

*Matematički fakultet*  
*Univerzitet u Beogradu*

# Elektronske lekcije o radnom okruženju Bootstrap

---

Master rad

**Student: Tanja Ivanović**

**Mentor: prof. dr Miroslav Marić**

# Sadržaj

Uvod.....	4
1. Korišćenje radnog okruženja Bootstrap.....	6
2. Klase omotača i mrežni sistem.....	8
2.1 Mrežna klasa <i>lg</i> .....	9
2.2 Mrežna klasa <i>md</i> .....	9
2.3 Mrežna klasa <i>sm</i> .....	9
2.4 Mrežna klasa <i>xs</i> .....	10
2.5 Kombinacije mrežnih klasa.....	10
3. Tipografija.....	11
3.1 Zadavanje boje teksta.....	13
3.2 Zadavanje boje pozadini teksta.....	13
3.3 Tipografske klase za formatiranje teksta i lista.....	14
4. Prikazivanje tabela.....	15
4.1 Zadavanje boje redovima tabele.....	16
5. Prikazivanje slika.....	17
5.1 Prilagođavanje slika rezoluciji uređaja.....	17
5.2 Prilagođavanje video zapisa rezoluciji uređaja.....	17
6. Prikazivanje dugmadi.....	19
6.1 Zadavanje boje dugmeta.....	19
6.2 Zadavanje veličine dugmeta.....	19
6.3 Aktivna i neaktivna dugmad.....	20
6.4 Zadavanje funkcionalnosti dugmeta.....	21
6.5 Grupisanje dugmadi.....	23
6.6 Veličina grupisanih dugmadi.....	23
7. Omotači tekstualnog sadržaja.....	24
7.1 Klasa <i>jumbotron</i> .....	24
7.2 Klasa <i>well</i> .....	24
7.3 Klasa <i>panel</i> .....	25
8. Klase za prikazivanje grafičkih i numeričkih oznaka.....	26
8.1 Klasa <i>glyphicon</i> .....	26
8.2 Klasa <i>badge</i> .....	26
8.3 Klasa <i>label</i> .....	27
9. Klase za obeležavanje stranica.....	28
9.1 Klasa <i>pagination</i> .....	28
9.2 Klasa <i>pager</i> .....	29
10. Elementi forme.....	30
10.1 Grupisanje elemenata forme sa labelom.....	30
10.2 Prikazivanje elemenata forme.....	31
10.3 Prikazivanje elementa input.....	31
10.4 Visina elemenata forme.....	33
10.5 Širina elemenata forme.....	33
11. Traka za navigaciju.....	35
11.1 Klasa <i>navbar-default</i> .....	35
11.2 Klasa <i>navbar-inverse</i> .....	36
11.3 Fiksiranje trake za navigaciju.....	36
12. Lista opcija.....	37
12.1 Komponenta <i>tab</i> .....	37
12.2 Komponenta <i>pill</i> .....	37

12.3	Lista opcija sa dinamičkim komponentama .....	38
13.	Polje alert.....	39
13.1	Upotreba atributa pri zatvaranju alert polja.....	39
13.2	Upotreba biblioteke jQuery pri zatvaranju alert polja.....	39
13.3	Događaji alert polja.....	40
13.4	Zadavanje boje alert polja.....	41
14.	Progress bar traka.....	42
15.	Grupisanje elemenata.....	43
15.1	Grupisanje unutar elementa <i>div</i> .....	43
15.2	Grupisanje unutar neuređene liste.....	43
15.3	Zadavanje boje grupisanim elementima.....	44
16.	Dodaci radnog okruženja Bootstrap.....	45
16.1	Dodatak dropdown.....	45
16.1.1	Upotreba atributa pri kreiranju padajućeg menija.....	45
16.1.2	Upotreba biblioteke jQuery pri kreiranju padajućeg menija.....	46
16.1.3	Metodi dodatka dropdown.....	46
16.1.4	Događaji dodatka dropdown.....	46
16.2	Dodatak collapse.....	47
16.2.1	Prikazivanje elementa <i>div</i> .....	47
16.2.2	Sakrivanje elementa <i>div</i> .....	48
16.2.3	Opcije dodatka collapse.....	49
16.2.4	Metodi dodatka collapse.....	49
16.2.5	Događaji dodatka collapse.....	50
16.3	Dodatak carousel .....	51
16.3.1	Indikator.....	51
16.3.2	Omotič slajdova.....	51
16.3.3	Kontrole za listanje slajdova.....	51
16.3.4	Upotreba atributa pri kreiranju komponente carousel.....	52
16.3.5	Upotreba biblioteke jQuery pri kreiranju komponente carousel.....	52
16.3.6	Opcije dodatka carousel.....	53
16.3.7	Metodi dodatka carousel.....	54
16.3.8	Događaji dodatka carousel.....	55
16.4	Dodatak modal .....	55
16.4.1	Kreiranje dijaloga prozora.....	55
16.4.2	Upotreba atributa pri kreiranju dijaloga prozora.....	56
16.4.3	Upotreba biblioteke jQuery pri kreiranju dijaloga prozora.....	56
16.4.4	Opcije dodatka modal .....	57
16.4.5	Metodi dodatka modal.....	57
16.4.6	Događaji dodatka modal.....	58
16.4.7	Veličina dijaloga prozora.....	59
16.5	Dodatak affix .....	59
16.5.1	Upotreba atributa za fiksiranje elemenata.....	59
16.5.2	Upotreba biblioteke jQuery za fiksiranje elemenata.....	60
16.5.3	Opcije dodatka affix.....	60
16.5.4	Fiksiranje trake za navigaciju.....	60
16.5.5	Događaji dodatka affix.....	61
16.6	Dodatak tooltip.....	62
16.6.1	Upotreba atributa pri kreiranju komponente tooltip.....	62
16.6.2	Upotreba biblioteke jQuery pri kreiranju komponente tooltip.....	63
16.6.3	Opcije dodatka tooltip .....	63

16.6.4 Metodi dodatka tooltip.....	65
16.6.5 Događaji dodatka tooltip.....	65
16.7 Dodatak popover.....	66
16.7.1 Upotreba atributa pri kreiranju komponente popover.....	66
16.7.2 Upotreba biblioteke jQuery pri kreiranju komponente popover.....	67
16.7.3 Opcije dodatka popover.....	67
16.7.4 Metodi dodatka popover.....	68
16.7.5 Događaji dodatka popover.....	69
Zaključak.....	71
Literatura.....	72

# Uvod

Pojava interneta i opšta dostupnost računara doveli su do velikog napretka u oblasti informacionih tehnologija. Zbog opšteg razvoja interneta, savremene veb tehnologije su konstantno u ekspanziji. Elektronske platforme za učenje veb tehnologija su postali značajni i praktični izvori znanja. Jedna takva platforma, dostupna na srpskom jeziku je *eŠkola veba*. Elektronska platforma *eŠkola veba* je elektronska škola koji omogućava interaktivno učenje veb tehnologija. Platforma predstavlja projekat koji se razvija na Matematičkom fakultetu u Beogradu, koji je javno dostupan na serveru fakulteta na adresi [http://www.edusoft.matf.bg.ac.rs/eskola\\_veba/](http://www.edusoft.matf.bg.ac.rs/eskola_veba/). Platforma je organizovana prema kursevima. Svaki kurs se sastoji iz elektronskih lekcija koje obrađuju jednu veb tehnologiju ili radno okruženje. Trenutno su dostupni sledeći kursevi:

- *HTML* (eng. *Hyper Text Markup Language* - jezik namenjen za definisanje strukture veb stranica)
- *CSS* (eng. *Cascading Style Sheets* - jezik namenjen za opisivanje elemenata veb stranica)
- *JavaScript* (skriptni programski jezik namenjen za definisanje funkcionalnosti veb stranica koje se mogu izvršavati na klijentskoj i serverskoj strani)
- *jQuery* (biblioteka programskog jezika *JavaScript* dizajnirana da pojednostavi sintaksu tog jezika)
- *TypeScript* (strogi nadskup programskog jezika *JavaScript* dizajniran za razvoj velikih veb aplikacija)
- *AngularJS* (radni okvir programskog jezika *JavaScript*)
- i *Bootstrap*,

a u planu je uvođenje i drugih kurseva. Elektronske lekcije o radnom okruženju *Bootstrap* predstavljaju javno dostupan priručnik na srpskom jeziku koji obrađuje i sistematizuje osnovne koncepte radnog okruženja *Bootstrap*.

*Bootstrap* predstavlja besplatan skup alata za kreiranje veb sajtova i veb aplikacija, zasnovan na *HTML*, *CSS* i *JavaScript* strukturi. Razvili su ga Mark Oto (eng. *Mark Otto*) i Džejkob Tornton (eng. *Jacob Thornton*) u kompaniji *Twitter* i objavili ga kao proizvod otvorenog koda 19. avgusta 2011. godine na veb servisu *GitHub*. Prva verzija nastaje iz potrebe da se objedini funkcionalnost biblioteka koje su se koristile za razvoj interfejsa u okviru same kompanije. Druga verzija (*Bootstrap 2.0*) je objavljena 31. januara 2012. godine i ona uvodi prilagodljivi dizajn i podršku novih komponenata. Osnovna namena ove verzije je kreiranje stranica koje mogu da prilagođavaju svoj izgled rezoluciji uređaja na kom se prikazuju. Prvo izdanje verzije *Bootstrap 3.0* je objavljeno 19. avgusta 2013. godine kada je uveden princip dizajna „mobilni uređaji pre svega” (eng. *mobile-first*). Ova verzija naglašava prilagodljivi veb dizajn kao podrazumevanu opciju. Poslednje stabilno izdanje izlazi u verziji 3.3.7. Prvo izdanje verzije *Bootstrap 4.0* je objavljeno 19. avgusta 2015. i uvodi podršku za *Sass*<sup>1</sup>. U vreme pisanja ovog rada, dostupna su alpha i beta izdanja verzije *Bootstrap 4.0* ali se ona još

---

<sup>1</sup> *Sass* (eng. *Syntactically Awesome Stylesheets*) je skriptni jezik koji se interpretira u *CSS*. Svaka validna *CSS* datoteka je i validna *Sass* datoteka.

nalazi u pripremi. Ovaj rad prati elektronske lekcije o radnom okruženju Bootstrap verzije 3.0, kao njegove poslednje stabilne verzije.

Bootstrap se nalazi u klasi najpopularnijih radnih okruženja za razvoj veb stranica pre svega jer pruža mogućnost uštede vremena u njihovom kreiranju. Uvodi mnoštvo gotovih dizajnerskih rešenja i komponenata. Može se reći da sadrži uglavnom sve osnovne komponente koje su veb sajtovima potrebne, zbog čega ima značajnu ulogu u njihovom kreiranju. Stekao je veliku popularnost kada je u niz svojih funkcionalnosti uveo prilagodljivi veb dizajn. Prilagodljivi veb dizajn omogućava da se sadržaj veb stranica prilagođava uređajima različitih rezolucija (savremenim telefonima, tabletima i desktop računarima) čija upotreba postaje sve veća. Pravljenje posebnih aplikacija za ove uređaje bio bi dug i neisplativ proces, pa je Bootstrap sa prilagodljivim veb dizajnom postao popularno rešenje ovog problema. Pre pojave radnog okruženja Bootstrap, za implementaciju prilagodljivog veb dizajna se u početku koristio samo *JavaScript*. *JavaScript* je korišćen da menja *CSS* atribute pri promeni veličine prozora i pri samom paljenju veb aplikacije. Dolaskom *CSS3*<sup>2</sup>, u tu svrhu se još koristio i modul *media*<sup>3</sup>*queries*.

Kao i svako radno okruženje, Bootstrap ima svojih nedostataka ali i prednosti. Glavni nedostatak radnog okruženja Bootstrap je to što ne daje originalan izgled veb aplikacijama. Često se dešava da veb aplikacije koje ga koriste vizuelno podsećaju jedna na drugu pa su zbog toga neophodne izmene u konfiguraciji njegovih komponenata. Učitavanje veb stranica koje koriste Bootstrap je nešto sporije jer se pri svakom učitavanju učitava i Bootstrap. Korišćenje radnog okruženja Bootstrap može da onemogući rad pojedinih funkcija biblioteke *jQuery*, tako da njegova integracija sa kompleksnim veb aplikacijama nije jednostavna. Sa druge strane, prednosti radnog okruženja Bootstrap su brojne. Osim što je projekat otvorenog koda, relativno je jednostavan za upotrebu. Predstavlja kolekciju razvijenih *JavaScript* i *CSS* biblioteka i alata koje olakšavaju i ubrzavaju razvoj veb sajtova i veb aplikacija. Omogućava jednostavniji razvoj veb formi i interfejsa namenjenih za prikazivanje na mobilnim uređajima. Kompatibilan je sa svim modernim pregledačima i prikazuje sadržaj identično na svima. Radno okruženje Bootstrap podržava veliki broj stilova i funkcionalnosti koji se mogu primeniti na različitim *HTML* elementima i komponentama samog okruženja. Stilovi radnog okruženja Bootstrap su predstavljeni preko pojma klase. Klase predstavljaju skup vrednosti *CSS* svojstava koje se mogu primeniti na različitim *HTML* elementima i komponentama zadajući im isti stil. Sa druge strane strane, funkcionalnosti radnog okruženja Bootstrap su predstavljene pojmovima opcija, metoda i događaja koje se vezuju za određenu komponentu okruženja. Opcijama se zadaju specifična svojstva komponente. Metod komponente predstavlja funkciju koja se može nad njom primeniti dok događaji komponente definišu akcije koje se realizuju u tačno određenim uslovima. U nastavku ovog rada će biti prikazani osnovni stilovi i funkcionalnosti radnog okruženja Bootstrap.

---

<sup>2</sup> *CSS3* predstavlja poslednji standard jezika *CSS* koji je kompatibilan sa perthodnim verzijama tog jezika.

<sup>3</sup> *Media queries* je *CSS3* modul koji omogućava definisanje sadržaja koji se prilagođava uslovima kao što je rezolucija uređaja. Postao je preporučeni standard u junu 2012. od strane međunarodne organizacije zadužene za izradu standarda interneta (*World Wide Web Consortium, W3C*).

# 1. Korišćenje radnog okruženja Bootstrap

Bootstrap se može koristiti na dva načina: upotrebom instalacionog paketa ili povezivanjem na mrežu za dostavu sadržaja – *CDN*.<sup>4</sup> U prvom slučaju, instalacioni paket se može preuzeti sa adrese [getbootstrap.com](http://getbootstrap.com) i predstavlja iskompajliranu verziju radnog okruženja Bootstrap. Paket se sastoji iz fontova i *CSS* i *JavaScript* fajlova. Uvođenje radnog okruženja Bootstrap u veb stranicu ovim putem se realizuje tako što se u zaglavlju *HTML* dokumenta navodi link koji pokazuje na *bootstrap.min.css* fajl iz tog paketa. Više informacija o instalaciji radnog okruženja Bootstrap se može videti u [1].

Drugi način da se uvede u veb stranicu je kreiranjem veze sa *CDN*-om. To se realizuje tako što se u zaglavlju *HTML* dokumenta navode linkovi pomoću kojih se pristupa udaljenim fajlovima. Linkovi koji se u ovom slučaju navode pokazuju na:

- poslednju iskompajliranu verziju jezika *CSS*,
- iskompajliranu verziju biblioteke *jQuery*,
- i poslednju iskompajliranu verziju jezika *JavaScript*.

Kôd 1.1 prikazuje linkove ka navedenim fajlovima.

```
<!--Poslednji iskompajlirani CSS-->
<link rel="stylesheet"
href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

<!--jQuery biblioteka-->
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"> </script>

<!--Poslednji iskompajlirani JavaScript-->
<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"> </script>
```

*Kôd 1.1 - Uvođenje radnog okruženja Bootstrap u okviru HTML dokumenta*

Korišćenje radnog okruženja Bootstrap u okviru veb stranice zahteva njegovo uvođenje na jedan od pomenutih načina. Da bi bilo omogućeno njegovo pravilno funkcionisanje, na nivou veb stranice je neophodno navođenje elementa *meta* sa sledećim atributima: `<meta name="viewport" content="width=device-width, initial-scale=1">`.

Upotreba ovog elementa omogućava prilagođavanje širini ekrana. Dajući instrukcije pregledaču kako da kontroliše i skalira dimenziju veb stranice, ovaj element određuje njen prikaz. Svojstvo *width* određuje širinu tog prikaza i može biti zadato konkretnom vrednošću piksela (npr. *width=300*) ili specijalnom vrednošću kao što je *width=device-width*. Svojstvo *width=device-width* određuje da širina prikaza veb stranice prati širinu ekrana na kom se prikazuje. Svojstvo *initial-scale=1* određuje da je uveličanje prikaza veb stranice postavljeno na inicijalnu vrednost prilikom njenog prvog učitavanja.

---

<sup>4</sup> *CDN* (content delivery network ili content distribution network) je distributivna mreža servera čiji je cilj da krajnjim korisnicima dostavi sadržaj uz visoke performanse i maksimalnu dostupnost.

Kôd 1.2 prikazuje kreiranje prve veb stranice korišćenjem radnog okruženja Bootstrap.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap</title>
    <link rel="stylesheet"
      href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.4/jquery.min.js"> </script>
    <script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"> </script>
    <meta name="viewport" content="width=device-width,initial-scale=1">
  </head>
  <body> Zdravo svete! </body>
</html>
```

*Kôd 1.2 - Kreiranje prve veb stranice u radnom okruženju Bootstrap*

Korišćenje radnog okruženja Bootstrap u elektronskim lekcijama je realizovano kreiranjem veze sa *CDN*-om, tj. upotrebom linkova iz koda 1.1. U svim narednim primerima će se podrazumevati da su uslovi za kreiranje veb stranice u radnom okruženju Bootstrap ispunjeni, pa se taj deo koda neće ponavljati.



## 2. Klase omotača i mrežni sistem

Prikazivanje elemenata u radnom okruženju Bootstrap se najčešće sprovodi u okviru omotača, koji doprinose boljem prikazu elementa, dajući mu veću širinu. Omotač se predstavlja elementom *div* klase *container* ili *container-fluid*. Klasa *container* obezbeđuje omotač fiksne širine dok *container-fluid* obezbeđuje omotač u širini ekrana. Unutar ovih omotača je moguće smestiti tekst kao i druge različite elemente ali treba imati u vidu da se jedan omotač ne može smestiti unutar nekog drugog omotača.

Mrežni sistem (ili mreža) se sastoji iz 12 kolona i proizvoljnog broja redova smeštenih unutar omotača. Na taj način, redovi i kolone grade mrežu sastavljenu iz ćelija, između kojih se kreiraju praznine. Ćelije istog reda se mogu grupisati u jednu, koja se prostire kroz više kolona. Ukoliko se ćelije grupišu, treba imati u vidu da se ćelije jednog reda zajedno prostiru kroz 12 kolona. Slika 2.1 prikazuje primer jedne mreže.



Slika 2.1 – Mreža

Mreža u radnom okruženju Bootstrap je fleksibilna. To znači da se ista mreža može prikazati različito na uređajima različitih rezolucija. Više informacija o mreži radnog okruženja Bootstrap se može videti u [2]. Radi boljeg razumevanja narednih lekcija, biće izvršena klasifikacija uređaja prema rezoluciji, što je prikazano u tabeli 2.1.

Tabela 2.1 – Klasifikacija uređaja prema rezoluciji

Naziv:	Rezolucija:
Uređaji najmanje rezolucije	Manja od 768 px.
Uređaji male rezolucije	Veća ili jednaka 768 px a manja od 992 px.
Uređaji srednje rezolucije	Veća ili jednaka 992 px a manja od 1200 px.
Uređaji velike rezolucije	Veća ili jednaka 1200 px.

Korišćenje mreže zahteva korišćenje mrežnih klasa. One određuju ponašanje mreže na uređajima različitih rezolucija. Klasifikacija mrežnih klasa prikazana je u tabeli 2.2.

Tabela 2.2 – Klasifikacija mrežnih klasa

Klasa:	Opis:
<i>xs</i>	Određuje ponašanje mreže kod uređaja najmanje rezolucije.
<i>sm</i>	Određuje ponašanje mreže kod uređaja male rezolucije.
<i>md</i>	Određuje ponašanje mreže kod uređaja srednje rezolucije.
<i>lg</i>	Određuje ponašanje mreže kod uređaja velike rezolucije.

Kôd 2.1 prikazuje uopštenu strukturu jedne mreže.

```

<div class="container">
  <div class="row">
    <div class="col-mrežnaKlasa-broj1"></div>
  </div>
  <div class="row">
    <div class="col-mrežnaKlasa-broj2"></div>
    <div class="col-mrežnaKlasa-broj3"></div>
  </div>
  <div class="row">
    <div class="col-mrežnaKlasa-broj4"></div>
    <div class="col-mrežnaKlasa-broj5"></div>
    <div class="col-mrežnaKlasa-broj6"></div>
  </div>
  ...
</div>

```

Kôd 2.1 – Uopštena struktura mreže

Oznaka „mrežnaKlasa” predstavlja konkretnu mrežnu klasu (*xs*, *sm*, *md* ili *lg*) dok oznake brojeva predstavljaju broj kolona kroz koje se konkretna ćelija prostire.

## 2.1 Mrežna klasa *lg*

Klasa *lg* određuje ponašanje mreže kod uređaja čija je rezolucija veća ili jednaka 1200 px. Označava da će svi uređaji velike rezolucije prikazati željeni raspored ćelija zadat strukturom mreže. Kod uređaja manje rezolucije doći će do automatskog prelamanja redova pa će se ćelije slagati jedna ispod druge. Slika 2.2 prikazuje mrežu nastalu korišćenjem ove klase.

col-lg-6

col-lg-6

Slika 2.2 – Mreža klase *lg*

Struktura mreže sa slike 2.2 je prikazana u kodu 2.2.

