

# Podešavanje Bootstrap grida

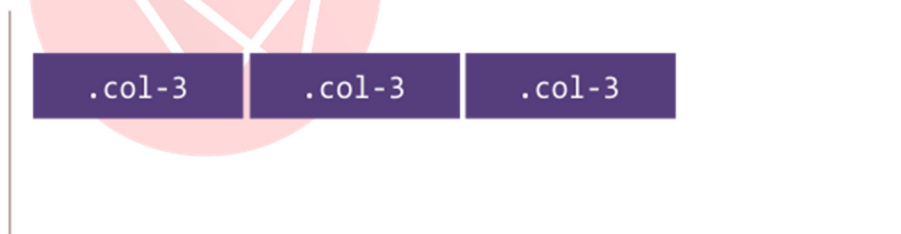
Prethodna lekcija je ilustrovala osnovne elemente Bootstrap grida i pristupe za njihovo ispravno korišćenje. Ipak, Bootstrap u sebi krije još mnoštvo opcija za podešavanje grida i njegovo prilagođavanje potrebama korisnika. Stoga će u lekciji pred vama biti reči o nekoliko veoma značajnih oblasti kontrole Bootstrap grida – horizontalnom i vertikalnom poravnanju i uticanju na redosled, položaj i vidljivost kolona. Na kraju lekcije biće prikazani i napredni pristupi kreiranja složenih grid sistema, kombinovanjem više sistema u jedan.

## Horizontalno poravnanje kolona

Kako bi horizontalno poravnanje kolona uopšte bilo moguće, neophodno je da kolone ne zauzimaju kompletnu dostupnu širinu unutar jednog reda. Takvu situaciju ćemo dobiti sledećim primerom:

```
<div class="container">
  <div class="row">
    <div class="col-3">
      .col-3
    </div>
    <div class="col-3">
      .col-3
    </div>
    <div class="col-3">
      .col-3
    </div>
  </div>
</div>
```

Na ovaj način biće kreiran Bootstrap grid sa jednim redom i tri kolone, pri čemu će kolone zauzimati 9/12 ukupne širine grida (3 + 3 + 3). Kompletan grid će izgledati kao na slici 3.1.



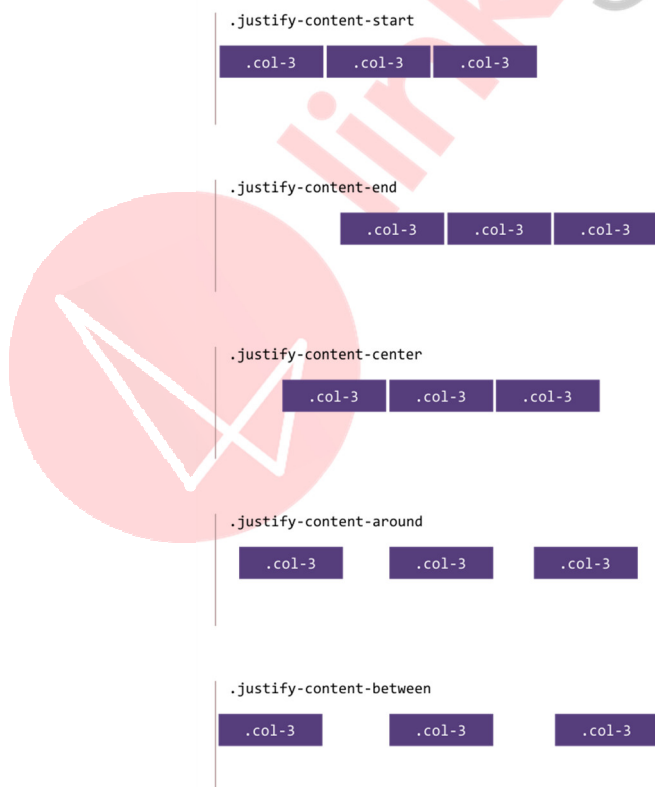
*Slika 3.1. Grid sa jednim redom i tri kolone od kojih svaka zauzima po 1/4 širine; tako 1/4 širine ostaje neiskorišćena*

Horizontalno poravnanje svih kolona unutar jednog reda se može postići jednostavnim dodavanjem jedne od nekoliko klasa na `.row` element (tabela 3.1).

