

Nastavni predmet	SKRIPTNI JEZICI I WEB PROGRAMIRANJE
Naslov cjeline	CSS – uređivanje web sjedišta
Naslov jedinice	Vježba 23: Responzivni web dizajn

## CILJ VJEŽBE

Učenik će objasniti i demonstrirati, te primjenjivati i oblikovanje responzivnih mrežnih stranica, pomoću CSS pravila kreirati i oblikovati izgled responzivnih mrežnih stranica i mrežnog sjedišta, kreirati i koristiti prekidne točke na mrežnoj stranici.

## PRIPREMA ZA VJEŽBU

Prisjetite se i odgovorite pisano u bilježnicu:

- ▶ 1. Navedite uređaje na kojima je moguće pregledavati mrežne stranice.
- ▶ 2. Navedite neke tipične širine ekrana (širina x visina) za pojedine skupine uređaja (u pikselima).

## IZVOĐENJE VJEŽBE

### CSS responzivni web dizajn

- S obzirom da se mrežne stranice pregledavaju na različitim uređajima, responzivni web dizajn omogućuje da mrežna stranica izgleda dobro na svim tim uređajima, te da se koristi jednostavno neovisno o uređaju.
- Responzivni web dizajn koristi samo HTML i CSS.
- Mrežne stranice ne bi smjele odbacivati informacije kako bi se mogle prikazati na manjim uređajima, nego bi trebale prilagoditi svoj sadržaj prema uređaju.
- Dakle, pod nazivom responzivni web dizajn podrazumjeva se korištenje HTML-a i CSS-a kako bi se mijenjalo veličinu, skrivalo, smanjivalo, povećavalo ili pomicalo sadržaj kako bi izgledao dobro neovisno o uređaju.



Desktop



Tablet



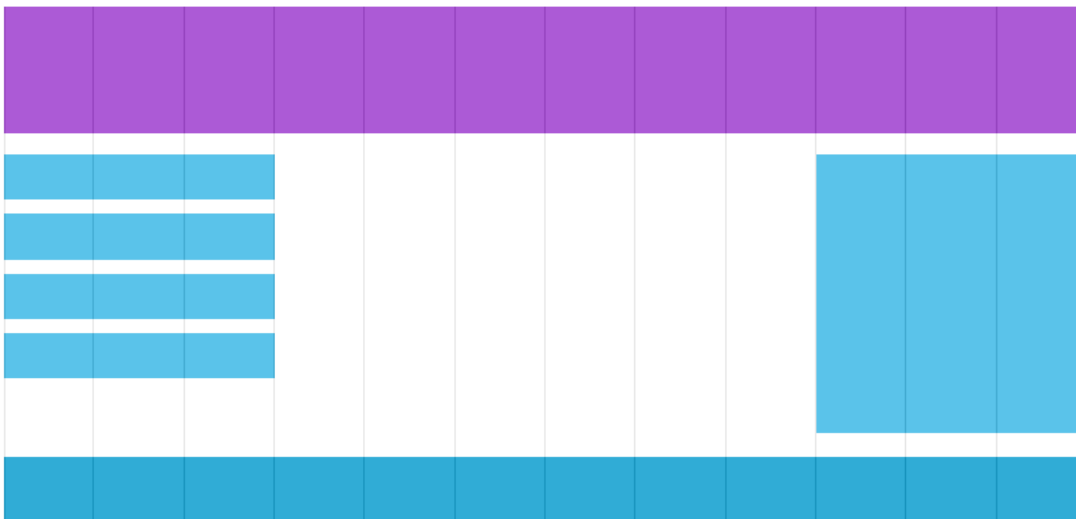
Phone

•

- Osnovni pojam koji je potrebno shvatiti kod RWD (responzivnog web dizajna) je viewport
- Viewport je korisniku vidljivo područje web stranice, ovisi o uređaju i manje je na mobilnom telefonu nego na stolnom računalu. Prije pojave tablet i mobilnih telefona, mrežne su stranice bile dizajnirane samo za ekrane računala i bilo je uobičajeno da imaju statički dizajn i točno određenu veličinu. Tako definirane stranice sada više ne stanu na nove, male uređaje koje koristimo za pretraživanje interneta. Kako bi to riješio, preglednik pribjegava smanjenju kompletne stranice (quick fix).
- HTML je uveo novu metodu kako bi web dizajnerima omogućio kontrolu viewporta kroz `<meta>` element, tako da bi na svim stranicama trebalo uključiti:
  - `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- Ovo pregledniku daje uputu kako da upravlja dimenzijama i promjenom veličine stranice.
- `width=device-width` – postavlja širinu stranice da bude prilagođena širini ekrana uređaja
- `initial-scale=1.0` – postavlja inicijalnu ili početnu razinu zooma-a za trenutak kada je stranica prvi puta učitana od strane preglednika.
- Korisnici mrežne stranice pomiču okomito bez obzira na uređaj i nisu ih navikli pomicati vodoravno, pa postupak takvog pomicanja ili potrebe za zoomiranjem kako bi vidjeli čitavu stranicu, rezultira lošim korisničkim iskustvom.
- Vrijede slijedeća dodatna pravila:
  1. NEMOJTE KORISTITI velike elemente točno određene veličine – npr., ako je slika prikazana na širini koja je šira od viewport-a, to bi za posljedicu imalo vodoravno skrolanje. Takv sadržaj treba biti prilagođen širini viewport-a.
  2. NE DOZVOLJAVAJTE da se sadržaj oslanja na točno određenu širinu viewport-a kako bi se ispravno prikazao – naime, dimenzije i širina ekrana u CSS pikselima se jako razlikuju među uređajima.
  3. KORISTITE CSS media queries kako biste primijenili različita oblikovanja za male i velike ekrane – postavljanje velikih apsolutnih CSS širina za elemente stranice uzrokovat će da će element biti preširok za širinu viewport-a na malom uređaju. Umjesto toga, razmotrite korištenje relativno zadanih vrijednosti širina, npr. 100%. Također, čuvajte se upotrebe velikih vrijednosti za apsolutno pozicioniranje, jer one mogu za posljedicu imati da element ispadne izvan viewport-a na malim uređajima.

