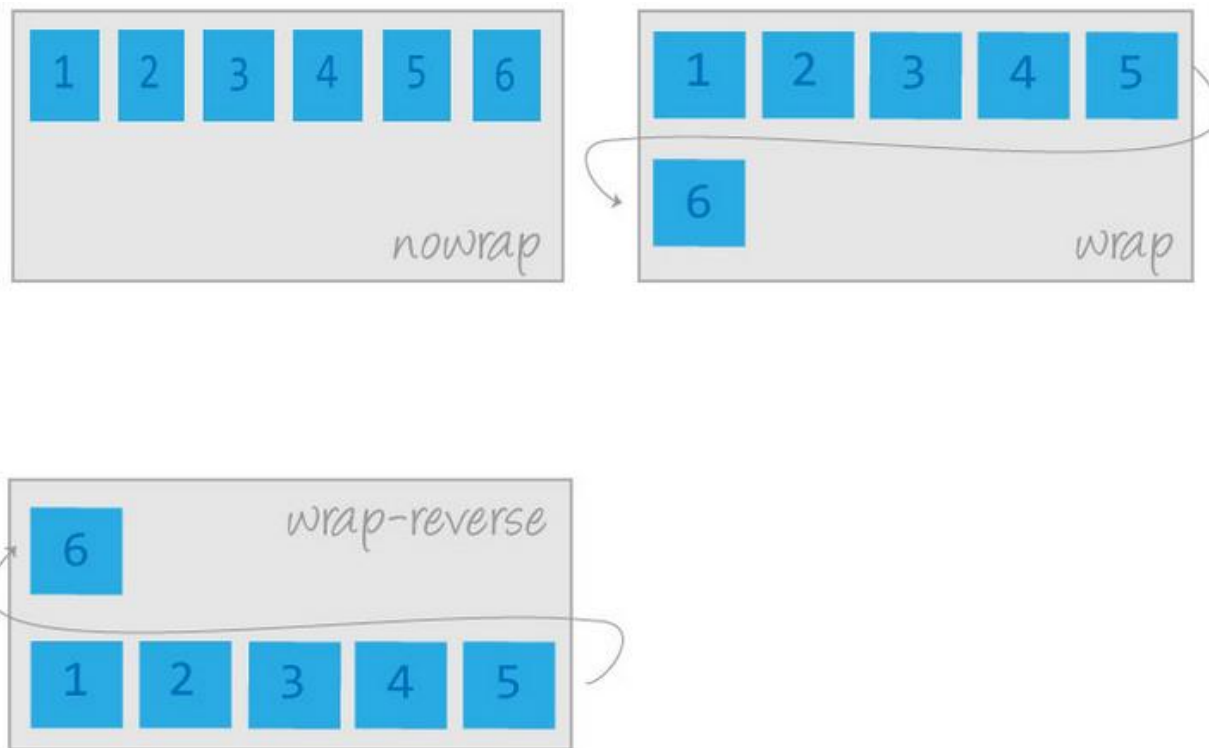
---



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Flex-wrap

---



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Justify-content

---

*flex-start*



*flex-end*



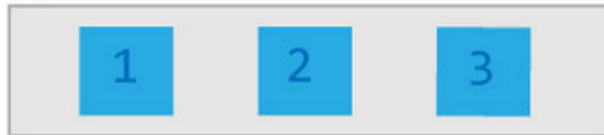
*center*



*space-between*



*space-around*



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Align-items

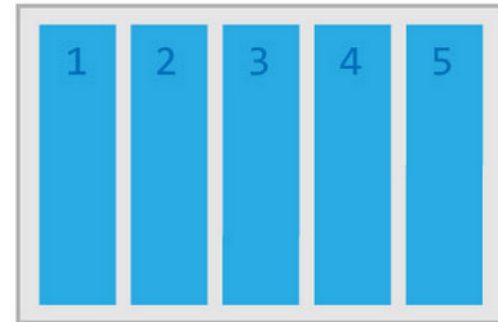
*flex-start*



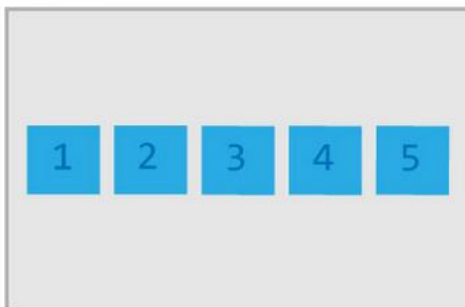
*flex-end*



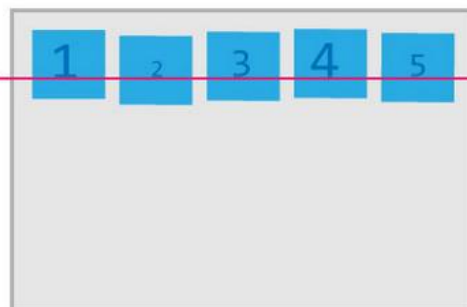
*stretch*



*center*



*baseline*



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Példa

---

- A flex-konténerek tulajdonságainak segítségével például nagyon könnyedén megoldható a teljes középre igazítás gyakori problémája.

```
.flex-container {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
}
```

