

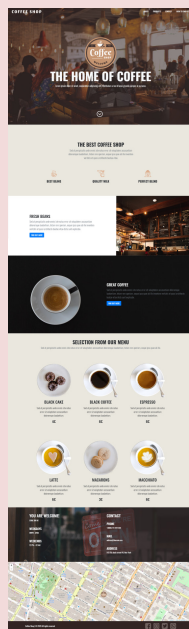
Uvod u Bootstrap

Razvoj web sajtova podrazumeva poznavanje tri osnovna jezika frontend razvoja. Reč je o jezicima HTML, CSS i JavaScript. Ovo su jezici koje web pregledači direktno razumeju, te se upravo zbog toga kaže da su oni osnovni kada je u pitanju kreiranje web sajtova. Ipak, posao kreiranja web sajtova prešao je ogroman put od nastanka weba, početkom 90-ih godina prošlog veka, pa sve do danas. Tokom vremena menjali su se pristupi, dobre prakse i alati koji su korišćeni za razvoj, tako prateći vrtoglavi razvoj weba. Iako je nekada bilo sasvim dovoljno poznavati jezike HTML, CSS i JavaScript, današnji pristupi za kreiranje sajtova nalažu korišćenje i mnogih drugih jezika, biblioteka, alata ili okruženja.

Kurs koji je pred vama približiće vam svet modernog razvoja web sajtova, koji pored osnovnih jezika weba podrazumeva poznavanje još nekih softverskih okvira, biblioteka i pristupa, koji značajno olakšavaju razvoj te tako omogućavaju da se jedan moderan, potpuno funkcionalan web sajt kreira za kratko vreme. Osnovno poznavanje jezika HTML, CSS i JavaScript svakako je preduslov. Ovi jezici predstavljaju jezgro na čijoj osnovi se temelje svi ostali pristupi koji će biti ilustrovani u lekcijama koje slede.

Coffee Shop web sajt

Tokom trajanja ovoga kursa, različiti pristupi koji budu ilustrovani biće iskorišćeni za kreiranje jednog kompletnog, realnog web sajta. Tema sajta će biti Coffee Shop.



Slika 1.1. Coffee Shop web sajt koji će biti kreiran tokom ovog kursa

U nastavku će, nakon završetka svake od oblasti, naučeno biti inkorporirano u kreiranje upravo spomenutog sajta. Prvi deo ovog kursa biće posvećen kreiranju layouta. Stoga će, nakon nekoliko narednih lekcija, prvo biti ilustrovano kreiranje osnovne strukture Coffee Shop web sajta.

Prvi deo ovog kursa biće posvećen jednom od najpopularnijih softverskih okvira za kreiranje sajtova današnjice. Reč je o Bootstrapu.

Šta je Bootstrap?

Bootstrap su kreirali Mark Otto i Jacob Thornton, kao interni alat kompanije Twitter za postizanje uniformnog izgleda i ponašanja web aplikacija ove kompanije. Bootstrap je zatim izašao iz okvira kompanije Twitter i postao javno dostupan proizvod, otvorenog koda, potpuno besplatan za korišćenje.



Slika 1.2. Bootstrap logo

Bootstrap olakšava kreiranje web sajtova tako što poseduje veliku količinu već napisane stilizacije, koju programeri mogu koristiti prilikom kreiranja web sajtova. Takva stilizacija pomaže da se stvori uniforman prikaz na svim web pregledačima, olakša raspoređivanje elemenata i ubrza optimizacija sajtova za različite uređaje. Ipak, Bootstrap je mnogo više od skupa stilova. Reč je, zapravo, o softverskom okviru koji je namenjen frontend razvoju.

Bootstrap poseban akcenat stavlja na dva pristupa:

- *mobile first design* i
- *responsive design*.

Responsive design podrazumeva pristup za kreiranje sajtova koji se prilagođavaju osobinama uređaja na kojima se prikazuju (slika 1.3).