## CSS Grid-view (rešetkasti pogled)

- Mnoge su stranice temeljene na tzv. grid pogledu, što znači da je stranica podijeljena u stupce:



- Upotreba grid-pogleda jako pomaže kod dizajniranja mrežnih stranica jer pojednostavnjuje smještanje elemenata na stranicu.
- Responzivni grid-view često ima 12 stupaca i ukupnu širinu od 100% te se može sužavati ili širiti prilikom promjene veličine prozora preglednika. Pogledajte primjer na [poveznici](#).
- Kako bi se napravio responzivni grid-view potrebno je prvo osigurati da svi HTML elementi imaju svojstvo box-sizing postavljeno na border-box. Time osiguravamo da su padding i border uključeni u ukupnu širinu i visinu elemenata:
- ```
* {
  box-sizing: border-box;
}
```
- Jednostavna mrežna stranica sa samo dva stupca:
- ```
.menu {
  width: 25%;
  float: left;
}
.main {
  width: 75%;
  float: left;
}
```
- Obično želimo koristiti 12 stupaca kako bismo imali veću kontrolu nad izgledom mrežne stranice. Prvo treba izračunati postotak za jedan stupac:  $100\% / 12 \text{ stupaca} = 8.33\%$ . Nakon toga potrebno je napraviti klasu za svaki od 12 stupaca, class="col-" i dodaje se broj koji definira koliko stupaca ta sekcija obuhvaća:
- ```
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```
- Svi ti stupci trebaju biti smješteni lijevo:
- ```
[class*="col-"] {
  float: left;
  padding: 15px;
  border: 1px solid red;
}
```
- Svaki se stupac omata sa div elementom, a u svakom retku broj stupaca uvijek mora biti 12, npr.:
- ```
<div class="row">
  <div class="col-3">...</div> <!-- 25% -->
  <div class="col-9">...</div> <!-- 75% -->
</div>
```
- Stupci u retku smješteni su lijevo te su time izuzeti iz tijeka stranice, pa će drugi elementi biti smješteni kao da ti stupci ne postoje. Da bi to spriječili treba dodati oblikovanje koje će očistiti tijek:

- ```
.row::after {
  content: "";
  clear: both;
  display: table;
}
```
- Zatim se može za bolji izgled dodati i malo boja i oblikovanja:
- Pogledajte i proučite primjer na [stranici](#).

## CSS media queries

- Media queries je tehnika uvedena sa CSS3. Ona koristi @media pravilo kako bi uključila skup (blok) CSS svojstava samo ako je određeni uvjet ispunjen, npr. ako je prozor preglednika 600 piksela ili manji, pozadinska boja je svijetlo plava:
- ```
@media only screen and (max-width: 600px) {
  body {
    background-color: lightblue;
  }
}
```
- Media queries mogu pomoći da mrežna stranica koja ima retke i stupce bude responzivna i da dobro izgleda na malim ekranima. Kod toga možemo dodati prekidnu točku (breakpoint) na mjestima na kojima se određeni dijelovi dizajna trebaju ponašati različito na obje strane te prekidne točke, npr. kada ekran (prozor preglednika) postane manji od 768px, svaki stupac treba imati širinu 100%:
- ```
/* For desktop: */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
.col-4 {width: 33.33%;}
.col-5 {width: 41.66%;}
.col-6 {width: 50%;}
.col-7 {width: 58.33%;}
.col-8 {width: 66.66%;}
.col-9 {width: 75%;}
.col-10 {width: 83.33%;}
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}