# Flex-elemek

---

- ❑ A flex-konténerek közvetlen gyermekei automatikusan flex-elemek lesznek.
- ❑ Esetükben azt kell meghatároznunk, hogy a szülő elem mekkora részét foglalják el, milyen sorrendben jelenjenek meg, növekedjenek vagy zsugorodjanak-e és egyenként vonatkozik-e rájuk valamilyen egyedi elrendezési szabály.
- ❑ Fontos, hogy flex-elem lehet egyszerre flex-konténer is, ami további flex-elemeket tartalmaz. Ez lényegében a módszer alapja.



# Flex-elemek tulajdonságai

---

- ❑ order: a flex-elemek sorrendje határozható meg a segítségével
- ❑ flex-grow: relatív érték, azt adja meg, hogy a testvér flex-elemekhez képest milyen mértékben növekedjen az adott elem
- ❑ flex-shrink: szintén relatív érték, a zsugorodást szabályozza
- ❑ flex-basis: elem kezdeti hossza
- ❑ flex: flex-grow, -shrink és -basis röviden
- ❑ align-self: általános align-items felülírása

# Order

---

*(default)*



*order: 1*



*order: 3*



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Flex-grow

---

*(default)*



*flex-grow: 1 (all)*



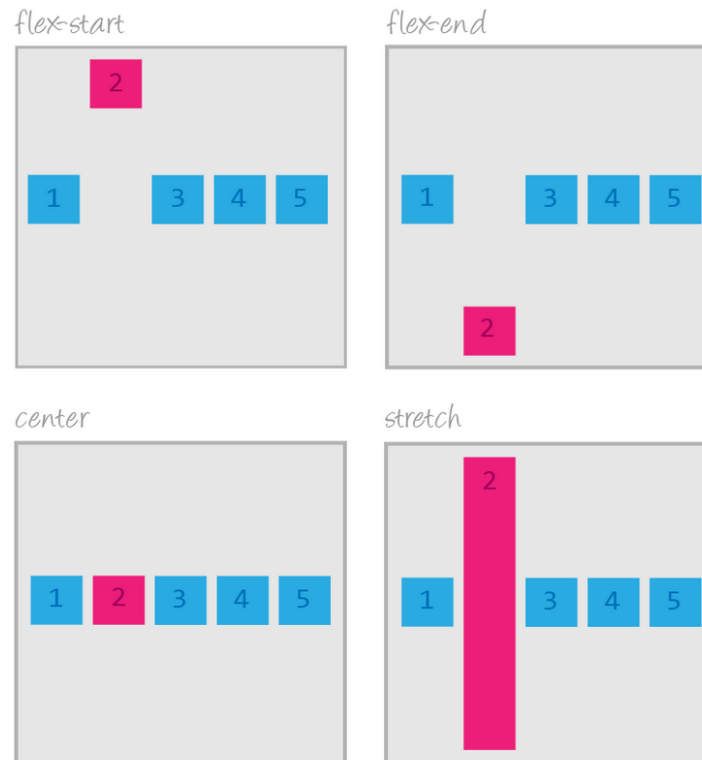
*flex-grow: 2 (on second item)*



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Align-self

---



Forrás: [https://tympanus.net/codrops/css\\_reference/flexbox/](https://tympanus.net/codrops/css_reference/flexbox/)

# Órai responszív layout

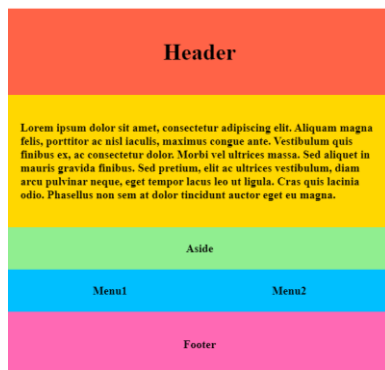
## PC nézet



## Tablet nézet



## Mobil nézet



# Hasznos linkek

---

- ❑ <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- ❑ Flexbox kurzus: <https://scrimba.com/g/gflexbox>  
(43 perc)
- ❑ W3school CSS flexbox  
[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)
- ❑ Flexbox in CSS (könyv):  
<https://www.oreilly.com/library/view/flexbox-in-css/9781491981474/>