Klasa	Efekat
<code>.justify-content-start</code>	poravnanje po levoj ili desnoj ivici grida, koje zavisi od usmerenja teksta; kada je usmerenje teksta <u>ltr</u> , poravnanje se obavlja po levoj osi; za <u>rtl</u> usmerenje teksta, poravnanje je po desnoj osi; usmerenje teksta se može menjati korišćenjem HTML atributa <code>dir</code>
<code>.justify-content-end</code>	poravnanje po levoj ili desnoj ivici grida, koje zavisi od usmerenja teksta
<code>.justify-content-center</code>	centriranje kolona unutar reda
<code>.justify-content-around</code>	centriranje kolona sa jednakim razmakom između svih kolona
<code>.justify-content-between</code>	centriranje kolona tako da se prva i poslednja kolona poravnavaju po levoj i desnoj ivici, o ostale kolone se raspoređuju sa jednakim prostorom između

*Tabela 3.1. Klase za horizontalno poravnanje*

Efekti upravo prikazanih klasa na primeru tri nešto ranije prikazane kolone ilustrovani su slikom 3.2.



*Slika 3.2. Efekat različitih klasa za poravnanje Bootstrap kolona*

## Vertikalno poravnanje kolona

Realizacija vertikalnog poravnanja HTML elemenata oduvek je predstavljala izazov, zbog normalnog toga HTML dokumenta, odnosno načina na koji se elementi raspoređuju. Ipak, pojavom Flexboxa kao sastavnog dela CSS jezika, obavljanje vertikalnog poravnanja donekle je olakšano. S obzirom na to da se Bootstrap zasniva na Flexboxu, vertikalno poravnanje kolona je moguće postići na prilično jednostavan način.

Preduslov za obavljanje vertikalnog poravnanja jeste postojanje viška vertikalnog prostora unutar Bootstrap reda. Bitno je znati da Bootstrap redovi podrazumevano ne poseduju sopstvenu visinu, odnosno da njihova visina zavisi od sadržaja koji se nalazi unutar kolona. Stoga je na samom početku, kako bi bilo moguće uvideti efekat vertikalnog poravnanja, potrebno ručno definisati visinu Bootstrap reda.

Visina Bootstrap reda se može definisati na različite načine. Moguće je koristiti osnovne CSS pristupe koji podrazumevaju upotrebu `height` svojstva i bilo koje apsolutne jedinice (`px`, `pt`, `pc`, `in`...). Takođe, moguće je koristiti i jedinicu `vh` (*viewport height*). Ipak, pored ovih standardnih CSS pristupa, moguće je koristiti i neku od pomoćnih Bootstrap klasa koje omogućavaju lako definisanje veličina.

### Pomoćne Bootstrap klase za definisanje veličine

Bootstrap poseduje veliku količinu pomoćne stilizacije, koja olakšava pisanje osnovnog CSS koda. Takva pomoćna stilizacija smeštena je unutar *Utilities* odeljka Bootstrap softverskog okvira. Mi ćemo se sa takvom, pomoćnom stilizacijom upoznavati tokom narednih lekcija, kada se njenim korišćenjem javi opravdana potreba.

Pomoćna Bootstrap stilizacija, između ostalog, omogućava da se veoma lako definiše veličina bilo kog HTML elementa, jednostavnim postavljanjem unapred pripremljenih klasa (tabela 3.2).

Klasa	Efekat
<code>.w-25</code>	<code>width: 25%;</code>
<code>.w-50</code>	<code>width: 50%;</code>
<code>.h-75</code>	<code>height: 75%;</code>
<code>.h-100</code>	<code>height: 100%;</code>
<code>.vw-100</code>	<code>width: 100vw;</code>
<code>.vh-100</code>	<code>height: 100vh;</code>

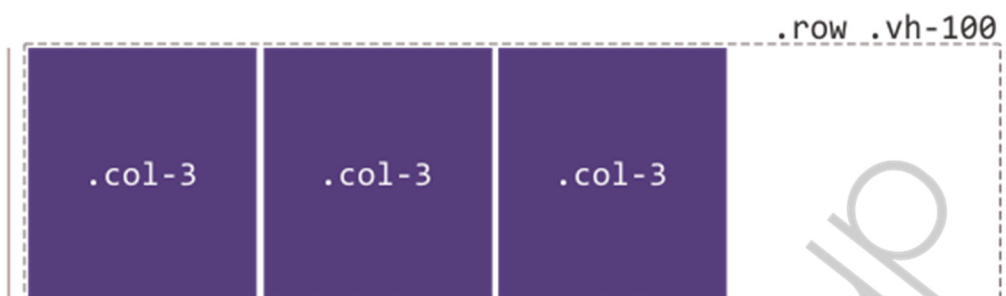
Tabela 3.2. Bootstrap klase za definisanje veličine HTML elemenata

Nakon upoznavanja klasa za lako definisanje veličine HTML elemenata, visina `.row` elementa se može definisati na sledeći način:

```
<div class="row vh-100">
  <div class="col-3">
    .col-3
  </div>
  <div class="col-3">
    .col-3
  </div>
  <div class="col-3">
    .col-3
  </div>
</div>
```

Bitno je da primetite klasu `.vh-100` koja je dodata na `.row` element.