Slika 1.3. Koncept responsive designa

Sam pojam *responsive* odnosi se na brzo i adekvatno reagovanje na nastale promene. Stoga je *responsive design* pristup koji garantuje da će korisničko okruženje sajtova izgledati adekvatno na svim uređajima. Tako se *responsive design* značajno razlikuje od jednog sličnog pristupa – adaptivnog dizajna. Adaptivni dizajn podrazumeva kreiranje više verzija jednog istog sajta, kao i određenu klijentsku i serversku logiku koja se koristi za detekciju osobina klijentskih uređaja. Za razliku od adaptivnog dizajna, *responsive design* uvek podrazumeva samo jednu verziju web sajta, koji se prilagođava osobinama uređaja na kome se prikazuje. Pored toga, *responsive design* podrazumeva i sledeće:

- prilagođavanje rasporeda komponenata osobinama displeja uređaja;
- optimizaciju tekstualnog sadržaja, sa posebnim akcentom na čitljivost;
- optimizaciju veličine, rezolucije i preglednosti slika, fotografija i svih grafičkih elemenata;
- optimizaciju video-sadržaja i
- optimizaciju korisničkog doživljaja sa posebnim akcentom na upotrebljivost.

Mobile first je pristup koji diktira način na koji se postiže *responsive design*. Tako *mobile first* podrazumeva inicijalno kreiranje korisničkog okruženja za uređaje sa najmanjim displejima (pametni telefoni, časovnici i slično). Displeji ovakvih uređaja značajno su manji od onih na desktop ili laptop kompjuterima (tipičan displej desktop kompjutera poseduje 80% više prostora za prikaz sadržaja od displeja na pametnim telefonima).



Slika 1.4. *Mobile first* pristup

Slika 1.4. ilustruje osnovnu logiku *mobile first* dizajna. Može se videti da takav pristup podrazumeva da se layout web sajta prvo kreira za pametne telefone. Zatim se takav raspored komponenata prilagođava uređajima sa većim displejima. S obzirom na to da je broj mobilnih uređaja već odavno nadmašio broj konvencionalnih, desktop uređaja na webu, korišćenje *mobile first* pristupa ima puno smisla.

Zbog čega nam je potreban Bootstrap?

Svako ko je kreirao makar nekoliko web sajtova korišćenjem osnovnih pristupa koji podrazumevaju korišćenje čistog HTML-a, CSS-a i JavaScripta primetio je da se određene komponente iznova i iznova ponavljaju. Stoga njihovo kreiranje podrazumeva ponavljanje identičnih operacija, odnosno pisanje identičnog koda, što na kraju rezultuje gubljenjem vremena i usporavanjem razvoja. Mnogi programeri ovakav problem rešavaju prikupljanjem programskog koda koji se često ponavlja, koji zatim integrišu u svaki novi projekat. Ipak, mnogo jednostavniji pristup podrazumeva korišćenje gotovih skupova funkcionalnosti, odnosno softverskih okvira. Jedan od takvih softverskih okvira je i Bootstrap. Najznačajnije prednosti korišćenja Bootstrapa navedene su u nastavku.

Ušteda vremena

Bootstrap omogućava da se posao kreiranja korisničkog okruženja sajtova obavi za neuporedivo kraći vremenski period u poređenju sa pristupom koji podrazumeva samostalno pisanje kompletnog koda. Naime, unutar Bootstrapa nalazi se velika količina stilizacije, spremne za korišćenje.

Bootstrap obezbeđuje i značajnu uštedu vremena kada projekat uđe u fazu održavanja. Zbog uniformnog sistema imenovanja i unapred utvrđene logike rada, olakšano je snalaženje kako u projektima koje smo samostalno kreirali tako i u projektima koje su kreirali neki drugi programeri.

Prevazilaženje razlika između pregledača

Cilj svakog frontend programera jeste postizanje konzistentnog izgleda i ponašanja sajtova na svim uređajima i pregledačima. Ipak, koliko god se proizvođači različitih pregledača trudili da maksimalno ispoštuju specifikaciju HTML, CSS i JavaScript jezika, razlike između interpretacije programskog koda i načina rada pregledača uvek postoje. Poznavanje svih takvih razlika i načina za postizanje konzistentnog ponašanja gotovo je nemoguće.

Upravo opisani problem mnogo lakše se rešava korišćenjem Bootstrapa. Iza njegovog razvoja stoji veliki razvojni tim i open-source zajednica, koja se brine za to da se eventualni problemi nekonzistentnog ponašanja brzo detektuju i otklone.

