

Nastavni predmet	SKRIPTNI JEZICI I WEB PROGRAMIRANJE
Naslov cjeline	CSS – uređivanje web sjedišta
Naslov jedinice	Vježba 21m: CSS media queries i flexbox layout modul

Antonio Matek, Mihael Modrić 3.B

PRIPREMA ZA VJEŽBU

Odgovorite pisano u bilježnicu:

- ▶ 1. Objasnite div element.

<div> je HTML element koji se koristi za grupiranje drugih elemenata zajedno kako bi ih lakše stilizirali ili manipulirali. Na primjer, koristi se za stvaranje sekcija na web stranici.

IZVOĐENJE VJEŽBE

CSS FlexBox modul

- [Isprobajte](#)
- Prije Flexbox Layout modula, postojala su četiri načina za oblikovanje izgleda: blok (za sekcije na mrežnoj stranici), inline (za tekst), tablice (za dvodimenzionalne tabelarne podatke), pozicioniranje (za eksplicitnu poziciju elementa).
- Ovaj modul pojednostavljuje dizajniranje fleksibilne responzivne strukture izgleda bez korištenja float ili pozicioniranja.
- Flexbox elementi: Flexbox model započinje definiranjem flex kontejnera (roditeljskog elementa), npr. za tri flexbox stavke (u nastavku). Flex kontejner postaje fleksibilan postavljanjem svojstva display na flex:

```
<div class="flex-container">  
  <div>1</div>  
  <div>2</div>  
  <div>3</div>  
</div>
```

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

Flex kontejner svojstva su:

- [flex-direction](#) – definira smjer u kojem će se slagati flex stavke vertikalno (od vrha do dna), a [column-reverse](#) obrnuto. Uz to postoje još i mogućnosti [row](#) i [row-reverse](#). Vrijednost [column](#) postavlja ih
- [flex-wrap](#) – određuje hoće li se prema potrebi flex stavke omotavati ili prebacivati u novi red. Vrijednost može biti [wrap](#) i [nowrap](#), te [wrap-reverse](#).
- [flex-flow](#) – ovo svojstvo je skraćeno svojstvo za [flex-direction](#) i [flex-wrap](#).
- [justify-content](#) – koristi se za poravnanje flex stavki ([center](#), [flex-start](#), [flex-end](#), [space-around](#), [space-between](#))
- [align-items](#) – koristi se za poravnanje flex stavki ([center](#), [flex-start](#), [flex-end](#), [stretch](#), [baseline](#)).
- [align-content](#) – za poravnanje flex linija (npr. [space-between](#) vrijednost ispisuje flex linije sa jednakim razmakom između njih, [space-around](#) ispisuje sa razmakom prije, između i nakon njih, [stretch](#) razvuče flex linije kako bi zauzele preostali prostor (podrazumijevano), vrijednost [center](#) ispisuje flex linije u sredini kontejnera, [flex-start](#) vrijednost ispisuje flex linije na početku kontejnera, [flex-end](#) ispisuje na kraju)

Savršeno centriranje:

```
.flex-container {
display: flex; height:
300px; justify-content:
center;
align-items: center;
}
```

- Izravni potomci flex kontejnera automatski postaju fleksibilne stavke (flex).

Svojstva flex stavki su:

- [order](#) – određuje poredak flex stavki. Prva stavka u kodu ne mora se pojaviti kao prva u izgledu. Vrijednost za poredak mora biti broj, podrazumijevana vrijednost je 0. Primjer: `<div style="order: 1">4</div>`
- [flex-grow](#) – određuje koliko će se flex stavka proširivati relativno prema ostalima. Vrijednost mora biti broj, podrazumijevana vrijednost je 0. Primjer: `<div style="flex-grow: 8">3</div>`
- [flex-shrink](#) – određuje koliko će se flex stavka sužavati relativno prema ostalima. Vrijednost mora biti broj, podrazumijevana vrijednost je 1.
- [flex-basis](#) – određuje inicijalnu dužinu flex stavke. Primjer: `<div style="flex-basis: 200px">3</div>`
- [flex](#) – skraćeno svojstvo za [flex-grow](#), [flex-shrink](#) i [flex-basis](#). Primjer: `<div style="flex: 0 0 200px">3</div>`
- [align-self](#) – određuje poravnanje za odabranu stavku unutar flex spremnika. Ovo svojstvo pregazi podrazumijevano poravnanje postavljeno sa [align-items](#) svojstvom spremnika.

