

WEBY – CSS 8

Flexbox

Flexbox

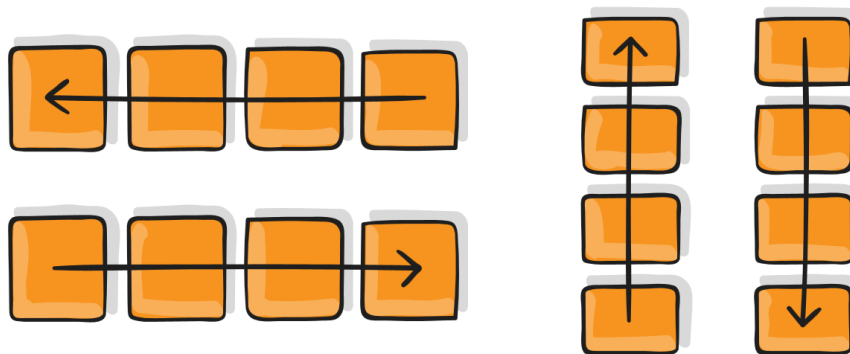
- Metóda na umiestňovanie obsahu po riadkoch alebo stĺpcoch
- Rodičovský komponent hovorí svojim deťom do
 - ich veľkosti
 - ich umiestnení v rodičovi
 - ich umiestnení navzájom
 - distribúciu zvyšného voľného miesta v sebe

Flexbox layout

- základný koncept – kontajner, ktorý rozmiestňuje svoje deti - položky (items)

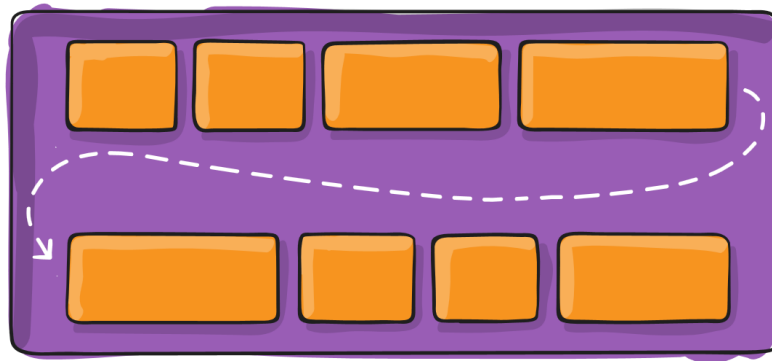
```
.container {  
  display: flex;  
}
```

- kontajneru môžeme určiť v akom smere budú položky rozmiestňované (default je row)
 - **flex-direction**: row | row-reverse | column | column-reverse
 - smer závisí od jazyka (napr. hebrejčina spôsobí, že row ide doľava)
 - zdefinujeme tým **hlavnú os**



Flexbox container

- defaultne sa všetky položky snažia vopchať do jedného riadku/stĺpca. Môžem ich však dať zalamovať – správajú sa ako text
 - **flex-wrap**: nowrap | wrap | wrap-reverse;
 - wrap-reverse usporiadaava riadky zdola nahor



- zjednotený zápis pre direction a wrap
 - **flex-flow**: <direction> <wrap>
 - default je “row nowrap”

Flexbox container

- rozmiestňovanie pozdĺž hlavnej osi
- v kontajneri nastavíme
 - **justify-content**: flex-start | flex-end | center | space-between | space-around | space-evenly
 - default je flex-start

