

# Interakcija čovek-računar

# Organizatori kursa:

- ▶ Predavanja

- ▶ Zoran Ćirović, kab. 514

- ▶ Vežbe

- ▶ Martina Nikolić
  - ▶ Biljana Bojičić

# Provera znanja

- ▶ Kontinuirana...
- ▶ Predavanja:
  - ▶ Rad na predavanjima.
  - ▶ Testovi (zatvoreni oblik pitanja).
- ▶ Vežbe
  - ▶ Obavezne
  - ▶ Rad na vežbama
  - ▶ Projekat

## Literatura:

1. Materijali sa predavanja i
  2. Materijali sa vežbi
- ▶ “*Dizajniranje korisničkog interfejsa*”, Ben Shneiderman, Catherine Plaisant, CET 2006
  - ▶ “*Human-Computer Interaction*”, A.Dix, J Finlay, G Abowd, R Beale, Prentice Hall 2006
  - ▶ <http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/>

# Organizacija predavanja

- ▶ 3 časa predavanja nedeljno - po rasporedu
- ▶ Vreme na predavanju se deli na:
  - ▶ izučavanje gradiva koje obuhvata teorijska znanja
  - ▶ Izučavanje alata koji obezbeđuju predznanje za rad na vežbama i sticanje praktičnih veština iz određenih oblasti
  - ▶ Posebne prezentacije iz nekih aktuelnih oblasti



Pitanja?

# Upotreba JS biblioteka u projektovanju veb aplikacija

JQuery

# Uvod

- ▶ JQuery je JS biblioteka. Mala, jednostavna za upotrebu, istovremeno brza pri radu. JQuery olakšava kreiranje JS koda:
  - ▶ rukovanje događajima,
  - ▶ kreiranje animacija,
  - ▶ **asinhroni rad.**
- ▶ jQuery biblioteka:
  - ▶ uživa podršku velikog broja programera okupljenih u jQuery zajednicu (blogovi, uputstva, knjige)
  - ▶ OpenSource - dostupan izvorni kod potpuno besplatno
  - ▶ Podržan od strane velikih kompanija: Google, Microsoft, IBM, Amazon...
  - ▶ Lako je proširiv



# Uključivanje jQuery biblioteke

- ▶ Kako bi se koristila jQuery biblioteka mora se uključiti na mestu gde se referiše spoljašnji JavaScript kod u projektu.
- ▶ `<script src="putanja"> </script>`
- ▶ To se radi na dva načina:
  - ▶ 1. Tako što se iskoristi biblioteka koja je već hostovana negde na Internetu - jQuery CDN (engl. Content Delivery Network).  
**putanja =**
    - ▶ <https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js> // Google
    - ▶ <https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.3.1.min.js> // Microsoft
  - ▶ 2. Preuzeti verziju biblioteke sa sajta [www.jquery.com](http://www.jquery.com) i postaviti je na web serveru ili lokalnom fajl sistemu. A onda, slično kao u prethodnom slučaju, navesti putanju do fajla gde je skinuti fajl  
**putanja =**
    - ▶ `./jQuery/jquery-3.3.1.min.js` // Lokalno u folderu jQuery

# Minimalna verzija

- ▶ Umanjena jQuery biblioteka znatno otežava eventualno eventualnu izmenu ili ispravku koda. Zato je preporuka da se u toku razvoja koristi neumanjenu verziju svih skripti, a da se pri postavljanju na sajt tj. u produkciju koriste minimalne veličine fajlova.

## Početak - jQuery()

- ▶ Oznaka **\$()**, predstavlja alias za **jQuery()** funkciju. Ova funkcija vraća objekat kojim se mogu prihvatiti DOM elementi HTML dokumenta, po redosledu po kome su definisani u dokumentu a koji odgovaraju prosleđenom selektoru.
- ▶ Vraćeni objekat ove funkcije poseduje veliki broj korisnih unapred definisanih metoda. Na primer jQuery() funkcija se može koristiti:
  - ▶ Za selekciju i obavljanje DOM elemenata nad kojima treba izvršiti neku operaciju;
  - ▶ Za korišćenje globalnih funkcija;
  - ▶ Za kreiranje novih DOM elemenata u HTML jeziku;
  - ▶ Asinhroni rad sa udaljenim servisima.

## Primeri selektovanja - adekvatni CSS selektorima

- ▶ `$(this).hide()` // skriva tekući element.
- ▶ `$("p").hide()` // skriva svi `<p>` elemente - `getElementsByTagName`.
- ▶ `$(".test").hide()` // elementi klase `class="test"` - `getElementsByClassName`.
- ▶ `$("#test").hide()` // element čiji je `id="test"` - `getElementById`.
- ▶ `$("[name='lozinka']").hide()` // element naziva `name="test"` - `getElementsByName`.

## Učitana stranica kao uslov

- ▶ Pri korišćenju JS funkcija u radu sa elementima dokumenta važno je da taj rad usledj pošto su učitani svi DOM elementi stranice u potpunosti, jer bi u suprotnom moglo da se primeni operacija nad elementom koji još nije učitao.
- ▶ Može se koristiti događaj `onLoad` kako bi omogućio izvršavanje tek nakon što se stranica u potpunosti učita. Sintaksa naredbe je obično sledeća:

```
$(fja)
$(document).ready(fja)
$("document").ready(fja)
$("img").ready(fja)
$().ready(fja)
```

```
$(document)
.ready(function () {
    // metode ...
});

// skraceno ...
$(function () {
    // metode ...
});
```

# Primer 1

```
▶ // 1....
▶ window.onload = promenaBoje;
▶ function promenaBoje() {
▶     $("h2").css("color", "red");
▶ }

▶ // 2....
▶ $(promenaBoje);
▶ function promenaBoje() {
▶     $("h2").css("color", "red");
▶ }

▶ // 3....
▶ $(function () {
▶     $("h2").css("color", "red");
▶ });
```

# Primeri selektora

- ▶ `$("*")` - svi elementi
- ▶ `$(this)` - trenutni HTML element
- ▶ `$("#divID")` - element za `id = "divID"`
- ▶ `$(".naglasen")` - `<p>` elementi koji imaju klasu "naglasen"
- ▶ `$(p:first)` - prvi `<p>` element
- ▶ `$(ul li:first-child)` - prvi `<li>` element svakog `<ul>` elementa
- ▶ `$("[href]")` - svi elementi sa href atributom
- ▶ `$(a[href!='_xyz'])` - svi `<a>` elementi kod kojih je atribut href različit od "\_ xyz"
- ▶ `$(":button")` - svi `<button>` elementi i `<input>` elemente koji imaju `type="button"`
- ▶ `$(tr:even)` - svi parni `<tr>` elementi

# Ulančavanje

- ▶ Karakteristika jQuery metoda jeste da povratna vrednost vraća isti vrstu objekata kao objekti nad kojim se funkcija izvršavala. Na taj način moguće nadovezati drugu funkciju koja treba da se izvrši nad istom grupom elemenata.
- ▶ Na primer, ukoliko želimo da dodamo novu klasu removed svakom od elemenata, pored toga što ćemo ih sakriti, napisali bismo sledeće:

Stilovi:

```
.stl1{  
    color: red;  
}  
.stl2 {  
    color: gray;  
}
```

HTML: `<h3 class="stl1">Naslov</h3>`

JS:

```
$(function () {  
    $("h3.stl1")  
        .addClass("stl2")  
        .removeClass("stl1");  
});
```

## Kontekсни upiti / drugi argument

- ▶ Ukoliko se pri pisanju upita, navede drugi argument koji je jQuery objekat, pretraga se vrši nad sadržajem tog objekta, a ne nad celim dokumentom. Na primer:
- ▶ `$div1 = $("div.vazno"); // div elementi sa klasom .vazno`
- ▶ `$ul1 = $(".smerovi", $div1); // elementi klase .smerovi, ul elementi, u nadjenom div1 objektu`
- ▶ `$li = $("li", $ul1); // li stavke u ul1 objektu`

# Događaji

- ▶ jQuery poseduje predefinisane metode za određene događaje. Na primer:

Događaji miša	Događaji tastature	Događaji forme	Document/Window događaji
click	keypress	submit	load
dblclick	keydown	change	resize
hover	keyup	focus	scroll
mouseenter, mouseleave, mouseout		blur	unload



## Primer 2:

// 1. Pomocu metode dogadjaja

```
$("#li").dblclick(function () {  
    $(this).hide();  
});
```

// hover() metod ima 2 funkcije i kombinacija je mouseenter() i mouseleave() metoda.

// Prva funkcija se izvršava kada se miš naše iznad elementa a druga kada miš napušta element.

```
$("#h3").hover(  
    function () {  
        alert("Nalazite se iznad h3 elementa!");  
    },  
    function () {  
        alert("Napuštanje elementa");  
    }  
);
```

// 2. Pomocu metode on()

// Metoda on() dodaje jednu ili više funkcija događaja za selektovane elemente.

// Dodajemo klik događaja za svaki < li > element:

```
$("#li").on({  
    mouseenter: function () {  
        $(this).css("background-color", "lightgreen");  
    },  
    mouseleave: function () {  
        $(this).css("background-color", "lightyellow");  
    },  
    click: function () {  
        $(this).css("background-color", "orange");  
    }  
});
```

## Uključivanje događaja

- ▶ Ako se poziva ista funkcija za više događaja, kod se još pojednostavljuje:
- ▶ `var brojac = 0;`
- ▶ `$("#div1").on("mouseenter mouseleave click", function () {`
- ▶  `$("#div2").text(brojac++);`
- ▶ `});`

## Isključivanje događaja

- ▶ Kao što se događaji mogu dodati takođe je moguće i isključiti neke već dodate događaje:
  - ▶ Isključivanje metode poruka vezane za događaj `mouseenter`
  - ▶ `$("#div1").off({"mouseenter", poruka });`
  - ▶ Isključivanje svih metoda za događaj `mouseenter`
  - ▶ `$("#div1").off({"mouseenter"});`

# Sadržaj elementa

- ▶ Tri metode za manipulaciju sadržajem DOM elemenata su:
  - ▶ `text()` - postavlja ili vraća tekstualni sadržaj
  - ▶ `html()` - postavlja ili vraća sadržaj uključujući HTML markup
  - ▶ `val()` - postavlja ili vraća vrednost form polja.

```
<p>Ime: <input type="text" id="ime" value="Petar P"></p>
<p id="testParagraf">ovo je <i>iskoseno</i> u ovom
paragrafu.</p>
```

```
<button id="btn0">cita val</button>
<button id="btn1">cita Text</button>
<button id="btn2">cita HTML</button> <br/>
<button id="btn3">postavlja val</button>
<button id="btn4">postavlja Text</button>
<button id="btn5">postavlja HTML</button>
```

```
$("#btn0").click(function () {
    alert("Value: " + $("#ime").val());
});
$("#btn1").click(function () {
    alert("Text: " + $("#testParagraf").text());
});
$("#btn2").click(function () {
    alert("HTML: " + $("#testParagraf").html());
});
```

```
$("#btn3").click(function () {
    $("#ime").val("Unesi ime");
});
$("#btn4").click(function () {
    $("#testParagraf").text("tekstualni sadrzaj");
});
$("#btn5").click(function () {
    $("#testParagraf").html("<b>html</b> sadrzaj");
});
```

# Izmena sadržaja

- ▶ Metode za izmenu sadržaja elementa su:
- ▶ `append()` - dodaje sadržaj na kraj postojećeg
- ▶ `prepend()` - ubacuje sadržaj na početak postojećeg
- ▶ `after()` - ubacuje sadržaj nakon selektovanog elementa
- ▶ `before()` - ubacuje sadržaj ispred sel. elementa.

```
$("#li").append("Dodato na kraj.");  
$("#li").prepend("Dodato ispred.");
```

```
function dinamicBtns() {  
    var btn1 = "<button id='btn1'>Prvo</button>";  
    var btn2 = "<button id='btn2'>Drugo</button>";  
    var btn3 = document.createElement("button");  
    btn3.innerHTML = "Treće";  
    $("body").append(btn1, btn2, btn3);  
}
```

## Izbacivanje elementa

- ▶ Metod `remove()` izbacuje selektovane elemente i odgovarajuće deca elemente.
- ▶ `$("#div1").remove();`

## Pražnjenje elementa

- ▶ Metod `empty()` vrši pražnjenje tj izbacivanje dece elemenata nad selektovanim elementima.
- ▶ `$("#div1").empty();`

## CSS klase

- ▶ Dodavanje i izbacivanje CSS klase:
- ▶ `$("#button").click(function () {`
- ▶  `$("#li").addClass("zeleno");`
- ▶  `$("#li, p").removeClass("standardno");`
- ▶ `});`

## Toglovanje CSS klase

- ▶ Često je potrebno obezbediti naizmenično dodavanja odnosno izbacivanje iste CSS klase. Na primer:
- ▶ `$("#button").click(function () {`
- ▶  `$("#li, p").toggleClass("zeleno");`
- ▶ `});`

## Čitanje vrednosti svojstva

- ▶ Čitanje određenog svojstva koje je važeće za element:
- ▶ `alert("Background color = " + $("#body").css("background-color"));`

## Definisanje svojstava

- ▶ Neposredno dodavanje svojstava selektovanim elementima se takođe vrši primenom funkcije `css()`. Na primer:
- ▶ 

```
$("#h3").css({  
  "background-color": "lightgreen",  
  "color": "darkgreen",  
  "font-size": "200%"  
});
```

## Funkcije koje koriste DOM stablo:

- ▶ 

```
$("#span").parent(); // roditeljski el.
```
- ▶ 

```
$("#span").parents(); // svi rodit. el.
```
- ▶ 

```
$("#span").parents("div"); // svi rod. el koji su div
```
- ▶ 

```
$("#span").parentsUntil("div"); // svi rod. el do div
```



# BOOTSTRAP

Verzija 4



# Šta je *Bootstrap*?

Bootstrap je jedno kompletno, moćno, u praksi često korišćeno okruženje za brzu, laku izradu „mobile first“ web orijentisanih aplikacija. Okruženje je zasnovano na primeni HTML, CSS odnosno JavaScript-a.