```

<div class="container-fluid">
  <div class="row">
    <div class="col-lg-6" style="background-color:rgb(255,185,220);">col-lg-6</div>
    <div class="col-lg-6" style="background-color:rgb(193,255,255);">col-lg-6 </div>
  </div>
</div>

```

Kôd 2.2 – Upotreba klase *lg*

## 2.2 Mrežna klasa *md*

Klasa *md* određuje ponašanje mreže kod uređaja čija je rezolucija veća ili jednaka 992 px. Ona označava da će se željeni raspored ćelija zadat strukturom mreže prikazati jedino kod uređaja srednje i velike rezolucije. Kod uređaja sa rezolucijom manjom od 992 px doći će do automatskog prelamanja redova.

## 2.3 Mrežna klasa *sm*

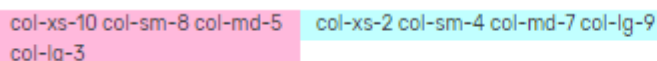
Klasa *sm* određuje ponašanje mreže kod uređaja čija je rezolucija veća ili jednaka 768 px. Označava da će uređaji male, srednje i velike rezolucije prikazati željeni raspored ćelija zadat strukturom mreže dok će kod ostalih doći do automatskog prelamanja redova.

## 2.4 Mrežna klasa `xs`

Klasa `xs` označava da će uređaji svih rezolucija prikazati željeni raspored ćelija zadat strukturom mreže. Upotrebom ove klase, obezbeđen je jedinstven prikaz mreže na svim uređajima, tj. neće doći do automatskog prelamanja redova ni na jednom uređaju.

## 2.5 Kombinacije mrežnih klasa

U strukturi jedne mreže, moguće je zadati više mrežnih klasa. Koja od njih će odrediti prikaz mreže, zavisi od rezolucije samog uređaja. Slika 2.3 prikazuje primer takve mreže.



`col-xs-10 col-sm-8 col-md-5 col-lg-3` `col-xs-2 col-sm-4 col-md-7 col-lg-9`

Slika 2.3 – Mreža klase `xs`, `sm`, `md` i `lg`

Struktura mreže sa slike 2.3 je prikazana u kodu 2.3.

```
<div class="container-fluid">
  <div class="row">
    <div class="col-xs-10 col-sm-8 col-md-5 col-lg-3" style="background-color:rgb(255,185,220);">
      col-xs-10 col-sm-8 col-md-5 col-lg-3
    </div>
    <div class="col-xs-2 col-sm-4 col-md-7 col-lg-9" style="background-color:rgb(193,255,255);">
      col-xs-2 col-sm-4 col-md-7 col-lg-9
    </div>
  </div>
</div>
```

Kod 2.3 - Upotreba klase `xs`, `sm`, `md` i `lg`

Ovaj primer određuje mrežu čiji je odnos širina ćelija jednak 3:9 na uređajima velike rezolucije, odnosno 5:7 na uređajima srednje rezolucije, odnosno 8:4 na uređajima male rezolucije, odnosno 10:2 na uređajima najmanje rezolucije. Na elektronskoj platformi *eŠkola veba* su dostupni primeri pomenutih klasa.

### 3. Tipografija

Bootstrap podržava skoro sve tipografske *HTML* elemente i stilizuje ih na poseban način. U nastavku ovog poglavlja za svaki od njih biće prikazan primer. Poređenja radi, isti primer *HTML* elementa biće prikazan u svom podrazumevanom obliku i u slučaju kada je stilizovan u radnom okruženju Bootstrap.

Elementi *h1*, *h2*, *h3*, *h4*, *h5* i *h6* se koriste za definisanje zaglavlja različitih nivoa. Slika 3.1 (a) prikazuje primer onih naslova koji su stilizovani u radnom okruženju Bootstrap, a slika 3.1(b) onih koji nisu. Naslovi se razlikuju u fontu.

Naslov nivoa 1

Naslov nivoa 2

Naslov nivoa 3

**Naslov nivoa 1**

**Naslov nivoa 2**

**Naslov nivoa 3**

Slika 3.1 (a) – Naslovi koji koriste Bootstrap stil

Slika 3.1 (b) – Naslovi koji ne koriste Bootstrap stil

Element *blockquote* se koristi za označavanje citata u tekstu. U okviru njega se koristi element *footer* u kom se navedi tekst koji se citira. Primer ovog elementa stilizovanog u radnom okruženju Bootstrap prikazan je na slici 3.2 (a), dok slika 3.2 (b) prikazuje isti element koji nije stilizovan. Pored razlike u fontu, može se primetiti razlika u tekstu navedenog unutar elementa *footer*. Bootstrap odvajava ovaj element zadajući mu drugu boju.

HTML (engl. HyperText Markup Language)  
— Preuzeto sa Wikipedije

HTML (engl. HyperText Markup Language)  
Preuzeto sa Wikipedije

Slika 3.2 (a) – Element *blockquote* koji koristi Bootstrap stil

Slika 3.2 (b) - Element *blockquote* koji ne koristi Bootstrap stil

Element *mark* se koristi za isticanje teksta. Razlika u njegovom prikazu kada koristi stil radnog okruženja Bootstrap (slika 3.3 (a)) i kada ga ne koristi (slika 3.3 (b)) je u boji pozadine tog elementa. Bootstrap stilizuje efekat isticanja u skladu sa modernim dizajnom.

Običan tekst.

Istaknuti tekst.

Običan tekst.

Istaknuti tekst.

Slika 3.3 (a) – Element *mark* koji koristi Bootstrap stil

Slika 3.3 (b) - Element *mark* koji ne koristi Bootstrap stil

Element *dl* se upotrebljava za označavanje definicione liste. Primer upotrebe definicione liste koja koristi stil radnog okruženja Bootstrap je prikazan na slici 3.4 (a). Pored razlike u fontu, primećuje se da Bootstrap naglašava pojmove koji se definišu. Isti primer koji ne koristi Bootstrap stil je prikazan na slici 3.4 (b).

**HTML**  
HyperText Markup Language  
**CSS**  
Cascading Style Sheets

HTML  
HyperText Markup Language  
CSS  
Cascading Style Sheets

Slika 3.4 (a) – Element *dl* koji koristi Bootstrap stil

Slika 3.4 (b) – Element *dl* koji ne koristi Bootstrap stil

Element *abbr* se upotrebljava za označavanje skraćenica i akronima u tekstu. Podrazumevani izgled ovog elementa je prikazan u primeru sa slike 3.5 (b). Stil koji Bootstrap zadaje ovom elementu se ogleda u naglašavanju pomoću tačaka, što se vidi na a slici 3.5 (a).

HTML

HTML

Slika 3.5 (a) – Element *abbr* koji koristi Bootstrap stil

Slika 3.5 (b) – Element *abbr* koji ne koristi Bootstrap stil

Element *code* se upotrebljava za označavanje delova koda unutar nekog teksta. Slika 3.6 (a) prikazuje primer upotrebe tog elementa stilizovanog u radnom okruženju Bootstrap. U poređenju sa podrazumevanim izgledom ovog primera sa slike 3.6 (b), može se primetiti da Bootstrap naglašava element *code* crvenom bojom.

Element `span` je linijski  
element a `div` je blokovski u HTML-u.

Element `span` je linijski  
element a `div` je blokovski u HTML-u.

Slika 3.6 (a) – Element *code* koji koristi Bootstrap stil

Slika 3.6 (b) – Element *code* koji ne koristi Bootstrap stil

Element *kbd* se upotrebljava za prikazivanje unosa sa tastature. Ovaj element u radnom okruženju Bootstrap jasno naglašava unos sa tastature, zadajući mu crnu pozadinu kao na slici 3.7 (a). Podrazumevani izgled istog primera je prikazan na slici 3.7 (b).

Prečica `ctrl+s` je naglašena.

Prečica `ctrl+s` je naglašena.

Slika 3.7 (a) – Element *kbd* koji koristi Bootstrap stil

Slika 3.7(b) – Element *kbd* koji ne koristi Bootstrap stil

Element *pre* se upotrebljava za prikaz teksta u kome se čuvaju beline i prelomi redova. Primer njegove upotrebe korišćenjem Bootstrap stila je dat na slici 3.8 (a) dok je podrazumevani izgled istog tog primera prikazan na slici 3.8 (b).

Ovaj element  
se može koristiti za  
pisanje linija koda.

Ovaj element  
se može koristiti za  
pisanje linija koda.

Slika 3.8 (a) – Element *pre* koji koristi Bootstrap stil

Slika 3.8 (b) – Element *pre* koji ne koristi Bootstrap stil

U nastavku ovog rada biće prikazane tipografske klase kojima je moguće zadati boju teksta i njegove pozadine kao i tipografske klase za formatiranje teksta i lista.

### 3.1 Zadavanje boje teksta

U tabeli 3.1 su prikazane tipografske klase kojima se zadaje boja teksta u skladu sa značenjem konteksta u kom se pojavljuje. Za svaku klasu je prikazan primer.

Tabela 3.1 – Klase za zadavanje boje teksta

Klasa:	Primer:	Izgled primera:
<i>text-muted</i>	<code>&lt;p class="text-muted"&gt;Ovaj tekst je isključen.&lt;/p&gt;</code>	Ovaj tekst je isključen.
<i>text-primary</i>	<code>&lt;p class="text-primary"&gt;Ovaj tekst je bitan.&lt;/p&gt;</code>	Ovaj tekst je bitan.
<i>text-success</i>	<code>&lt;p class="text-success"&gt;Uspešna akcija!&lt;/p&gt;</code>	Uspešna akcija!
<i>text-info</i>	<code>&lt;p class="text-info"&gt;Informacija.&lt;/p&gt;</code>	Informacija.
<i>text-warning</i>	<code>&lt;p class="text-warning"&gt;Upozorenje!&lt;/p&gt;</code>	Upozorenje!
<i>text-danger</i>	<code>&lt;p class="text-danger"&gt;Opasnost!&lt;/p&gt;</code>	Opasnost!

### 3.2 Zadavanje boje pozadini teksta

U tabeli 3.2 su prikazane tipografske klase kojima se zadaje boja pozadini teksta u skladu sa značenjem konteksta.

Tabela 3.2 – Klase za zadavanje boje pozadini teksta

Klasa:	Primer:	Izgled primera:
<i>bg-primary</i>	<code>&lt;p class="bg-primary"&gt;Ovaj tekst je bitan.&lt;/p&gt;</code>	Ovaj tekst je bitan.
<i>bg-success</i>	<code>&lt;p class="bg-success"&gt;Uspešna akcija!&lt;/p&gt;</code>	Uspešna akcija!
<i>bg-info</i>	<code>&lt;p class="bg-info"&gt;Informacija.&lt;/p&gt;</code>	Informacija.
<i>bg-warning</i>	<code>&lt;p class="bg-warning"&gt;Upozorenje!&lt;/p&gt;&lt;/div&gt;</code>	Upozorenje!
<i>bg-danger</i>	<code>&lt;p class="bg-danger"&gt;Opasnost!&lt;/p&gt;</code>	Opasnost!

### 3.3 Tipografske klase za formatiranje teksta i lista

Tipografske klase za formatiranje teksta su prikazane u tabeli 3.3.

Tabela 3.3 – Tipografske klase za formatiranje teksta

<b>Klasa:</b>	<b>Opis:</b>
<i>Lead</i>	Čini tekst istaknutim.
<i>Small</i>	Čini tekst umanjenim.
<i>text-left</i>	Pomera tekst ulevo.
<i>text-center</i>	Centrira tekst.
<i>text-right</i>	Pomera tekst udesno.
<i>text-justify</i>	Poravnjava tekst obostrano.
<i>text-nowrap</i>	Ne dozvoljava prelom reči na kraju reda.
<i>text-lowercase</i>	Ispisuje tekst malim slovima.
<i>text-uppercase</i>	Ispisuje tekst velikim slovima.
<i>text-capitalize</i>	Svaku reč iz teksta započinje velikim slovom.

Tipografske klase za formatiranje lista su prikazane u tabeli 3.4.

Tabela 3.4 – Tipografske klase za formatiranje lista

<b>Klasa:</b>	<b>Opis:</b>
<i>list-unstyled</i>	Uklanja grafičke oznake stavki liste.
<i>list-inline</i>	Postavlja stavke jedne liste u liniju.
<i>dl-horizontal</i>	Postavlja u jednu liniju stavke iz definicione liste kada se prozor pregledača raširi.

**Napomena:** klase *list-unstyled* i *list-inline* se primenjuju kod uređenih i neuređenih lista, a *dl-horizontal* samo kod definicionih. Kompletna lista svih tipografskih klasa i elemenata može se videti na adresi [http://www.w3schools.com/bootstrap/bootstrap\\_ref\\_css\\_text.asp](http://www.w3schools.com/bootstrap/bootstrap_ref_css_text.asp). Na elektronskoj platformi *eŠkola veba* su dostupni primeri za svaku od tipografskih klasa.

## 4. Prikazivanje tabela

Podrazumevani izgled tabela u *HTML*-u se može promeniti upotrebom klasa koje Bootstrap obezbeđuje za stilizovanje tabela. Dostupne su klase: *table*, *table-striped*, *table-bordered*, *table-hover*, *table-condensed* i *table-responsive*. Svaka od ovih klasa se navodi na nivou elementa *table*, osim poslednje, što će biti razmotreno u nastavku.

Upotrebom klase *table*, tabelle dobijaju svetlu pozadinu i okvire između redova. Primer takve tabelle dat je na slici 4.1.

Kontinent	Država	Glavni grad
Evropa	Austrija	Beč
Azija	Kina	Peking
Afrika	Egipat	Kairo

Slika 4.1 – Tabela klase *table*

Struktura tabelle sa slike 4.1 je prikazana u kodu 4.1.

```
<div class="container-fluid">
  <table class="table">
    <thead>
      <tr> <th>Kontinent</th> <th>Država</th> <th>Glavni grad</th> </tr>
    </thead>
    <tbody>
      <tr> <td>Evropa</td> <td>Austrija</td> <td>Beč</td> </tr>
      <tr> <td>Azija</td> <td>Kina</td> <td>Peking</td> </tr>
      <tr> <td>Afrika</td> <td>Egipat</td> <td>Kairo</td> </tr>
    </tbody>
  </table>
</div>
```

Kôd 4.1 – Upotreba klase *table*

Upotrebom klase *table-striped*, tabelle dobijaju prugaste redove pri čemu tabela nema nikakvih okvira. Tabela će dobiti okvire redova ako se istovremeno navedu klase *table* i *table-striped* na nivou elementa *table* i prikaz tabelle će biti širi (slika 4.2).

Kontinent	Država	Glavni grad
Evropa	Austrija	Beč
Azija	Kina	Peking
Afrika	Egipat	Kairo

Slika 4.2 – Tabela klase *table-striped* i *table*

Klasa *table-bordered* zadaje okvire između redova i kolona tabelle. Kao i u prethodnom slučaju, za širi prikaz tabelle, treba navesti klase *table* i *table-bordered* istovremeno.

Upotrebom klase *table-hover*, pozadina konkretnog reda tabelle postaje tamnija prelaskom kursora preko njega i tabela se prikazuje bez okvira. Tabela će dobiti okvire redova ako se sa klasom *table-hover* istovremeno navede i klasa *table*.

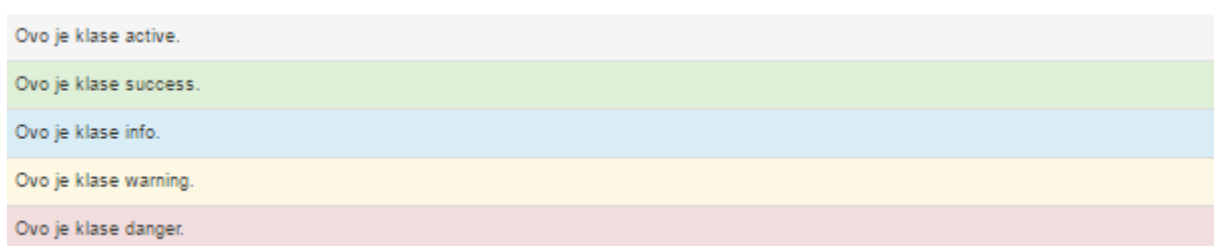


Klasa *table-condensed* sužava prikaz tabele. Tačnije, ćelijama tabele se smanjuje vrednost atributa *padding* a okviri tabele se ne prikazuju. Kao i u prethodnom slučaju, tabela dobija okvire redova ako se pored klase *table-condensed* navede klasa *table*.

Klasa *table-responsive* se navodi na nivou elementa *div* koji okružuje tabelu. Upotrebom ove klase postiže se prilagodljivost tabele rezoluciji uređaja. Ako je rezolucija uređaja manja od 768 px, tabela će se skrolovati horizontalno, a u slučaju da je veća od 768px, ostaje nepromenjena.

## 4.1 Zadavanje boje redovima tabele

U tabeli je moguće zadati boju ćelija i redova primenom klasa: *active*, *success*, *info*, *warning* i *danger*. Slika 4.3 ilustruje upotrebu ovih klasa na nivou redova. Analogno se mogu upotrebiti nad ćelijama.



Ovo je klase active.
Ovo je klase success.
Ovo je klase info.
Ovo je klase warning.
Ovo je klase danger.

Slika 4.3 – Tabela sa redovima klase *active*, *success*, *info*, *warning* i *danger*

Na elektronskoj platformi *eškola veba* su dostupni primeri tabela za svaku od pomenutih klasa. Više informaciji o stilovima tabela u radnom okruženju Bootstrap se može videti u [4].

## 5. Prikazivanje slika

Osnovni kod za ubacivanje slika je isti kao u *HTML*-u. Ono što Bootstrap uvodi su klase pomoću kojih možemo kreirati slike ovalnog<sup>5</sup> oblika, pravougaonog oblika i slike sa zaobljenim uglovima. Tabela 5.1 daje pregled ovih klasa sa primerima.

Tabela 5.1 – Klase za zadavanje oblika slike

Klasa:	Primer:	Izgled primera:
<i>img-circle</i>	<pre>&lt;img src="courses/bs/images/sl1.jpg" class="img-circle" alt="Slika1" width="304" height="236"&gt;</pre>	
<i>img-thumbnail</i>	<pre>&lt;img src="courses/bs/images/sl2.jpg" class="img-thumbnail" alt="Slika2" width="304" height="236"&gt;</pre>	
<i>img-rounded</i>	<pre>&lt;img src="courses/bs/images/sl3.jpg" class="img-rounded" alt="Slika3" width="304" height="236"&gt;</pre>	

### 5.1 Prilagođavanje slika rezoluciji uređaja

Upotrebom klase *img-responsive* na nivou elementa *img*, slika se automatski prilagođava rezoluciji uređaja na kom se prikazuje. Klasa *img-responsive* primenjuje CSS stil koji je prikazan u kodu 5.1.

```
display:block; max-width:100%; height:auto;
```

Kod 5.1 – CSS stil klase *img-responsive*

### 5.2 Prilagođavanje video zapisa rezoluciji uređaja

Prilagođavanje video zapisa rezoluciji uređaja na kom se prikazuje se postiže upotrebom klase *embed-responsive-item*. Ovu klasu je moguće primeniti na elementima: *iframe*, *embed*, *video* i *object*. Za svaki od ovih elemenata potrebno je uvesti okružujući element *div* klase *embed-responsive* koji definiše odnos prikaza. Odnos prikaza neke slike opisuje proporcionalni odnos između njene širine i visine. Za prikaz video sadržaja dostupni su odnosi: 4:3 (univerzalni video format 20.veka) i 16:9 (univerzalni video format za HD televiziju i evropsku digitalnu televiziju). Odnos 4:3 se primenjuje tako što se na nivou pomenutog elementa *div* pored klase *embed-responsive* navodi još i *embed-responsive-4by3*, a za odnos 16:9 klasa *embed-responsive-16by9*. Kôd 5.2 prikazuje umetanje video zapisa korišćenjem elementa *iframe*, sa odnosom prikaza 16:9.

<sup>5</sup> Internet Explorer 8 ne podržava slike ovalnog oblika.

