**Responsive design**

**00:20:36**

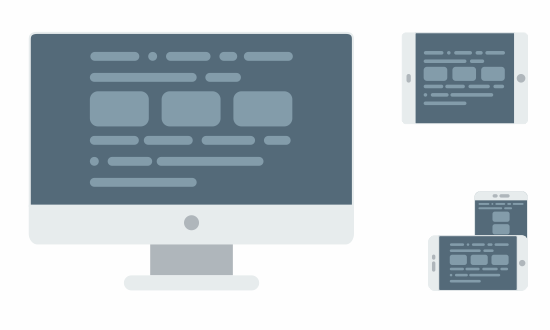
Jedinica: 17 od 30

**+Rezime**

Kada je raspoređivanje elemenata na HTML stranici u pitanju, ne može se prenebregnuti problem koji nastaje prilikom prikaza jedne HTML stranice na uređajima koji poseduju displeje različitih veličina i rezolucija. Prilikom izgradnje HTML aplikacije, potrebno je voditi računa da prikaz podataka uvek bude optimalan, tako da korisnik u svakoj situaciji i bez obzira na uređaj koji koristi, bude u stanju da na jednostavan i pregledan način rukuje podacima. Da bi se tako nešto postiglo, neophodno je poznavati načela prilagodljivog dizajna korisničkog interfejsa, odnosno responsive design-a.

**Šta je responsive design?**

Sam pojam *responsive* odnosi se na brzo i adekvatno reagovanje na nastale promene. Stoga je *responsive design* pristup kreiranju korisničkog interfejsa aplikacija, tako da on izgleda adekvatno na svim uređajima (desktop računarima, tabletima i telefonima). Takođe, pored pukog izgleda, responsive dizajn se bavi opštim korisničkim ugođajem prilikom korišćenja aplikacije, tako da on bude uniforman, bez obzira na veličinu displeja, platformu i orijentaciju uređaja. Ukoliko korisnik pristupa aplikaciji sa različitih uređaja, responsive design garantuje odgovarajuću reakciju aplikacije, po kojoj se ona prilagođava hardverskim i softverskim osobinama uređaja koji korisnik koristi. Koncept responsive designa ilustrovan je slikom 17.1.



*Slika 17.1 – Koncept responsive designa*

Koncept responsive dizajna potekao je iz arhitekture, gde je nastao kao potreba da se fizički prostori u kojima ljudi žive i rade prilagode prisustvu ljudi koji u takvom prostoru provode vreme. Kombinacijom različitih senzora, pametnih materijala koji imaju mogućnost ugibanja i istezanja i mikrokontrolera, stvorili bi se prostori koji bi se u svakom trenutku mogli prilagoditi ljudskim potrebama. Upravo na takvim načelima zasniva se i responsive design korisničkih interfejsa aplikacija. Umesto da se za svaki različiti tip uređaja kreira nova verzija aplikacije, responsive design omogućava da se jedna aplikacija prilagođava različitim situacijama u kojima se nađe.

**Fleksibilni raspored komponenata**

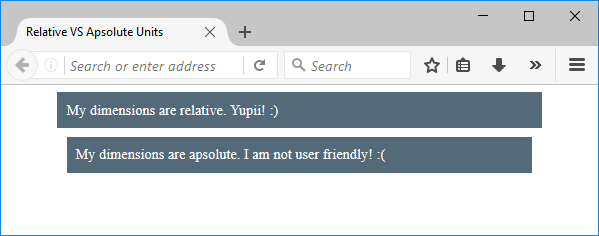
Kreiranje fleksibilnog rasporeda komponenata jeste prvo načelo responsive web designa. Zasniva se na praksi kreiranja elemenata koji svoju veličinu i poziciju mogu da prilagode situacijama u kojima se nalaze.