## Istorija

Bootstrap su razvili Mark Otto i Jacob Thornton za **Twitter**. Objavljen je kao projekat otvorenog koda u avgustu 2011 na repozitorijumu **GitHub**.

# Zašto baš Bootstrap?

- Pristup „Mobile first“:

Od verzije Bootstrap 3 okruženje sadrži stil kojim se projektuje prvo sajt za mobilne uređaje, **engl. Mobile first**, i to kroz celu biblioteku Bootstrap-a.

- Podrška web čitača:

Svi popularni web čitači podržavaju ovo okruženje.

- Lako za početnike:



Sa osnovnim znanjem iz HTML i CSS može se naučiti i uspešno koristiti. Oficijelni sajt za Bootstrapa sadrži veoma dobru dokumentaciju.

- Prilagodljivi dizajn:

Bootstrap dizajn je prilagodljiv (pripadajući CSS se podešava da sajt bude prilagodljiv za Desktop, Tabelet i Mobilne uređaje).

# Šta okruženje Bootstrap uključuje?

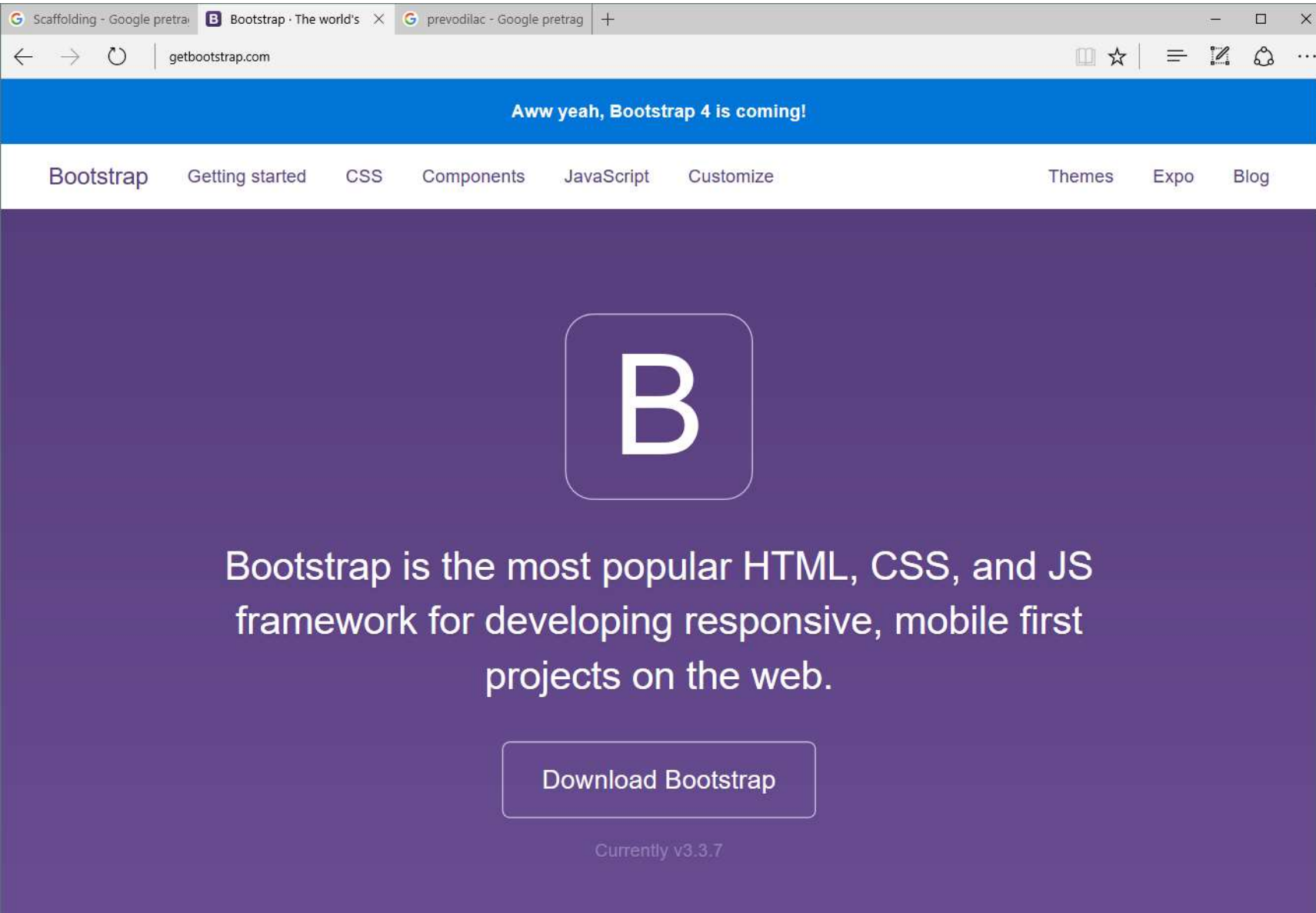
- ▶ **Struktura:** Bootstrap koristi osnovnu strukturu zasnovanu na mrežastom sistemu tzv. Grid struktura. Osim mreže u osnovnoj strukturi nudi stilizaciju linkova i pozadine.
- ▶ **CSS:** Bootstrap sadrži CSS podešavanja od osnova sa globalnim CSS podešavanjima, stilizacijom osnovnih HTML elemenata i dodacima na osnovni izgled koja su izvedena na osnovu proširenja osnovnih klasa.
- ▶ **Komponente:** Bootstrap sadrži na desetine komponenata namenjenih za višestuko korišćenje. Ove komponente omogućavaju na primer sopstvenu ikonografiju, padajuće liste, navigaciju, pomoćne prozore (engl. alerte), promene na osnovu pozicije miša itd.

## Šta okruženje Bootstrap uključuje?

- ▶ **JavaScript Plugins**: Bootstrap sadrži i na desetina posebnih komponenata iz biblioteke jQuery. Lako se uključuju sve ili jedna po jedna.
- ▶ **Customize**: Prilagođenje Bootstrap komponenata je moguće kao i jQuery plugin-ova i tako dobijanje sopstvenih komponenata.

## Download Bootstrap

- ▶ Poslednja verzija Bootstrap-a može se dobiti na lokaciji: <http://getbootstrap.com/>.



# File struktura

## ► PRECOMPILED BOOTSTRAP

Ako se koristi kompajlirana verzija Bootstrapa, vrši se najpre preuzimanje ZIP fajla, a zatim ekstrakcija svih fajlova. Dobija se struktura fajlova/direktorijuma kao na slici:

```
bootstrap/  
├── css/  
│   ├── bootstrap.css  
│   ├── bootstrap.css.map  
│   ├── bootstrap.min.css  
│   ├── bootstrap.min.css.map  
│   ├── bootstrap-grid.css  
│   ├── bootstrap-grid.css.map  
│   ├── bootstrap-grid.min.css  
│   ├── bootstrap-grid.min.css.map  
│   ├── bootstrap-reboot.css  
│   ├── bootstrap-reboot.css.map  
│   ├── bootstrap-reboot.min.css  
│   └── bootstrap-reboot.min.css.map  
└── js/  
    ├── bootstrap.bundle.js  
    ├── bootstrap.bundle.min.js  
    ├── bootstrap.js  
    └── bootstrap.min.js
```

Kao što se vidi, postoje kompajlirane verzije CSS odnosno JS (bootstrap.\*), kao i umanjene verzije CSS odnosno JS (bootstrap.min.\*).