Nakon upravo prikazane intervencije nad Bootstrap gridom, ispoljava se još jedna njegova značajna osobina. Moguće je primetiti da visina svih kolona, bez obzira na količinu sadržaja unutar njih, prati promenu visine reda (slika 3.3).



*Slika 3.3. Podrazumevano, sve kolone imaju identičnu visinu kao i red unutar koga se nalaze*

Ovakva osobina je direktna posledica korišćenja Flexboxa za realizaciju Bootstrapa. S obzirom na to da su Bootstrap redovi zapravo Flexbox kontejneri, a kolone Flexbox elementi, kolone podrazumevano nastoje da zauzmu kompletno dostupan vertikalni prostor. Ipak, uskoro ćete moći da vidite da se takvo ponašanje može menjati, korišćenjem različitih klasa za vertikalno centriranje.

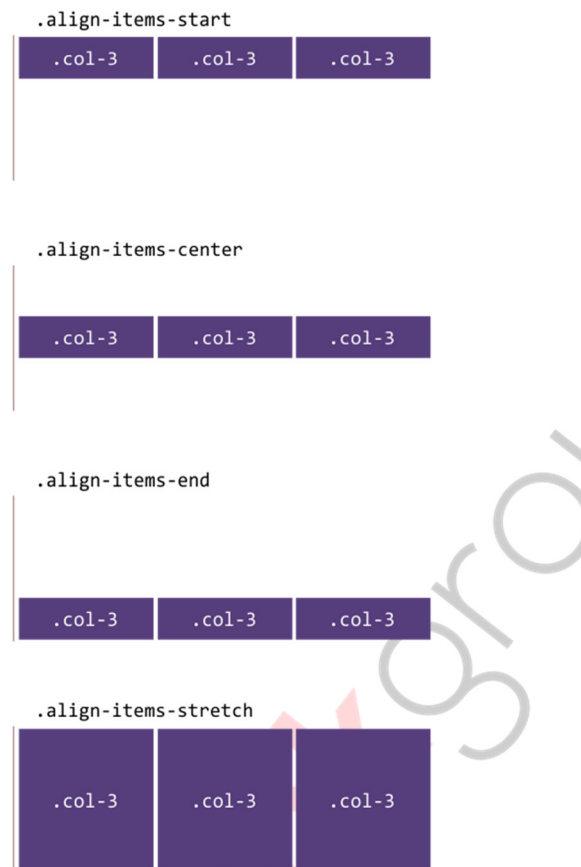
Vertikalno centriranje kolona unutar Bootstrapa moguće je postići na dva načina:

- grupnim centriranjem svih kolona unutar jednog reda, definisanjem odgovarajuće klase na `.row` elementu;
- pojedinačnim centriranjem elemenata, navođenjem odgovarajuće klase na samoj koloni.
- Grupno vertikalno poravnanje definisanjem klase na `.row` elementu se može obaviti korišćenjem klasa prikazanih tabelom 3.3.

Klasa	Efekat
<code>.align-items-start</code>	poravnava sve kolone po vrhu reda
<code>.align-items-center</code>	vertikalno centrira sve kolone unutar reda
<code>.align-items-end</code>	poravnava sve kolone po dnu reda
<code>.align-items-stretch</code>	razvlači sve kolone kako bi popunile dostupan vertikalni prostor unutar reda

*Tabela 3.3. Bootstrap klase za vertikalno centriranje svih kolona unutar redova*

Efekti klasa iz tabele 3.3. ilustrirani su slikom 3.4.



*Slika 3.4. Efekti klasa za vertikalno centriranje svih kolona unutar jednog reda*

Vertikalno poravnanje kolona je moguće postići i na svakoj koloni pojedinačno, korišćenjem klasa iz tabele 3.4.