Osnovno pravilo koje je neophodno poštovati kako bi se kreirao prilagodljivi layout jeste davanje prednosti relativnim jedinicama u odnosu na apsolutne. Relativne i apsolutne jedinice pomenute su u jednoj od prethodnih lekcija, kada je rečeno da su relativne jedinice one koje se odnose na neke druge vrednosti u svom okruženju. Tako se relativnim jedinicama vrednost ne postavlja eksplicitno, već na osnovu neke druge vrednosti. Najpoznatije relativne jedinice su procenti i em-ovi. Sledeći primer ilustruje razliku između apsolutnih i relativnih jedinica:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | <!DOCTYPE html>  <html>  <head>   <title>Relative VS Apsolute Units</title>   <style type="text/css">      div{     margin: 0px auto 10px auto;     padding: 10px;     background-color: #546A79;     color: white;    }      #div1{     width: 80%;    }      #div2{     width: 500px;    }     </style>  </head>  <body>   <div id="div1">My dimensions are relative. Yupii! :)</div>   <div id="div2">My dimensions are apsolute. I am not user friendly! :(</div>  </body>  </html> |

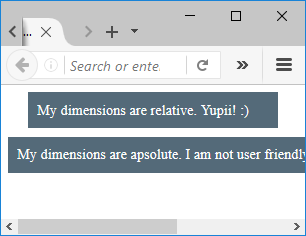
|  |
| --- |
| **Napomena**    *Svi primeri iz ove lekcije nalaze se u prilogu.* |

U primeru su kreirana dva div elementa, od kojih je jednom širina postavljena korišćenjem apsolutnih jedinica, a drugom korišćenjem relativnih. Div sa apsolutnom širinom uvek će imati istu širinu, bez obzira na veličinu [viewporta](https://www.link-elearning.com/linkdl/opisPojma.php?id=144341). Kada je širina viewporta veća od 500 px, situacija će biti kao na slici 17.2.



*Slika 17.2 – Različiti efekat relativnih i apsolutnih jedinica*

Sa slike 17.2 se može videti da se širina prvog div elementa povećava sa povećavanjem dostupne širine. Situacija koja nastaje kada se viewport smanji ilustrovana je slikom 17.3.



*Slika 17.3 – Različiti efekat relativnih i apsolutnih jedinica (2)*

Kada se viewport smanji, kao na slici 17.3, simuliraju se uslovi koji postoje na uređajima sa manjim displejima. Prvi div element se adekvatno prilagođava veličini dostupnog prostora, dok tako nešto nije slučaj sa drugim div elementom koji uvek ima širinu od 500 px. To rezultuje odsecanjem dela informacija iz vidnog polja, što narušava opšti korisnički ugođaj.  
  
U prethodnom primeru iskorišćena je relativna jedinica, procenat (%). Pored em-ova, to je najkorišćenija relativna jedinica. Ipak, pored ove dve jedinice, CSS3 uvodi još nekoliko vrlo korisnih relativnih jedinica. One su prikazane u tabeli 17.1.

|  |  |
| --- | --- |
| **Jedinica** | **Opis** |
| vw | jedinica relativna širini vidnog polja (viewporta) |
| vh | jedinica relativna visini vidnog polja (viewporta) |
| vmin | jedinica relativna manjoj dimenziji |
| vmax | jedinica relativna većoj dimenziji |

*Tabela 17.1 – Viewport jedinice*

Jedinice vw i vh odnose se na x i y ose vidnog polja browsera. Jedinica vw relativna je širini vidnog polja browsera, dok je jedinica vh relativna visini vidnog polja browsera.

S obzirom da se orijentacija prenosnih uređaja lako može promeniti iz *portrait* u *landscape* mod jednostavnim okretanjem uređaja, može se desiti da u jednom trenutku visina bude veća od širine, a drugom da širina bude veća od visine. Kako bi se odgovorilo na ovakve promene, postoje jedinice vmax i vmin. Jedinica vmin relativna je dimenziji vidnog polja koja je manja, bilo da je to visina ili širina. S druge strane, jedinica vmax relativna je većoj dimenziji vidnog polja. Tabela 17.2 ilustruje različite vrednosti iskazane viewport jedinicama.