## ▶ BOOTSTRAP SOURCE CODE

- ▶ Ako se preuzme izvorni kod okruženja Bootstrap tada je struktura fajlova kao na sledećoj slici:

```
bootstrap/  
├── dist/  
│   ├── css/  
│   └── js/  
├── docs/  
│   └── examples/  
├── js/  
└── scss/
```

- ▶ Fajlovi u folderima **less/**, **js/**, odnosno **fonts/** su izvorni kod za Bootstrap CSS, JS, odnosno icon fontove respektivno.
- ▶ Folder **dist/** sadrži sve što je navedeno u prekompajliranoj verziji koja je već opisana.
- ▶ Folderi **docs-assets/**, **examples/**, odnosno **\*.html** sadrže Bootstrap dokumentaciju.

# Package managers

- Moguće je priključiti izvorne datoteke Bootstrap-a u skoro svaki projekat sa nekim od najpopularnijih menadžera paketa.
- Bez obzira na menadžer paketa, Bootstrap će zahtevati kompajler Sass i Autoprefixer za podešavanje.

## **npm**

- Instalirajte Bootstrap u svoje aplikacije pomoću Node.js paketa:
- `npm install bootstrap --save`
- `npm install jquery --save`

## **NuGet**

- Ako razvijate .NET aplikacije takođe možete instalirati Bootstrap's CSS ili Sass i JavaScript koristeći NuGet:
- `Install-Package bootstrap`
- `Install-Package bootstrap.sass`



# HTML šablon za preuzeti Bootstrap:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Bootstrap 101 Template</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Bootstrap -->
    <link href="css/bootstrap.min.css" rel="stylesheet">

  </head>
  <body>
    <h1>Hello, world!</h1>
    <!-- jQuery -->
    <script src="https://code.jquery.com/jquery.js">
    </script>
    <!-- Include all compiled plugins -->
    <script src="js/bootstrap.min.js">
    </script>

  </body>
</html>
```

```
<!--[if lt IE 9]>
  <script src="https://oss.maxcdn.com/
    libs/html5shiv/3.7.0/
      html5shiv.js">
  </script>
  <script src="https://oss.maxcdn.com/
    libs/respond.js/1.3.0/respond.min.js">
  </script>
<![endif]-->
```

# HTML šablon za linkovani Bootstrap - 1

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet"
  integrity="sha384-Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
  crossorigin="anonymous">
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
  JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>
</head><body>

</body>
</html>
```

# HTML šablon za linkovani Bootstrap - 2

```
<!doctype html>
<html lang="en">
<head>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aowXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
  <title>Šablon BS!</title>
</head>
<body>
  <h1>Šablonska stranica</h1>
  <!-- Optional JavaScript -->
  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>
</body>
</html>
```

# HTML šablon za linkovani Bootstrap - OLD:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>naslov stranice</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
</head>
<body>

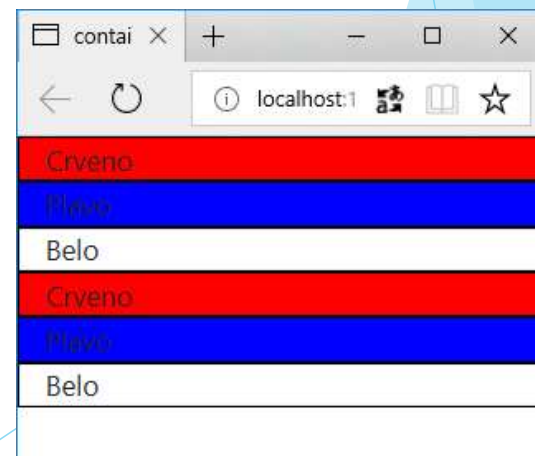
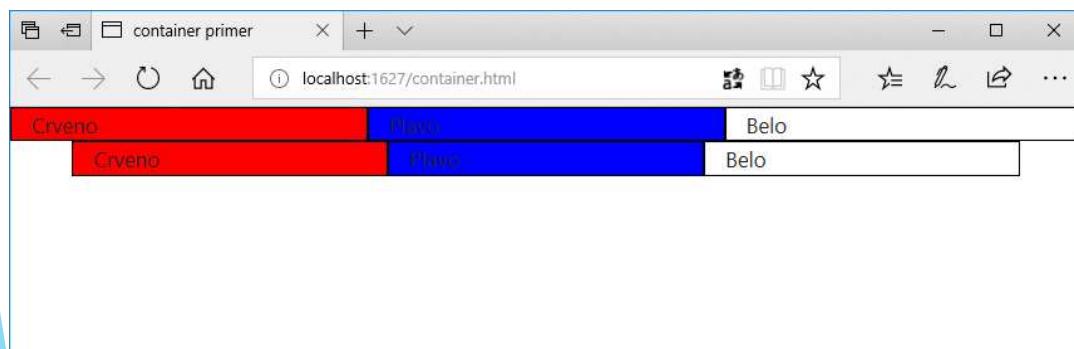
</body>
</html>
```

# Kontejneri

- ▶ Kontejneri su osnovni element drugih elemenata u Bootstrap-u. Kontejneri se zahtevaju prilikom korišćenja standardnog mrežnog (engl. grid) sistema. Može se izabrati kontejner sa fiksnom širinom (što znači da ima promene maksimalne samo širine na tačkama prekida) ili kontejner fluidne širine, što znači da neprekidno menja širini i uvek je na 100% širine.
- ▶ Kontejneri mogu biti ugneženi. Ipak, većini rasporeda nije potreban ugrađeni kontejner.



```
<div class="container-fluid">  
  <div class="row">...</div>  
</div>  
<div class="container">  
  <div class="row">...</div>  
</div>
```



# Mrežasta struktura

Osnovna struktura stranice zasnovane na Bootstrap sistemu uključuje prilagodljivu fluidnu mrežu sastavljenu od **maksimalno 12 kolona**. Sistem uključuje predefinisane klase za lako definisanje izgleda.

Mrežasti sistem se koristi za kreiranje vizuelnog prikaza zasnovanog na redovima i kolonama u kojima je smešten sadržaj za prikaz.

- ▶ Redovi (rows) se koriste za kreiranje horizontalne grupe kolona.
- ▶ Kolone (cols) čuvaju sadržaj koji se prikazuje. Kolone su sadržane u redovima.
- ▶ Formatiranje (poravnavanje, padding) redova se obavlja u `.container` klasi ako je definisane širine ili u `.container-fluid` (pune širine).
- ▶ Mrežasta struktura je predefinisana postojanjem klasa. Tako klasa `.row` ili `.col-4` predstavljaju predefinisane klase za red odnosno kolonu širine 4 od 12 jedinica. Međutim, koristeći određenu sintaksu mogu se dobiti i drugačiji prikazi.
- ▶ Kolone sadrže u sebi i rastojanje između kolona preko `padding-a`.