@media only screen and (max-width: 768px) {
  /* For mobile phones: */
  [class*="col-"] {
    width: 100%;
  }
}
```
- Uvijek dizajnirajte stranicu metodom mobile first. Mobile First znači oblikovanje za mobilne uređaje prije oblikovanja za stolni ili bilo koji drugi uređaj (na taj će se način stranica ispisivati brže na manjim uređajima).
- U tom slučaju, umjesto da se oblikovanje mijenja kad širina postane manja od 768px, dizajn se mora mijenjati kad širina postane veća od 768px:
- ```
/* For mobile phones: */
[class*="col-"] {
```

```

width: 100%;
}

@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
  .col-12 {width: 100%;}
}

```

- Prekidne se točke dodaju prema potrebi (za više različitih vrsta, odnosno veličina uređaja) i može ih biti i više:

- ```
/* For mobile phones: */
[class*="col-"] {
  width: 100%;
}
```

```

@media only screen and (min-width: 600px) {
  /* For tablets: */
  .col-s-1 {width: 8.33%;}
  .col-s-2 {width: 16.66%;}
  .col-s-3 {width: 25%;}
  .col-s-4 {width: 33.33%;}
  .col-s-5 {width: 41.66%;}
  .col-s-6 {width: 50%;}
  .col-s-7 {width: 58.33%;}
  .col-s-8 {width: 66.66%;}
  .col-s-9 {width: 75%;}
  .col-s-10 {width: 83.33%;}
  .col-s-11 {width: 91.66%;}
  .col-s-12 {width: 100%;}
}

```

```

@media only screen and (min-width: 768px) {
  /* For desktop: */
  .col-1 {width: 8.33%;}
  .col-2 {width: 16.66%;}
  .col-3 {width: 25%;}
  .col-4 {width: 33.33%;}
  .col-5 {width: 41.66%;}
  .col-6 {width: 50%;}
  .col-7 {width: 58.33%;}
  .col-8 {width: 66.66%;}
  .col-9 {width: 75%;}
  .col-10 {width: 83.33%;}
  .col-11 {width: 91.66%;}
}

```

```
.col-12 {width: 100%;}
}
```

- **HTML kod će izgledati npr. ovako:**

```
<div class="row">
  <div class="col-3 col-s-3">...</div>
  <div class="col-6 col-s-9">...</div>
  <div class="col-3 col-s-12">...</div>
</div>
```

- **Tipične prekidne točke za pet grupa uređaja:**

```
/* Extra small devices (phones, 600px and down) */
@media only screen and (max-width: 600px) {...}
```

```
/* Small devices (portrait tablets and large phones, 600px and up) */
@media only screen and (min-width: 600px) {...}
```

```
/* Medium devices (landscape tablets, 768px and up) */
@media only screen and (min-width: 768px) {...}
```

```
/* Large devices (laptops/desktops, 992px and up) */
@media only screen and (min-width: 992px) {...}
```

```
/* Extra large devices (large laptops and desktops, 1200px and up) */
@media only screen and (min-width: 1200px) {...}
```

- **Media queries mogu se koristiti i za promjenu izgleda stranice ovisno o orijentaciji preglednika:**

```
@media only screen and (orientation: landscape) {
  body {
    background-color: lightblue;
  }
}
```

- **Također, mogu se koristiti za skrivanje elemenata na različitim veličinama ekrana ili za promjenu veličine fonta:**

```
/* If the screen size is 600px wide or less, hide the element */
@media only screen and (max-width: 600px) {
  div.example {
    display: none;
  }
}
```

```
/* If the screen size is 601px or more, set the font-size of <div> to 80px */
@media only screen and (min-width: 601px) {
  div.example {
    font-size: 80px;
  }
}
```

```
/* If the screen size is 600px or less, set the font-size of <div> to 30px */
@media only screen and (max-width: 600px) {
  div.example {
    font-size: 30px;
  }
}
```

## PONAVLJANJE I PRAKTIČNA VJEŽBA:

1. U online [editoru](#) ili editoru koji uobičajeno koristite provježbajte naučeno.
2. Napravite rješenje za mrežnu stranicu koja će (prema uzoru iz primjera za responzivni web dizajn – grad Chania), predstavljati korisniku neku odabranu biljku ili životinju. Pri tome uvažite slijedeće upute:
  - Stranica treba imati osobine responzivne mrežne stranice
  - U naslovnom dijelu stranice stoji naziv biljke ili životinje
  - U lijevom dijelu smještene su poveznice vezane uz sadržaj, odnosno poglavlja (npr. o staništu biljke/životinje, o osobinama, prehrani, razmnožavanju i sl.)
  - Središnji (najveći) dio stranice obuhvaća glavni sadržaj
  - Tekst koji čini glavni sadržaj trebao bi biti popraćen slikama (o smještaju slika i radu sa pozadinom vidi na [poveznici](#))
  - Dodavanje video materijala je proizvoljno, a može se pogledati na [poveznici](#)
  - Na stranici je potrebno prilagoditi boje i fontove
  - Prikaz stranice je potrebno provjeriti za veličine viewport-a kod najmanje tri grupe uređaja (pri tom definirajte prekidne točke): mobilni telefon, tablet i prijenosno, odnosno stolno računalo