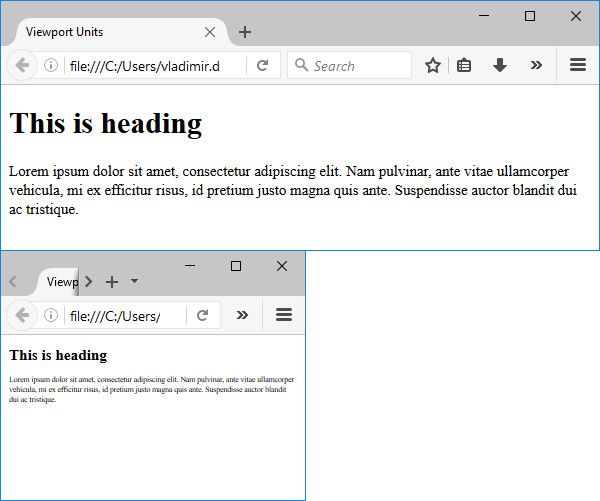
|  |  |
| --- | --- |
| **Vrednost** | **Značenje** |
| 1vw | 1% širine vidnog polja |
| 33vw | 33% širine vidnog polja |
| 1vh | 1% visine vidnog polja |
| 45vh | 45% visine vidnog polja |
| 30vmin | 30% visine ili širine, u zavisnosti od toga koja je veličina manja |
| 30vmax | 30% visine ili širine, u zavisnosti od toga koja je veličina veća |

*Tabela 17.2 – Viewport jedinice i vrednosti*

 Relativne viewport jedinice često se koriste za definisanje veličine teksta:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | <!DOCTYPE html>  <html>  <head>   <title>Viewport Units</title>   <style type="text/css">      h1{     font-size: 4vmax;    }    p{     font-size: 1.5vmax;    }     </style>  </head>  <body>   <h1>This is heading</h1>   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam pulvinar, ante vitae ullamcorper vehicula, mi ex efficitur risus, id pretium justo magna quis ante. Suspendisse auctor blandit dui ac tristique. </p>  </body>  </html> |

Prikazani primer proizvodi efekat kao na slici 17.4.



*Slika 17.4 – Postavljanje veličine teksta korišćenjem viewport jedinica*

U nastavku lekcije biće prikazano kako se primer jednog layouta sa definisanim apsolutnim jedinicama može transformisati korišćenjem relativnih jedinica. Primer statičkog layouta je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | <!DOCTYPE html>  <html>  <head>   <title>Viewport Units</title>   <style type="text/css">      .main-container {     margin: auto;     width: 578px;    }    section,    aside {     margin: 10px;     padding: 10px;     font-size: 20px;     background-color: #546A79;     color:white;     text-align: center;     font-family: Verdana;    }    section {     float: left;     width: 340px;    }    aside {     float: right;     width: 158px;    }     </style>  </head>  <body>   <div class="main-container">    <section>SECTION</section>    <aside>SIDEBAR</aside>   </div>  </body>  </html> |

Stranica iz primera sadrži jedan glavni kontejner označen id-jem, *main-container*, koji unutar sebe sadrži jedan section i jedan aside element. Širine svih elemenata su postavljene korišćenjem apsolutnih jedinica, odnosno piksela. Pritom je širina div elementa dobijena na sledeći način:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | section width + aside width + section margin\*2 + aside margin\*2 + section padding \*2 + aside padding \*2  =  340 + 158 + 10\*2 + 10\*2 + 10\*2 + 10\*2  =  340 + 158 + 20 + 20 + 20 + 20  = 578 |

Formula koja se može koristiti za konvertovanje apsolutnih vrednosti u relativne je sledeća:

vrednost na elementu / roditeljska vrednost \* 100

Korišćenjem prikazane formule, apsolutne vrednosti iz primera se mogu konvertovati u relativne na sledeći način:

|  |  |
| --- | --- |
| 1  2  3 | section: 340 / 578 \* 100% = 58.8235294  aside: 158 / 578 \* 100% = 27.3356401  padding, margin: 10 / 578 \* 100% = 1.7301038 |

Stilizacija sa relativnim jedinicama, nakon primene svih opisanih tehnika, može da izgleda ovako:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | .main-container {   margin: auto;   width: 75%;  }  section,  aside {   margin: 1.7301038%;   padding: 1.7301038%;   font-size: 1.5vmax;   background-color: #546A79;   color:white;   text-align: center;   font-family: Verdana;  }  section {   float: left;   width: 58.8235294%;  }  aside {   float: right;   width: 27.3356401%;  } |

Sve apsolutne jedinice zamenjene su relativnim, tako da je layout u potpunosti prilagodljiv svim mogućim situacijama.