Klasa	Efekat
.align-self-start	poravnava kolonu po vrhu reda
.align-self-end	poravnava kolonu po dnu reda
.align-self-center	vertikalno centrira kolonu unutar reda
.align-self-stretch	razvlači kolonu kako bi popunila dostupan vertikalni prostor unutar reda

*Tabela 3.4. Bootstrap klase za vertikalno centriranje pojedinačnih kolona*

Efekti upravo prikazanih klasa ilustrovani su slikom 3.5.



Slika 3.5. Efekti klasa za vertikalno centriranje pojedinačnih kolona

#### Pitanje

Vertikalno poravnanje pojedinačnih kolona se može obaviti korišćenjem klasa sa prefiksom:

- a) `.justify-content-`
- b) `.justify-self-`
- c) `.align-items-`
- d) **`.align-self-`**

Vertikalno poravnanje kolona je moguće postići na svakoj koloni pojedinačno, korišćenjem klasa koje započinju prefiksom `.align-self-`.

#### Definisanje redosleda kolona

Kolone unutar jednog reda podrazumevano se raspoređuju redosledom kojim su definisane unutar HTML koda. Ipak, Bootstrap omogućava da se na njihov vizuelni raspored utiče korišćenjem klasa koje započinju prefiksom `order-` nakon koga se navodi željeni redni broj (slika 3.6).

**`.order-*`**

Slika 3.6. Struktura klasa za definisanje redosleda kolona

Zvezdica sa slike 3.6. se može zameniti brojem u rasponu od 1 do 12. Razlog je prilično očigledan – maksimalni broj kolona unutar Bootstrap grida je upravo 12.

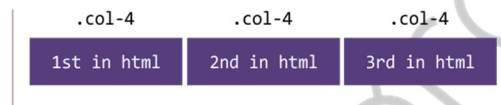
Za demonstraciju efekta klasa za definisanje redosleda kolona poslužiće sledeći primer:

```

<div class="container">
  <div class="row">
    <div class="col-4">
      1st in html
    </div>
    <div class="col-4">
      2nd in html
    </div>
    <div class="col-4">
      3rd in html
    </div>
  </div>
</div>

```

Kod ilustruje Bootstrap grid sa jednim redom i tri kolone, koje podjednako dele dostupan prostor unutar reda. Bez ikakvog dodatnog podešavanja, ovaj kod će stvoriti efekat kao na slici 3.7.



*Slika 3.7. Tri kolone raspoređene redosledom kojim su definisane u HTML kodu*

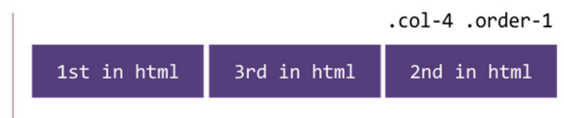
Sada će na jednu kolonu ovakvog grida biti postavljena jedna od klasa za definisanje redosleda:

```

<div class="container">
  <div class="row">
    <div class="col-4">
      1st in html
    </div>
    <div class="col-4 order-1">
      2nd in html
    </div>
    <div class="col-4">
      3rd in html
    </div>
  </div>
</div>

```

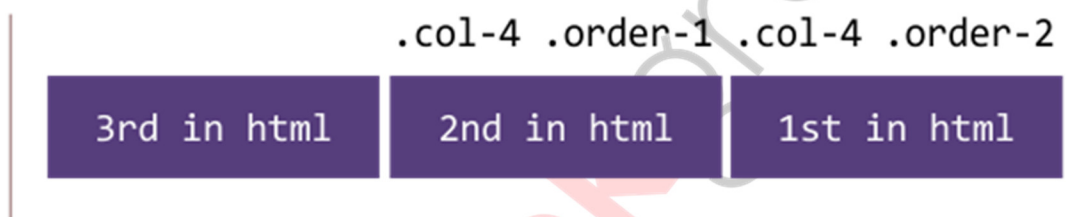
Na drugoj koloni sada je postavljena klasa `.order-1`. To će stvoriti nešto drugačiji prikaz (slika 3.8).