```
<div class="embed-responsive  
embed-responsive-16by9">  
  <iframe class="embed-responsive-item"  
  src="https://www.youtube.com/embed/XGSy3_Czz8k">  
  </iframe>  
</div>
```

*Kôd 5.2 – Umetanje video zapisa*

Prikazani primeri su dostupni na elektronskoj platformi *eŠkola veba*. Više informacija o stilovima slika u radnom okruženju Bootstrap se može videti u [3].

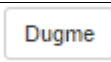


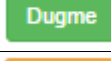


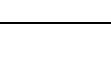
## 6. Prikazivanje dugmadi

Prikazivanje dugmadi u *HTML*-u se realizuje upotrebom elemenata *button* ili *input*. Za opisivanje dugmadi Bootstrap uvodi klasu *btn* koja se može primeniti, kako na elementima *button* i *input*, tako i na elementu za označavanje linkova, elementu *a*. Klasa *btn* zadaje osnovni stil dugmeta i ukoliko se primeni na nivou linka, link će se prikazati u formi dugmeta. Osim ove klase, Bootstrap obezbeđuje klase za zadavanje boje i veličine dugmeta. U nastavku ovog rada je prikazana njihova upotreba na primeru elementa *button*. Na elektronskoj platformi *eŠkola veba* je prikazana njihova upotreba na elementima *button*, *input* i *a*.

### 6.1 Zadavanje boje dugmeta

Zadavanje boje dugmeta se realizuje upotrebom klasa: *btn-default*, *btn-primary*, *btn-info*, *btn-success*, *btn-warning* i *btn-danger*. Pored pomenutih klasa, Bootstrap obezbeđuje klasu *btn-link* čijom primenom dugme dobija izgled linka. Svaka od ovih klasa se koristi istovremeno sa klasom *btn*, čija je upotreba obavezna (tabela 6.1).





Tabela 6.1 – Klase za zadavanje boje dugmeta

Klasa:	Primer:	Izgled primera:
<i>btn-default</i>	<pre>&lt;button type="button" class="btn btn-default"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-primary</i>	<pre>&lt;button type="button" class="btn btn-primary"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-info</i>	<pre>&lt;button type="button" class="btn btn-info"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-success</i>	<pre>&lt;button type="button" class="btn btn-success"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-warning</i>	<pre>&lt;button type="button" class="btn btn-warning"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-danger</i>	<pre>&lt;button type="button" class="btn btn-danger"&gt;Dugme&lt;/button&gt;</pre>	
<i>btn-link</i>	<pre>&lt;button type="button" class="btn btn-link"&gt;Dugme&lt;/button&gt;</pre>	

### 6.2 Zadavanje veličine dugmeta

Veličina dugmeta se zadaje upotrebom klasa: *btn-xs*, *btn-sm*, *btn-md* i *btn-lg*. Upotreba ovih klasa na nivou elementa *button* je prikazana u tabeli 6.2.

Tabela 6.2 – Klase za zadavanje veličine dugmeta

Klasa:	Primer:	Izgled primera:
<i>btn-xs</i>	<code>&lt;button type="button" class="btn btn-warning btn-xs"&gt;Dugme&lt;/button&gt;</code>	
<i>btn-sm</i>	<code>&lt;button type="button" class="btn btn-warning btn-sm"&gt;Dugme&lt;/button&gt;</code>	
<i>btn-md</i>	<code>&lt;button type="button" class="btn btn-warning btn-md"&gt;Dugme&lt;/button&gt;</code>	
<i>btn-lg</i>	<code>&lt;button type="button" class="btn btn-warning btn-lg"&gt;Dugme&lt;/button&gt;</code>	

Pored navedenih klasa za zadavanje veličine dugmeta, Bootstrap uvodi još i klasu *btn-block*. Upotrebom ove klase dobija se dugme nivoa bloka. Dugme nivoa bloka je dugme koje se prostire celom širinom svog okružujućeg elementa. Primer takvog dugmeta je dat na slici 6.1 koje se može dobiti pomoću koda 6.1.



Slika 6.1 – Dugme nivoa bloka

```
<div class="container">
  <button type="button" class="btn btn-warning btn-block">Dugme</button>
</div>
```

Kôd 6.1 – Upotreba klase *btn-block*

### 6.3 Aktivna i neaktivna dugmad

Neaktivno dugme se od aktivnog dugmeta razlikuje po karakterističnom bledom izgledu. Da bi dugme bilo aktivno potrebno je na nivou elementa *button* navesti klasu *active*, a da bi bilo neaktivno, klasu *disabled*. Slika 6.2 prikazuje aktivno i neaktivno dugme čiji prikaz se dobija pomoću koda 6.2.



Slika 6.2 – Aktivno i neaktivno dugme

```
<div class="container">
  <button type="button" class="btn btn-warning active">Aktivno dugme</button>
  <button type="button" class="btn btn-warning disabled">Neaktivno dugme</button>
</div>
```

Kôd 6.2 – Upotreba klasa *active* i *disabled*

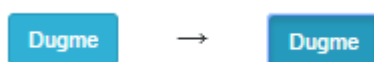
## 6.4 Zadavanje funkcionalnosti dugmeta

Funkcionalnost radnog okruženja Bootstrap u radu sa dugmadima se zasniva na upotrebi biblioteke *jQuery* ili atributa na nivou tog dugmeta.

*jQuery* je biblioteka programskog jezika *JavaScript* dizajnirana da olakša sintaksu tog jezika uvodeći metode za selektovanje, pretragu i manipulaciju *DOM*<sup>6</sup> elemenata. Osim toga, ona još omogućava izmenu njihovih svojstava kao i pridruživanje događaja selektovanom elementu.

Pomoću biblioteke *jQuery*, Bootstrap uvodi opcije, metode i događaje za svoje komponente i određene *HTML* elemente. Njegova funkcionalnost u radu sa dugmadima se ogleda kroz upotrebu posebno određenih metoda, dok opcije i događaji nisu podržani. Metod dugmeta koji Bootstrap obezbeđuje predstavlja funkciju koja se nad njim poziva nakon što se dugme pritisne i može se primeniti isključivo na nivou tog dugmeta. Podržana su 4 metoda: *button('toggle')*, *button('loading')*, *button('reset')* i *button('string')*.

Metod *button('toggle')* čini da dugme izgleda pritisnuto. Isti efekat se može postići bez upotrebe biblioteke *jQuery*, upotrebom atributa *data-toggle='button'* na nivou dugmeta. Slika 6.3 ilustruje upotrebu ovog metoda na dugmetu pre nego što je pritisnuto i nakon što je pritisnuto pomoću koda 6.3.

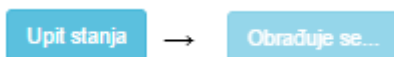


Slika 6.3 – Efekat upotrebe metoda *button('toggle')*

```
<button type="button" class="btn btn-info">Dugme</button>
<script>
$(document).ready(function(){
  $(".btn-info").click(function(){ $(this).button('toggle'); });
});
</script>
```

Kôd 6.3 – Upotreba metoda *button('toggle')*

Metod *button('loading')* onemogućava aktivnost dugmeta i postavlja mu novi tekst. Novi tekst se zadaje kao vrednost atributa *data-loading-text*. Ukoliko se atribut ne navede, podrazumevano se postavlja tekst „loading...”. Slika 6.4 ilustruje upotrebu ovog metoda pomoću koda 6.4, prikazujući dugme pre i nakon što se pritisne.



Slika 6.4 – Efekat upotrebe metoda *button('loading')*

<sup>6</sup> *DOM* (Document Object Model) predstavlja strukturnu prezentaciju *HTML* ili *XML* dokumenata čiji su elementi organizovani u stablo u kome svaki čvor predstavlja jedan element.

```

<button type="button" class="btn btn-info" data-loading-text="Obrađuje se...">Upit stanja
</button>
<script>
$(document).ready(function(){
  $(".btn-info").click(function(){ $(this).button('loading'); });
});
</script>

```

Kôd 6.4 – Upotreba metoda `button('loading')`

Metod `button('reset')` vraća originalni tekst na dugmetu, ukoliko je bio promenjen. Slika 6.5 prikazuje dva dugmeta u različitim trenucima: na početku, nakon što se pritisne levo dugme i nakon što se pritisne desno dugme koje vraća izgled levog dugmeta na početno stanje pomoću metoda `button('reset')`. Primer je nastao pomoću koda 6.5.



Slika 6.5 - Efekat upotrebe metoda `button('reset')`

```

<button type="button" class="btn btn-warning" data-loading-text="Obrađuje se...">Upit stanja
</button>
<button type="button" class="btn btn-info">Prekid</button>
<script>
$(document).ready(function(){
  $(".btn-warning").click(function(){ $(this).button('loading'); });
  $(".btn-info").click(function(){ $(' .btn-warning').button('reset'); });
});
</script>

```

Kôd 6.5 – Upotreba metoda `button('reset')`

Metod `button('string')` postavlja novi tekst na dugmetu. Novi tekst se zadaje kao vrednost atributa `data-string-text`. Slika 6.6 prikazuje dva dugmeta u različitim stanjima: na početku, nakon što se pritisne levo dugme i nakon što se pritisne desno koje postavlja novi tekst levom dugmetu pomoću metoda `button('string')`. Primer je dobijen pomoću koda 6.6.



Slika 6.6 – Efekat upotrebe metoda `button('string')`

```

<button type="button" class="btn btn-warning" data-loading-text="Obrađuje se..."
data-string-text="Obrađeno">Upit stanja</button>
<button type="button" class="btn btn-danger">Gotovo</button>
<script>
$(document).ready(function(){
  $(".btn-warning").click(function(){$(this).button('loading');});
  $(".btn-danger").click(function(){$(' .btn-warning').button('string');});
});
</script>

```

Kôd 6.6 – Upotreba metoda `button('string')`

## 6.5 Grupisanje dugmadi

Dugmad se može grupisati u okviru okružujućeg elementa linijski i vertikalno. U nastavku rada će biti prikazano grupisanje u okviru elementa *div*. Linijsko grupisanje se postiže upotrebom klase *btn-group*, a vertikalno upotrebom klase *btn-group-vertical* na nivou okružujućeg elementa. Slika 6.7 prikazuje linijsko a slika 6.8 vertikalno grupisanje tri dugmeta.



Slika 6.7 – Linijski grupisana dugmad



Slika 6.8 – Vertikalno grupisana dugmad

Primer linijski grupisanih dugmadi sa slike 6.7 se može dobiti pomoću koda 6.7. Kôd slike 6.8 bio bi sličan, tj. klasu *btn-group* bi trebalo zameniti klasom *btn-group-vertical*.

```
<div class="btn-group">
  <button type="button" class="btn btn-info">dugme 1</button>
  <button type="button" class="btn btn-info">dugme 2</button>
  <button type="button" class="btn btn-info">dugme 3</button>
</div>
```

Kôd 6.7 – Linijsko grupisanje dugmadi

## 6.6 Veličina grupisanih dugmadi

Umesto zadavanja veličine svakom pojedinačnom dugmetu, na nivou grupe je moguće zadati veličinu svima. To se postiže upotrebom klasa: *btn-group-lg*, *btn-group-md*, *btn-group-sm* i *btn-group-xs* kojima se zadaje redom dugmad najvećih, srednjih, malih i najmanjih dimenzija. Svaka od klasa se zadaje na nivou okružujućeg elementa *div*. Kod 6.8 prikazuje upotrebu klase *btn-group-lg*, kojom se grupiše velika dugmad. Analogno se upotrebljavaju i ostale klase koje grupišu dugmad manjih dimenzija.

```
<div class="btn-group btn-group-lg">
  <button type="button" class="btn btn-info">dugme 1</button>
  <button type="button" class="btn btn-info">dugme 2</button>
  <button type="button" class="btn btn-info">dugme 3</button>
</div>
```

Kôd 6.8 – Upotreba klase *btn-group-lg*

U okviru okružujućeg elementa *div*, dugme se može grupisati sa bilo kojim drugim elementima, ne obavezno sa dugmadima. Grupisanjem dugmeta sa neuređenom listom je moguće kreirati padajući meni koji se prikazuje klikom na dugme. Kreiranje padajućeg menija na nivou dugmeta zahteva upotrebu klasa iz dodatka dropdown, koji je detaljno objašnjen u sekciji 16.1. Na elektronskoj platformi *eŠkola veba* je prikazana upotreba svih pomenutih klasa kao i kreiranje padajućeg menija na nivou dugmeta.

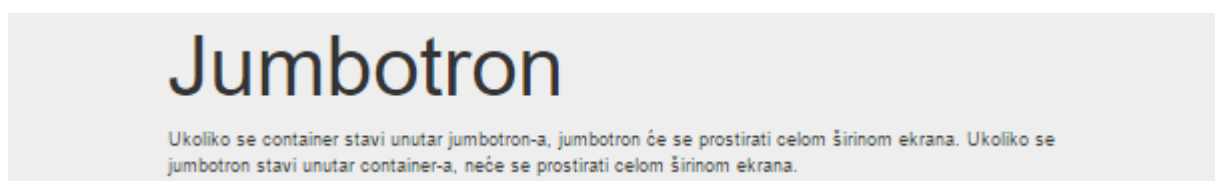


## 7. Omotači tekstualnog sadržaja

U radnom okruženju Bootstrap postoje klase koje se koriste za kreiranje omotača oko tekstualnog sadržaja. To su klase: *jumbotron*, *well* i *panel* i svaka od njih se upotrebljava nad elementom *div*. Za razliku od klasa omotača *container* i *container-fluid*, ove tri klase daju specifičan izgled pozadini teksta koji zahvataju.

### 7.1 Klasa *jumbotron*

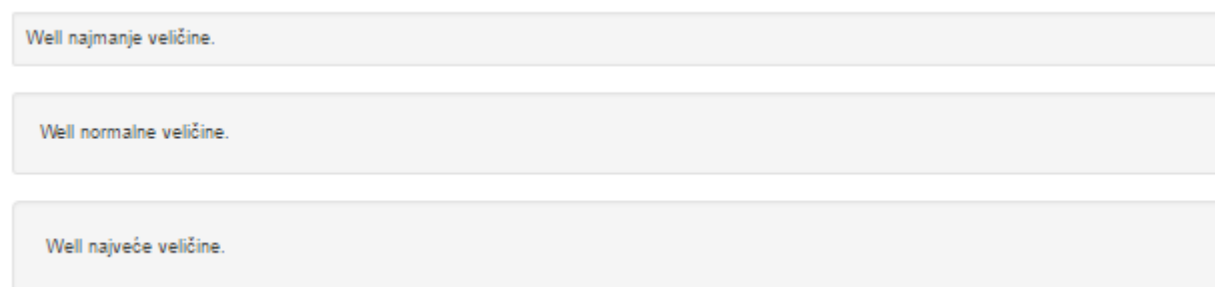
*Jumbotron* je klasa koja se dodeljuje elementu *div* kada je potrebno istaći sadržaj unutar njega. Element *div* klase *jumbotron* se prikazuje kao velika površina sive boje sa zaobljenim ivicama. Unutar njega je moguće staviti bilo koji validan *HTML* element i klasu iz radnog okruženja Bootstrap, a tekst koji se u njemu nalazi je uvećan u odnosu na spoljašnji (slika 7.1).



Slika 7.1 – Element *div* klase *jumbotron*

### 7.2 Klasa *well*

*Well* je klasa koja se dodeljuje elementu *div* kada je potrebno odvojiti neki tekstualni sadržaj. Upotrebom ove klase, element *div* dobija svetlo sivu pozadinu, zaobljene ivice i uvećanu vrednost atributa *padding*. Element *div* klase *well* se može prikazati u 3 različite veličine, što je određeno klasama: *well*, *well-sm* i *well-lg* (slika 7.2).



Slika 7.2 – Element *div* klase *well*, *well-sm* i *well-lg*

Primer sa slike 7.2 dobijen je pomoću koda 7.1.

```
<div class="container-fluid">
  <div class="well well-sm">Well najmanje veličine.</div>
  <div class="well">Well normalne veličine. </div>
  <div class="well well-lg">Well najveće veličine.</div>
</div>
```

Kôd 7.1 – Upotreba klase *well*, *well-sm* i *well-lg*

## 7.3 Klasa *panel*

*Panel* je klasa koja se dodeljuje elementu *div* kada je potrebno uokviriti sadržaj unutar njega. U daljem tekstu, element *div* klase *panel* nazivaćemo panel. Panel se prikazuje kao uokvireno polje bele boje i zaobljenih ivica. Unutar njega je moguće staviti bilo koji validan *HTML* element tekstualnog sadržaja. Podrazumevani izgled panela se realizuje upotrebom klase *panel-default*. Slika 7.3 prikazuje podrazumevani izgled panela dobijenog pomoću koda 7.2.



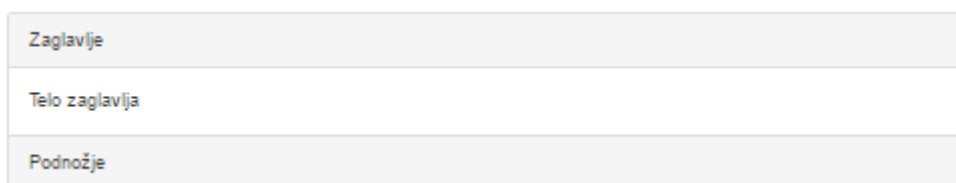
Slika 7.3 – Podrazumevani izgled panela

```
<div class="container">
  <div class="panel panel-default">
    <div>Ovo je div element.</div>
    <p>Ovo je pasus.</p>
    <span>Ovo je span element.</span>
  </div>
</div>
```

Kôd 7.2 - Upotreba klase *panel-default*

Veličina panela svakako zavisi od veličine njegovog sadržaja. Međutim, na veličinu panela se može uticati upotrebom klase *panel-body*. Ova klasa se navodi na nivou elementa koji se nalazi unutar panela. U slučaju prethodnog primera, to bi bili elementi *div*, *p* i *span*. Upotrebom klase *panel-body* uvećava se vrednost atributa *padding* tih elemenata.

Tekstualni sadržaj panela je moguće rasporediti u 3 celine: zaglavlje, podnožje i telo panela. Da bi se određeni tekst pojavio u zaglavlju, potrebno je staviti ga unutar elementa *div* klase *panel-heading*. Slično, tekst će biti u podnožju, odnosno telu panela, kada se stavi unutar elementa *div* klase *panel-footer*, odnosno klase *panel-body*. Slika 7.4 prikazuje panel u kome je izdvojeno zaglavlje, podnožje i telo panela.



Slika 7.4 – Panel sa zaglavljem, telom i podnožjem

Bootstrap omogućava grupisanje više panela u jedan. Da bi se više panela grupisalo u jedan, potrebno je staviti ih unutar elementa *div* klase *panel-group*, što je prikazano na elektronskoj platformi *eŠkola veba*.

## 8. Klase za prikazivanje grafičkih i numeričkih oznaka

U radnom okruženju Bootstrap postoje klase koje se koriste za kreiranje grafičkih i numeričkih oznaka koje se mogu pridružiti tekstu ili *HTML* elementima. To su klase: *glyphicon*, *badge* i *label*.

### 8.1 Klasa *glyphicon*

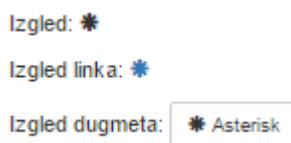
*Glyphicon* je klasa koja obezbeđuje prikazivanje sličica na dugmetu, linkovima, tekstu, formama i ostalim elementima. Bootstrap obezbeđuje 260 različitih sličica. Svaka sličica ima svoj naziv i svakoj odgovara određena klasa *glyphicon-\** u kojoj *\** predstavlja njen naziv. Prikazivanje konkretne sličice se realizuje upotrebom elementa *span* klase: *glyphicon* i *glyphicon-\**.

Kôd 8.1 predstavlja osnovni kôd za formiranje sličice.

```
<span class="glyphicon glyphicon-*"></span>
```

Kôd 8.1 – Struktura sličice

Slika 8.1 prikazuje upotrebu sličice *asterisk* nad linkom i dugmetom, dobijenu pomoću koda 8.2.



Slika 8.1 – Sličica asterisk na linku i dugmetu

```
<p>Izgled: <span class="glyphicon glyphicon-asterisk"> </span> </p>
<p>Izgled linka:
  <a href="#"> <span class="glyphicon glyphicon-asterisk"> </span> </a>
</p>
<p>Izgled dugmeta:
  <button type="button" class="btn btn-default btn-sm">
    <span class="glyphicon glyphicon-asterisk"> </span> Asterisk
  </button>
</p>
```

Kôd 8.2 – Upotreba klase *glyphicon* i *glyphicon-asterisk*

Na elektronskoj platformi *eŠkola veba* prikazano je 60 sličica sa svojom upotrebom nad linkom i dugmetom. Ostale se mogu videti na adresi [www.glyphicons.com](http://www.glyphicons.com).

### 8.2 Klasa *badge*

*Badge* je klasa koja obezbeđuje prikazivanje numeričkih oznaka nad linkovima ili dugmadima. Prikazivanje numeričke oznake nad linkom ili dugmetom se realizuje upotrebom elementa *span* klase *badge* unutar tog linka, odnosno dugmeta. Slika 8.2 prikazuje numeričku oznaku nad linkom i dugmetom i dobijena je pomoću koda 8.3.

Novosti **6**

Informacije **6**

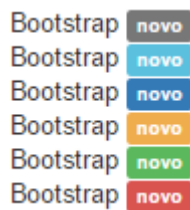
Slika 8.2 – Numeričke oznake nad linkom i dugmetom

```
<a href="#">
Novosti <span class="badge">6</span>
</a>
<button type="button" class="btn btn-info">
Informacije <span class="badge">6</span>
</button>
```

Kôd 8.3 – Upotreba klase badge nad linkom i dugmetom

### 8.3 Klasa *label*

*Label* je klasa koja obezbeđuje prikazivanje labela. Labele se koriste sa tekstovima i obezbeđuju dodatne informacije o njemu. Prikazivanje labele pored teksta se realizuje upotrebom elementa *span* klase *label* pored tog teksta. Klasa *label* se uvek koristi u kombinaciji sa nekom od klasa za zadavanje boje labela. Klase za zadavanje boje labela su: *label-default*, *label-primary*, *label-success*, *label-info*, *label-warning* i *label-danger*. Slika 8.3 prikazuje labele različitih boja dobijenih pomoću koda 8.4.



Slika 8.3 - Labele