Ukoliko je potrebno preuzeti još veću kontrolu nad veličinom različitih komponenata, moguće je uposliti i četiri CSS svojstva koja omogućavaju definisanje maksimalnih i minimalnih vrednosti visine i širine. Ta svojstva su prikazana u tabeli 17.3.

|  |  |
| --- | --- |
| **Svojstvo** | **Opis** |
| min-width | Minimalna širina elementa |
| max-width | Maksimalna širina elementa |
| min-height | Minimalna visina elementa |
| max-height | Maksimalna visina elementa |

*Tabela 17.3 – CSS svojstva za definisanje maksimalne i minimalne visine i širine*

**Kako bi sajt bio prilagodljiviji, prednost je uvek potrebno dati:**

 relativnim jedinicama

 apsolutnim jedinicama

**Medija upiti (media queries)**

Kreiranje korisničkog interfejsa korišćenjem isključivo relativnih jedinica nekada može biti vrlo zahtevan posao. Stoga CSS poznaje još jednu tehniku koja dozvoljava da se određeni CSS opisi primene samo u slučaju zadovoljenja nekog uslova. Tako nešto se postiže korišćenjem  medija upita (*engl. Media Query*), i slobodno se može reći da su medija upiti najmoćniji pristup realizovanju responsive designa.

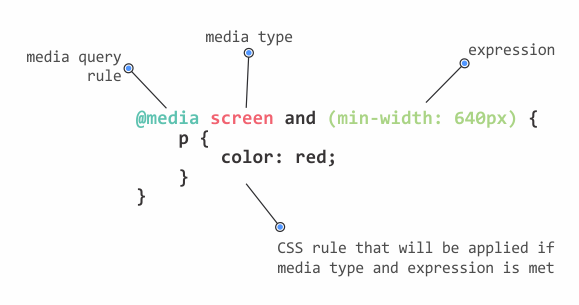
Medija upit se sastoji iz tipa medija na koji će se definisani CSS opisi odnositi, i jednog ili više uslovnih izraza. Pritom je tip medija opcioni podatak koji se ne mora navesti. Opšta sintaksa medija upita je sledeća:

|  |  |
| --- | --- |
| 1 | @media not|only mediatype and (media feature) {    CSS-Code; } |

Primer jednog medija upita je sledeći CSS kod:

|  |  |
| --- | --- |
| 1  2  3  4  5 | @media screen and (min-width: 640px) {      p {          color: red;      }  } |

Različiti elementi prikazanog medija upita predstavljeni su slikom 17.5.



*Slika 17.5 – Različiti segmenti medija upita*

Medija upit započinje @media ključnom rečju, koja je na slici 17.5 obojena plavom bojom. Nakon nje, unutar medija upita može se naći tip medija na koji će se upit odnositi. Na slici 17.5, tip medija je screen, i obojen je crvenom bojom. Medija upiti poznaju sledeće tipove medija:

* all
* screen
* print
* speech

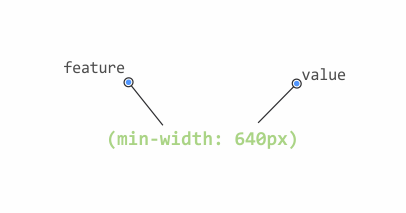
Tip medija definiše tip uređaja za koje će se CSS opisi definisani unutar medija upita primenjivati. Tako se u slučaju medija tipa print, stilovi primenjuju samo prilikom štampe. Tip screen definiše da se stilizacija primenjuje kada se HTML dokument prikazuje na displejima kompjutera, tableta i mobilnih telefona. Tip speech se koristi za uređaje koji čitaju sadržaj stranice naglas, takozvane čitače ekrana. Na kraju, tip all se odnosi na sve nabrojane situacije korišćenja jednog HTML dokumenta. Ukoliko se vrednost medija tipa ne navede, podrazumevana vrednost je screen.

Mnogo značajniji deo medija upita jeste deo koji se odnosi na kreiranje izraza. Svaki medija upit se može sastojati iz jednog ili više izraza, koji definišu u kojim situacijama će se stilizacija unutar medija upita aktivirati.