```
<div class="container">
  <div class="row">
    <div class="col-??-*"></div>
    <div class="col-??-*"></div>
  </div>
  <div class="row">
    <div class="col-??-*"></div>
    <div class="col-??-*"></div>
    <div class="col-??-*"></div>
  </div>
  <div class="row">
    ...
  </div>
</div>
```

# Media upiti

Na osnovu širine ekrana selektivno se primenjuju stilovi.....

Bootstrap je **mobile-first** stilova. Zato se prekidne tačke uglavnom zasnivaju na minimalnim širinama pogleda i omogućavaju da se uvećavaju elementi prilikom promene prikaza.

```
/* (tablets, 768px i širi) */
```

```
@media (min-width: @screen-sm-min) { ... }
```

```
/* (desktops, 992px i širi) */
```

```
@media (min-width: @screen-md-min) { ... }
```

```
/* (large desktops, 1200px i širi) */
```

```
@media (min-width: @screen-lg-min) { ... }
```

```
@media (min-width: @screen-sm-min) and (max-width: @screen-sm-max) { ... }
```

```
@media (min-width: @screen-md-min) and (max-width: @screen-md-max) { ... }
```

# Klase za mrežastu strukturu

- ▶ Koriste se 4 klase:
- ▶ **(xs)** - za telefone. Ova klasa je podrazumevana. U verziji 4 se ne koristi.
- ▶ **sm** - za tablete
- ▶ **md** - za desktop računare
- ▶ **lg** - za velike desktop računare
- ▶ **xl** - za jako široke ekrane
- ▶ **Napomena:** Svaka klasa koja se primenjuje odnosi se ne samo za definisanu veličinu već i za veličine ekrane koje su veće od te. Na primer ako je definisana klasa xs (extra small) a nije definisana sm (small) onda se klasa xs primenjuje za ekrane na koje bi se odnosila klasa sm.



# Promena iz vertikalnog u horizontalni prikaz za različite veličine ekrana - Primer1

```
<div class="container-fluid">
  <div class="row">
    <div id="crveno" class="col-sm-4">
      Crveno
    </div>
    <div id="plavo" class="col-sm-4">
      Plavo
    </div>
    <div id="belo" class="col-sm-4">
      Belo
    </div>
  </div>
</div>
```

```
<style>
  [class]="col"
  {
    border: solid 1px black;
  }
  #crveno{
    background-color:red;
  }
  #plavo{
    background-color:blue;
  }
  #belo{
    background-color:white;
  }
</style>
```



Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	Extra large ≥1200px	
Max širina kontejnera	None (auto)	540px	720px	960px	1140px
Prefiks	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
Broj kolona	12				
Gutter širina	30px (15px sa svake strane kolone)				

**col-xs-\*** je odbačen u verziji Bootstrap 4, a umesto istog koristi se **col-\***

Kolone imaju horizontalni peding koji se naziva **guter** između pojedinih kolona. Moguće je ukoniti margine između redova i peding u kolonama sa klasom **.no-gutters** uz **.row**.

## Prateće objašnjenje za Primer1

- ▶ Tri kolone. Svaka kolona širine 4 jedinice. Ukupan broj jedinica po kolonama za jedan red je 12. `class="col-sm-4"`
- ▶ Klasa `sm` je odabrana tako da se 3 kolone vide u prikazu predviđenom za desktop ekrane i šire, dok za uže prikaz će biti promenjen tako da svaka stavka bude jedna ispod druge.

# Više prelomnih tačaka

- ▶ Kada je potrebno dodati više prelomnih tačaka, odnosno poseban prikaz za mobilne, table, dekstop ... Onda se dodaje više klasa.

xs - 3:4:5

sm - 4:4:4

md - 5:4:3

## Primer 2

```
<div class="container-fluid">
  <div class="row">
    <div id="crveno" class="col-3 col-sm-4 col-md-5">
      Crveno
    </div>
    <div id="plavo" class="col-4 col-sm-4 col-md-4">
      Plavo
    </div>
    <div id="belo" class="col-5 col-sm-4 col-md-3">
      Belo
    </div>
  </div>
</div>
```

- ▶ Kolike su širine kolona za različite širine ekrana?

# Offset klase

- ▶ U mrežastom prikazu moguće je i preskakati određene kolone (ćelije). Za to se koriste tzv offset klase. Sintaksa je:
- ▶ `col-??-offset-*`
- ▶ gdje `??` predstavlja oznaku `sm`, `md`, `lg`, `xl` a `*` broj kolona u jedinicama 1/12.

## Primer 2a

```
<div class="row">
  <div class="col-md-4">.col-md-4</div >
  <div class="col-md-4 col-md-offset-4">.col-md-4 .col-md-offset-4</div >
</div >

<div class="row">
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div >
  <div class="col-md-3 col-md-offset-3">.col-md-3 .col-md-offset-3</div >
</div >

<div class="row">
  <div class="col-md-6 col-md-offset-3">.col-md-6 .col-md-offset-3</div>
</div >
```

Primer2a.html

.col-md-4		.col-md-4 .col-md-offset-4	
	.col-md-3 .col-md-offset-3		.col-md-3 .col-md-offset-3
	.col-md-6 .col-md-offset-3		

# Ugnježdene kolone

- ▶ Ugnježdavanje sadržaja u već postojeću mrežu vrši se dodavanjem novog reda tj odeljka klase `.row` i postavljanjem `.col-??-*` kolona unutar postojeće `.col-??-*` kolone.
- ▶ Ugnježđenje redova podrazumeva uključivanje kolona do 12 ili manje.

```
<div class="row">
  <div style="background-color: green" class="col-sm-3">
    Roditeljski red: .col-sm-3
  </div>
  <div style="background-color: yellow" class="col-sm-9">
    Roditeljski red: .col-sm-9
    <div class="row">
      <div class="col-8 col-sm-6">
        Dete 1: .col-8 .col-sm-6
      </div>
      <div class="col-4 col-sm-6">
        Dete 2: .col-4 .col-sm-6
      </div>
    </div>
  </div>
</div>
```

# Zadatak

- ▶ Napisati kod koji odgovara primeru na slici:



# Promena redosleda prikaza

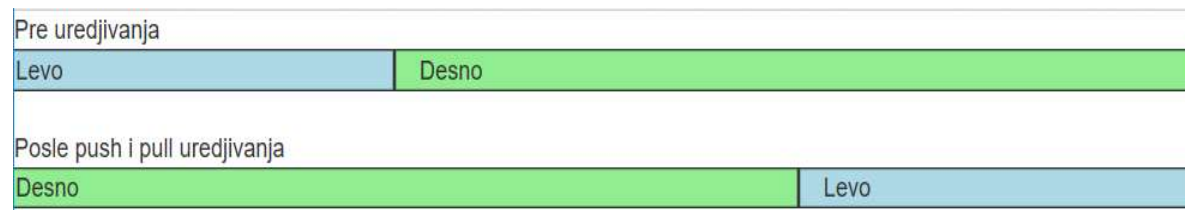
- ▶ Ako su kolone zapisane u jednom redosledu moguće ih je prikazati u drugačijem. Ovo se postiže primenom klasa `.col-??-push-*` odnosno `.col-??-pull-*` gde je `*` u opsegu od 1 do 11. Ovo je u verziji BS 3.
- ▶ U primeru sa desne strane vrši se zamena redosleda ovih kolona koristeći pomenuta svojstva.

Verzija BS 3