Korišćenje dokazanih pristupa i dobrih praksi

Upravo spomenuta činjenica koja podrazumeva da na razvoju Bootstrapa učestvuje veliki razvojni tim, potpomognut zajednicom korisnika, donosi još jednu prednost. Reč je o korišćenju dokazanih pristupa i dobrih praksi. Naime, preslikavanje jednog mockupa u funkcionalnu web stranicu može se obaviti na stotinu načina. Drugim rečima, postoji veliki broj načina na koje se može dobiti identičan vizuelni i funkcionalni efekat na webu. Ipak, to ne znači da su svi takvi pristupi ujedno i dobri, kada se u obzir uzmu i druge kvalitativne kategorije softverskih proizvoda (performanse, optimizacija, lakoća održavanja...). Zaključak je da je prilikom razvoja softvera veoma značajan i pristup koji je upotrebljen za rešavanje nekog problema. Bootstrap osigurava korišćenje dokazanih pristupa i dobrih praksi.

Smanjenje mogućnosti konflikata

Bootstrap je tokom vremena iz jednostavne CSS biblioteke narastao u veliki softverski okvir pomoću koga se mogu pokriti brojni aspekti razvoja web sajtova. Naime, pored pukog stilizovanja, Bootstrap poseduje i veliki broj ugrađenih komponenata koje kombinuju veći broj HTML elemenata i prateću JavaScript logiku za njihovo funkcionisanje. Sve to omogućava da se bez ikakvih problema svi elementi web sajta realizuju korišćenjem Bootstrapa, što je mnogo bolji pristup nego korišćenje većeg broja pojedinačnih, manjih biblioteka. Svako ko je na jednom sajtu koristio veći broj različitih biblioteka i softverskih okvira može posvedočiti da takav pristup često stvara neočekivane efekte, usled konflikata koji nastaju kao produkt neusaglašenosti različitih biblioteka.

Olakšavanje održavanja i timskog rada

Programerska profesija veoma često podrazumeva ažuriranje i nadogradnju postojećih softverskih proizvoda. Snalaženje u velikoj količini fajlova sa izvornim kodom koji su pisali neki drugi programeri može predstavljati veliki izazov. Pogotovu ukoliko su za pisanje koda korišćena interna pravila jednog ili više programera. Ukoliko je, sa druge strane, potrebno ažurirati ili nadograditi sajt u čijem kreiranju je učestvovao Bootstrap, posao programera je uglavnom znatno jednostavniji. Bootstrap funkcioniše po jasno utvrđenim pravilima, pa je mnogo lakše razumeti napisan kod. Sve nepoznanice jednostavno se mogu otkloniti uvidom u odličnu dokumentaciju na zvaničnom sajtu Bootstrapa.

Pitanje

Bootstrap je nastao kao interni alat kompanije:

- a) **Twitter**
- b) Facebook
- c) Microsoft
- d) Google

Objašnjenje:

Bootstrap su kreirali Mark Otto i Jacob Thornton, kao interni alat kompanije Twitter za postizanje uniformnog izgleda i ponašanja web aplikacija ove kompanije.

Struktura Bootstrapa

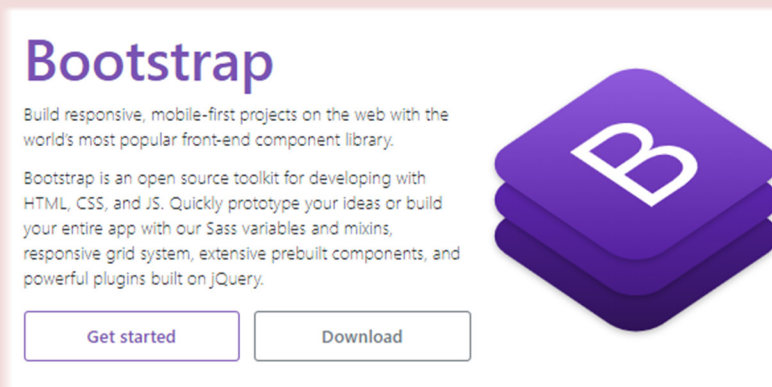
Bootstrap je sačinjen iz nekoliko CSS i JavaScript fajlova. Pre nego što budemo u mogućnosti da takve fajlove na pravi način upotrebimo, potrebno je razumeti njihovu svrhu unutar Bootstrap softverskog okvira. Njihovu svrhu je najbolje razumeti analizom paketa koji se dobija preuzimanjem Bootstrapa sa zvaničnog sajta.