Primjer: `<div class="flex-container">`
`<div>1</div>`
`<div style="align-self: flex-start">2</div>`
`<div style="align-self: flex-end">3</div>`
`<div>4</div>`
`</div>`

CSS media queries

- CSS2 uvodi tipove medija. @media rule omogućuje definiranje različitih pravila za stilove koji se primjenjuju za različite tipove medija. Npr, može se postaviti jedan skup pravila oblikovanja za računalne ekrane, drugi za pisače, treći za neke druge uređaje itd.
- CSS3 je uveo Media Queries. Media queries u CSS3 proširuje CSS2 media types ideju: Umjesto da se prati tip uređaje, usredotočuje se na njegove mogućnosti. Media queries mogu se koristiti za provjeru:
 - Širine i visine viewport-a
 - Širine i visine uređaje
 - orijentacije (landscape/portrait)
 - rezolucije
- Upotreba media queries je popularna tehnika za primjenjivanje kod mobilnih uređaja (npr. iPhone i Android telefona).

Media Query Syntax

Media query sastoji se od media type i može sadržavati jedan ili više izraza koji mogu poprimiti vrijednost true ili false.

```
@media not|only mediatype and (expressions) {  
  CSS-Code;  
}
```

Rezultat upita je istina ako navedeni tip medija odgovara tipu uređaja na kojem se dokument ispisuje i ako su svi izrazi u upitu istiniti. Kad je media query true, primjenjuje se odgovarajući stil ili pravila oblikovanja kao što je to uobičajeno.

Ako se ne koristi **not** ili **only** operatore, media type je opcionalan i primjenjuju se svi tipovi.

Mogu postojati različita oblikovanja (stylesheets) za različite medije:

```
<link rel="stylesheet" media="mediatype and|not|only (expressions)" href="print.css">
```

Jednostavni media queries primjeri

```
@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
} //mijenja boju pozadine
```

```
@media screen and (min-width: 480px) {  
  #leftsidebar {width: 200px; float: left;}  
  #main {margin-left: 216px;}  
} //mijenja izgled navigacije
```

- Proučite i isprobajte [primjer1](#), [primjer2](#), te [primjer3](#).

- Proučite [primjere](#) za realizaciju responzivnosti.

PONAVLJANJE I PRAKTIČNA VJEŽBA:

1. U online [editoru](#) provježbajte naučeno.
2. Prema [primjeru](#) napravite mrežnu stranicu (galeriju) sa tematski i koloristički usklađenim odabirom od najmanje dvanaest kvalitetnih slika jednake širine, a različitih visina.
3. Prema [primjeru](#) napravite mrežnu stranicu o sebi.

Napomena: prezentirane vježbe se za svakog učenika trebaju razlikovati.

```
<!DOCTYPE html>
<html>
<head>
<style>
* {
  box-sizing: border-box;
}

body {
  margin: 0;
  font-family: Arial;
}

.header {
  text-align: center;
  padding: 32px;
}

.row {
  display: flex;
  flex-wrap: wrap;
  padding: 0 4px;
}

/* Create four equal columns that sits next to each other */
.column {
  flex: 25%;
  max-width: 25%;
  padding: 0 4px;
}

.column img {
  margin-top: 8px;
  vertical-align: middle;
}
```

```

/* Responsive layout - makes a two column-layout instead of four columns */
@media screen and (max-width: 800px) {
  .column {
    flex: 50%;
    max-width: 50%;
  }
}

/* Responsive layout - makes the two columns stack on top of each other instead of next to each other */
@media screen and (max-width: 600px) {
  .column {
    flex: 100%;
    max-width: 100%;
  }
}
</style>
</head>
<body>

<!-- Header -->
<div class="header">
  <h1>Responsive Image Grid</h1>
  <p>Resize the browser window to see the responsive effect.</p>
</div>

<!-- Photo Grid -->
<div class="row">
  <div class="column">
    
    
    
    
    
    
    
  </div>

  <div class="column">
    

```


</div>

<div class="column">

<img src="https://www.celebritycruises.com/blog/content/uploads/2020/09/