```
<div class="row">
  <p>Pre uredjivanja</p>
  <div class="col-md-3" style="background-color: lightblue;
    border: solid 1px">
    Levo
  </div>
  <div class="col-md-6" style="background-color: lightgreen;
    border: solid 1px">
    Desno
  </div>
</div><br>
<div class="row">
  <p>Posle push i pull uredjivanja</p>
  <div class="col-md-3 col-md-push-6" style="background-color: lightblue;
    border: solid 1px">
    Levo
  </div>
  <div class="col-md-6 col-md-pull-3" style="background-color: lightgreen;
    border: solid 1px">
    Desno
  </div>
</div><br>
```



- ▶ Međtun, sa novom verzijom Bootstrap je uveo pojednostavljenje promene redosleda primenom novih klasa **.order**. Tako bi u novoj verziji preuređivanje bilo:
- ▶ `<div class="col-md-3 order-2">`
- ▶ `<div class="col-md-6 order-1">`



# Svojstvo prikaza \*\*\*\*

- ▶ BS obezbeđuje sopstvene klase za promene CSS svojstva display. Bez obzira što se radi o prilagodljivom dizajnu i rešenju tipa mobile-first obezbeđuju se klase za sve vrste prikaza.
- ▶ Sintaksa za definisanje prikaza počinje sa d- a sledi tačka prekida i vrednost:
- ▶ `.d-{tačkaPrekida}-{val}` : ovo važi za sve klase osim za xs gde se tačka prekida ne navodi.
- ▶ `val` može imati sledeće moguće vrednosti:
  - `none`
  - `inline`
  - `inline-block`
  - `block`
  - `table`
  - `table-cell`
  - `table-row`
  - `flex`
  - `inline-flex`

# Prikaz i skrivanje sadržaja

- ▶ Klase koje se koriste za prikaz ili skrivanje sadržaja u zavisnosti od širine ekrana (tj uređaja) prikazane su u tabeli. Koristi se neka od kombinacija.
- ▶ Ovo je važeće za verziju 3 BS.

Classes	Extra small (<768px)	Small (≥768px)	Medium (≥992px)	Large (≥1200px)
.visible-xs-*	Visible	Hidden	Hidden	Hidden
.visible-sm-*	Hidden	Visible	Hidden	Hidden
.visible-md-*	Hidden	Hidden	Visible	Hidden
.visible-lg-*	Hidden	Hidden	Hidden	Visible
.hidden-xs	Hidden	Visible	Visible	Visible
.hidden-sm	Visible	Hidden	Visible	Visible
.hidden-md	Visible	Visible	Hidden	Visible
.hidden-lg	Visible	Visible	Visible	Hidden

# Primer za verziju 3

```
<h1 class="visible-xs"> EXTRA SMALL Ekran.</h1>  
<h1 class="visible-sm"> SMALL Ekran.</h1>  
<h1 class="visible-md"> MEDIUM Ekran.</h1>  
<h1 class="visible-lg"> LARGE Ekran.</h1>
```

Vidljiv samo jedan tekst i to onaj koji odgovara širini ekrana

```
<span class="hidden-xs"> EXTRA SMALL Ekran.</span>  
<span class="hidden-sm"> SMALL Ekran.</span>  
<span class="hidden-md"> MEDIUM Ekran.</span>  
<span class="hidden-lg"> LARGE Ekran.</span>
```

Nije vidljiv samo jedan tekst i to onaj koji odgovara širini ekrana

## Primer za verziju 4

```
<h1 class="d-none d-sm-block">Nevidljiv za EXTRA SMALL</h1>  
<h1 class="d-block d-sm-none"> Vidljiv samo za SMALL Ekran</h1>  
<h1 class="d-none d-lg-block"> Nevidljiv za MEDIUM Ekran i manje</h1>  
<h1 class="d-none d-md-block d-lg-none"> Vidljiv samo za MEDIUM</h1>
```

### ► Primeri upotrebe ovih klasa u novoj verziji BS

hidden-xs-down = d-none d-sm-block  
hidden-sm-down = d-none d-md-block  
hidden-md-down = d-none d-lg-block  
hidden-lg-down = d-none d-xl-block  
hidden-xl-down = d-none (same as hidden)  
hidden-xs-up = d-none (same as hidden)  
hidden-sm-up = d-sm-none  
hidden-md-up = d-md-none

visible-xs (only) = d-block d-sm-none  
visible-sm (only) = d-none d-sm-block d-md-none  
visible-md (only) = d-none d-md-block d-lg-none  
visible-lg (only) = d-none d-lg-block d-xl-none  
visible-xl (only) = d-none d-xl-block

# Isključivanje svojstva plutanja

- ▶ Bootstrap ima svoju klasu kojom obezbeđuje primenu isključivanja plutajućeg svojstva.
- ▶ Pogledajte primer za verziju 3. Za verziju 4 primer nije dobar zbog drugačijeg prikaza druge ćelije koja obuhvata visinu celog reda.

```
<div class="container">
```

```
  <div class="row">
```

```
    <div class="col-6 " style="background-color:lightblue;">col 1.<br />
```

U zavisnosti od primenjenog svojstva clearfix nakon druge koloe biće i formiran prikaz treće i četvrte kolone. Ako se primeni clearfix nakon druge kolone ove kolone idu ispod prethodnog sadržaja a ako ne onda se sadržaj popunjava na plutajući način.