Preuzimanje Bootstrap fajlova

Bootstrap je moguće preuzeti sa zvaničnog web sajta:

<https://getbootstrap.com/>

Na stranici koja se otvara, potrebno je odabrati *Download* (slika 1.5).



Slika 1.5. Preuzimanje Bootstrapa

Nakon klika na *Download*, otvara se stranica sa linkovima za preuzimanje Bootstrapa u nekoliko različitih varijanti. Nama je potrebna varijanta koja poseduje CSS i JS fajlove spremne za korišćenje, te je stoga neophodno odabrati dugme *Download*, u okviru odeljka Compiled CSS i JS (slika 1.6).



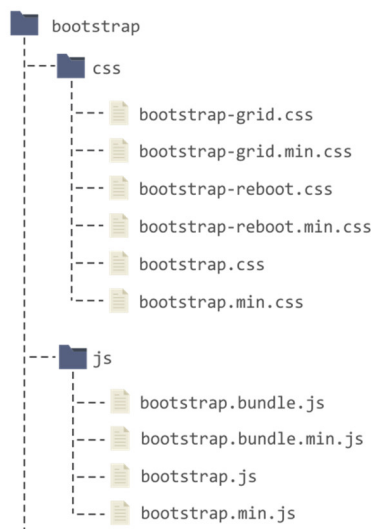
Slika 1.6. Preuzimanje Bootstrapa (1)

Klikom na *Download* dugme, preuzimanje .zip arhive sa Bootstrap fajlovima započinje.

Bootstrap je konceptualno podeljen na sledeće oblasti:

- **Layout** – stilizacija i komponente za kreiranje moćnog Grid sistema za raspoređivanje elemenata web sajtova;
- **Content** – stilizacija koja se odnosi na osnovne HTML elemente za prikazivanje sadržaja, čime se postiže uniforman izgled na svim pregledačima;
- **Componentes** – skup gotovih komponentata za realizaciju manjih ili većih delova korisničkog okruženja web sajtova (dugmići, modalni prozori, kartice, navigacija...);
- **Utilities** – skup pomoćne stilizacije opšte namene, koju je moguće primeniti na gotovo svakom elementu i time uštedeti vreme.

Četiri upravo navedene oblasti predstavljaju grubu podelu svega onoga što postoji unutar Bootstrapa. Ipak, za početak je ovakva podela i više nego korisna, jer nam omogućava da razumemo ulogu različitih fajlova koji se nalaze unutar Bootstrap paketa. Naime, kada se preuzeta .zip arhiva sa Bootstrap fajlovima raspakuje, dobija se struktura ilustrovana slikom 1.7.



Slika 1.7. Struktura Bootstrap fajlova

Bootstrap fajlovi su podeljeni u dva foldera: `css` i `js`. Naravno, unutar foldera `css` nalaze se fajlovi sa CSS kodom, a unutar `js` foldera fajlovi sa kodom jezika JavaScript.

Svi Bootstrap fajlovi započinju prefiksom `bootstrap-`. Nakon ovog prefiksa, fajlovi poseduju nekoliko različitih odrednica, koje ukazuju na namenu fajlova. Sve ovo na kraju omogućava da se Bootstrap koristi u nekoliko različitih oblika, odnosno da se koriste sve Bootstrap funkcionalnosti ili samo neke od njih.

Uloge različitih CSS Bootstrap fajlova predstavljene su u tabeli 1.1.