```
Bootstrap <span class="label label-default">novo</span>
Bootstrap <span class="label label-info">novo</span>
Bootstrap <span class="label label-primary">novo</span>
Bootstrap <span class="label label-warning">novo</span>
Bootstrap <span class="label label-success">novo</span>
Bootstrap <span class="label label-danger">novo</span>
```

Kôd 8.4 – Upotreba klasa *label-default*, *label-info*, *label-primary*, *label-warning*, *label-success* i *label-danger*

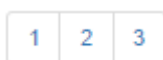
Na elektronskoj platformi *eŠkola veba* se mogu videti prikazani primeri.

## 9. Klase za obeležavanje stranica

Postoje dve klase za obeležavanje stranica u radnom okruženju Bootstrap: klasa `pagination` i klasa `pager`. Ove klase stilizuju izgled dugmadi koja se koriste za prelazak sa jedne stranice na drugu.

### 9.1 Klasa `pagination`

`Pagination` je klasa koja se koristi za obeležavanje stranica u vidu numerisanih dugmadi. Obeležavanje stranica se koristi kod velikih veb sajtova. Numerisana dugmad se obično koriste za prelazak sa jedne stranice na drugu. Prikazivanje dugmeta se realizuje upotrebom neuredjene liste `ul` klase `pagination` čije su stavke linkovi koji odvide na željenu stranicu. Slika 9.1 prikazuje numerisanu dugmad dobijenu upotrebom ove klase pomoću koda 9.1.



Slika 9.1 – Numerisana dugmad

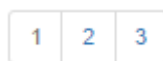
```
<ul class="pagination">
  <li><a href="#">1</a></li>
  <li><a href="#">2</a></li>
  <li><a href="#">3</a></li>
</ul>
```

Kôd 9.1 – Upotreba klase `pagination`

Klasa `pagination` se može koristiti za obeležavanje kako aktivnih tako i neaktivnih stranica. Obeležavanje aktivne stranice se ostvaruje sa dugmetom u plavoj boji a neaktivne, sa dugmetom koje je neaktivno. To se postiže upotrebom klase `active` nad elementom `li` koji označava aktivnu stranicu, odnosno upotrebom klase `disabled` nad elementom `li` koji označava neaktivnu stranicu. Dugme koje označava aktivnu stranicu je prikazano na slici 9.2, a dugme koje označava neaktivnu, na slici 9.3.



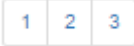

Slika 9.2 – Numerisana dugmad sa aktivnom stranicom



Slika 9.3 – Numerisana dugmad sa neaktivnom stranicom

Dugmad za obeležavanje se može prikazati u tri veličine: podrazumevanoj, maloj i velikoj. Tabela 9.1 prikazuje klase kojima se zadaje njihova veličina.

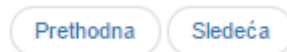
Tabela 9.1 – Klase za zadavanje veličine numerisanog dugmeta

Klasa:	Primer:	Izgled primera:
<i>pagination-sm</i>	<pre>&lt;ul class="pagination pagination-sm"&gt;   &lt;li&gt;&lt;a href="#"&gt;1&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;&lt;a href="#"&gt;2&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;&lt;a href="#"&gt;3&lt;/a&gt;&lt;/li&gt; &lt;/ul&gt;</pre>	
<i>pagination-lg</i>	<pre>&lt;ul class="pagination pagination-lg"&gt;   &lt;li&gt;&lt;a href="#"&gt;1&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;&lt;a href="#"&gt;2&lt;/a&gt;&lt;/li&gt;   &lt;li&gt;&lt;a href="#"&gt;3&lt;/a&gt;&lt;/li&gt; &lt;/ul&gt;</pre>	

Ukoliko se ne upotrebi nijedna klasa iz tabele 9.1, dugmad će se prikazati u podrazumevanoj veličini.

## 9.2 Klasa *pager*

*Pager* je klasa koja se koristi za obeležavanje isključivo prethodne i naredne stranice u vidu dugmadi. U odnosu na dugmad za obeležavanje iz prethodne lekcije, ova dugmad je drugačije stilizovana. Prikazivanje dugmadi se realizuje upotrebom neuređene liste *ul* klase *pager* sa dve stavke koje sadrže linkove na prethodnu i narednu stranicu. Slika 9.4 prikazuje dugmad dobijenu upotrebom klase *pager* pomoću koda 9.2.



Slika 9.4 – Obeležavanje prethodne i naredne stranice

```
<ul class="pager">
  <li><a href="#">Prethodna</a></li>
  <li><a href="#">Sledeća</a></li>
</ul>
```

Kôd 9.2 – Upotreba klase *pager*

Pored ovoga, Bootstrap obezbeđuje klase *previous* i *next*. Klase *previous* i *next* se koriste nad stavkama neuređene liste klase *pager*. Klasa *previous* pozicionira dugme na levu stranu a klasa *next* pozicionira na desnu (slika 9.5).



Slika 9.5 – Upotreba klase *previous* i *next*

Na elektronskoj platformi *eŠkola veba* su dostupni primeri za sve navedene klase.

## 10. Elementi forme

Forme (formulari) predstavljaju osnovu interaktivnosti veb stranica. One omogućavaju unos podataka koji se mogu iskoristiti na klijentskom računaru ili proslediti serveru. Kreiranje forme se postiže upotrebom elementa *form* u okviru kog je moguće koristiti elemente forme i druge *HTML* elemente. Elementi forme su *textarea*, *select* i *input*. Oni se mogu prikazati u širini celog ekrana upotrebom klase *form-control*.

### 10.1 Grupisanje elemenata forme sa labelom

Elementi forme se najčešće prikazuju sa svojom labelom sa kojom se grupišu. Njihovo grupisanje se sprovodi kada se stave unutar elementa *div* klase *form-group*. Na nivou elementa forme potrebno je zadati klasu *form-control* i identifikator, dok je na nivou elementa *label* potrebno navesti atribut *for* čija je vrednost jednaka pomenutom identifikatoru. Ovo važi za sve elemente forme osim za kontrole *checkbox* i *radio*, koje se izdvajaju kao posebni tipovi<sup>7</sup> elementa *input*. One se grupišu korišćenjem posebnih klasa.

Slika 10.1 prikazuje element *textarea* grupisan sa labelom pomoću koda 10.1.

Polje za unos:



Slika 10.1 – Element *textarea* sa labelom

```
<form>
<div class="form-group">
  <label for="txt">Polje za unos:</label>
  <textarea class="form-control" rows="2" id="txt"> </textarea>
</div>
</form>
```

Kôd 10.1 - Grupisanje elementa *textarea* sa labelom

Checkbox kontrola omogućava korisniku da štiklira više ponuđenih opcija sa liste. Grupisanje labele sa checkbox kontrolom se realizuje upotrebom klase *checkbox* pomoću koda 10.2.

```
<form>
<div class="checkbox"> <label><input type="checkbox" value="">Opcija1</label> </div>
</form>
```

Kôd 10.2 – Grupisanje *checkbox* kontrole sa labelom

Radio dugme omogućava korisniku da izabere jednu od više ponuđenih opcija. Grupisanje labele sa radio kontrolom se realizuje upotrebom klase *radio* pomoću koda 10.3.

---

<sup>7</sup> Osnovni atribut elementa *input* je *type* koji određuje njegov tip. *HTML5* obezbeđuje 16 tipova elementa *input*: *checkbox*, *radio*, *text*, *password*, *datetime*, *datetime-local*, *date*, *month*, *time*, *week*, *number*, *email*, *url*, *search*, *tel* i *color*.

```
<form>
<div class="radio"> <label><input type="radio">Opcija1</label> </div>
</form>
```

Kôd 10.3 – Grupisanje radio dugmeta sa labelom

Na elektronskoj platformi *eŠkola veba* dostupni su primeri za svaki od pomenutih elemenata forme, gde je prikazano njihovo grupisanje sa labelom.

## 10.2 Prikazivanje elemenata forme

Bootstrap obezbeđuje klase za vertikalno, horizontalno i linijsko prikazivanje forme. Vertikalno prikazivanje je podrazumevano prikazivanje forme i njime se elementi forme prikazuju jedan ispod drugog, svaki u posebnom redu (slika 10.2).

Email:

  
 Zapamti email  

Slika 10.2 – Vertikalno prikazivanje elemenata forme

Horizontalno prikazivanje elemenata forme podrazumeva da se grupisani elementi forme prikazuju u istom redu (slika 10.3). Realizuje se upotrebom klase *form-horizontal* na nivou elementa *form*.

Email:

Zapamti

Slika 10.3 – Horizontalno prikazivanje elemenata forme

Linijsko prikazivanje elemenata forme podrazumeva da se svi elementi forme prikazuju u istom redu i da su levo-poravnati (slika 10.4). Postiže se navođenjem klase *form-inline* na nivou elementa *form* i omogućeno je kod uređaja čija rezolucija nije manja od 768px.

Email:   Zapamti email

Slika 10.4 – Linijsko prikazivanje elemenata forme

Na elektronskoj platformi *eŠkola veba* dostupni su primeri za svaku od prikazanih formi.

## 10.3 Prikazivanje elementa *input*

Element *input* je moguće grupisati sa pomoćnim elementima: tekstom, sličicom ili dugmetom. To se realizuje kada se pomoćni element smesti u element *div* klase *input-group* zajedno sa elementom *input*. U slučaju da se element *input* grupiše sa tekstom ili



sličicom, potrebno je još navesti klasu `input-group-addon` na nivou teksta ili sličice. Slika 10.5 prikazuje element `input` grupisan sa sličicom, pomoću koda 10.4.



Slika 10.5 – Element `input` grupisan sa sličicom

```
<form>
<div class="input-group">
  <span class="input-group-addon"> <span class="glyphicon glyphicon-user"></span> </span>
  <input type="email" class="form-control" placeholder="Email">
</div>
</form>
```

Kôd 10.4 – Grupisanje elementa `input` sa sličicom

Grupisanje sa tekстом se sprovodi na analogan način, samo se unutar elementa `span` klase `input-group-addon` navodi željeni tekst umesto sličice. Slika 10.6 prikazuje grupisanje `input` elementa sa tekстом.



Slika 10.6 - Element `input` grupisan sa tekстом

U slučaju da se grupiše sa dugmetom, navodi se klasa `input-group-btn` na nivou elementa `div` koji okružuje dugme. Slika 10.7 prikazuje grupisanje sa dugmetom pomoću koda 10.5.



Slika 10.7 - Element `input` grupisan sa dugmetom

```
<form>
<div class="input-group">
  <input type="text" class="form-control" placeholder="Traži">
  <div class="input-group-btn">
    <button type="button" class="btn btn-info">
      <span class="glyphicon glyphicon-search"></span>
    </button>
  </div>
</div>
</form>
```

Kôd 10.5 – Grupisanje elementa `input` sa dugmetom

Element `input` se može pojaviti u 3 različita stanja: stanju fokusa, stanju disabled ili stanju readonly. Stanje fokusa je podrazumevano stanje elementa `input` i ogleda se u trenutku unosa kada ono dobija fokus. Preostala dva stanja se aktiviraju navođenjem ključnih reči `disabled`, odnosno `readonly`, na nivou elementa `input`. Stanje `disabled` onemogućava unos teksta, dok stanje `readonly` onemogućava unos i dozvoljava samo čitanje iz njega. Bootstrap obezbeđuje klase kojima se zadaje boja elementa `input`: `has-success`, `has-warning` i `has-error`. Klase se zadaju na nivou elementa `div` koji okružuje element `input`. Na elektronskoj platformi *eŠkola veba* su dostupni primeri za svaku od pomenutih klasa.

## 10.4 Visina elemenata forme

Visina elementa `textarea` se zadaje vrednošću atributa `rows` na nivou samog elementa. Visina elemenata `input` i `select` je podrazumevano zadata, ali se može menjati upotrebom klasa: `input-lg` i `input-sm` koje se zadaju na nivou samih elemenata. Upotrebom klase `input-lg` se dobija element najveće visine, a upotrebom klase `input-sm`, element najmanje visine. Slika 10.8 prikazuje elemente `input` klase `input-lg` i `input-sm` dobijene pomoću koda 10.6.

Polje za unos:

Polje za unos:

Slika 10.8 – Elementi `input` različitih visina

```
<form>
  <div class="form-group container">
    <label for="id1">Polje za unos:</label>
    <input class="form-control input-lg" id="id1" type="text">
    <label for="id2">Polje za unos:</label>
    <input class="form-control input-sm" id="id2" type="text">
  </div>
</form>
```

Kôd 10.6 – Upotreba klase `input-lg` i `input-sm` nad elementima `input`

Visina elemenata `input` i `select` se može zadati još i upotrebom klasa `form-group-lg` i `form-group-sm` koje se navode na nivou okružujućeg elementa `div`. Elementi sa slike 10.8 se mogu dobiti pomoću koda 10.7.

```
<form>
  <div class="form-group form-group-lg">
    <label for="id1">Polje za unos:</label><input type="text" class="form-control" id="id1">
  </div>
  <div class="form-group form-group-sm">
    <label for="id2">Polje za unos:</label> <input type="text" class="form-control" id="id2">
  </div>
</form>
```

Kôd 10.7 - Upotreba klase `form-group-lg` i `form-group-sm`

## 10.5 Širina elemenata forme

Širina elemenata forme se može smanjiti upotrebom mrežnih klasa i zadaje se kada se element smesti unutar elementa `div` željene mrežne klase, a potom taj element `div` smesti u `div` klase `form-group`. Mrežna klasa na nivou pomenutog elementa `div` elementa određuje broj kolona kroz koje se element prostire. Slika 10.9 prikazuje elemente forme različitih širina dobijenih pomoću koda 10.8.

Input element:

Select element:

Textarea element:

Slika 10.9 – Elementi forme različitih širina

```

<form>
  <div class="form-group">
    <div class="col-xs-2">
      <label for="id1">Input element:</label> <input class="form-control" type="text" id="id1">
    </div>
  </div>
  <div class="form-group">
    <div class="col-xs-4">
      <label for="id2">Select element:</label>
      <select class="form-control" id="id2"> <option>1</option> <option>2</option> </select>
    </div>
  </div>
  <div class="form-group">
    <div class="col-xs-5">
      <label for="id3">Textarea element:</label> <textarea class="form-control" rows="4" id="id3">
    </div>
  </div>
</form>

```

Kôd 10.8 – Upotreba mrežnih klasa kod zadavanja širine elemenata forme

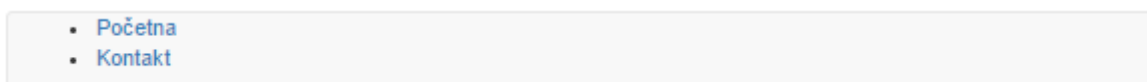
Na elektronskoj platformi *eŠkola veba* su dostupni primeri koji ilustruju upotrebu pomenutih klasa na svim elementima forme.

## 11. Traka za navigaciju

Kreiranje trake za navigaciju se realizuje upotrebom klase `navbar` nad elementom `nav`. Element `nav` je uveden u *HTML5*<sup>8</sup> i predstavlja sekciju neke stranice koja sadrži linkove ka drugim stranicama ili delovima u okviru iste. Traka za navigaciju se obično smešta na vrhu ili na dnu stranice. U zavisnosti od rezolucije uređaja, njeni linkovi se prikazuju ili u jednom redu ili jedan ispod drugog. Upotreba klase `navbar` se sprovodi u kombinaciji sa jednom od klasa `navbar-default` ili `navbar-inverse`.

### 11.1 Klasa `navbar-default`

Upotrebom klase `navbar-default`, traka za navigaciju dobija okvire i pozadinu svetlo-sive boje. Slika 11.1 prikazuje ovakvu traku, dobijenu pomoću koda 11.1.

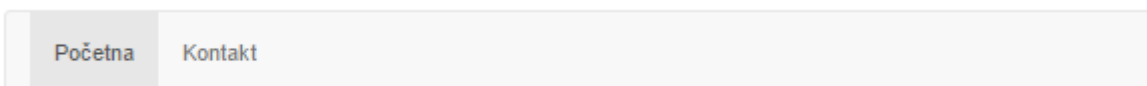


Slika 11.1 – Traka za navigaciju

```
<nav class="navbar navbar-default">
  <div class="container">
    <ul>
      <li class="active"><a href="#">Početna</a></li>
      <li><a href="#">Kontakt</a></li>
    </ul>
  </div>
</nav>
```

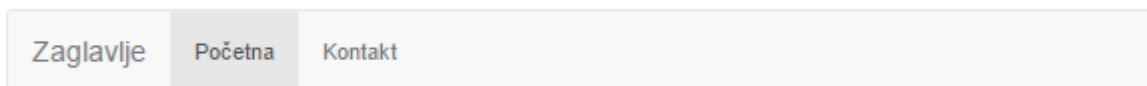
Kód 11.1 – Upotreba klase `navbar-default`

Prethodna traka za navigaciju dobija bolji izgled kada se na nivou elementa `ul` navedu klase `nav` i `navbar-nav`. Ove klase stilizuju prikaz linkova, što je prikazano na slici 11.2.



Slika 11.2 – Stilizovana traka za navigaciju

Traka za navigaciju može imati zaglavlje koje se predstavlja elementom `div` klase `navbar-header` u okviru elementa `nav`. Najčešće samo zaglavlje sadrži link koji se može stilizovati klasom `navbar-brand`. Ovakva traka se smatra standardnom trakom za navigaciju (slika 11.3) i nastaje pomoću koda 11.2.



Slika 11.3 – Standardna traka za navigaciju u svetlo-sivoj boji

<sup>8</sup> *HTML5* predstavlja poslednji standard jezika *HTML* koji je kompatibilan sa prethodnim verzijama tog jezika.

```

<nav class="navbar navbar-default">
  <div class="container">
    <div class="navbar-header"> <a class="navbar-brand" href="#">Zaglavlje</a> </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Početna</a></li>
      <li><a href="#">Kontakt</a></li>
    </ul>
  </div>
</nav>

```

Kôd 11.2 – Kreiranje standardne trake za navigaciju u svetlo-sivoj boji

## 11.2 Klasa *navbar-inverse*

Klasa *navbar-inverse* se koristi u kombinaciji sa klasom *navbar* na nivou elementa *nav*, i njenom upotrebom traka za navigaciju se prikazuje u crnoj boji. Standardna traka za navigaciju u crnoj boji nastaje pomoću koda 11.2 kada se klasa *navbar-default* zameni klasom *navbar-inverse*. Slika 11.4 prikazuje primer ovakve trake za navigaciju.



Slika 11.4 – Standardna traka za navigaciju u crnoj boji

## 11.3 Fiksiranje trake za navigaciju

Traka za navigaciju se može fiksirati na vrhu ili na dnu strane i ona ostaje vidljiva na fisnoj poziciji nezavisno od skrolovanja strane. Fiksiranje na vrhu strane se sprovodi kada se standardnoj traci doda klasa *navbar-fixed-top* na nivou elementa *nav*. Slično, fiksiranje na dnu strane se sprovodi upotrebom klase *navbar-fixed-bottom*.

Traka za navigaciju se različito prikazuje na uređajima različitih rezolucija. Kod uređaja male rezolucije, komponente će se posložiti jedna ispod druge, dok se kod uređaja veće rezolucije komponente prikazuju u istom redu. Na elektronskoj platformi *eŠkola veba* se mogu videti primeri upotrebe svih pomenutih klasa. Takođe je prikazano kako se može kreirati traka za navigaciju koja sadrži padajući meni.

## 12. Lista opcija

Lista opcija u *HTML*-u je predstavljena neuređenom listom (*ul*) čije su stavke linkovi koji odvođe na određenu adresu. Podrazumevano, stavke iz liste opcija se prikazuju jedna ispod druge. Korišćenjem radnog okruženja Bootstrap ovu listu opcija je moguće na jednostavan način prikazati u horizontalnom položaju, upotrebom klase *list-inline* na nivou elementa *ul*. Slika 12.1 prikazuje listu opcija dobijenu upotrebom ove klase.



Slika 12.1 – Lista opcija koja koristi klasu *list-inline*

Alternativni način za kreiranje horizontalne liste opcija je upotrebom klasa: *nav*, *nav-tabs* i *nav-pills* koje obezbeđuje radno okruženje Bootstrap. Ove klase stilizuju izgled linkova iz liste opcija, prikazujući ih kao posebne komponente. U zavisnosti od toga koje klase se koriste, moguće je kreirati listu opcija sa komponentama *tab*, odnosno listu opcija sa komponentama *pill*.

### 12.1 Komponenta tab

Komponenta *tab* je stavka liste opcija koja se prikazuju kao uokvirena kartica bele boje. Lista opcija sa ovim komponentama se formira upotrebom neuređene liste klase *nav* i *nav-tabs*. Slika 12.1 prikazuje takvu listu opcija, dobijenu pomoću koda 12.1.



Slika 12.1 – Lista opcija sa komponentama *tab*

```
<ul class="nav nav-tabs">
  <li class="active"><a href="#">HTML</a></li>
  <li><a href="#">CSS</a></li>
  <li><a href="#">JavaScript</a></li>
  <li><a href="#">jQuery</a></li>
</ul>
```

Kôd 12.1 – Upotreba klase *nav* i *nav-tabs*

U kodu 12.1 korišćena je klasa *active*, koja se navodi na nivou elementa *li* i predstavlja aktivnu komponentu iz liste opcija.

### 12.2 Komponenta pill

Komponenta *pill* je stavka liste opcija koja se prikazuju kao kartica plave boje. Lista opcija sa ovim komponentama se formira upotrebom neuređene liste klase *nav* i *nav-pills*. Slika 12.2 prikazuje takav meni, dobijen pomoću koda 12.2.



Slika 12.2 – Lista opcija sa komponentama *pill*

```

<ul class="nav nav-pills">
  <li class="active"><a href="#">HTML</a></li>
  <li><a href="#">CSS</a></li>
  <li><a href="#">JavaScript</a></li>
  <li><a href="#">jQuery</a></li>
</ul>

```

Kôd 12.2 – Upotreba klase *nav* i *nav-pills*

Komponente liste opcija se mogu prikazati i u vertikalnom položaju. To se postiže upotrebom klase *nav-stacked* na nivou elementa *ul*, pored već pomenutih klasa *nav* i *nav-tabs*, odnosno *nav* i *nav-pills*.

## 12.3 Lista opcija sa dinamičkim komponentama

Lista opcija sa dinamičkim komponentama je lista opcija koja obezbeđuje da se klikom na njene komponente prikazuje tekstualni sadržaj u okviru iste veb strane, i dostupan je isključivo kada se klikne na tu komponentu. Ona se formira iz dve celine. Prvu celinu čini sama lista opcija, predstavljena neuređenom listom. Na nivou svake stavke nalazi se link koji pokazuje na sadržaj u okviru iste strane. Obavezni atributi tog linka su: *datatoggle="tab"* i *href*. Vrednost atributa *href* je jednaka nazivu identifikatora onog elementa na koji link pokazuje, sa oznakom # ispred.

Drugu celinu čini element *div* klase *tab-content* u okviru kog se nalaze elementi *div* na koje linkovi pokazuju. Svaki od njih je klase *tab-pane* i svaki je označen identifikatorom. U praksi je uvek jedna komponenta iz liste opcija aktivna i prikazan je sadržaj koji je u vezi sa njom. To se postiže tako što se na nivou željene komponente navodi klasa *active* a nad odgovarajućim elementom *div* klase *active* i *in*. Slika 12.3 prikazuje listu opcija sa dinamičkim komponentama pill, dobijenu pomoću koda 12.3.



Slika 12.3 – Lista opcija sa dinamičkim komponentama pill

```

<ul class="nav nav-pills">
  <li class="active"><a data-toggle="pill" href="#html3">HTML</a></li>
  <li><a data-toggle="pill" href="#css3">CSS</a></li>
  <li><a data-toggle="pill" href="#js3">JS</a></li>
  <li><a data-toggle="pill" href="#jq3">jQ</a></li>
</ul>
<div class="tab-content">
  <div id="html3" class="tab-pane in active">Hyper Text Markup Language</div>
  <div id="css3" class="tab-pane">Cascading Style Sheets</div>
  <div id="js3" class="tab-pane">JavaScript</div>
  <div id="jq3" class="tab-pane">jQuery</div>
</div>

```

Kôd 12.3 – Kreiranje liste opcija sa dinamičkim komponentama pill

Na elektronskoj platformi *eŠkola veba* su prikazani primeri za svake navedene klase, primenjene na listama opcija sa komponentama tab i pill.

## 13. Polje alert

Poljem alert nazivamo element `div` klase `alert`. Alert polje se koristi kada je potrebno prikazati neki tekst u vidu poruke. Osnovni kod za ubacivanje alert polja je prikazan u kodu 13.1 i to alert polje nema mogućnost zatvaranja, već se prikazuje kao statična poruka.