```
  </div>
```

```
  <div class="col-6 " style="background-color:lightcoral;">col 2.</div>
```

```
  <!--<div class="clearfix "></div-->
```

```
  <div class="col-6 " style="background-color:lightgreen;">col 3.</div>
```

```
  <div class="col-6 " style="background-color:lightgray;">col 4.</div>
```

```
  </div>
```

```
</div>
```

col 1.

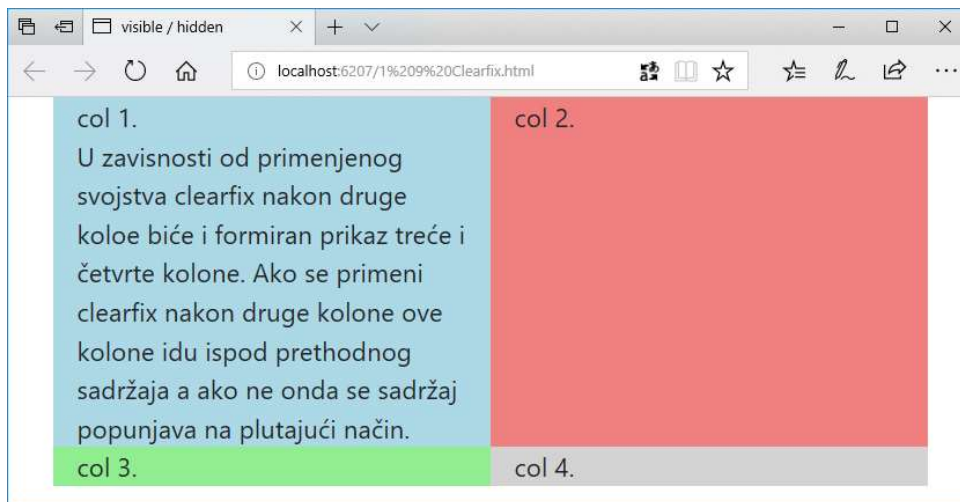
U zavisnosti od primenjenog svojstva clearfix nakon druge koloe biće i formiran prikaz treće i četvrte kolone. Ako se primeni clearfix nakon druge kolone ove kolone idu ispod prethodnog sadržaja a ako ne onda se sadržaj popunjava na plutajući način.

col 2.

col 3.

col 4.

► Nakon primene clearfix-a iza druge kolone:



# Primer isključivanje svojstva plutanja

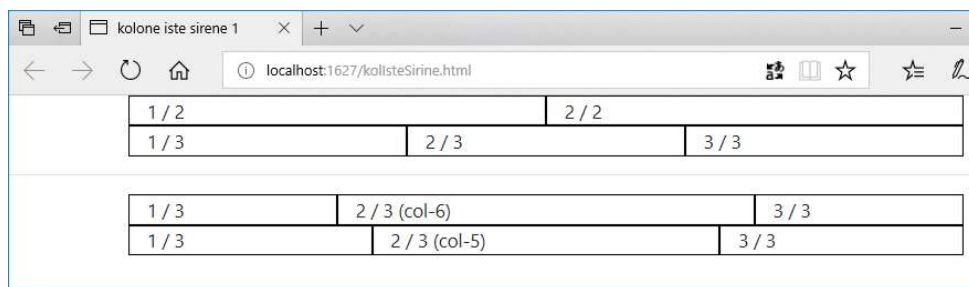
Bez upotrebe klase clearfix odeljak koji obuhvata oba dugmeta neće obuhvatiti dugmad tj imaće prekid u prikazu.

```
<div class="bg-info <!--clearfix-->">  
  ovo je  
  <button class="btn float-left">levo dugme</button>  
  <button class="btn float-right">desno dugme</button> između  
</div>
```



# Postavljanje kolona jednake širine

- ▶ Ako širina kolone nije definisana onda će širina biti ista za takve kolone i ravnomerno raspoređena.
- ▶ Ako je neka kolona definisane širine onda se preostali prostor raspodeljuje na takve kolone.



1 / 2	2 / 2	3 / 2
1 / 3	2 / 3	3 / 3

1 10 kollsteSirine

```
<div class="container">
  <div class="row">
    <div class="col">
      1 / 2
    </div>
    <div class="col">
      2 / 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 / 3
    </div>
    <div class="col">
      2 / 3
    </div>
    <div class="col">
      3 / 3
    </div>
  </div>
</div>
<hr/>
<div class="container">
```

```
<div class="row">
  <div class="col">
    1 / 3
  </div>
  <div class="col-6">
    2 / 3 (col-6)
  </div>
  <div class="col">
    3 / 3
  </div>
</div>
<div class="row">
  <div class="col">
    1 / 3
  </div>
  <div class="col-5">
    2 / 3 (col-5)
  </div>
  <div class="col">
    3 / 3
  </div>
</div>
</div>
```

# Podešavanje margina i pedinga(engl. spacing)

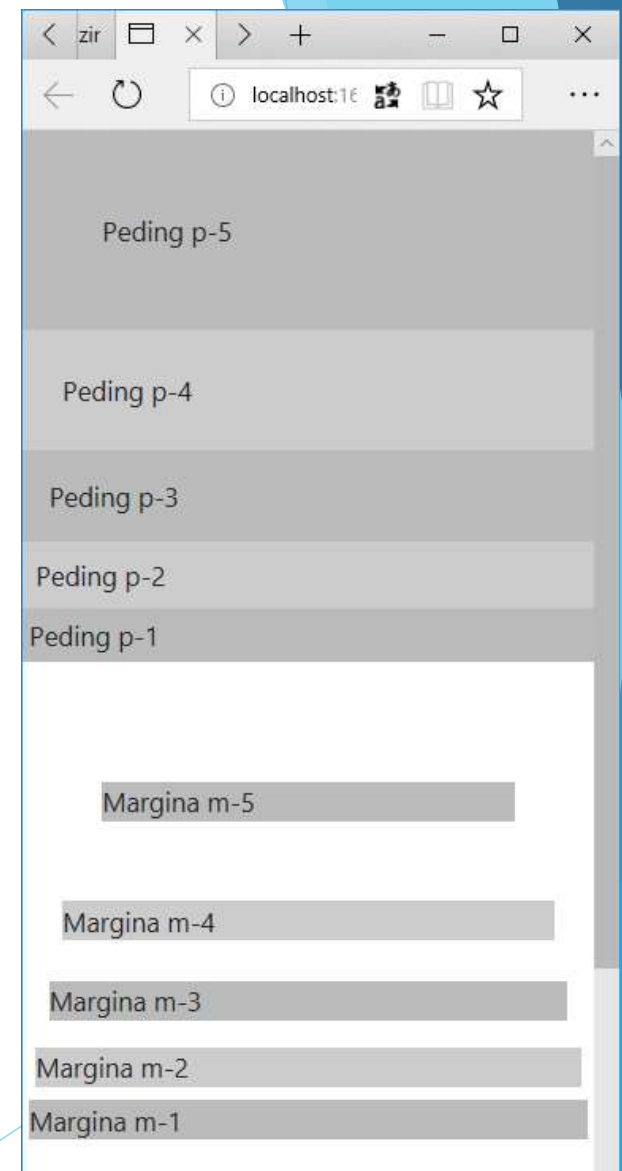
- ▶ Mogu se podesiti odgovarajuće vrednosti margine ili pedinga elementu. Postoji podrška u rasponu od .25rem do 3rem.
- ▶ Ove klase su imenovane po konvenciji:
- ▶ `{svojstvo}{strana}-{vel}` odnosno ako treba da imaju tačke prekida (engl. breakpoint) onda se definiše
- ▶ `{svojstvo}{strana}-{breakpoint}-{vel}` za: sm, md, lg, and xl (xs je podrazmevan i ne definiše se).
- ▶ `svojstvo` može biti:
- ▶ `m` - za margine
- ▶ `p` - za peding

- ▶ Gde je **strana** :
  - ▶ t - za klase koje definišu margin-top ili padding-top
  - ▶ b - za klase koje definišu margin-bottom ili padding-bottom
  - ▶ l - za klase koje definišu margin-left ili padding-left
  - ▶ r - za klase koje definišu margin-right ili padding-right
  - ▶ x - za klase koje definišu \*-left i \*-right
  - ▶ y - za klase koje definišu \*-top i \*-bottom
  - ▶ - za klase koje definišu sve strane
- 
- ▶ A vrednost **vel** je vrednost :
  - ▶ 0 - prostor se izbacuje 0
  - ▶ 1 - \* .25
  - ▶ 2 - \* .5
  - ▶ 3 - podrazumevano
  - ▶ 4 - \* 1.5
  - ▶ 5 - \* 3

# Primer - spacing.html