Fajlovi	Značenje
bootstrap.css bootstrap.min.css	fajlovi koji omogućavaju korišćenje svih Bootstrap funkcionalnosti; fajl sa odrednicom <code>min</code> poseduje <u>minimiziran</u> kod; tako oba fajla sadrže identične funkcionalnosti, a u <u>produkcionim</u> uslovima je najbolje koristiti <code>min</code> fajl, zbog njegove manje veličine
bootstrap-grid.css bootstrap-grid.min.css	fajlovi koji omogućavaju korišćenje isključivo Bootstrap Grid sistema za raspoređivanje elemenata sajta; ostali delovi (stilizacija sadržaja, komponente i pomoćna stilizacija) nisu uključeni u ove fajlove; uloga <code>min</code> fajla identična je kao i u prethodnom redu
bootstrap-reboot.css bootstrap-reboot.min.css	fajlovi koji poseduju samo stilizaciju za osnovne HTML elemente, čime se postiže uniforman izgled na svim web pregledačima; tako ovi fajlovi ne omogućavaju raspoređivanje elemenata ili korišćenje Bootstrap komponenta

Tabela 1.1. Uloga različitih CSS Bootstrap fajlova

Većina Bootstrap mogućnosti realizovana je korišćenjem CSS, odnosno Sass jezika. Ipak, određena ponašanja nije moguće postići isključivo upotrebom stilizacije, pa su stoga sastavni deo Bootstrapa i neki JavaScript fajlovi. Njihova uloga je ilustrovana tabelom 1.2.

Fajlovi	Značenje
bootstrap.js bootstrap.min.js	fajlovi sa osnovnom JavaScript logikom koja se koristi za funkcionisanje nekih od komponenata, ali i za programabilnu kontrolu
bootstrap.bundle.js bootstrap.bundle.min.js	fajlovi koji pored osnovne JavaScript logike sadrže i Popper.js JavaScript biblioteku, koja pomaže prilikom pozicioniranja svih onih prozora koji se prikazuju preko osnovnog sadržaja web dokumenta (<u>tooltip</u> , <u>popover</u> , <u>drop-down</u>)

Tabela 1.2. Uloga različitih JavaScript Bootstrap fajlova

Integracija Bootstrapa u projekat

Nakon upoznavanja sa ulogom različitih fajlova koji se dobijaju sa Bootstrapom, moguće je preći na njihovu integraciju u projekat. Preduslov za korišćenje Bootstrapa jeste referenciranje jednog ili više fajlova u HTML dokumentu koji će koristiti Bootstrap funkcionalnosti. Koliko fajlova će biti uključeno u HTML dokument – isključivo zavisi od komponenata Bootstrapa koje će biti korišćene. Tako postoji dosta kombinacija, koje u ovom trenutku mogu delovati zbunjujuće. Ipak, boljim upoznavanjem Bootstrapa tokom narednih lekcija, bićete u mogućnosti da uvidite razlike između različitih načina za integraciju:

- ukoliko će se Bootstrap koristiti samo za osnovno raspoređivanje elemenata web sajta, dovoljno je u HTML dokument uključiti samo `bootstrap-grid.css` ili `bootstrap-grid.min.css` fajl;
- kada se Bootstrap koristi samo za postizanje uniformnog izgleda osnovnih HTML elemenata, dovoljno je obaviti uključivanje `bootstrap-reboot.css` ili `bootstrap-reboot.min.css` fajla;
- ukoliko se planira korišćenje punog skupa Bootstrap funkcionalnosti, potrebno je u HTML dokument uključiti `bootstrap.css` ili `bootstrap.min.css` fajl, ali i jedan od nešto ranije prikazanih JavaScript fajlova; s obzirom na to da se JavaScript Bootstrap logika zasniva na jQuery i Popper.js bibliotekama, neophodno je osigurati i njihovo uključivanje; kada se iskoristi `bootstrap.bundle.js` ili `bootstrap.bundle.min.js` fajl, Popper.js je automatski uključen; ipak, nijedan od Bootstrap JavaScript fajlova ne poseduje jQuery, pa je ovu biblioteku svakako neophodno samostalno referencirati.

Napomena

U nastavku će biti ilustrovani pristupi za integraciju Bootstrapa sa punim skupom funkcionalnosti. Pristupe koji obezbeđuju redukovani skup funkcionalnosti potrebno je koristiti onda kada se unapred zna da se neke funkcionalnosti Bootstrapa neće koristiti. U svakom slučaju, tokom rada na projektu, uvek je moguće promeniti način na koji je Bootstrap integrisan, pa tako ukinuti nepotrebne ili dodati nove Bootstrap funkcionalnosti.