```
<div class="alert"> Ovo je neka poruka. </div>
```

Kôd 13.1 – Struktura alert polja

Zatvaranje alert polja se postiže klikom na neko dugme ili klikom na link koji se navodi unutar alert polja. Neophodno je da element, čijim klikom se alert polje zatvara, bude klase `close`, koja mu zadaje funkciju. U narednim primerima za zatvaranje alert polja biće korišćeno dugme. Sama akcija zatvaranja alert polja sprovodi se upotrebom atributa ili pomoću biblioteke *jQuery*.

### 13.1 Upotreba atributa pri zatvaranju alert polja

Zatvaranje alert polja se može realizovati upotrebom atributa `data-dismiss="alert"` koji se navodi na nivou dugmeta čija je funkcija da zatvori alert polje. Slika 13.1 prikazuje takvo alert polje nastalo pomoću koda 13.2.

Ovo je neka poruka.



Slika 13.1 – Alert polje

```
<div class="alert">  
<button type="button" class="close" data-dismiss="alert">&times;</button> Ovo je neka poruka.  
</div>
```

Kôd 13.2 – Upotreba atributa `data-dismiss` kod alert polja

### 13.2 Upotreba biblioteke *jQuery* pri zatvaranju alert polja

Funkcionalnost radnog okruženja Bootstrap u radu sa alert poljem se ogleda u upotrebi biblioteke *jQuery*, pomoću koje Bootstrap uvodi metode i događaje za rad sa njim. Opcije alert polja nisu podržane. Metod alert polja podržan u radnom okruženju Bootstrap predstavlja funkciju koja se poziva nad tim alert poljem nakon što se pritisne dugme koje se nalazi u okviru njega. Podržan je jedan metod, metod `alert ("close")`. Uloga ovog metoda je zatvaranje alert polja.

Zatvaranje alert polja upotrebom biblioteke *jQuery* se postiže pozivanjem metoda `alert ("close")` nad alert poljem, u trenutku kada se pritisne dugme u okviru njega. Alert polje sa slike 13.1 dobija mogućnost zatvaranja pomoću biblioteke *jQuery* kada se primeni kôd 13.3.



```

<div class="alert" id="polje"> <button type="button" class="close">&times; </button>
Ovo je nekakva poruka.
</div>
<script>
$(document).ready(function(){
  $(".close").click(function(){
    $("#polje").alert("close");
  });
});
</script>

```

Kôd 13.3 – Upotreba biblioteke jQuery kod alert polja

## 13.4 Događaji alert polja

Drugi vid funkcionalnosti radnog okruženja Bootstrap u radu sa alert poljem se zasniva na događajima, uvedenih pomoću biblioteke jQuery. Događaj alert polja predstavlja definisanu akciju na veb stranici koja se realizuje u tačno određenim uslovima, koji su određeni nazivom tog događaja.

Zadavanje događaja alert polja se realizuje pozivom metoda *on* nad alert poljem, koji prihvata dva parametra. Prvi parametar je naziv događaja alert polja, a drugi parameter je funkcija koja će se realizovati na veb stranici. Naziv događaja, kao prvi parameter metoda *on*, određuje trenutak u kom se poziva funkcija navedena kao drugi parameter tog metoda.

Tabela 13.1 prikazuje upotrebu događaja alert polja pod pretpostavkom da je ono kreirano pomoću koda 13.3. Akcija koja se realizuje kroz pozive događaja u primerima iz tabele je pojavljivanje dijalog prozora.

Tabela 13.1 – Događaji alert polja

Događaj:	Opis:	Primer:
close.bs.alert	Događaj nastupa neposredno pre nego što će alert polje biti zatvoreno.	<pre> &lt;script&gt; \$(document).ready(function(){   \$(".close").click(function(){     \$("#polje").alert("close");   });   \$("#polje").on('close.bs.alert', function(){     alert('Alert polje će uskoro biti zatvoreno.');</pre>
closed.bs.alert	Događaj nastupa nakon što je alert polje zatvoreno.	<pre> &lt;script&gt; \$(document).ready(function(){   \$(".close").click(function(){     \$("#polje").alert("close");   });   \$("#polje").on('closed.bs.alert', function(){     alert('Alert polje je već zatvoreno.');</pre>

## 13.5 Zadavanje boje alert polja

Alert polje je moguće prikazati u 4 različite boje koje su u vezi sa značenjem njegovih poruka. Za to se koriste klase: *alert-success*, *alert-info*, *alert-warning* i *alert-danger*. Svaka od ovih klasa se koristi u kombinaciji sa klasom *alert*. Na slici 13.2 je prikazano alert polje klase *alert-success* koje se zatvara klikom na link pomoću koda 13.4. Analogno se upotrebljavaju klase *alert-info*, *alert-warning* i *alert-danger*.



Slika 13.2 – Alert polje klase *alert-success*

```
<div class="alert alert-success">
  <a href="#" class="close" data-dismiss="alert">&times;</a> Uspešna akcija!
</div>
```

Kôd 13.4 – Upotreba klase *alert-success*

U svim prethodnim primerima, klikom na dugme ×, alert polje se izgubi ne ostavljajući trag za sobom. Zatvaranje alert polja je moguće animirati tako da se uočava trag pri njegovom nestajanju i to se postiže korišćenjem klasa *fade* i *in* na nivou okružujućeg elementa *div*.

## 14. Progress bar traka

Progress bar je traka koja se koristi da prikaže stepen uspešnosti nekog procesa. Najčešće je korišćena sa procentualnim podacima koji opisuju taj proces. Prikazivanje trake progress bar<sup>9</sup> se realizuje upotrebom elementa `div` klase `progress` unutar kog se nalazi element `div` klase `progress-bar` sa atributima:

- `role="progressbar"` (označava ulogu elementa `div`)
- `aria-valuenow` (označava trenutnu vrednost obojene površine)
- `aria-valuemin` (označava minimalnu moguću vrednost obojene površine)
- `aria-valuemax` (označava maksimalnu moguću vrednost obojene površine)
- `style="width:70%"` (kreira obojenu traku u skladu sa zadatom širinom).

Slika 14.1 prikazuje primer jedne progress bar trake, koja se može dobiti pomoću koda 14.1.



Slika 14.1 – Traka progress bar

```
<div class="progress">
  <div class="progress-bar" role="progressbar" aria-valuenow="70" aria-valuemin="0"
    aria-valuemax="100" style="width:70%"> 70% Complete </div>
</div>
```

Kôd 14.1 – Upotreba klase `progress-bar`

U zavisnosti od konteksta u kom se pojavljuje, moguće je zadati željenu boju progress bar trake. Bootstrap obezbeđuje sledeće klase za zadavanje boje: `progress-bar-success`, `progress-bar-info`, `progress-bar-warning` i `progress-bar-danger`. Pored ovih klasa, dostupne su još i klase `progress-bar-striped` i `active`. Sve navedene klase se navode na nivou pomenutog elementa `div` klase `progress-bar`. Slika 14.2 prikazuje progress bar traku klase `progress-bar-striped`, koja joj daje prugasti izgled.



Slika 14.2 – Progress bar traka klase `progress-bar-striped`

Upotrebom klase `active` u kombinaciji sa `progress-bar-striped`, progress bar traka postaje aktivna, što se ogleda u kretanju prugastih linija. Na elektronskoj platformi *eŠkola veba* dostupni su primeri za sve pomenute klase.

<sup>9</sup> Internet Explorer 9 i ranije verzije Internet Explorer-a ne podržavaju progress bar trake.

## 15. Grupisanje elemenata

Elementi se mogu grupisati unutar elemenata *div* i *ul* i to se realizuje upotrebom klase *list-group* nad tim elementima. Unutar elementa *div* je moguće grupisati linkove, pasuse i elemente *span* i *div*. Ukoliko je potrebno grupisati naslov (*h1*, *h2*, *h3*, *h4*, *h5*, *h6*), potrebno je smestiti ga unutar elementa *div* i potom grupisati taj element *div*. Unutar neuređene liste moguće je grupisati isključivo stavke te liste, pri čemu se u stavkama mogu pojaviti različiti elementi (linkovi, pasusi, elementi *span* i *div* i naslovi).

### 15.1 Grupisanje unutar elementa *div*

Grupisanje različitih elemenata unutar elementa *div* se realizuje upotrebom elementa *div* klase *list-group* unutar kog su svi elementi koji se grupišu klase *list-group-item*. Ovakvo grupisanje se smatra podrazumevanim. Ukoliko je potrebno grupisati naslov, treba ga smestiti unutar elementa *div* klase *list-group-item* i obezbediti da sam naslov bude klase *list-group-item-heading*. Ukoliko je potrebno grupisati element *div* u kome se nalazi naslov i pasus, potrebno je da taj element *div* bude klase *list-group-item*, pasus klase *list-group-item-text* a naslov klase *list-group-item-heading*. Kod 15.1 prikazuje podrazumevano grupisanje različitih elemenata unutar elementa *div*.

```
<div class="list-group">
  <a href="#" class="list-group-item"> Link </a>
  <p class="list-group-item"> Pasus </p>
  <span class="list-group-item"> Span </span>
  <div class="list-group-item"> Div </div>
  <div class="list-group-item"> <h2 class="list-group-item-heading"> Naslov </h2> </div>
  <div class="list-group-item">
    <h2 class="list-group-item-heading"> Naslov </h2>
    <p class="list-group-item-text"> Pasus ispod naslova</p>
  </div>
</div>
```

Kôd 15.1 Podrazumevano grupisanje unutar elementa *div*

### 15.2 Grupisanje unutar neuređene liste

Grupisanje različitih elemenata unutar neuređene liste se realizuje upotrebom elementa *ul* klase *list-group* unutar koje su sve stavke klase *list-group-item*. Ovo grupisanje se smatra podrazumevanim. Kod 15.2 prikazuje podrazumevano grupisanje unutar neuređene liste.

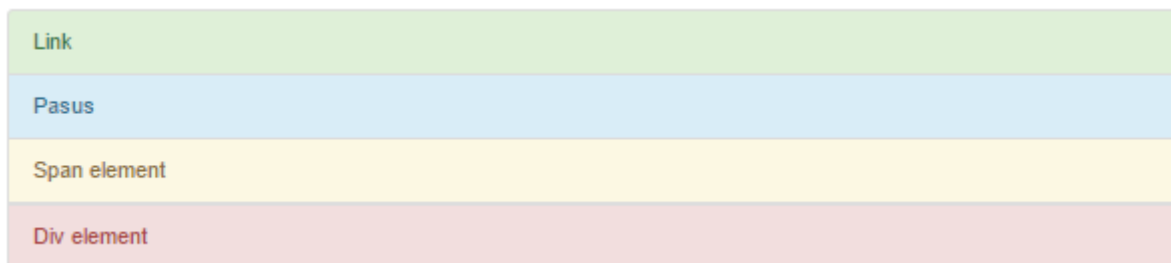
```
<ul class="list-group">
  <li class="list-group-item">1.stavka</li>
  <li class="list-group-item"> <a href="#"> Link</a> </li>
  <li class="list-group-item"> <p>Pasus</p> </li>
  <li class="list-group-item"> <span>Span element</span> </li>
  <li class="list-group-item"> <div>Div element</div> </li>
  <li class="list-group-item"> <h3>Naslov</h3> </li>
  <li class="list-group-item"> <h3>Naslov</h3> <p>Pasus ispod naslova</p> </li>
</ul>
```

Kôd 15.2 – Podrazumevano grupisanje unutar neuređene liste

**Napomena:** u svim prethodnim primerima, vrednost atributa *href* je postavljena na vrednost # kako ne bi došlo do greške 404. U realnoj situaciji, umesto toga treba staviti pravi link.

## 15.3 Zadavanje boje grupisanim elementima

Zadavanje boje grupisanim elementima se postiže upotrebom klasa: *list-group-item-success*, *list-group-item-info*, *list-group-item-warning* i *list-group-item-danger*. Slika 15.1 prikazuje grupisanje različito obojenih elemenata unutar elementa *div* pomoću koda 15.3.



Slika 15.1 – Grupisanje obojenih elemenata

```
<div class="list-group">
  <a href="#" class="list-group-item list-group-item-success">Link</a>
  <p class="list-group-item list-group-item-info"> Papus </p>
  <span class="list-group-item list-group-item-warning"> Span element </span>
  <div class="list-group-item list-group-item-danger"> Div element </div>
</div>
```

Kôd 15.3 – Upotreba klasa za zadavanje boje grupisanih elemenata

Na elektronskoj platformi *eŠkola veća* dostupni su primeri koji prikazuju upotrebu pomenutih klasa, kao i grupisanje neaktivnih linkova i grupisanje sa oznakama unutar elemenata *div* i *ul*.

## 16. Dodaci radnog okruženja Bootstrap

Korišćenje dodataka se definiše navođenjem poslednje iskompajlirane verzije *JavaScript*-a prilikom uvođenja radnog okruženja Bootstrap u *HTML* stranicu preko *CDN*-a, što je prikazano na početku ovog rada u kodu 1.1. Dodaci se mogu koristiti individualno (upotrebom individualnih *\*.js* fajlova) ili u kompletu (upotrebom fajla *bootstrap.js* ili *bootstrap.min.js*). Ukoliko se dodaci koriste u kompletu, navođenjem jednog od pomenuta dva fajla biće obezbeđena funkcionalnost svih dodataka.

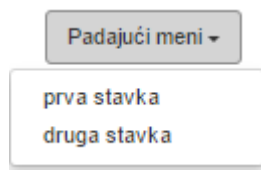
### 16.1 Dodatak dropdown

Dropdown je dodatak koji omogućava prikazivanje padajućeg menija. Ukoliko se individualno uvodi, potrebno je navođenje fajla *dropdown.js*. Padajući meni se kreira pomoću elementa *div* klase *dropdown* ili klase *dropup*, u kom se nalaze obavezni elementi: dugme i neuređena lista.

Dugme služi za otvaranje padajućeg menija. Zadaje se upotrebom elementa *button* klase *dropdown-toggle*. Unutar elementa *button* preporučuje se navođenje elementa *span*, klase *caret*, jer obezbeđuje grafički prikaz strelice na dugmetu. Neuređena lista unutar padajućeg menija prikazuje opcije koje padajući meni nudi. Zadaje se upotrebom elementa *ul* klase *dropdown-menu*. Uobičajeno je da se unutar svake stavke nalazi link. Ovim je navedeno opšte uputstvo za kreiranje padajućeg menija ali time nije obezbeđena njegova funkcionalnost i očekivano ponašanje. Funkcionalnost padajućeg menija se može zadati upotrebom atributa ili upotrebom biblioteke *jQuery*.

#### 16.1.1 Upotreba atributa pri kreiranju padajućeg menija

Jedan od načina da se obezbedi funkcionalnost padajućeg menija je da se na nivou pomenutog dugmeta navede atribut *data-toggle="dropdown"*. Slika 16.1.1 prikazuje padajući meni čija je funkcionalnost obezbeđena na ovaj način pomoću koda 16.1.1.



Slika 16.1.1 – Padajući meni

```
<div class="dropdown">
  <button class="btn btn-default dropdown-toggle" type="button" data-toggle="dropdown">
    Padajući meni <span class="caret"></span>
  </button>
  <ul class="dropdown-menu">
    <li><a href="#">prva stavka</a> </li>
    <li><a href="#">druga stavka</a> </li>
  </ul>
</div>
```

Kôd 16.1.1 – Upotreba atributa *data-toggle* pri kreiranju padajućeg menija

## 16.1.2 Upotreba biblioteke *jQuery* pri kreiranju padajućeg menija

Drugi način da se obezbedi funkcionalnost padajućeg menija je upotrebom biblioteke *jQuery*, pozivom metoda *dropdown()* nad dugmetom koje se nalazi unutar njega. U ovom slučaju je upotreba atributa *data-toggle="dropdown"* na nivou dugmeta takođe neophodna. Kôd 16.1.2 prikazuje kreiranje padajućeg menija sa slike 16.1.1, čija je funkcionalnost obezbeđena pozivom metoda *dropdown()*.

```
<div class="dropdown">
  <button class="btn btn-default dropdown-toggle" type="button" data-toggle="dropdown">
    Padajući meni <span class="caret"></span> </button>
  <ul class="dropdown-menu">
    <li><a href="#">prva stavka</a></li>
    <li><a href="#">druga stavka</a></li>
  </ul>
</div>
<script> $(document).ready(function(){$(".dropdown-toggle").dropdown();}); </script>
```

Kôd 16.1.2 – Upotreba biblioteke *jQuery* pri kreiranju padajućeg menija

Funkcionalnost radnog okruženja Bootstrap u radu sa padajućim menijem se zasniva na metodama i događajima dodatka *dropdown*, uvedenih pomoću biblioteke *jQuery*. Opcije dodatka *dropdown* nisu podržane.

## 16.1.3 Metodi dodatka *dropdown*

Metod dodatka *dropdown* predstavlja funkciju koja se poziva nad dugmetom iz padajućeg menija. Podržan je jedan metod, metod *dropdown("toggle")*. Ovaj metod omogućava naizmenično prikazivanje i sakrivanje neuređene liste padajućeg menija pritiskom na njegovo dugme.

Kôd 16.1.3 prikazuje upotrebu metoda *dropdown("toggle")* nad dugmetom u okviru padajućeg menija, pod pretpostavkom da je padajući meni kreiran kao u kodu 16.1.2.

```
<script>
  $(document).ready(function(){
    $(".dropdown-toggle").dropdown("toggle");
  });
</script>
```

Kôd 16.1.3 - Upotreba metoda *dropdown("toggle")* nad dugmetom padajućeg menija

## 16.1.4 Događaji dodatka *dropdown*

Dodatna funkcionalnost radnog okruženja Bootstrap u radu sa padajućim menijem se zasniva na događajima dodatka *dropdown*.

Zadavanje događaja dodatka *dropdown* se realizuje pozivom metoda *on* nad padajućim menijem za koji se vezuje taj događaj. Prvi parametar je naziv događaja dodatka *dropdown*, a drugi parametar je funkcija koja će se realizovati na veb stranici. Kôd 16.1.4 prikazuje upotrebu događaja *show.bs.dropdown* se nad padajućim menijem, pod pretpostavkom da je on kreiran pomoću koda 16.1.2.

Ovaj događaj opisuje akciju koja se realizuje neposredno pre nego što se padajući meni prikaže. Akcija koja se realizuje kroz poziv tog događaja je pojavljivanje dijaloga prozora.