flex-start



flex-end



center



space-between



space-around

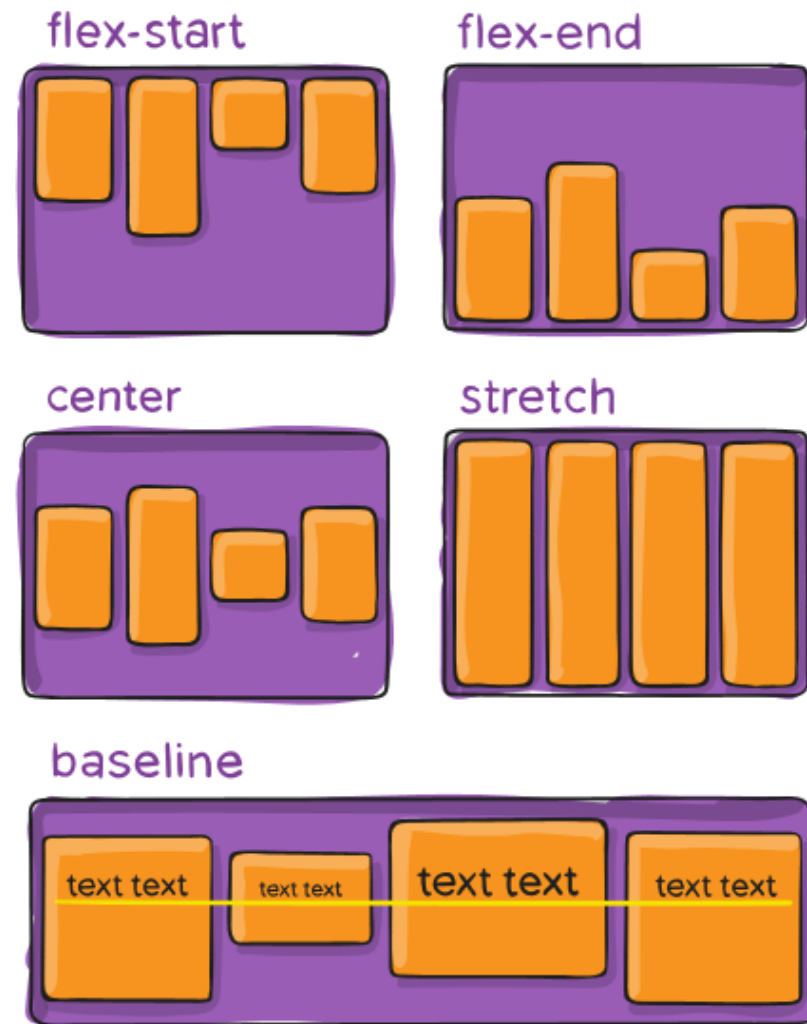


space-evenly



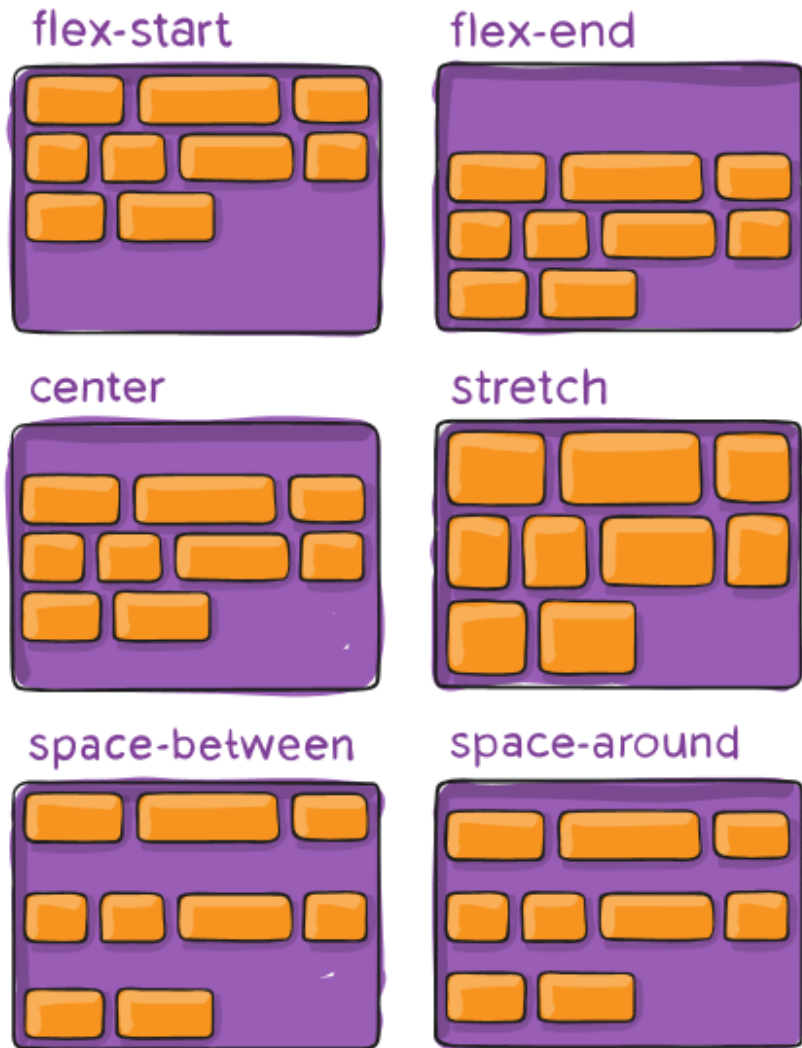
Flexbox container

- rozmiestňovanie kolmo na hlavnú os
- v kontajneri nastavíme
 - ▣ **align-items:** stretch | flex-start | flex-end | center | baseline
 - ▣ default je stretch