Bootstrap fajlove je u HTML dokument moguće uključiti na dva načina:

- korišćenjem CDN-a i
- lokalno, smeštanjem fajlova na lokaciju na kojoj se nalazi i sam sajt.

Osnovna razlika između dva navedena pristupa ogleda se u lokaciji na kojoj se Bootstrap fajlovi nalaze. CDN omogućava korišćenje fajlova koji se nalaze na Bootstrap ili nekim drugim udaljenim serverima, dok lokalni pristup podrazumeva smeštanje fajlova na server na kome se nalazi i sam sajt.

CDN integracija

CDN pristup podrazumeva korišćenje fajlova koji se nalaze na specijalizovanim serverima čija je uloga da omoguće visoku raspoloživost fajlova. Prvo je neophodno obaviti uključivanje Bootstrap CSS fajla, dodavanjem link elementa unutar head sekcije:

```
<link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-
Vkoo8x4CGsO3+Hhvxv8T/Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"
crossorigin="anonymous">
```

Već je rečeno da mnoge Bootstrap komponente zahtevaju i JavaScript fajlove, koji se naposljetku oslanjaju na jQuery i Popper.js biblioteke. Stoga je i ove fajlove potrebno uključiti u HTML, dodavanjem script elemenata na kraj body dela:

```
<script src="https://code.jquery.com/jquery-3.4.1.slim.min.js"
integrity="sha384-
J6qa4849bLE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwjlyYfoRSJoZ+n"
crossorigin="anonymous"></script>

<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js" integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>

<script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js" integrity="sha384-
wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYdliqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
```

Lokalna integracija

Nešto ranije su ilustrovani koraci za preuzimanje Bootstrap fajlova na sopstveni kompjuter. Upotreba takvih fajlova, odnosno njihovo fizičko smeštanje unutar projekta i referenciranje u HTML dokumentu, podrazumeva lokalnu integraciju Bootstrapa. Za lokalnu integraciju biće iskorišćeni fajlovi koji obezbeđuju kompletan skup Bootstrap funkcionalnosti (pogledajte tabele 1.1. i 1.2).

Preporuka je da se unutar projekta prvo kreiraju zasebni folderi za CSS i JS fajlove, sa nazivima `css` i `js` (slika 1.8).



Slika 1.8. Struktura projekta

Unutar foldera `css` potrebno je smestiti `bootstrap.min.css` fajl, a unutar foldera `js` fajl `bootstrap.bundle.min.js`. S obzirom na to da Bootstrap zahteva i jQuery biblioteku, koja se ne isporučuje kao sastavni deo Bootstrapa, nju je potrebno posebno preuzeti i smestiti unutar foldera `js`.

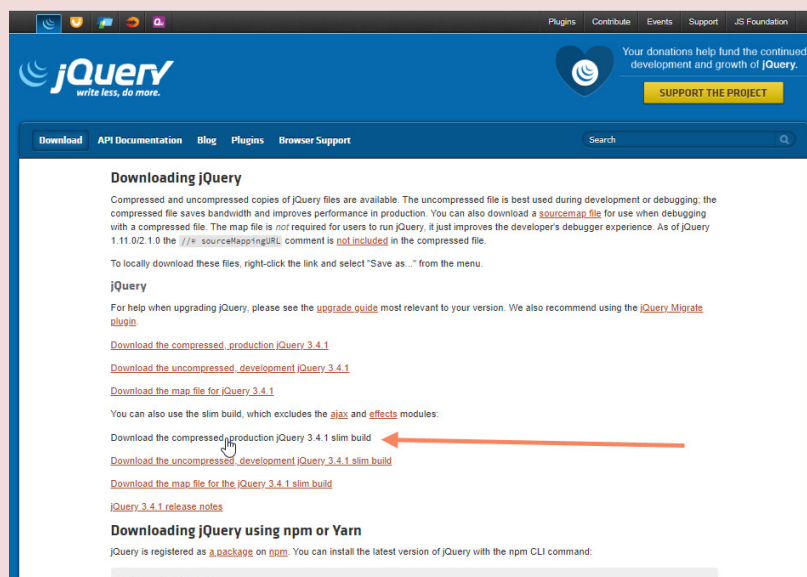
Preuzimanje jQuery biblioteke

jQuery je moguće preuzeti sa zvaničnog sajta:

<https://jquery.com/download/>

Baš kao i Bootstrap, i jQuery postoji u nekoliko različitih verzija. Osnovna podela je na punu i *slim* varijantu. Za korišćenje Bootstrapa, dovoljna je *slim* varijanta, iz koje su isključeni moduli za rad sa Ajaxom i efektima.

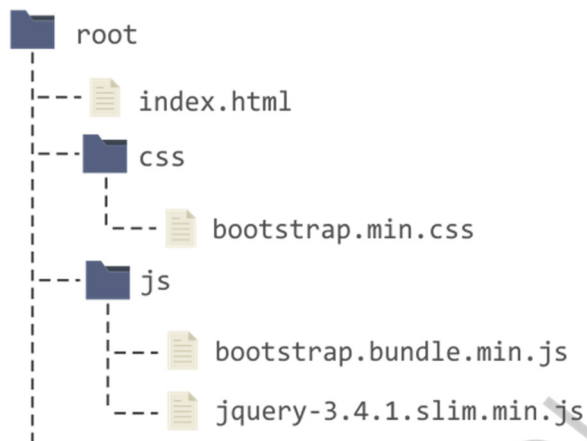
Kako bi se osigurala minimalna veličina fajla, potrebno je odabrati produkcionu, kompresovanu varijantu (slika 1.9).



Slika 1.9. Preuzimanje jQuery biblioteke

Kako bi se `.js` fajl koji predstavlja jQuery sačuvao na lokalnom kompjuteru, potrebno je izvršiti desni klik na link sa slike 1.9 i odabrati opciju *Save link as...* Na taj način će fajl koji predstavlja jQuery biblioteku biti preuzet na Vaš kompjuter.

Nakon preuzimanja jQuery biblioteke, nju je potrebno smestiti unutar `js` foldera koji je kreiran nešto ranije. Na taj način biće stvorena struktura projekta kao na slici 1.10.



Slika 1.10. Struktura projekta unutar koga se koristi Bootstrap

Nakon postavljanja fajlova na njihovo mesto, potrebno je uključiti ih u HTML dokument (`index.html`). Tako je unutar `head` sekcije dokumenta potrebno smestiti sledeći link element:

```
<link rel="stylesheet" href="css/bootstrap.min.css">
```

Na kraj `body` elementa, potrebno je dodati sledeće `script` elemente:

```
<script src="js/jquery-3.4.1.slim.min.js"></script>
<script src="js/bootstrap.bundle.min.js"></script>
```

Redosled uključivanja CSS i JS fajlova

Redosled uključivanja CSS i JS fajlova neophodnih za funkcionisanje Bootstrapa veoma je bitan. Kada je reč o CSS fajlu (`bootstrap.min.css`), preporuka je da se on uključi pre bilo kog drugog fajla za stilizovanje. Naime, prilikom razvoja web sajtova, bez obzira na to da li se koristi Bootstrap ili ne, uglavnom se određeni deo stilizacije samostalno definiše. Takva stilizacija se obično smešta unutar eksternog fajla `style.css` ili nekog drugog fajla sličnog naziva. Kako bi stilizacija unutar takvog fajla uspešno funkcionisala, odnosno kako bi se sprečilo da Bootstrap stilizacija dobije prioritet nad našom stilizacijom, fajl sa stilizacijom koju samostalno definišemo je potrebno uključiti nakon Bootstrap CSS fajla. Razlog je i više nego jasan – ukoliko unutar dva različita CSS fajla postoje opisi sa identičnim selektorima, uvek će biti iskorišćen onaj koji je kasnije naveden, odnosno onaj koji se nalazi unutar fajla koji je kasnije uključen u HTML.