```
<script>
$('.dropdown').on('show.bs.dropdown', function(){
  alert('Padajući meni će uskoro biti prikazan.');
```

Kôd 16.1.4 - Upotreba događaja `show.bs.dropdown`

Dodatak `dropdown` podržava 4 događaja i njihove karakteristike su prikazane u tabeli 16.1.1.

Tabela 16.1.1 – Događaji dodatka `dropdown`

Događaj:	Opis:
<code>show.bs.dropdown</code>	Događaj se realizuje neposredno pre nego što se padajući meni prikaže.
<code>shown.bs.dropdown</code>	Događaj se realizuje nakon što je padajući meni prikazan.
<code>hide.bs.dropdown</code>	Događaj se realizuje neposredno pre nego što će padajući meni biti sakriven.
<code>hidden.bs.dropdown</code>	Događaj se realizuje nakon što je padajući meni sakriven.

U tabeli 16.1.2 su prikazane preostale klase koje dodatak `dropdown` podržava.

Tabela 16.1.2 – Klase dodatka `dropdown`

Klasa:	Opis:
<code>Dropup</code>	Omogućava izlistavanje padajućeg menija nagore.
<code>dropdown-menu-right</code>	Padajući meni se prikazuje desno poravnat prilikom otvaranja.
<code>dropdown-header</code>	Tekstualni sadržaj se prikazuje većim fontom, u vidu zaglavlja.
<code>Disabled</code>	Stavka padajućeg menija postaje neaktivna.
<code>Divider</code>	Prikazuje se horizontalna linija koja razdvaja preostale stavke liste.

Na elektronskoj platformi *eŠkola veba* se mogu videti primeri njihove upotrebe. Više informacija o dodatku `dropdown` radnog okruženja Bootstrap se može videti u [4].

## 16.2 Dodatak `collapse`

`Collapse` je dodatak koji omogućava prikazivanje i sakrivanje elementa pritiskom na neko dugme ili link. Ukoliko se uvodi individualno, potrebno je navođenje fajla `collapse.js`.

Korišćenje dodatka `collapse` se svodi na korišćenje klasa: `collapse`, `in` i `collapsing`. Korišćenje navedenih klasa se ostvaruje upotrebom atributa ili upotrebom biblioteke *jQuery*. Na elektronskoj platformi *eŠkola veba*, upotreba ovih klasa je u oba slučaja prikazana nad panelom i nad elementom `div`, koji se prikazuju, odnosno sakrivaju klikom na dugme i na link. U ovom radu će biti prikazana upotreba nad elementom `div` koji se prikazuje, odnosno sakriva klikom na dugme.

### 16.2.1 Prikazivanje elementa `div`

U ovom slučaju se podrazumeva da je element `div` inicijalno sakriven i da se klikom na dugme prikazuje. Da bi se ovo postiglo, neophodno je da element `div` bude klase `collapse`.



U slučaju da se njegovo prikazivanje vrši upotrebom atributa, potrebno je zadati mu identifikator, a na nivou dugmeta navesti attribute: `data-toggle="collapse"` i `data-target`. Atribut `data-toggle="collapse"` zadaje funkciju dugmeta a `data-target` vrši povezivanje dugmeta sa sakrivenim elementom, pa se za vrednost ovog atributa navodi identifikator pomenutog elementa `div`. Slika 16.2.1 prikazuje dugme pre i nakon što se pritisne. Njegova funkcionalnost je obezbeđena upotrebom atributa `data-toggle` pomoću koda 16.2.1.



Slika 16.2.1 – Prikazivanje teksta klikom na dugme

```
<button data-toggle="collapse" data-target="#id1">Kliknite</button>
<div class="collapse" id="id1"> Tekst koji prikazujemo.</div>
```

Kôd 16.2.1 – Upotreba atributa `data-toggle` za prikazivanje teksta klikom na dugme

Sa druge strane, u slučaju da se prikazivanje elementa `div` vrši upotrebom biblioteke `jQuery`, potrebno je obezbediti da se klikom na dugme poziva metod `collapse()` nad elementom `div`, pomoću koda 16.2.2.

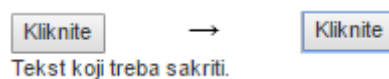
```
<button type="button" class="btn btn-warning">Prikaži</button>
<div class="collapse">Ovo je inicijalno sakriveno.</div>
<script>
$(document).ready(function(){
  $(".btn-warning").click(function(){ $(".collapse").collapse();});
});
</script>
```

Kôd 16.2.2 – Upotreba biblioteke `jQuery` za prikazivanje teksta klikom na dugme

## 16.2.2 Sakrivanje elementa `div`

U ovom slučaju se podrazumeva da je element `div` inicijalno prikazan i da se klikom na dugme sakriva. Da bi se ovo postiglo, neophodno je da element `div` bude klase `collapse` i `in`.

U slučaju da se njegovo sakrivanje vrši upotrebom atributa `data-toggle`, potrebno je ispuniti iste uslove kao u sekciji 16.2.1. Slika 16.2.2 prikazuje dugme pre i nakon što se pritisne. Njegova funkcionalnost je obezbeđena u kodu 16.2.3.



Slika 16.2.2 – Sakrivanje teksta klikom na dugme

```
<button data-toggle="collapse" data-target="#id2">Kliknite</button>
<div class="collapse in" id="id2">Tekst koji treba sakriti.</div>
```

Kôd 16.2.3 - Upotreba atributa `data-toggle` za sakrivanje teksta klikom na dugme

U slučaju da se sakrivanje elementa `div` vrši upotrebom biblioteke `jQuery`, potrebno je obezbediti da se klikom na dugme poziva metod `collapse()` nad tim elementom `div`, odnosno postupiti na isti način kao u sekciji 16.2.1.

**Napomena:** upotreba atributa *data-toggle* omogućava naizmenično sakrivanje, odnosno prikazivanje elemenata klikom na dugme. Sa druge strane, upotreba metoda *collapse()* dozvoljava samo jedno sakrivanje ili jedno prikazivanje elementa.

Funkcionalnost radnog okruženja Bootstrap u sakrivanju i prikazivanju elemenata se zasniva na opcijama, metodama i događajima dodatka *collapse*, uvedenih pomoću biblioteke jQuery.

### 16.2.3 Opcije dodatka *collapse*

Opcije dodatka *collapse* predstavljaju svojstva radnog okruženja Bootstrap koja se dodeljuju elementu pomoću kog se element *div* sakriva, odnosno prikazuje, što je najčešće link ili dugme. Opcije se mogu koristiti kao atributi koji počinju prefiksom *data-* a mogu se koristiti i kao parametri metoda *collapse()*.

Elektronske lekcije radnog okruženja Bootstrap prikazuju opciju *parent*. Ova opcija se upotrebljava kada postoji više elemenata za sakrivanje, odnosno prikazivanje, koji se nalaze u okviru jednog roditeljskog elementa u okviru kog su grupisani. Pretpostavimo da je identifikator roditeljskog elementa *id="h"*. Opcija *parent* obezbeđuje da prikazivanje jednog podelementa u okviru roditeljskog elementa, automatski povlači sakrivanje ostalih podelemenata.

U slučaju da se opcija upotrebljava u formi atributa, navodi se kao atribut *data-parent* na nivou linka ili dugmeta koje služi za sakrivanje, odnosno prikazivanje podelementa. Za vrednost ovog atributa se navodi identifikator roditeljskog elementa, dakle *data-parent="#h"*. U slučaju da se opcija koristi kao parameter metoda *collapse()*, ista opcija se koristi u sledećoj formi: *collapse({parent:"#h"})*. Ovaj metod se poziva nad podelementom kog treba sakriti, odnosno prikazati, u trenutku pritiska na dugme ili link koje služi za sakrivanje, odnosno prikazivanje.

### 16.2.4 Metodi dodatka *collapse*

Metod dodatka *collapse* predstavlja funkciju koja se poziva nad elementom *div* kojeg treba sakriti, odnosno prikazati u trenutku kada se pritisne dugme. Kôd 16.2.4 prikazuje upotrebu metoda *collapse("toggle")* nad elementom *div*, koji je inicijalno sakriven. Uloga ovog metoda je prikaže element *div*, ukoliko je on sakriven i da ga sakrije ukoliko je prikazan, pritiskom na dugme. Ovaj metod omogućava da se pritiscima na dugme element *div* naizmenično prikazuje i sakriva.

```
<button type="button" class="btn btn-default">Prikaži/Sakrij</button>
<div class="collapse">Ovo je neki tekst.</div>
<script>
$(document).ready(function(){
$(".btn-default").click(function(){ $(".collapse").collapse('toggle'); });
});
</script>
```

Kôd 16.2.4 - Upotreba metoda *collapse("toggle")*

Ostali metodi dodatka *collapse* se koriste analogno i navedeni su u tabeli 16.2.1.

Tabela 16.2.1 – Metodi dodatka collapse

Metod:	Opis:
collapse(opcije)	Parametar ovog metoda je opcija.
collapse("toggle")	Metod sakriva element ukoliko je bio prikazan, odnosno prikazuje ukoliko je bio sakriven.
collapse("show")	Metod prikazuje element.
collapse("hide")	Metod sakriva element.

## 16.2.5 Događaji dodatka collapse

Događaji dodatka collapse su akcije koje se mogu realizovati isključivo kada je omogućeno prikazivanje ili sakrivanje elementa *div*.

Zadavanje događaja dodatka collapse se realizuje pozivom metoda *on* nad elementom koji se sakriva, odnosno prikazuje. Prvi parametar je naziv događaja dodatka collapse, a drugi parametar je funkcija koja će se realizovati na veb stranici. Kôd 16.2.5 prikazuje upotrebu događaja *show.bs.collapse* nad elementom *div*, koji je inicijalno sakriven.

Ovaj događaj opisuje akciju koja se realizuje neposredno pre nego što će sakriveni element biti prikazan. Akcija koja se realizuje kroz poziv tog događaja je pojavljivanje dijaloga prozora. U ovom kodu se može primetiti da zadavanju događaja prethodi poziv metoda *collapse("toggle")*, koji obezbeđuje prikazivanje sakrivenog elementa, što je osnovni preduslov da se događaj realizuje.

```
<button type="button" class="btn btn-default">Prikazi/Sakrij</button>
<div class="collapse"> Ovo je neki tekst. </div>
<script>
$(document).ready(function(){
  $(".btn-default").click(function(){
    $(".collapse").collapse('toggle'); });
  $(".collapse").on('show.bs.collapse', function(){
    alert('Element će uskoro biti prikazan.');
```

Kôd 16.2.5 - Upotreba događaja show.bs.collapse

Dodatak collapse podržava 4 događaja i njihove karakteristike su prikazane u tabeli 16.2.2.

Tabela 16.2.2 – Događaji dodatka collapse

Događaj:	Opis:
show.bs.collapse	Događa se neposredno pre nego što će sakriveni element biti prikazan.
shown.bs.collapse	Događa se nakon što je element prikazan.
hide.bs.collapse	Događa se neposredno pre nego što će element biti sakriven.
hidden.bs.collapse	Događa se nakon što je element sakriven.

Na elektronskoj platformi *eŠkola veba* se mogu videti primeri za sve navedene opcije, metode i događaje.

## 16.3 Dodatak carousel

Carousel je dodatak koji omogućava automatsko i manuelno prikazivanje slajdova. Ukoliko se uvodi individualno, potrebno je navođenje fajla *carousel.js*. Korišćenje dodatka carousel se svodi na korišćenje istoimene klase *carousel*<sup>10</sup>.

Komponentom carousel ćemo zvati svaki element *div* klase *carousel*. Komponenta carousel se koristi za prikazivanje slajdova. Slajdovi se podrazumevano prikazuju automatski, ali je moguće i manuelno prikazivanje. Osnovni delovi jedne komponente carousel su: indikatori, omotač slajdova i kontrole za listanje slajdova.

Klasa *carousel* naglašava da ova komponenta koristi dodatak carousel i njena upotreba je obavezna. Na nivou komponente carousel se obavezno zadaje identifikator i opciono koristi klasa *slide*. Klasa *slide* uvodi poseban efekat u prikazivanju slajdova.

### 16.3.1 Indikatori

Indikatori se formiraju kao stavke uređene liste klase *carousel-indicators*. Ova klasa zadaje indikatore komponente carousel. Prva stavka te liste je klase *active*, kako bi komponenta carousel pravilno funkcionisala. Indikatori predstavljaju male kružnice na dnu slajda pri čemu se svaka od njih vezuje za odgovarajući slajd. Onaj indikator čiji slajd je trenutno prikazan se prikazuje belom bojom. Na ovaj način se beleži prolazak kroz slajdove.

### 16.3.2 Omotač slajdova

Omotač slajdova se predstavlja elementom *div* klase *carousel-inner*. Ova klasa označava da se u okviru tog elementa *div* zadaju slajdovi komponente carousel. Unutar omotača se navode slajdovi kao zasebni elementi *div* klase *item*. Klasa *item* označava da se u okviru tog elementa zadaje sadržaj slajda što može biti slika ili tekst. Na nivou prvog slajda, neophodno je navođenje klase *active* pored postojeće klase *item*. U suprotnom, komponenta carousel neće biti prikazana.

### 16.3.3 Kontrole za listanje slajdova

Postoje leva i desna kontrola za listanje slajdova. Leva kontrola se upotrebljava za prelazak na prethodni slajd, dok se desna upotrebljava za prelazak na sledeći. Kontrole se predstavljaju linkom klase *carousel-control*. Ova klasa daje linku značenje kontrole za listanje slajdova. Link klase *carousel-control* i klase *left* dobija značenje leve kontrole, dok link klase *carousel-control* i klase *right* dobija značenje desne.

Sve ovo predstavlja opšte uputstvo za kreiranje osnovne komponente carousel, ali njena funkcionalnost time nije obezbeđena. Funkcionalnost komponente carousel se ostvaruje na dva načina: upotrebom atributa ili upotrebom biblioteke *jQuery*. Opšta priča važi u oba slučaja. Razmotrimo pojedinačne slučajeve.

---

<sup>10</sup> Dodatak carousel nije podržan u Internet Explorer 9 i njegovim ranijim verzijama.

### 16.3.4 Upotreba atributa pri kreiranju komponente carousel

Jedan od načina da se kreira funkcionalna komponenta carousel sa svojim osnovnim delovima je upotrebom različitih atributa. Na nivou same komponente carousel je neophodno navođenje atributa `data-ride="carousel"`. Ovaj atribut obezbeđuje da se prikazivanje slajdova započne odmah po učitavanju veb strane. Sledeći neophodni atributi se zadaju na nivou indikatora. Tačnije, na nivou stavki uređene liste su obavezni atributi `data-target` i `data-slide-to`. Atribut `data-target` povezuje stavku sa komponentom pa se za vrednost ovog atributa navodi identifikator komponente carousel. Atribut `data-slide-to` označava koji će se slajd prikazati pritiskom na tu kružnicu. Vrednost ovog atributa predstavlja se odgovarajućim brojem slajda. I na kraju, neophodna je upotreba atributa `data-slide` na nivou leve i desne kontrole za listanje slajdova. Dopuštene vrednosti atributa su `"prev"` i `"next"`. Prva se zadaje na nivou leve kontrole i označava da se njome mogu listati prethodni slajdovi, a druga se zadaje na nivou desne i označava listanje narednih slajdova. Slika 16.3.1 prikazuje komponentu carousel sa svim osnovnim delovima, čiji su slajdovi slike i koja je dobijena pomoću koda 16.3.1.



Slika 16.3.1 – Komponenta carousel čiji su slajdovi slike

```
<div id="id1" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#id1" data-slide-to="0" class="active"></li>
    <li data-target="#id1" data-slide-to="1"> </li>
  </ol>
  <div class="carousel-inner" role="listbox">
    <div class="item active">
      
    </div>
    <div class="item">
      
    </div>
  </div>
  <a class="left carousel-control" href="#id1" role="button" data-slide="prev">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
  </a>
  <a class="right carousel-control" href="#id1" role="button" data-slide="next">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
  </a>
</div>
```

Kôd 16.3.1 – Kreiranje komponente carousel upotrebom atributa

### 16.3.5 Upotreba biblioteke *jQuery* pri kreiranju komponente carousel

Drugi način da se obezbedi funkcionalnost komponente carousel je upotrebom biblioteke *jQuery*. U ovom slučaju, potrebno je stavkama uređene liste zadati identifikator pa:

1. pozvati metod `carousel()` nad komponentom carousel (čime se ona aktivira)
2. obezbediti da se klikom na svaku stavku liste poziva metod `carousel()` nad komponentom, sa parametrom koji predstavlja broj odgovarajućeg indikatora

3.obezbediti da se klikom na levu kontrolu poziva metod `carousel("prev")` nad komponentom  
4.obezbediti da se klikom na desnu kontrolu poziva metod `carousel("next")` nad komponentom carousel.

Komponenta carousel sa slike 16.3.1 se može kreirati upotrebom biblioteke *jQuery* pomoću koda 16.3.2.

```
<div id="id1" class="carousel slide">
  <ol class="carousel-indicators">
    <li id="id2" class="active"></li>
    <li id="id3"></li>
  </ol>
  <div class="carousel-inner" role="listbox">
    <div class="item active">
      
    </div>
    <div class="item">
      
    </div>
  </div>
  <a class="left carousel-control" href="#id1" role="button">
    <span class="glyphicon glyphicon-chevron-left" aria-hidden="true"></span>
  </a>
  <a class="right carousel-control" href="#id1" role="button">
    <span class="glyphicon glyphicon-chevron-right" aria-hidden="true"></span>
  </a>
</div>
<script>
$(document).ready(function(){
  $("#id1").carousel();
  $("#id2").click(function(){ $("#id1").carousel(0); });
  $("#id3").click(function(){ $("#id1").carousel(1); });
  $(".left").click(function(){ $("#id1").carousel("prev"); });
  $(".right").click(function(){ $("#id1").carousel("next"); });
});
</script>
```

Kôd 16.3.2 – Kreiranje komponente carousel upotrebom biblioteke *jQuery*

Funkcionalnost radnog okruženja Bootstrap u prikazivanju slajdova pomoću komponente carousel se zasniva na opcijama, metodama i događajima dodatka carousel, uvedenih pomoću biblioteke *jQuery*.

### 16.3.6 Opcije dodatka carousel

Opcije dodatka carousel predstavljaju svojstva radnog okruženja Bootstrap koja opisuju komponentu carousel. U slučaju kada je komponenta carousel kreirana preko atributa, opcije se zadaju takođe preko atributa sa prefiksom „*data-*” na nivou same komponente carousel. U slučaju kada je komponenta kreirana pomoću biblioteke *jQuery*, opcije se zadaju kao parametri metoda `carousel()` koji se poziva nad komponentom carousel.

U tabeli 16.3.1 su prikazane opcije dodatka carousel. Za svaku od njih je prikazan odgovarajući atribut, odgovarajući poziv metoda, i dat je opis vrednosti koje ta opcija prihvata.

Tabela 16.3.1 – Opcije dodatka carousel

Opcija:	Opis:
interval	Određuje razmak između slajdova u milisekundama. Može se koristiti kao atribut <i>data-interval</i> ili u vidu metoda <i>carousel({interval:500})</i> . Podrazumevana vrednost opcije je 5000. Vrednost se može postaviti na <i>false</i> i na taj način se zaustavlja automatsko prikazivanje slajdova.
pause	Zaustavlja prikazivanje slajdova kada kursor miša uđe u komponentu carousel, i nastavlja prikazivanje kada je kursor napusti. Može se koristiti kao atribut <i>data-pause</i> ili u vidu metoda <i>carousel({pause:vrednost})</i> . Podrazumevana vrednost je <i>hover</i> . Vrednost se može postaviti na <i>false</i> i na taj način položaj kursora neće uticati na prikazivanje slajdova.
wrap	Određuje način prikazivanja slajdova. Može se koristiti kao atribut <i>data-wrap</i> ili u vidu metoda <i>carousel({wrap:vrednost})</i> . Dozvoljene su vrednosti <i>true</i> i <i>false</i> . Vrednost <i>true</i> označava konstantno prikazivanje slajdova kružnim redosledom, dok <i>false</i> označava da će se prikazivanje zaustaviti na poslednjem slajdu. Podrazumevano je <i>true</i> .

### 16.3.7 Metodi dodatka carousel

Metod dodatka carousel predstavlja funkciju koja se može pozvati nad komponentom carousel.

Kôd 16.3.3 prikazuje upotrebu metoda *carousel("pause")*, pod pretpostavkom da je carousel komponenta kreirana na način prikazan u kodu 16.3.2. Ovaj metod zaustavlja prikazivanje slajdova carousel komponente, pritiskom na dugme sa identifikatorom *id="dugme1"*, koje je dodatno kreirano za potrebe ovog primera.

```
<button type="button" id="dugme1" class="btn btn-info">Zaustavi slajdove</button>
<script>
$(document).ready(function(){
    $("#dugme1").click(function(){
        $("#id1").carousel("pause");
    });
    ...
});
</script>
```

Kôd 16.3.3 - Upotreba metoda carousel("pause")

Ostali metodi dodatka collapse se koriste analogno i prikazani su u tabeli 16.3.2.

Tabela 16.3.2 – Metodi dodatka carousel

Metod:	Opis:
carousel(opcije)	Aktivira komponentu carousel korišćenjem opcija datih u prethodnoj tabeli.
carousel("pause")	Zaustavlja prikazivanje slajdova.
carousel("cycle")	Prikazuje slajdove idući sa leva na desno.
carousel(broj)	Prikazuje slajdove idući sa leva na desno.
carousel("prev")	Prikazuje prethodni slajd.
carousel("next")	Prikazuje naredni slajd.

## 16.3.8 Događaji dodatka carousel

Događaji dodatka carousel su akcije koje se realizuju isključivo kada je omogućeno prikazivanje slajdova.

Zadavanje događaja dodatka carousel se realizuje pozivom metoda `on` nad carousel komponentom. Prvi parametar je naziv događaja, a drugi parameter je funkcija koja će se realizovati na veb stranici. Kôd 16.3.4 prikazuje upotrebu događaja `slide.bs.carousel` nad carousel komponentom, pod pretpostavkom da je ona kreirana u kodu 16.3.2. Ovaj događaj zadaje akciju koja se realizuje neposredno pre nego što se izvrši prelaz sa jednog slajda na drugi. Akcija koja se realizuje kroz poziv tog događaja u konkretnom primeru je pojavljivanje dijaloga prozora.