U prikazanom primeru, izraz medija upita je sledeći:

|  |  |
| --- | --- |
| 1 | (min-width: 640px) |

Unutar izraza medija upita, zapravo se vrši proveravanje neke do funkcionalnosti uređaja na kome se HTML stranica čita. U konkretnom prikazanom primeru, proverava se da li je viewport uređaja na kome se HTML dokument čita jednak ili veći od 640 px. Tako se jedan izraz unutar medija upita sastoji iz testirane funkcionalnosti i vrednosti, kao na slici 17.6.



*Slika 17.6 – Različiti segmenti izraza medija upita*

CSS omogućava testiranje različitih funkcionalnosti uređaja koji parsiraju HTML dokument. Najznačajnije osobine uređaja koje se mogu testirati prikazane su u tabeli 17.4.

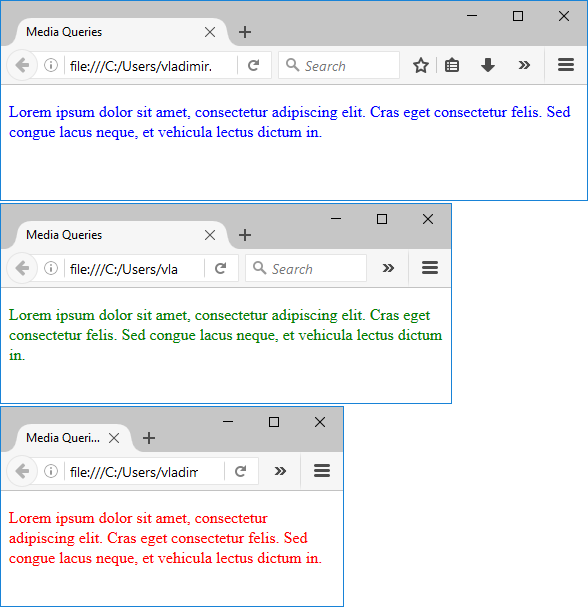
|  |  |
| --- | --- |
| **Osobina** | **Opis** |
| aspect-ratio | odnos širine i visine vidnog polja |
| device-aspect-ratio | odnos širine i visine uređaja za prikaz slike |
| color | broj bita po jednoj boji izlaznog uređaja za prikaz slike |
| device-height | visina displeja za prikaz slike |
| device-width | širina displeja za prikaz slike |
| height | visina vidnog polja |
| width | širina vidnog polja |
| orientation | orijentacija vidnog polja, odnosno da li je uređaj u portret ili landscape modu |
| resolution | rezolucija izlaznog uređaja, odnosno gustina tačaka, koja se izražava korišćenjem jedinica dpi ili dpcm |

*Tabela 17.4 – Različite osobine uređaja koje se mogu testirati unutar medija upita*

Pogledajmo jedan primer sa definisanih nekoliko medija upita:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 | <!DOCTYPE html>  <html>  <head>   <title>Media Queries</title>   <style type="text/css">      @media screen and (max-width: 399px) {     p {      color: red;     }    }      @media screen and (max-width: 549px) and (min-width: 400px) {     p {      color: green;     }    }        @media screen and (min-width: 550px) {     p {      color: blue;     }    }     </style>  </head>  <body>   <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras eget consectetur felis. Sed congue lacus neque, et vehicula lectus dictum in. </p>  </body>  </html> |

Ovoga puta su unutar primera definisana tri medija upita, za različite širine vidnog polja uređaja koji čita HTML sadržaj. Efekat prikazanog koda će biti kao na slici.



*Slika 17.7 – Medija upiti na delu*

Na slici 17.7, može se videti kako se boja teksta paragrafa menja u zavisnosti od širine vidnog polja.