*Slika 3.8. Kolona koje je u HTML kodu druga po redosledu na stranici se nalazi na trećem mestu*

Na slici 3.8. se može videti vrlo zanimljivo ponašanje. Kolona koja je unutar HTML koda druga po redu sada je prebačena na treću poziciju i to upravo zbog postavljanja klase `.order-1`. Razlog je vrlo jednostavan. Bootstrap kolone se podrazumevano ponašaju kao da na sebi imaju klasu `.order-0`. Broj 1 je veći od broja 0, pa se upravo zbog toga dve kolone bez `.order-` klase vizuelno prikazuju pre kolone sa klasom `.order-1`. Ipak, definisanjem neke od `.order-` klase na još jednom elementu, situacija se menja (slika 3.9):

```
<div class="container">
  <div class="row">
    <div class="col-4 order-2">
      1st in html
    </div>
    <div class="col-4 order-1">
      2nd in html
    </div>
    <div class="col-4">
      3rd in html
    </div>
  </div>
</div>
```



Slika 3.9. Kolone čiji je vizuelni raspored potpuno drugačiji od rasporeda u kodu

Sada je na prvu kolonu u HTML kodu postavljena klasa `.order-2`. Na taj način se dobija raspored koji je prikazan slikom 3.9, odnosno, kolone se vizuelno prikazuju redosledom 3, 2, 1.

Redosled kolona je moguće definisati i u zavisnosti od osobina displeja. Tako Bootstrap omogućava integraciju responsive odrednica `sm`, `md`, `lg` i `xl` unutar klase za kontrolisanje redosleda kolona (slika 3.10).

**.order-\*-\***

Slika 3.10. Struktura klase za definisanje redosleda kolona u zavisnosti od osobina displeja

Slika 3.10. ilustruje formu za definisanje klase kojima se redosled kolona kontroliše u zavisnosti od osobina displeja. Na mesto prve zvezdice je moguće dodati neku od odrednica `sm`, `md`, `lg` ili `xl`. Na mesto druge zvezdice, kao i do sada, postavlja se broj u rasponu od 1 do 12:

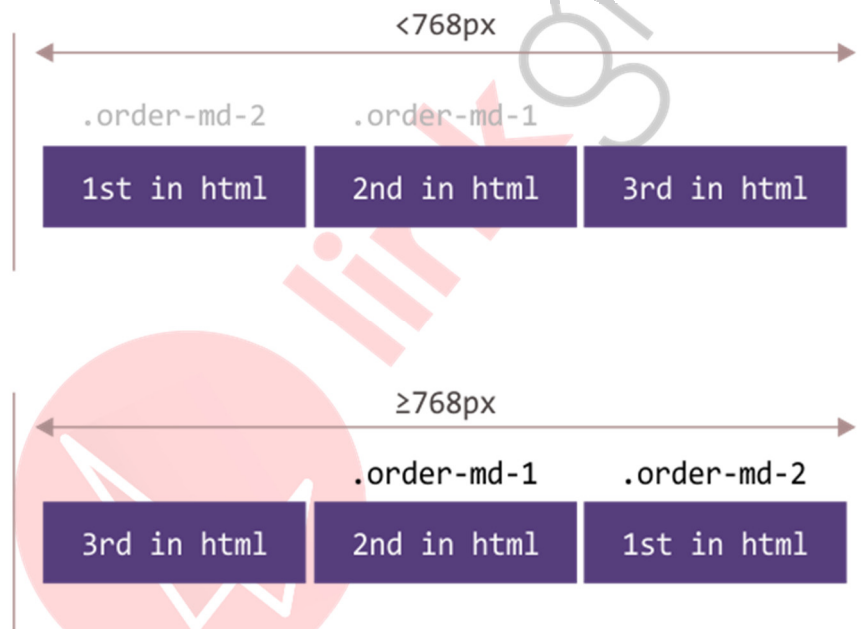


```

<div class="container">
  <div class="row">
    <div class="col-4 order-md-2">
      1st in html
    </div>
    <div class="col-4 order-md-1">
      2nd in html
    </div>
    <div class="col-4">
      3rd in html
    </div>
  </div>
</div>

```

Na postojećem primeru napravljena je mala izmena. Unutar dve `order` klase dodata je odrednica `md`. Redni brojevi su ostali identični. Sada će vizuelni raspored kolona koji smo samostalno definisali biti primenljiv samo na displejima srednje veličine i većim (*medium*, *large* i *extra large*). Na malim (*small*) i najmanjim displejima (*extra small*), raspored koji smo samostalno definisali uopšte se neće primenjivati (slika 3.11).