```
<div class="p-5" style="background-color: #bbb;">Peding p-5</div>
<div class="p-4" style="background-color: #ccc;">Peding p-4</div>
<div class="p-3" style="background-color: #bbb;">Peding p-3</div>
<div class="p-2" style="background-color: #ccc;">Peding p-2</div>
<div class="p-1" style="background-color: #bbb;">Peding p-1</div>
<br/>
<div class="m-5" style="background-color: #bbb;">Margina m-5</div>
<div class="m-4" style="background-color: #ccc;">Margina m-4</div>
<div class="m-3" style="background-color: #bbb;">Margina m-3</div>
<div class="m-2" style="background-color: #ccc;">Margina m-2</div>
<div class="m-1" style="background-color: #bbb;">Margina m-1</div>
```

1 11 spacing



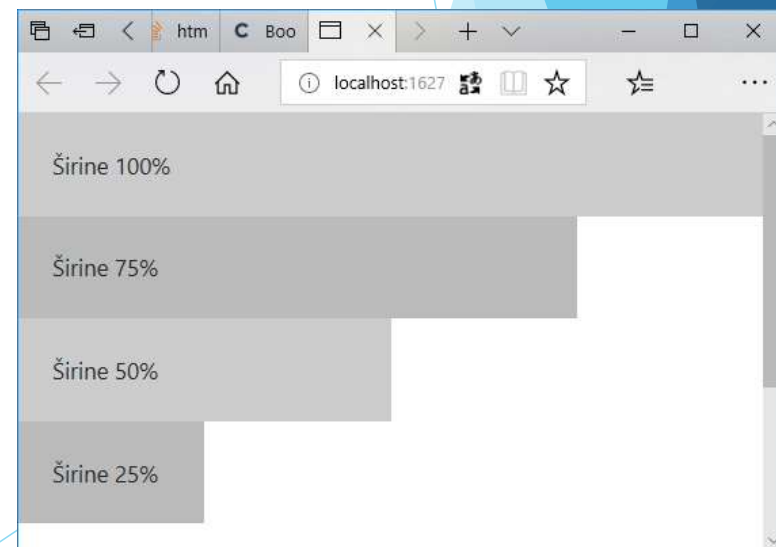
# Podešavanje veličine (engl. sizing)

- ▶ Širina i visina se lako mogu podesiti tako da budu relativne u odnosu na roditelja. Postoje klase za 25%,50%,75% i 100%. Sintaksa klase je:
- ▶ `w-{proc}` - gdje je proc upravo jedna od vrednosti 25,50,75 ili 100.

## ▶ Primer:

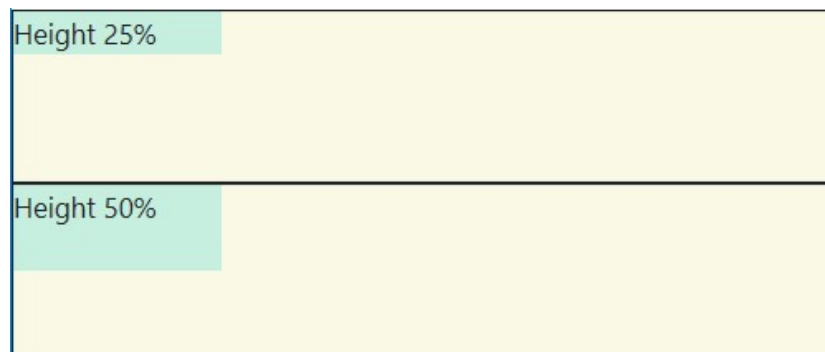
```
<div class="w-100 p-4" style="background-color: #ccc;">Širine 100%</div>  
<div class="w-75 p-4" style="background-color: #bbb;">Širine 75%</div>  
<div class="w-50 p-4" style="background-color: #ccc;">Širine 50%</div>  
<div class="w-25 p-4" style="background-color: #bbb;">Širine 25%</div>
```

1 12 sizing



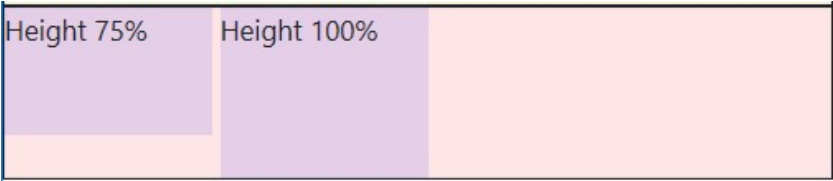
- ▶ Podešavanje se može izvesti i po vertikali tj po visini. Naravno za to je neophodno da postoji kontejner definisane visine. Postoje klase za 25%,50%,75% i 100% visine kontejnera. Sintaksa klasa je:
- ▶ **h-{proc}** - gde je proc jedna od vrednosti 25,50,75 ili 100.

```
<div style="height: 100px; background-color: rgba(200,200,0,0.1); border:solid 1px;">  
  <div class="h-25 " style="width: 120px; background-color: rgba(0,200,200,0.2)">Height 25%</div>  
</div>  
<div style="height: 100px; background-color: rgba(200,200,0,0.1); border:solid 1px;">  
  <div class="h-50 " style="width: 120px; background-color: rgba(0,200,200,0.2)">Height 50%</div>  
</div>
```



► Primer:

```
<div style="height: 100px; background-color: rgba(255,0,0,0.1); border:solid 1px;">  
  <div class="h-75 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 75%</div>  
  <div class="h-100 d-inline-block" style="width: 120px; background-color: rgba(0,0,255,.1)">Height 100%</div>  
</div>
```



# Tipografija

- ▶ Bootstrap koristi sopstvena podešavanja umesto podrazumevanih za web čitače. Ovo donosi prepoznatljive razlike u odnosu na podrazumevane izgledе.
- ▶ Bootstrap koristi “Helvetica Neue”, “Helvetica”, “Arial”, i sans-serif kao podrazumevane fontove respektivno.
- ▶ Primer: Razlika u fontu, veličini i bojama u odnosu na standardna podešavanja

```
<h1>h1 Naslov 1 <small>Mali <mark>tekst</mark></small></h1>  
<h2>h2 Naslov 2 <small>Mali <mark>tekst</mark></small></h2>  
<h3>h3 Naslov 3 <small>Mali <mark>tekst</mark></small></h3>  
...
```

**small** - može da se koristi kao podnaslov

h1 Naslov 1 Mali tekst

h2 Naslov 2 Mali tekst

h3 Naslov 3 Mali tekst

h4 Naslov 4 Mali tekst

h5 Naslov 5 Mali tekst

h6 Naslov 6 Mali tekst

**h1 Naslov 1 Mali tekst**

**h2 Naslov 2 Mali tekst**

**h3 Naslov 3 Mali tekst**

**h4 Naslov 4 Mali tekst**

**h5 Naslov 5 Mali tekst**

**h6 Naslov 6 Mali tekst**



# Klase za predefinisane boje teksta i pozadine

`.text-muted`  
`.text-primary`  
`.text-success`  
`.text-info`  
`.text-warning`  
`.text-danger`

`muted.`

`primary.`

`success.`

`info.`

`warning.`

`danger.`

`.bg-primary`  
`.bg-success`  
`.bg-info`  
`.bg-warning`  
`.bg-danger`

`important.`

`success.`

`information.`

`warning.`

`danger.`