Slična je situacija i sa JS fajlovima koji su neophodni za funkcionisanje Bootstrapa. jQuery je potrebno uključiti pre Bootstrap JavaScript fajla, pošto Bootstrap JavaScript logika zahteva postojanje jQuery biblioteke.

Podešavanje Bootstrap projekta

Kako bi se osiguralo ispravno funkcionisanje Bootstrapa, neophodno je obaviti još neke intervencije na HTML dokumentu unutar koga će biti korišćen Bootstrap. Za ispravno funkcionisanje Bootstrapa, na početku HTML dokumenta je neophodno navesti HTML5 doctype deklaraciju:

```
<!doctype html>
```

S obzirom na to da se Bootstrap zasniva na načelima *responsive dizajna*, unutar `head` dela je potrebno postaviti odgovarajući `meta` element za podešavanje vidnog polja (*viewport*):

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

`meta` element može se koristiti za definisanje različitih osobina HTML dokumenata, od kojih neke utiču na način na koji će pregledač obaviti parsiranje stranice. Kada se za vrednosti `meta` atributa `name` postavi `viewport`, pregledaču se stavlja do znanja da želimo da utičemo na osobine vidnog polja. Tako se prikazanom linijom obavlja sledeće:

- širina vidnog polja, koja se označava kao `width`, dobija vrednost `device-width`; tako se širina vidnog polja postavlja na vrednost softverske, CSS širine uređaja;
- inicijalno uvećanje se postavlja na vrednost 1, što praktično znači da neće biti nikakvog uvećavanja prikaza.

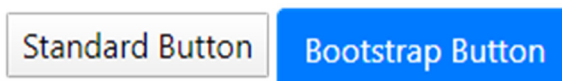
Provera funkcionisanja Bootstrapa

Za kraj ove lekcije, biće obavljena provera uspešnosti integracije Bootstrapa unutar HTML dokumenta. Na početak `body` dela potrebno je postaviti sledeći HTML kod:

```
<button type="button">Standard Button</button>  
<button type="button" class="btn btn-primary">Bootstrap  
Button</button>
```

Korišćenjem dve prikazane linije HTML koda, kreiraju se dva `button` elementa. Prvi je klasični `button` element, bez ikakvih dodatnih atributa (osim standardnog atributa `type`, kojim se definiše tip dugmeta). Ipak, na drugom `button` elementu postavljen je atribut `class`, kojim su ovom elementu dodeljene dve klase. Reč je o klasama koje postoje unutar Bootstrapa, a kojima se kreira jedno Bootstrap dugme. Inače, ovo je osnovni princip korišćenja Bootstrapa – na elemente se postavljaju specifične klase, čime se dobija posebna vizuelna reprezentacija ili ponašanje.

Bez obzira na pristup korišćen za integraciju (CDN ili lokalno smeštanje fajlova), rezultat treba da bude identičan (slika 1.11).



Slika 1.11. Standardno i Bootstrap dugme

Rezime

- Bootstrap su kreirali Mark Otto i Jacob Thornton, kao interni alat kompanije Twitter za postizanje uniformnog izgleda i ponašanja web aplikacija ove kompanije.
- Bootstrap olakšava kreiranje web sajtova, tako što poseduje veliku količinu već napisane stilizacije, koju programeri mogu koristiti prilikom kreiranja sajtova.
- Bootstrap poseban akcenat stavlja na *mobile first* i *responsive design*.
- Bootstrap je sačinjen iz nekoliko CSS i JavaScript fajlova.
- Osnovna funkcionalnost Bootstrapa jeste mogućnost lakog raspoređivanja elemenata korišćenjem Bootstrap Grida.
- Bootstrap je moguće preuzeti i njegove fajlove ručno integrisati u bilo koji web sajt, a mogu se koristiti i Bootstrap fajlovi koji su hostovani na različitim CDN serverima.
- HTML dokument unutar koga se koristi Bootstrap mora imati HTML5 doctype deklaraciju - `<!doctype html>`.
- HTML dokument unutar koga se koristi Bootstrap mora imati definisan meta atribut za podešavanje vidnog polja - `<meta name="viewport" content="width=device-width, initial-scale=1">`.
- Osnovni princip korišćenja Bootstrapa podrazumeva dekorisanje HTML elemenata različitim klasama.