Slika 3.11. Efekat raspoređivanja kolona u zavisnosti od osobina displeja

## Pomeranje kolona unutar grida

Prilikom raspoređivanja elemenata web sajtova korišćenjem Bootstrapa, veoma često se može javiti potreba za pomeranjem nekih kolona udesno, čime je određeni prostor unutar reda moguće preskočiti, odnosno ostaviti prazan. Do ovakve potrebe najčešće dolazi kada kolone ne zauzimaju kompletan prostor unutar jednog reda, a potrebno ih je rasporediti na specifičan način, koji ne podrazumeva njihovo sukcesivno ređanje. Na primer, pogledajte sledeći primer:

```

<div class="container">
  <div class="row">
    <div class="col-2">
      col-2
    </div>
    <div class="col-2">
      col-2
    </div>
    <div class="col-2">
      col-2
    </div>
  </div>
</div>

```

Prikazani primer podrazumeva Bootstrap grid sa jednim redom i tri kolone, pri čemu svaka od kolona zauzima 2/12 ukupne širine reda. Sve to praktično znači da je prva polovina reda zauzeta kolonama, dok je druga polovina slobodna (slika 3.12).



*Slika 3.12. Tri kolone koje zauzimaju jednu polovinu dostupne širine*

Unutar grida prikazanog slikom 3.12, šest kolona je zauzeto, dok je šest i dalje slobodno. Ipak, raspored zauzetih i slobodnih kolona ne mora biti baš ovakav kao na slici. Bootstrap omogućava da se kolone pomere udesno za određeni broj referentnih kolona, a sve to korišćenjem klasa koje započinju prefiksom `.offset-`.

## **.offset-\***

*Slika 3.13. Struktura klasa za pomeranje kolona udesno*

Na mesto zvezdice, dodaje se broj, koji definiše za koliko referentnih kolona je konkretnu kolonu potrebno pomeriti udesno:

```

<div class="container">
  <div class="row">
    <div class="col-2 offset-1">
      col-2 offset-1
    </div>
    <div class="col-2 offset-2">
      col-2 offset-2
    </div>
    <div class="col-2 offset-3">
      col-2 offset-3
    </div>
  </div>
</div>

```

Na primeru su sada upotrebljene klase `.offset-1`, `.offset-2` i `.offset-3`. Na prvoj koloni je postavljena klasa `.offset-1`, što znači da će ova kolona biti pomerena za jednu referentnu kolonu udesno. Na drugom `div` elementu koji predstavlja kolonu, postavljena je klasa `.offset-2`. Zbog toga će takva kolona biti pomerena za dva mesta udesno. Na kraju, poslednja kolona se klasom `.offset-3` pomerena je udesno za tri mesta (slika 3.14).



Slika 3.14. Efekat pomeranja kolona u desno; isprekidanim linijama su prikazane referentne kolone koje su preskočene, te su stoga prazne, odnosno u njima nema elemenata

#### Napomena

Prilikom pomeranja kolona bitno je voditi računa o tome da ukupan zbir kolona ne pređe 12. Naime, i vrednosti na `.offset-` klasama ulaze u ukupan zbir iskorišćenih kolona. Tako je u upravo prikazanom primeru iskorišćeno tačno 12 kolona ( $1 + 2 + 2 + 2 + 3 + 2$ ).

Baš kao i dosta puta do sada, i prilikom pomeranja kolona moguće je koristiti odrednice `sm`, `md`, `lg` i `xl` (slika 3.15). Na taj način je moguće kontrolisati pomeranje kolona za svaki od tipova uređaja pojedinačno.

## .offset-\*-\*

Slika 3.15. Struktura klasa za pomeranje kolona udesno, u zavisnosti od tipa displeja uređaja

Na mesto prve zvezdice postavlja se jedna od odrednica za različite tipove uređaja, dok se umesto druge zvezdice postavlja neki od brojeva između 1 i 12:

```
<div class="container">
  <div class="row">
    <div class="col-2 offset-xl-1 offset-md-2">
      col-2 offset-1
    </div>
    <div class="col-2 offset-xl-2 offset-md-1">
      col-2 offset-2
    </div>
    <div class="col-2 offset-3">
      col-2 offset-3
    </div>
  </div>
</div>
```

Primer je sada modifikovan dodavanjem klasa za pomeranje sa specijalnim odrednicama za različite uređaje. Efekat će biti objašnjen za svaku kolonu pojedinačno.