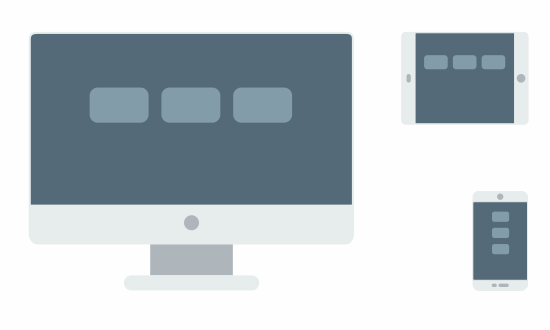
Na ovaj način se može postići moćna manipulacija rasporedom i veličinom elemenata na stranici i na taj način postići responsive design. Sledeći primer ilustruje način na koji se može optimizovati prikaz tri slike unutar HTML stranice za različite uređaje. HTML markup je sledeći:

|  |  |
| --- | --- |
| 1  2  3  4  5 | <div id="container">  <img src="placeholder.png" alt="placeholder">  <img src="placeholder.png" alt="placeholder">  <img src="placeholder.png" alt="placeholder">  </div> |

Stilizacija sa medija upitima za ovakve elemente je sledeća:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53 | #container{  margin: auto;  text-align: center;  }    @media screen and (max-width: 439px) {  #container{   width: 300px;  }    img{   width: 280px;  }      }    @media screen and (min-width: 440px) and (max-width: 699px) {  #container{   width: 400px;    }    img{   width: 125px;  }      }    @media screen and (min-width: 700px) and (max-width: 1099px) {  #container{   width: 640px;    }    img{   width: 200px;  }      }      @media screen and (min-width: 1100px) {  #container{   width: 960px;  }    img{   width: 300px;  }  } |

Ovoga puta su definisani medija upiti za četiri različite veličine vidnog polja. Medija upiti će stvoriti efekat koji je ilustrovan slikom 17.8.



*Slika 17.8 – Medija upiti na delu(2)*

Svi medija upiti prikazani u dosadašnjim primerima definisani su unutar same stilizacije. Medija upiti se mogu definisati na nekoliko različitih načina:

* korišćenjem media atributa na link elementu, prilikom uključivanja eksterne stilizacije
* korišćenjem @media pravila unutar interne ili eksterne stilizacije
* korišćenjem @import pravila, prilikom importovanja eksternog CSS fajla

Definisanje medija upita prilikom linkovanja:

|  |  |
| --- | --- |
| 1 | <link href="styles.css" rel="stylesheet" media="all and (max-width: 1024px)"> |

Definisanje medija upita korišćenjem @importa direktive:

|  |  |
| --- | --- |
| 1 | @import url(styles.css) all and (max-width: 1024px) {...} |

Na ovaj način će se svi CSS opisi definisani unutar fajla sa eksternom stilizacijom primenjivati samo na uređaje sa širinom vidnog polja do 1024 px.

|  |
| --- |
| **Napomena**    *Preporučuje se korišćenje @media pravila unutar postojeće stilizacije kako bi se smanjio broj zahteva koje je potrebno da browser uputi serveru, kako bi dobio sve potrebne fajlove za adekvatan prikaz stranice.* |

**JavaScript**

S obzirom da se responsive design tiče celokupnog korisničkog ugođaja na različitim uređajima, potrebno je o prilagodljivosti razmišljati i prilikom pisanja JavaScript koda. JavaScript omogućava praćenje različitih dešavanja u životu jedne aplikacije, odnosno praćenje pojava koje se nazivaju događaji. Ipak, različiti događaji se na različitim uređajima obrađuju na različite načine, odnosno, ne poseduju svi uređaji istovetne događaje. Jedan od JavaScript događaja jeste događaj do koga dolazi kada korisnik pritisne taster miša. Ali, šta je sa uređajima koji ne poseduju miš, kao što su pametni telefoni? Na telefonima sa ekranima osetljivim na dodir, događaj ekvivalentan pritisku tastera miša je događaj spuštanja prsta na displej, takozvani tap. Ukoliko naša aplikacija sluša samo događaj pritiska tastera miša, na uređajima sa ekranima osetljivim na dodir, funkcionalnost koja je povezana sa takvim događajem neće funkcionisati. O ovakvim i sličnim stvarima je potrebno voditi računa, kako bi se responsive design na najbolji mogući način implementirao.

|  |
| --- |
| **Napomena**    *JavaScript jeziku biće posvećen veliki deo ovoga kursa, počevrši od sledećeg modula.* |