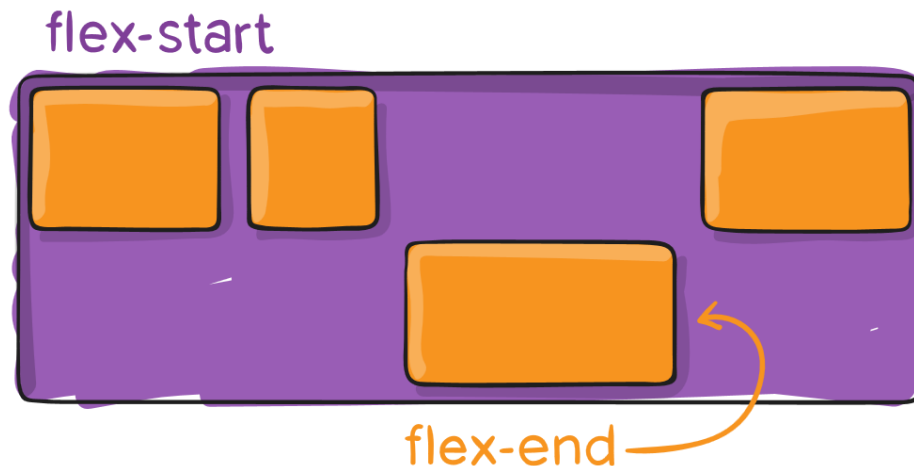
Flexbox container

- rozmiestňovanie kolmo na hlavnú os **ak máme wrap**
- riadok je taký vysoký, ako jeho najvyšší item plus koľko si ukradne z voľného miesta
- v kontajneri nastavíme
 - ▣ **align-content:** flex-start | flex-end | center | space-between | space-around | stretch
 - ▣ default je stretch (každý riadok kradne rovnako z voľného miesta)



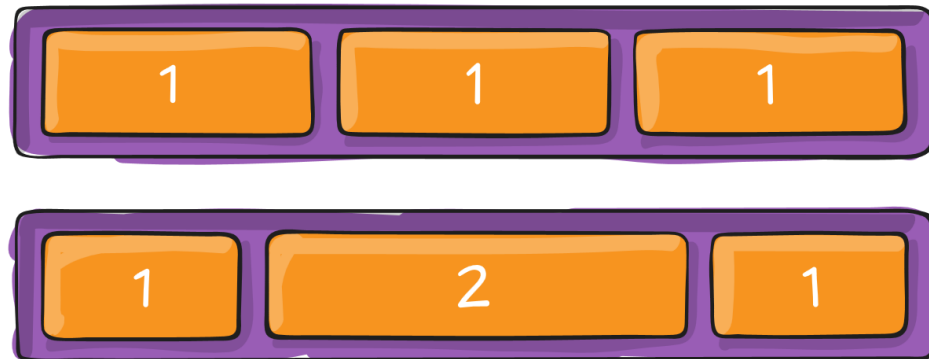
Flexbox item

- umiestnenie kolmo na hlavnú os pre konkrétnu položku inak ako pre zvyšok kontajnera
 - ▣ **align-self**: auto | flex-start | flex-end | center | baseline | stretch;



Flexbox item

- prispôsobenie veľkosti položky pozdĺž hlavnej osi v závislosti od voľného miesta, ak je kontajner širší ako súčet pôvodných širok položiek
 - ▣ **flex-grow**: <číslo>;
 - ▣ default je 0 – nezväčšuje sa
 - ▣ číslo hovorí aký pomer z voľného miesta si ukradne voči ostatným položkám (3 si ukradne 3x koľko miesta ako 1)



Flexbox item

- prispôsobenie veľkosti položky pozdĺž hlavnej osi v závislosti od nedostatku miesta, ak je kontajner užší ako súčet pôvodných širok položiek
 - **flex-shrink**: <číslo>;
 - default je 1 – znižuje sa v pomere 1
 - číslo hovorí ako veľmi sa položka zmenší pri úzkom kontajneri voči ostatným položkám (3 sa vzdá 3x viac miesta ako 1)
- počiatočná veľkosť položky pozdĺž hlavnej osi pred zväčšovaním/zužovaním
 - **flex-basis**: <veľkosť> | auto;
 - default je auto – ideme podľa originálnej šírky/výšky položky
- kombinácia všetkých troch (odporúčané)
 - **flex**: none | auto | flex-grow flex-shrink flex-basis;
 - initial (default) už vieme: "0 1 auto"
 - none: 0 0 auto, auto: 1 1 auto;
 - druhý a tretí parameter sú nepovinné

Flexbox item

- zmena poradia položiek v kontajneri
 - ▣ `order: <integer>;`
 - ▣ default je 0
 - ▣ položky najprv zoradí podľa hodnoty `order` a ak je ich viac s rovnakou hodnotou `order` ich vzájomné poradie je dané poradím v kontajneri