Na prvoj koloni su definisane klase `.offset-xl-1` i `.offset-md-2`. Ovim se postiže sledeće:

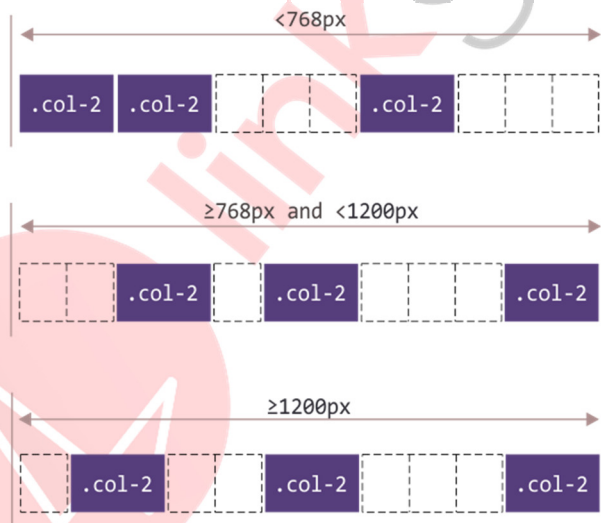
- na svim displejima sa vidnim poljem preko 1200px, prva kolona će biti pomerena za jednu poziciju udesno, zbog klase `.offset-xl-1`;
- za vidna polja širine između 768px i 1199px, prva kolona će biti pomerena za dve pozicije udesno, zbog klase `.offset-md-2`;
- za vidna polja širine ispod 768px, prva kolona neće uopšte biti pomerena.

Na drugoj koloni su definisane sledeće klase: `.offset-xl-2` i `.offset-md-1`. Ovim se postiže:

- na svim displejima sa vidnim poljem preko 1200px, druga kolona će biti pomerena za dve pozicije udesno, zbog klase `.offset-xl-2`;
- za vidna polja širine između 768px i 1199px, druga kolona će biti pomerena za jednu poziciju udesno, zbog klase `.offset-md-1`;
- za vidna polja širine ispod 768px, druga kolona neće uopšte biti pomerena.

Na treću kolonu postavljena je klasa `.offset-3`. To znači da će treća kolona uvek biti pomerena udesno za tri pozicije.

Efekat upravo ilustrovanog pomeranja ilustrovan je slikom 3.16.



Slika 3.16. Ponašanje tri Bootstrap kolone iz prethodnog primera na različitim displejima

## Uticanje na vidljivost kolona

Bootstrap poseduje skup klasa koje je moguće koristiti kako bi se uticalo na vidljivost elemenata. Bitno je razumeti da takve klase nisu striktno vezane za kolone, što je bio slučaj sa klasama koje su prikazane u dosadašnjem toku lekcije. Reč je, zapravo, o skupu pomoćnih klasa koje je moguće koristiti za kontrolisanje vidljivosti bilo kojeg elementa koji se nalazi unutar Bootstrap grida.

`.d-*`  
*or*  
`.d-*-*`

*Slika 3.17. Struktura klase za kontrolisanje vidljivosti elemenata*

Slika 3.17. ilustruje strukturu klasa kojima je moguće kontrolisati vidljivost elemenata unutar Bootstrap grida. Ovakve klase zapravo omogućavaju kontrolisanje `display` svojstva, pa upravo zbog toga započinju slovom `d`.

Klase za kontrolisanje vidljivosti postoje u dva oblika, u zavisnosti od toga da li se navodi responsive odrednica ili ne. Ukoliko se definiše tip displeja, na mesto prve zvezdice postavlja se jedna od responsive odrednica – `sm`, `md`, `lg` ili `xl`, dok je na mesto druge zvezdice moguće postaviti jednu od vrednosti `display` svojstva:

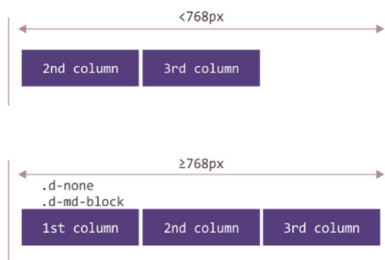
- `none`
- `inline`
- `inline-block`
- `block`
- `table`
- `table-cell`
- `table-row`
- `flex`
- `inline-flex`

Ukoliko se responsive odrednica izostavi, onda se na mestu prve zvezdice navodi jedna od upravo prikazanih vrednosti `display` svojstva.

Upravo prikazane klase omogućavaju veoma laku promenu vrednosti `display` svojstva bilo kojeg elementa unutar Bootstrap grida i to u zavisnosti od osobina uređaja. Tako je neke elemente na uređajima sa najmanjim displejima moguće u potpunosti ukloniti, a prikazati ih tek kada vidno polje dostigne određenu širinu:

```
<div class="container">
  <div class="row">
    <div class="col-4 d-none d-md-block">
      1st column
    </div>
    <div class="col-4">
      2nd column
    </div>
    <div class="col-4">
      3rd column
    </div>
  </div>
</div>
```

Primer ilustruje Bootstrap grid sa jednim redom i tri kolone. Pri tome su na prvoj koloni postavljene sledeće klase: `.d-none` i `.d-md-block`. Ovo praktično znači da će prva kolona imati `none` vrednost `display` svojstva, sve dok širina vidnog polja ne postane veća od 767px (slika 3.18).