```
$("#id1").on('slide.bs.carousel', function () {  
    alert('Uskoro će biti prikazan sledeći slajd.');
```

Kôd 16.3.4 - Upotreba događaja `slide.bs.carousel`

Zadavanju svih događaja prethodi poziv metoda `carousel()` nad komponentom carousel, koji obezbeđuje prikazivanje slajdova, što je osnovni preduslov da se događaj realizuje. Svi događaji dodatka carousel su prikazani u tabeli 16.3.3.

Tabela 16.3.3 – Događaji dodatka carousel

Događaj:	Opis:
slide.bs.carousel	Događa se neposredno pre nego što se izvrši prelaz sa jednog slajda na drugi.
slid.bs.carousel	Događa se nakon što se prikaže neki slajd.

Na elektronskoj platformi *eŠkola veba* su dostupni primeri za sve navedene opcije, metode i događaje.

## 16.4 Dodatak modal

Modal je dodatak koji omogućava prikazivanje dijaloga prozora na vrhu strane. Ukoliko se uvodi individualno, potrebno je navođenje fajla `modal.js`. Upotreba dodatka modal se svodi na korišćenje istoimene klase `modal`.

### 16.4.1 Kreiranje dijaloga prozora

Kreiranje dijaloga prozora se sprovodi na početku `HTML` stranice i podrazumeva kreiranje dve celine: njegovog sadržaja i dugmeta pomoću kog se otvara.

Sadržaj dijaloga prozora se sastoji iz tri elementa `div` poređana jedan unutar drugog. Prvi je klase `modal`, sa atributom `role="dialog"`. Klasa `modal` zadaje fokus dijalogu prozoru i obavezna je prilikom kreiranja. Na nivou ovog elementa opciono se koristi klasa `fade` koja zadaje efekat pri ulasku i izlasku iz dijaloga prozora. Upotreba atributa `role="dialog"` takođe nije obavezna ali se preporučuje zbog čitača ekrana. Drugi element `div` je klase `modal-dialog`. Ova klasa postavlja podrazumevanu širinu dijaloga prozora i prikazuje margine. I treći je klase `modal-content`, koja stilizuje izgled dijaloga prozora zadajući mu ivice i boju pozadine. Sadržaj može biti podeljen na zaglavlje, telo i podnožje. Svaki od tih delova predstavljen je



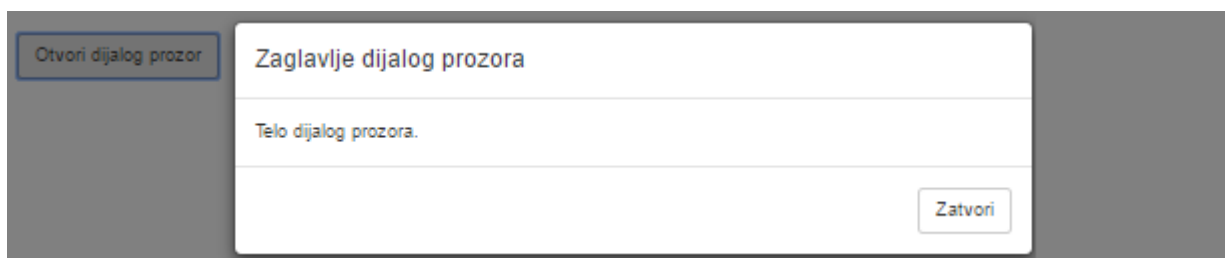
elementom *div* odgovarajuće klase: *modal-header*, *modal-body* ili *modal-footer* koje stilizuju, redom, izgled zaglavlja, tela i podnožja dijalog prozora.

Da bi imao mogućnost zatvaranja, na nivou podnožja potrebno je staviti dugme na čiji klik se dijalog prozor zatvara. Obavezan atribut na nivou dugmeta je *data-dismiss="modal"*. Treba imati u vidu da otvaranje više dijalog prozora istovremeno nije moguće. Potrebno je zatvoriti stari kako bi se otvorio novi dijalog prozor. Ovo predstavlja opšte uputstvo za kreiranje dijalog prozora, ali njegova funkcionalnost time nije obezbeđena. Funkcionalnost dijalog prozora se zadaje na dva načina: upotrebom atributa ili upotrebom biblioteke *jQuery*.

## 16.4.2 Upotreba atributa pri kreiranju dijalog prozora

Upotreba atributa pri kreiranju dijalog prozora i zadavanju njegove funkcionalnosti podrazumeva upotrebu atributa *data-toggle="modal"* i *data-target*, koji se navode na nivou dugmeta.

Atribut *data-target* povezuje dugme sa dijalog prozorom, pa se za vrednost ovog atributa navodi identifikator prvog elementa *div* iz sadržaja dijalog prozora, sa oznakom # ispred. Slika 16.4.1 prikazuje dijalog prozor kreiran na ovaj način primenom koda 16.4.1.



Slika 16.4.1 – Dijalog prozor

```
<button type="button" class="btn btn-default" data-toggle="modal" data-target="#id1">
Otvori dijalog prozor
</button>
<div class="modal fade" id="id1" role="dialog">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h4 class="modal-title">Zaglavlje dijalog prozora</h4>
      </div>
      <div class="modal-body">
        Telo dijalog prozora.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Zatvori</button>
      </div>
    </div>
  </div>
</div>
```

Kôd 16.4.1 – Kreiranje dijalog prozora upotrebom atributa *data-toggle*

## 16.4.3 Upotreba biblioteke *jQuery* pri kreiranju dijalog prozora

Funkcionalnost dijalog prozora upotrebom biblioteke *jQuery* se ostvaruje pozivom metoda *modal()* nad prvim elementom *div* iz sadržaja, kada se klikne na dugme.

Dijalog prozor sa slike 16.4.1 se može dobiti upotrebom biblioteke *jQuery* pomoću koda 16.4.2.

```
<button type="button" class="btn btn-default" id="dugme">Otvori dijalog prozor</button>
<div class="modal fade" id="sadrzaj">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header"> <h4 class="modal-title">Zaglavlje dijalog prozora</h4> </div>
      <div class="modal-body"> Telo dijalog prozora. </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-default" data-dismiss="modal">Zatvori </button>
      </div>
    </div>
  </div>
</div>
<script> $(document).ready(function(){$("#dugme").click(function(){$("#sadrzaj").modal();});});
</script>
```

Kôd 16.4.2 – Kreiranje dijalog prozora upotrebom biblioteke *jQuery*

Funkcionalnost radnog okruženja Bootstrap u prikazivanju dijalog prozora se zasniva na opcijama, metodama i događajima dodatka modal, uvedenih pomoću biblioteke *jQuery*.

#### 16.4.4 Opcije dodatka modal

Opcije dodatka modal predstavljaju svojstva radnog okruženja Bootstrap koja opisuju dijalog prozor. U slučaju kada je dijalog prozor kreiran preko atributa, opcije se zadaju takođe preko atributa sa prefiksom „*data-*”, na nivou dugmeta kojim se dijalog prozor otvara. U slučaju kada je komponenta kreirana pomoću biblioteke *jQuery*, opcije se zadaju kao parametri metoda *modal()*, koji se poziva nad sadržajem dijalog prozora u momentu pritiska na dugme koje služi za njegovo otvaranje.

U tabeli 16.4.1 su prikazane opcije dodatka modal. Za svaku je dat opis funkcionalnosti, odgovarajući atribut, odgovarajući poziv metoda i skup dozvoljenih vrednosti.

Tabela 16.4.1 – Opcije dodatka modal

Opcija:	Opis:
backdrop	Određuje da li će dijalog prozor imati osenčenu pozadinu ili ne. Opcija se može koristiti preko atributa <i>data-backdrop</i> ili u obliku metoda <i>modal({backdrop:vrednost})</i> . Dopusnene vrednosti opcije su: <i>true</i> (za osenčenu pozadinu), <i>false</i> (za neosenčenu) i „ <i>static</i> ” (za osenčenu). Vrednosti <i>false</i> i „ <i>static</i> ” ne dopuštaju zatvaranje dijalog prozora kada se klikne van njega.
keyboard	Određuje da li se dijalog prozor može zatvoriti pritiskom dugmeta sa tastature (ESC). Opcija se može koristiti preko atributa <i>data-keyboard</i> ili u obliku metoda <i>modal({keyboard:vrednost})</i> . Dopusnene vrednosti opcije su: <i>true</i> (omogućeno zatvaranje pritiskom na ESC) i <i>false</i> (nije omogućeno).

#### 16.4.5 Metodi dodatka modal

Metod dodatka modal predstavlja funkciju koja se može pozvati nad sadržajem dijalog prozora, pritiskom na dugme čija je uloga otvaranje tog prozora.

Kôd 16.4.3 prikazuje upotrebu metoda `modal("toggle")` nad sadržajem dijalog prozora pritiskom na dugme, pod pretpostavkom da su sadržaj i dugme dijalog prozora kreirani na način prikazan u kodu 16.4.2. Ovaj metod omogućava naizmenično otvaranje i zatvaranje dijalog prozora, pritiskom na dugme.

```
<script>
$(document).ready(function(){
  $("#dugme").click(function(){
    $("#sadrzaj").modal("toggle");
  });
});
</script>
```

Kôd 16.4.3 - Upotreba metoda `modal("toggle")`

Ostali metodi dodatka `modal` se koriste analogno i prikazani su u tabeli 16.4.2.

Tabela 16.4.2 – Metodi dodatka `modal`

Metod:	Opis:
<code>modal(opcije)</code>	Argumenti ovog metoda su opcije, opisane u prethodnoj sekciji.
<code>modal("toggle")</code>	Otvora dijalog prozor ako je on zatvoren, i zatvara ga ako je otvoren.
<code>modal("show")</code>	Prikazuje dijalog prozor.
<code>modal("hide")</code>	Zatvara dijalog prozor.

## 16.4.6 Događaji dodatka `modal`

Događaji dodatka `modal` su akcije koje se realizuju isključivo kada je omogućeno prikazivanje dijalog prozora.

Događaji dodatka `modal` se zadaju pozivom metoda `on` nad sadržajem dijalog prozora. Prvi parametar tog metoda je naziv događaja, a drugi parametar je funkcija koja će se realizovati na veb stranici.

Kôd 16.4.4 prikazuje upotrebu događaja `show.bs.modal` nad sadržajem dijalog prozora, pod pretpostavkom da su sadržaj dijalog prozora i dugme za njegovo prikazivanje kreirani u kodu 16.4.2. Ovaj događaj zadaje akciju koja se realizuje neposredno pre nego što će dijalog prozor biti otvoren. Primetimo da zadavanju ovog događaja prethodi pozivanje metoda `modal("show")` nad sadržajem dijalog prozora, čime se on prikazuje, što je osnovni preduslov da se događaj realizuje.

```
<script>
$(document).ready(function(){
  $("#dugme").click(function(){
    $("#sadrzaj").modal("show");
  });
  $("#sadrzaj").on('show.bs.modal', function () {
    alert('Dijalog prozor će uskoro biti otvoren.');
```

Kôd 16.4.4 - Upotreba događaja `show.bs.modal`

Ostali događaji dodatka `modal` se koriste analogno i prikazani su u tabeli 16.4.3.

Tabela 16.4.3 – Događaji dodatka modal

Događaj:	Opis:
show.bs.modal	Događa se neposredno pre nego što će dijalog prozor biti otvoren.
shown.bs.modal	Događa se nakon što je dijalog prozor otvoren.
hide.bs.modal	Događa se neposredno pre nego što će dijalog prozor biti zatvoren.
hidden.bs.modal	Događa se nakon što je dijalog prozor zatvoren.

### 16.4.7 Veličina dijalog prozora

Veličina dijalog prozora se zadaje upotrebom klasa `modal-sm` i `modal-lg`. Klasa `modal-sm` zadaje najmanju a klasa `modal-lg` najveću veličinu dijalog prozora. Navode se na nivou elementa `div` klase `modal-dialog`. Ukoliko se ne upotrebi nijedna od njih, veličina dijalog prozora će biti podrazumevana.

Na elektronskoj platformi *eŠkola veba* dostupni su primeri upotrebe svih pomenutih klasa, opcija, metoda i događaja dodatka modal.

## 16.5 Dodatak affix

Affix je dodatak koji se koristi kada je potrebno fiksirati određeni element na veb strani. Ukoliko se uvodi individualno, potrebno je navođenje fajla `affix.js`. Fiksiranje se najčešće primenjuje nad listom opcija ili nad trakom za navigaciju i obezbeđuje im konstantan prikaz na veb strani bez obzira na skrolovanje. Element se može fiksirati upotrebom atributa ili upotrebom biblioteke `jQuery`.

### 16.5.1 Upotreba atributa za fiksiranje elemenata

Upotreba atributa za fiksiranje elemenata podrazumeva upotrebu atributa `data-spy="affix"`, koji se navodi na nivou tog elementa. Pored toga, poželjno je navođenje opcionih atributa `data-offset-top` i `data-offset-bottom`.

Vrednost atributa `data-offset-top` predstavlja broj piksela koji je potrebno skrolovati od početka strane da bi se dati element fiksirao na određenoj poziciji. Slično, vrednost atributa `data-offset-bottom` predstavlja broj piksela koji je potrebno skrolovati od kraja strane da bi se dati element fiksirao. Pre nego što bude objašnjeno kako se zadaje pozicija na kojoj će se fiksirati, razmotrimo kako dodatak `affix` funkcioniše u pozadini. U pozadini funkcioniše tako što smenjuje 3 klase: `affix`, `affix-top` i `affix-bottom`. Ove klase se automatski smenjuju nad fiksnim elementom, u zavisnosti od njegove pozicije tokom skrolovanja.

Ako je naveden atribut `data-offset-top`, elementu se prvo pridružuje klasa `affix-top`, a kada skrolovanjem dostigne vrednost zadatu tim atributom, pridružuje mu se klasa `affix`. Sa druge strane, ako je na nivou elementa zadat atribut `data-offset-bottom`, pridružuje mu se prvo klasa `affix`. U trenutku kada dostigne vrednost zadatu atributom, pridružuje mu se klasa `affix-bottom`.

Klasu `affix` odlikuje CSS svojstvo `position:fixed`; dok preostale dve nemaju te vrednosti. Kao i svaku drugu klasu, klasu `affix` je moguće nadograditi drugim svojstvima. Ono što je obavezno je zadavanje CSS svojstva `top`, kao što je prikazano u kodu 16.5.1.

```
<style> .affix{top:100px;} </style>
```

#### Kôd 16.5.1 – Zadavanje CSS svojstva *top* na nivou klase *affix*

Ovo svojstvo, kao i podrazumevano svojstvo *position:fixed*, će se primeniti na elementu onog trenutka kada nastupi klasa *affix*, odnosno kada atribut *data-offset-top* (ili *data-offset-bottom*) dostigne zadatu vrednost.

Na osnovu ovoga se zaključuje da se pozicija fiksnog elementa, nakon što atribut *data-offset-top*, (ili *data-offset-bottom*) dostigne zadatu vrednost, zadaje pomoću CSS svojstva *top* klase *affix*, koje predstavlja njegovu udaljenost od vrha strane.

### 16.5.2 Upotreba biblioteke *jQuery* za fiksiranje elemenata

Drugi način da se element fiksira je da se korišćenjem biblioteke *jQuery* pozove metod *affix()* nad tim elementom. Pozicija na kojoj se element fiksira je isto određena CSS svojstvom *top* klase *affix*. Broj piksela koji je potrebno skrolovati od početka, odnosno od kraja strane, da bi se element fiksirao, zadaje se kao parametar metoda opcijom *offset*.

### 16.5.3 Opcije dodatka *affix*

Opcije dodatka *affix* predstavljaju svojstva radnog okruženja Bootstrap koja se dodeljuju elementu koji se fiksira. Elektronske lekcije prikazuju upotrebu jedne opcije *affix* dodatka – opcije *offset*.

U slučaju kada je element fiksiran pomoću atributa *data-spy="affix"*, opcija *offset* se zadaje preko atributa *data-offset-top* i *data-offset-bottom* na nivou fiksnog elementa. Sa druge strane, u slučaju da se element fiksira pomoću biblioteke *jQuery*, opcija se zadaje kao parametar metoda *affix()* koji se poziva nad fiksnim elementom. Tako, na primer atributu *data-offset-top="100"* odgovara poziv metoda: *affix({offset: {top: 100} })*. Slično, atributu *data-offset-bottom="100"* odgovara *affix({offset: {bottom: 100} })*.

### 16.5.4 Fiksiranje trake za navigaciju

Kreiranje trake za navigaciju je detaljno objašnjeno u 11. lekciji. Prikažimo njeno fiksiranje na oba pomenuta načina.

Fiksiranje upotrebom atributa *data-spy="affix"* se postiže kada se na nivou elementa *nav* navede atribut *data-spy="affix"*. Kôd 16.5.2 prikazuje kreiranje trake za navigaciju koja postaje fiksna kada se postigne vrednost od 100 piksela skrolovanih od vrha strane.

```
<nav class="navbar navbar-inverse"
data-spy="affix" data-offset-top="100">
  <ul class="nav navbar-nav">
    <li><a href="#">Sekcija 1</a></li>
    <li><a href="#">Sekcija 2</a></li>
    <li><a href="#">Sekcija 3</a></li>
  </ul>
</nav>
```

#### Kôd 16.5.2 – Fiksiranje trake za navigaciju upotrebom atributa *data-spy="affix"*

U ovom slučaju, u zaglavlju stranice treba obezbediti kôd 16.5.3.

```
<style>
  .affix{ top:0px; width:100%; }
</style>
```

Kôd 16.5.3 – Zadavanje CSS svojstava na nivou klase *affix*

Identična traka za navigaciju se može fiksirati upotrebom biblioteke *jQuery* pomoću koda 16.5.4.

```
<nav class="navbar navbar-inverse" id="id1">
  <ul class="nav navbar-nav">
    <li><a href="#">Sekcija 1</a></li>
    <li><a href="#">Sekcija 2</a></li>
    <li><a href="#">Sekcija 3</a></li>
  </ul>
</nav>
<script>
$(document).ready(function(){ $('#id1').affix({offset: {top: 100} });
});
</script>
```

Kôd 16.5.4 – Fiksiranje trake za navigaciju pomoću biblioteke *jQuery*

I u ovom slučaju je obavezno zadati *top* svojstvo klase *affix*.

### 16.5.5 Događaji dodatka *affix*

Događaji dodatka *affix* su akcije koje se mogu realizovati isključivo kada je omogućeno fiksiranje nekog elementa.

Događaj dodatka *affix* se realizuje pozivom metoda *on* nad elementom koji će biti fiksiran. Kôd 16.5.5 prikazuje upotrebu događaja *affix.bs.affix* nad trakom za navigaciju, pod pretpostavkom da je ona kreirana u kodu 16.5.4.

```
<script>
$(document).ready(function(){
  $('#id1').affix({offset: {top: 100} });
  $('#id1').on('affix.bs.affix', function(){
    alert('Traka za navigaciju će uskoro biti fiksirana.');
```

Kôd 16.5.5 - Upotreba događaja *affix.bs.affix*

Ovaj događaj nastupa neposredno pre nego što element postane fiksiran, tj. neposredno pre nego što se klasa *affix-top* zameni klasom *affix*. Akcija koja se realizuje usled tog događaja u konkretnom primeru je pojavljivanje dijaloga prozora.

Ostali događaji dodatka *affix* se koriste analogno i prikazani su u tabeli 16.5.1.

Tabela 16.5.1

Događaj:	Opis:
affix.bs.affix	Događaj nastupa neposredno pre nego što element postane fiksiran, tj. neposredno pre nego što se klasa <i>affix-top</i> zameni klasom <i>affix</i> .
affixed.bs.affix	Događaj nastupa nakon što element postane fiksiran, tj. nakon što se klasa <i>affix-top</i> zameni klasom <i>affix</i> .
affix-top.bs.affix	Događaj nastupa neposredno pre nego što se element vrati u svoju originalnu poziciju, tj. neposredno pre nego što se klasa <i>affix</i> zameni klasom <i>affix-top</i> .
affixed-top.bs.affix	Događaj nastupa nakon što se element vrati u svoju originalnu poziciju, tj. nakon što se klasa <i>affix</i> zameni klasom <i>affix-top</i> .
affix-bottom.bs.affix	Događaj nastupa neposredno pre nego što se element vrati u svoju originalnu poziciju, tj. neposredno pre nego što se klasa <i>affix</i> zameni klasom <i>affix-bottom</i> .
affixed-bottom.bs.affix	Događaj nastupa nakon što se element vrati u svoju originalnu poziciju, tj. nakon što se klasa <i>affix</i> zameni klasom <i>affix-bottom</i> .

Na elektronskoj platformi *eŠkola vebe* dostupni su primeri upotrebe svih pomenutih klasa i događaja gde je još prikazana njegova upotreba nad vertikalnim menijem.

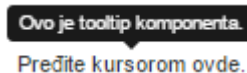
## 16.6 Dodatak tooltip

Tooltip je dodatak koji se koristi za prikazivanje komponente tooltip nad elementom. Ukoliko se uvodi individualno, potrebno je navođenje fajla *tooltip.js*.

Komponenta tooltip nekog elementa se prikazuje u vidu balončića prelaskom kursora preko tog elementa. Kreiranje komponente tooltip nad određenim elementom se realizuje upotrebom atributa ili upotrebom biblioteke *jQuery*.

### 16.6.1 Upotreba atributa pri kreiranju komponente tooltip

Jedan od načina da se kreira komponenta tooltip nad određenim elementom je upotrebom atributa *data-toggle="tooltip"* nad tim elementom, čime se ona aktivira. Sadržaj komponente tooltip se zadaje kao vrednost atributa *title* na nivou elementa koji je sadrži. Za potpunu funkcionalnost komponente tooltip dobijene na ovaj način, neophodno je korišćenje biblioteke *jQuery*. Korišćenjem biblioteke *jQuery* potrebno je pozvati metod *tooltip()* nad elementom koji sadrži tooltip komponentu. Slika 16.6.1 prikazuje komponentu tooltip nad linkom nastalu upotrebom atributa *data-toggle* pomoću koda 16.6.1.



Slika 16.6.1 – Komponenta tooltip

```
<span data-toggle="tooltip" id="id1" title="Ovo je tooltip komponenta.">Pređite kursom ovde.
</span>
<script>
$(document).ready(function(){ $('#id1').tooltip();
});
</script>
```

Kôd 16.6.1 – Kreiranje komponente tooltip upotrebom atributa *data-toggle*

## 16.6.2 Upotreba biblioteke *jQuery* pri kreiranju komponente tooltip

Drugi način da se kreira komponenta tooltip nad elementom je pozivanjem metoda *tooltip()* nad tim elementom, pri čemu se sadržaj komponente tooltip zadaje kao vrednost opcije *title* metoda *tooltip()*. Komponentu tooltip sa slike 16.6.1 je moguće kreirati na ovaj način pomoću koda 16.6.2.

