

CSS

Šesti dio

PREGLED

- Flex
 - Uvod
 - Container, parent element
 - Items, child elementi
 - Primjeri

FLEX, UVOD

- Prije flexbox layouta, koristila su se 4 layout moda
 - Block, za sekcije na stranici
 - Inline, za tekst
 - Table, za 2D tabele podataka
 - Positioned, za eksplicitno pozicioniranje elemenata
- Flexible Box Layout Module olakšava dizajniranje fleksibilnog layouta bez korišćenja float-a i pozicioniranja

FLEX IMPACT

- Neki svojstva (properties) mijenjaju default ponašanje ili su čak ignorisana
 - **margin**, margine susjednih itema se ne sudaraju (collapse)
 - **min-width** i **min-height**, default je auto, ne 0
 - **visibility**: collapse
 - **float**, **clear**, **vertical-align** su ignorisani

FLEX, CONTAINER

- Da biste mogli da koristite flexbox model, prvo treba definisati flex container (parent element), pogledati primjer 14,

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

- Container sadrži sledeće proprijetije:
 - flex-direction
 - flex-wrap
 - flex-flow
 - justify-content
 - align-items
 - align-content

FLEX, DIRECTION

- Property `flex-direction` služi da definišemo kako se elementi slažu u parent elementu
 - `flex-direction: column;` - slaganje itema vertikalno od vrha ka dnu
 - `flex-direction: column-reverse;` - slaganje itema vertikalno od dna ka vrhu
 - `flex-direction: row;` - slaganje itema horizontalno s lijeva u desno
 - `flex-direction: row-reverse;` slaganje itema horizontalno s desna u lijevo
- Isprobajmo na prethodnom primjeru ove varijacije

FLEX, WRAP I FLOW

- Property `flex-wrap` da li flex itemi mogu da se upakuju (wrapuju) ili ne
 - `flex-wrap: wrap;` - ako flex itemi mogu da se upakuju
 - `flex-wrap: nowrap;` - ako flex itemi ne mogu da se upakuju (default)
 - `flex-wrap: wrap-reverse;` - ako je potrebno da se flex itemi upakuju u obrnutnom redosledu
 - Iprobajmo [primjer](#) za sve varijacije
- Za kombinovanje `flex-direction` i `flex-wrap` koristi se `flex-flow: direction wrapping` property

FLEX, JUSTIFY CONTENT

- Property `justify-content` služi za horizontalno poravnanje flex itema u containeru
 - `justify-content: center;` centriranje itema na sredinu containera
 - `justify-content: flex-start;` poravnanje itema uz početak containera
 - `justify-content: flex-end;` poravnanje itema uz kraj containera
 - `justify-content: space-around;` space prije između i nakon itema
 - `justify-content: space-between;` space između itema
 - `justify-content: space-evenly;` slično kao space-around ali isti left i right space kao i između itema
 - Isprobati [primjer](#) za razne varijacije

FLEX, ALIGN ITEMS

- Property `align-items` služi za vertikalno poravnanje itema
 - `align-items: center;` za vertikalno centriranje
 - `align-items: flex-start;` za vertikalno poravnanje uz gornju ivicu
 - `align-items: flex-end;` za vertikalno poravnanje uz donju ivicu
 - `align-items: stretch;` za popunjavanje containera po visini
 - `align-items: baseline;` poravnanje u odnosu na content itema (u našem primjer poravnanje po tekstu)
 - Pogledati [primjer](#) i isrobati razne varijacije

FLEX, ALIGN CONTENT

- Property `align-content` se koristi za poravnanje flex linija (flex lines, tj. flex row)
 - `align-content: space-between;` prikazuje flex linije sa istim razmakom između njih
 - `align-content: space-around;` prikaz flex linije sa razmakom prije, nakon i između
 - `align-content: stretch;` zauzima sav prostor između linija (default)
 - `align-content: center;` centriranje flex linija (vrsta) po sredini containera
 - `align-content: flex-start;` flex linije zauzimaju container od početka
 - `align-content: flex-end;` flex linije zauzimaju container od kraja
 - Isporobati [primjer](#)

FLEX, IDEALNO CENTRIRANJE

- Postaviti `justify-content` (horizontalno centriranje itema) i `align-items` (vertikalno centriranje itema) na `center`
- Pogledajte [primjer](#)

FLEX, ITEMS

- Flex itemi ili flex children su svi child(samo direktni) elementi flex containera
- Flex itemi imaju sledeće proprijetije:
 - order
 - flex-grow
 - flex-shrink
 - flex-basis
 - flex
 - align-self

FLEX, ORDER

- Property **order** se koristi za slaganje flex itema. Prvi flex item ne mora da bude prvi item koji će se prikazati u layout-u
 - **order**: **number**
 - Pogledati i [primjer](#)

FLEX, FLEX GROW

- Property `flex-grow` definiše koliko flex item može relativno da raste u odnosu na ostale flex iteme, default vrijednost je 0
 - Pogledajmo `primjer` gdje zadnji flex item raste 8 puta brže od ostalih

FLEX, FLEX SHRINK

- Property `flex-shrink` kojim se definiše koliko flex item može da se skupi/smanji u odnosu na ostale flex iteme
 - Pogledajmo [primjer](#)

FLEX, FLEX BASIS I FLEX

- Property flex-basis služi da definišemo inicijalno veličinu flex itema
 - Pogledajte [primjer](#)
- Property flex kombinuje flex-grow, flex-shrink i flex-basis
 - Ako primijenimo `flex: 0 0 200px` na flex item šta dobijamo?
 - Šta ako za item postavimo `flex: 2`, a ostale iteme na `flex: 1`
 - Šta ako imamo procentualne vrijednosti za flex-basis, i ako je flex-grow i flex-shrink 0
 - Šta kad imamo konkretnu dimenziju elementa (basis), npr. 2 div elementa istih dimenzija, sa i bez grow i shrink, šta ako imamo procentualnu vrijednost, isprobajmo !!!

FLEX, ALIGN SELF

- Property `align-self` definiše poravnanje selektovanog itema unutar flex containera
- `Align-self` gazi default poravnanje koji može biti setovano na container element kroz property `align-items`
- `Align-self` može da ima iste vrijednosti kao `align-items`
- Pogledati [primjer](#)

FLEX, PRIMJERI

- Pogledajmo kako može flex da se iskoristi u praksi
 - [Primjer 1](#)
 - [Primjer 2](#)
 - Preporuka da pogledate [tutorial](#) za praktični dio, može vam pomoći i pri izradi 2. domaćeg

Pitanja