*Slika 3.18. Efekat skrivanja kolona na uređajima se extra small i small displejima*

Na slici 3.18, može se videti da se Bootstrap grid za širine vidnog polja ispod 768px ponaša kao da prva kolona ne postoji. Čim vidno polje dobije širinu od 768px ili više, prva kolona dobija `block` za vrednost `display` svojstva i postaje vidljiva.

## Gnežđenje

Prilikom kreiranja korisničkog okruženja web sajtova, nekada se može javiti potreba za postizanjem kompleksnijeg rasporeda komponenta, koji se ne može postići korišćenjem do sada prikazanih pristupa. Na primer, jedna takva situacija se može ilustrovati slikom 3.19.



*Slika 3.19. Primer jednog složenog layouta*

Poseban problem unutar prikazanog layouta predstavlja druga kolona prvog reda. Može se videti da se unutar takve kolone nalaze dva posebna reda, pri čemu prvi takav red poseduje tri kolone, a drugi samo jednu. Jasno je da se korišćenjem do sada prikazanih pristupa ne može postići ovakav raspored elemenata. Rešenje se krije u pojmu gnežđenja (*nesting, engl.*).

Gnežđenje je pojam koji podrazumeva smeštanje redova i kolona u već kreirane Bootstrap kolone. Drugim rečima, kada se unutar jedne Bootstrap kolone postave novi redovi i kolone, dobija se gnežđenje. Ovakav pristup je najbolje objasniti analizom primera:

```

<div class="container">
  <div class="row">
    <div class="col-8">

  </div>
  <div class="col-4">
    <div class="row">
      <div class="col-4"></div>
      <div class="col-4"></div>
      <div class="col-4"></div>
    </div>
    <div class="row">
      <div class="col-12">

    </div>
  </div>
  <div class="row">
    <div class="col-12">

  </div>
</div>

```

Prikazanim kodom kreira se layout baš kakav je prikazan slikom 3.19. Pažnju je potrebno usmeriti na drugu kolonu prvog reda. Unutar takve kolone postavljena su dva nova reda, sa svojim kolonama. Ovakav pristup jeste osnovni način za kreiranje kompleksnih layouta korišćenjem Bootstrapa.

Bitno je razumeti da se unutar kolona direktno postavljaju redovi sa svojim kolonama, odnosno da se prilikom gnežđenja **ne** kreiraju dodatni glavni kontejneri.

## Rezime

- Bootstrap u pozadini koristi Flexbox, te je stoga vertikalno i horizontalno poravnanje moguće postići korišćenjem Flexbox principa.
- Horizontalno poravnanje svih kolona unutar jednog reda se može postići jednostavnim dodavanjem neke od klasa sa prefiksom `.justify-content-`.
- Vertikalno poravnanje svih kolona unutar jednog reda može se postići korišćenjem klasa koje započinju prefiksom `.align-items-`.
- Vertikalno poravnanje pojedinačnih kolona može se postići korišćenjem klasa koje započinju prefiksom `.align-self-`.
- Vizuelni raspored kolona unutar redova može se definisati korišćenjem klasa sa prefiksom `.order-`.
- Ukoliko je potrebno definisati vizuelni raspored kolona u zavisnosti od veličine displeja, koriste se klase sa strukturom `.order-*-*`.
- Pomeranje kolona udesno se može obaviti korišćenjem različitih klasa sa strukturom `.offset-*-*`.

- Bootstrap poseduje pomoćne klase za definisanje veličine elemenata; za definisanje širine u procentima koriste se klase sa prefiksom `.w-`, a za definisanje visine – klase sa prefiksom `.h-`; ukoliko je potrebno definisati visinu ili širinu jednaku vidnom polju, koriste se klase `.vw-100` i `.vh-100`.
- Pomoćne klase za definisanje vidljivosti elemenata započinju prefiksom `.d-`, nakon koga se navode opciona responsive odrednica i neko od vrednosti `display` svojstva.
- Kreiranje složenih layouta se može postići smeštanjem dodatnih redova i kolona u već kreirane kolone Bootstrap grida.