```
<span id="id1">Predite kursom ovde.</span>
<script>
$(document).ready(function(){ $('#id1').tooltip({title:"Ovo je tooltip komponenta."});
});
</script>
```

*Kôd 16.6.2– Kreiranje komponente tooltip upotrebom biblioteke jQuery*

Funkcionalnost radnog okruženja Bootstrap u prikazivanju tooltip komponente se zasniva na opcijama, metodama i događajima dodatka popover, uvedenih pomoću biblioteke jQuery.

## 16.6.3 Opcije dodatka tooltip

Opcije dodatka tooltip predstavljaju svojstva radnog okruženja Bootstrap koja opisuju komponentu tooltip. U slučaju kada je komponenta tooltip kreirana preko atributa nad određenim elementom, sve opcije se zadaju kao atributi sa prefiksom „*data-*” na nivou tog elementa, osim opcije *title*, koja se navodi bez prefiksa „*data-*”. U slučaju kada je komponenta tooltip kreirana pomoću biblioteke *jQuery* nad nekim elementom, opcije se zadaju kao parametri metoda *tooltip()* koji se poziva nad tim elementom.

U tabeli 16.6.1 su prikazane opcije dodatka tooltip. Za svaku od njih je prikazan odgovarajući atribut, odgovarajući poziv metoda, i dat je opis vrednosti koje ta opcija prihvata.



Tabela 16.6.1 – Opcije tooltip dodatka

Opcija:	Opis:
title	Opcija se može koristiti kao atribut <i>title</i> ili u vidu metoda <i>tooltip({title:vrednost})</i> . Vrednost opcije određuje tekstualni sadržaj tooltip komponente.
animation	Opcija određuje da li se prikazuje efekat pri pojavljivanju i nestajanju komponente tooltip. Može se koristiti kao atribut <i>data-animation</i> ili u vidu metoda <i>tooltip({animation:vrednost})</i> . Dopusćene vrednosti su <i>true</i> , kojom se zadaje efekat i vrednost <i>false</i> , kojom se efekat onemogućava.
container	Opcija pridružuje komponentu tooltip elementu koji je određen vrednošću opcije. Opcija se može koristiti kao atribut <i>data-container</i> ili u vidu metoda <i>tooltip({container:vrednost})</i> .
delay	Određuje za koliko milisekundi se pojavljuje i povlači komponenta tooltip. Opcija se može koristiti kao atribut <i>data-delay</i> ili u vidu metoda <i>tooltip({delay:vrednost})</i> . Ukoliko se navodi u vidu metoda, moguće je zadati trenutak pojavljivanja ( <i>show</i> ) i trenutak povlačenja ( <i>hide</i> ) na sledeći način: <i>tooltip({delay:{show:500, hide:100}})</i> . Ukoliko se navodi preko atributa <i>data-delay</i> , vrednost opcije važi za oba trenutka.
html	Određuje da li komponenta tooltip može prihvatiti <i>HTML</i> elemente kao sadržaj. Opcija se može koristiti kao atribut <i>data-html</i> ili u vidu metoda <i>tooltip({html:vrednost})</i> . Dopusćene vrednosti su <i>true</i> i <i>false</i> .
placement	Određuje poziciju komponente tooltip. Može se koristiti kao atribut <i>data-placement</i> ili u vidu metoda <i>tooltip({placement:vrednost})</i> . Komponenta tooltip se podrazumevano pojavljuje iznad svog elementa ali se njena pozicija može promeniti upotrebom vrednosti: <ul style="list-style-type: none"> <li>• <i>top</i> (koja je pozicionira iznad)</li> <li>• <i>bottom</i> (ispod)</li> <li>• <i>left</i> (levo)</li> <li>• <i>right</i> (desno)</li> <li>• <i>auto</i></li> </ul> koje se navode kao vrednosti opcije <i>placement</i> . Vrednost <i>auto</i> dopušta pregledaču da sam odredi poziciju komponente tooltip. Ako je zadata vrednost <i>auto left</i> , to znači da će se komponenta tooltip pojaviti na levoj strani kad god je to moguće. Slično i za ostale.
trigger	Određuje način na koji se komponenta tooltip pojavljuje. Podrazumevano pojavljivanje je usled prelaska kursora preko elementa a moguća pojavljivanja su još usled levog i desnog klika na element. Opcija se može koristiti kao atribut <i>data-trigger</i> ili u vidu metoda <i>tooltip({trigger:vrednost})</i> . Dopusćene vrednosti opcije su: <ul style="list-style-type: none"> <li>• <i>click</i> (omogućava pojavljivanje nakon levog klika)</li> <li>• <i>hover</i> (pojavljivanje usled prelaska kursoram)</li> <li>• <i>focus</i> (pojavljivanje usled levog ili desnog klika) .</li> </ul>

## 16.6.4 Metodi dodatka tooltip

Metod dodatka tooltip predstavlja funkciju koja se poziva nad elementom nad kojim se pojavljuje komponenta tooltip.

Kôd 16.6.3 prikazuje upotrebu metoda `tooltip("show")` nad linkom nad kojim se prikazuje komponenta tooltip u trenutku pritiska na dugme. Ovaj metod omogućava prikazivanje komponente tooltip, u ovom slučaju pritiskom na dugme.

```
<a href="#" data-toggle="tooltip" title="Hej!">Nad ovim se prikazuje tooltip komponenta </a>
<button class="btn btn-info">Prikaži</button>
<script>
$(document).ready(function(){
    $(".btn-info").click(function(){
        $('[data-toggle='tooltip']").tooltip('show');
    });
});
</script>
```

Kôd 16.6.3 - Upotreba metoda tooltip("show")

Ostali metodi dodatka tooltip se koriste analogno i prikazani su u tabeli 16.6.2.

Tabela 16.6.2 – Metodi dodatka tooltip

Metod:	Opis:
tooltip(opcije)	Aktivira opcije koje se primenjuju nad komponentom tooltip.
tooltip("show")	Prikazuje komponentu tooltip.
tooltip("hide")	Sakriva komponentu tooltip.
tooltip("toggle")	Prikazuje komponentu tooltip, ako je bila sakrivena. Sakriva komponentu tooltip, ukoliko je bila prikazana.
tooltip("destroy")	Sakriva i deaktivira komponentu tooltip.

## 16.6.5 Događaji dodatka tooltip

Događaji dodatka tooltip su akcije koje se realizuju isključivo kada je omogućeno prikazivanje komponente tooltip. Zadaju se pozivom metoda `on` nad elementom nad kojim će se ona prikazivati. Prvi parametar tog metoda je naziv događaja, a drugi parametar je funkcija koja će se realizovati na veb stranici.

Kôd 16.6.4 prikazuje upotrebu događaja `show.bs.tooltip` nad linkom nad kojim se prikazuje komponenta tooltip pritiskom na dugme. Ovaj događaj zadaje akciju koja se realizuje neposredno pre nego što će komponenta tooltip biti prikazana.

```
<a href="#" data-toggle="tooltip" title="Hooray!">Nad ovim linkom će se pojaviti tooltip komponenta.</a>
<button type="button" class="btn btn-info">Pritisni</button>
<script>
$(document).ready(function(){
    $(".btn-info").click(function(){
        $('[data-toggle='tooltip']").tooltip('show'); });
    $('[data-toggle='tooltip']").on('show.bs.tooltip', function(){
        alert('Tooltip komponenta će uskoro biti prikazana.');
```

Kôd 16.6.4 - Upotreba događaja show.bs.tooltip

Ostali događaji dodatka tooltip se koriste analogno i prikazani su u tabeli 16.6.3.

Tabela 16.6.3 – Događaji dodatka tooltip

Događaj:	Opis:
show.bs.tooltip	Događaj nastupa neposredno pre nego što će komponenta tooltip biti prikazana.
shown.bs.tooltip	Događaj nastupa nakon što je komponenta tooltip prikazana.
hide.bs.tooltip	Događaj nastupa neposredno pre nego što će komponenta tooltip biti sakrivena.
hidden.bs.tooltip	Događaj nastupa nakon što je komponenta tooltip sakrivena.

Više informacija o dodatku tooltip radnog okruženja Bootstrap se može videti u [4].

## 16.7 Dodatak popover

Popover je dodatak koji se koristi za prikazivanje komponente popover nad elementom. Ukoliko se uvodi individualno, potrebno je navođenje fajla *popover.js* ali i fajla *tooltip.js*. Komponenta popover nekog elementa se prikazuje u vidu prozora tekstualnog sadržaja kada se klikne na taj element. Razlika između komponenta popover i tooltip je u tome što u popover može stati veći sadržaj. Komponenta popover se može kreirati upotrebom atributa ili upotrebom biblioteke *jQuery*.

### 16.7.1 Upotreba atributa pri kreiranju komponente popover

Kreiranje komponente popover upotrebom atributa podrazumeva korišćenje sledećih atributa:

- `data-toggle="popover"` (koji kreira komponentu)
- `data-content` (vrednost ovog atributa predstavlja sadržaj komponente)
- `title` (vrednost ovog atributa predstavlja zaglavlje komponente).

Navedeni atributi se koriste na nivou elementa koji sadrži komponentu popover. Funkcionalnost konkretne komponente popover se ostvaruje pomoću biblioteke *jQuery*, pozivom metoda *popover()* nad tom komponentom ili pozivom istog metoda nad svim komponentama popover na stranici. Kôd 16.7.1 prikazuje kreiranje komponente popover nad linkom i obezbeđuje funkcionalnost svih komponenta popover koje se nalaze na toj stranici.

```
<a href="#" data-toggle="popover" title="Zaglavlje" data-content="Tekstualni sadržaj.">
Kliknite
</a>
<script>
$(document).ready(function(){ $('[data-toggle="popover"]').popover();
});
</script>
```

Kôd 16.7.1 – Kreiranje komponente popover nad linkom upotrebom atributa `data-toggle`

## 16.7.2 Upotreba biblioteke *jQuery* pri kreiranju komponente popover

Kreiranje komponente popover upotrebom biblioteke *jQuery* se postiže pozivom metoda *popover()* sa obaveznim parametrima:

- *title* (čija vrednost predstavlja zaglavlje komponente popover)
- *content* (čija vrednost predstavlja sadržaj komponente popover).

Metod se poziva nad konkretnim elementom, kome se pristupa preko identifikatora ili na neki drugi način. Kôd 16.7.2 prikazuje kreiranje komponente popover nad linkom i obezbeđuje njenu funkcionalnost upotrebom biblioteke *jQuery*.

```
<a href="#" id="link">Kliknite</a>
<script>
$(document).ready(function(){
  $("#link").popover({title:"Zaglavlje",content:"Tekstualni sadržaj."});
});
</script>
```

*Kôd 16.7.2 - Kreiranje komponente popover upotrebom biblioteke jQuery*

Funkcionalnost radnog okruženja Bootstrap u prikazivanju popover komponente se zasniva na opcijama, metodama i događajima dodatka popover, uvedenih pomoću biblioteke *jQuery*.

## 16.7.3 Opcije dodatka popover

Opcije dodatka *popover* predstavljaju svojstva radnog okruženja Bootstrap koja opisuju komponentu popover. Kada je komponenta popover kreirana upotrebom atributa nad određenim elementom, sve opcije se zadaju kao atributi sa prefiksom „*data-*” na nivou tog elementa, osim opcije *title*, koja se navodi bez prefiksa „*data-*”. Kada je komponenta tooltip kreirana pomoću biblioteke *jQuery* nad nekim elementom, opcije se zadaju kao parametri metoda *popover()* koji se poziva nad tim elementom.

U tabeli 16.7.1 su prikazane opcije dodatka popover. Za svaku od njih je prikazan odgovarajući atribut, odgovarajući poziv metoda, i dat je opis vrednosti koje ta opcija prihvata.

Tabela 16.7.1 – Opcije dodatka popover

Opcija:	Opis:
title	Može koristiti kao atribut <i>title</i> ili u vidu metoda <i>popover({title:vrednost})</i> . Vrednost opcije određuje zaglavlje popover komponente.
content	Može se koristiti kao atribut <i>data-content</i> ili u vidu metoda <i>popover({content:vrednost})</i> . Vrednost opcije određuje sadržaj popover komponente.
animation	Određuje da li će se pojaviti efekat pri otvaranju, odnosno zatvaranju komponente popover. Može se koristiti kao atribut <i>data-animation</i> ili u vidu metoda <i>popover({animation:vrednost})</i> . Dopusštene vrednosti opcije su: <ul style="list-style-type: none"> <li>• <i>true</i> (sa efektom)</li> <li>• <i>false</i> (bez efekta).</li> </ul>
delay	Određuje broj milisekundi potrebnih da se komponenta popover otvori, odnosno zatvori. Može se upotrebiti kao atribut <i>data-delay</i> ili u vidu metoda <i>popover({delay:vrednost})</i> .
html	Određuje da li komponenta popover u svoj sadržaj može prihvatiti <i>HTML</i> element. Koristi se kao atribut <i>data-html</i> ili u vidu metoda <i>popover({html:vrednost})</i> . Dopusštene vrednosti su <i>true</i> i <i>false</i> . Napomena: <i>HTML</i> element može biti ubačen u sadržaj kao vrednost atributa <i>title</i> .
placement	Određuje poziciju komponente popover u odnosu na element uz koji se prikazuje. Može se koristiti kao atribut <i>data-placement</i> ili u vidu metoda <i>popover({placement:vrednost})</i> . Dopusštene vrednosti opcije su: <ul style="list-style-type: none"> <li>• <i>“top”</i> (iznad elementa)</li> <li>• <i>“bottom”</i> (ispod elementa)</li> <li>• <i>“left”</i> (sa leve strane elementa)</li> <li>• <i>“right”</i> (sa desne strane elementa)</li> <li>• <i>“auto”</i> (dopušta pregledaču da sam odredi poziciju).</li> </ul> Napomena: ako je vrednost jednaka <i>auto left</i> , to znači da će se komponenta popover pojaviti na levoj strani kad god je to moguće.
trigger	Određuje na koji način se komponenta popover uključuje, odnosno isključuje. Opcija se može koristiti kao atribut <i>data-trigger</i> ili u vidu metoda <i>popover({trigger:vrednost})</i> . Dopusštene vrednosti su: <ul style="list-style-type: none"> <li>• <i>“click”</i> (isključuje se, odnosno uključuje klikom na element)</li> <li>• <i>“hover”</i> (prelaskom kursora preko elementa)</li> <li>• <i>“focus”</i> (uključuje se kada element dobije fokus, a isključuje klikom van elementa)</li> <li>• <i>“manual”</i> (ručno).</li> </ul> Napomena: moguće je navesti više vrednosti razdvojene razmakom.

#### 16.7.4 Metodi dodatka popover

Metod dodatka popover predstavlja funkciju koja se može pozvati nad elementom nad kojim se pojavljuje komponenta popover.

Kôd 16.7.3 prikazuje upotrebu metoda *popover(“show”)* nad linkom nad kojim se prikazuje komponenta popover u trenutku pritiska na dugme. Ovaj metod omogućava prikazivanje komponente popover, u ovom slučaju pritiskom na dugme.

```

<a href="#" data-toggle="popover" title="Naslov" data-content="Sadržaj">Popover komponenta</a>
<button class="btn btn-warning">Prikazi</button>
<script>
$(document).ready(function(){
    $(".btn-warning").click(function(){
        $('[data-toggle='popover']').popover('show');
    });
});
</script>

```

Kôd 16.7.3 - Upotreba metoda popover("show")

Ostali metodi dodatka popover se koriste analogno i prikazani su u tabeli 16.7.2.

Tabela 16.7.2 – Metodi dodatka popover

Metod:	Opis:
popover(opcije)	Prikazuje komponentu popover koristeći opcije date u prethodnoj tabeli.
popover("show")	Prikazuje komponentu popover.
popover("hide")	Sakriva komponentu popover.
popover("toggle")	Prikazuje komponentu popover ukoliko je bila sakrivena, odnosno sakriva je ukoliko je bila prikazana.
popover("destroy")	Sakriva i deaktivira komponentu popover.

### 16.7.5 Događaji dodatka popover

Događaji dodatka popover su akcije koje se realizuju isključivo kada je omogućeno prikazivanje komponente popover. Zadaju se pozivom metoda *on* nad elementom nad kojim će se ona prikazivati.

Kôd 16.7.4 prikazuje upotrebu događaja *show.bs.popover* nad linkom nad kojim se prikazuje komponenta tooltip. Ovaj događaj zadaje akciju koja se realizuje neposredno pre nego što će komponenta tooltip biti prikazana. Primetimo da zadavanju događaja prethodi prikazivanje komponente popover, pritiskom na dugme, što je osnovni preduslov da se događaj realizuje.

```

<a href="#" data-toggle="popover" title="Naslov" data-content="Sadržaj"> Ovdje se pojavljuje popover komponenta </a>
<button class="btn btn-warning"> Kliknite </button>
<script>
$(document).ready(function(){
    $(".btn-warning").click(function(){
        $('[data-toggle='popover']').popover('toggle');
    });
    $('[data-toggle='popover']').on('show.bs.popover', function(){
        alert('Popover komponenta će uskoro biti prikazana.');

```

Kôd 16.7.4 - Upotreba događaja show.bs.popover

Ostali događaji dodatka popover se koriste analogno i prikazani su u tabeli 16.7.3.

Tabela 16.7.3 – Događaji dodatka popover

<b>Događaj:</b>	<b>Opis:</b>
show.bs.popover	Događa se neposredno pre nego što će komponenta popover biti prikazana.
shown.bs.popover	Događa se nakon što je komponenta popover prikazana.
hide.bs.popover	Događa se neposredno pre nego što će se komponenta popover sakriti.
hidden.bs.popover	Događa se nakon što je komponenta popover sakrivena.

Na elektronskoj platformi *eŠkola veba* su dostupni primeri upotrebe pomenutih opcija, metoda i događaja dodatka popover. Više informacija o dodatku popover radnog okruženja Bootstrap se može videti u [4].

## Zaključak

Elektronske lekcije sistematizuju osnovne koncepte radnog okruženja Bootstrap. Imaju za cilj da približe njegove mogućnosti i olakšaju proces njegovog usvajanja. Lekcije su javno dostupne i namenjene su osobama koje se prvi put sa njim susreću. Ovim putem svako sa osnovnim znanjem *HTML*-a i *CSS*-a može početi sa učenjem. Svaka lekcija je prilagođena početnicima i obrađena sa njihovog aspekta, što ide u prilog iskoristivom potencijalu ovih lekcija.

Sadržaj elektronskih lekcija pokriva osnovne dokumentovane informacije ovog radnog okruženja. Uređen je prema slobodnom metodičkom pristupu prema kom su slične funkcionalnosti radnog okruženja Bootstrap obrađene u istim lekcijama. Svaka lekcija obrađuje posebnu ili više sličnih komponenata opisujući mogućnosti i načine njihove upotrebe. Prikazivanje pojedinačnih komponenata iziskuje upotrebu specifičnih klasa, koje su deo bogate palete razvojnog okruženja Bootstrap. Elektronske lekcije prikazuju upotrebu ovih klasa na širokom spektru primera. Kroz obradu komponenata detaljno je objašnjena uloga korišćenih klasa i razmatrana njihova kombinacija sa drugim klasama. Upotreba svake klase je prikazana na velikom broju *HTML* elemenata, kako bi čitalac odmah dobio pregled kako se ta upotreba manifestuje na veb stranici. Veliki potencijal ovih lekcija se ogleda upravo u količini i raznolikosti primera. Pored klasa, pojedine komponente koriste metode i događaje, koji su deo radnog okruženja više zasnovanog na *JavaScript* tehnologiji. Elektronske lekcije prikazuju upotrebu svakog metoda i svakog događaja. Na ovaj način je jasno predstavljena funkcionalnost za svaki od njih, na svim obrađenim komponentama. Mnoštvo prikazanih klasa, metoda i događaja doprinose šarenolikosti ovih lekcija. Razumevanje savremenih tehnologija najčešće podrazumeva tumačenje strane literature, što otežava i usporava proces učenja. Posebna pogodnost ovih lekcija se ogleda u tome što su dostupne na srpskom jeziku. Elektronske lekcije radnog okruženja Bootstrap imaju za cilj da osposobe svakog čitaoca za efiksano kreiranje veb stranica. Bogatstvo i raznolikost dizajnerskih rešenja čini Bootstrap posebnim i neiscrpnim izvorom kvalitetnog veb dizajna a elektronske lekcije daju sistematičan pristup za njegovo korišćenje.



## Literatura

[1] Zvaničan sajt razvojnog okruženja *Bootstrap 3*:

<http://bootstrapdocs.com/v3.0.3/docs/>

[2] Sajt *w3schools* servisa:

<https://www.w3schools.com/bootstrap/default.asp>

[3] Sajt *tutorialspoint* servisa:

<https://www.tutorialspoint.com/bootstrap/>

[4] Sajt *w3resource* servisa:

<http://www.w3resource.com/twitter-bootstrap/tutorial.php>