



WordPress

na

Bootstrap 3.x



Igor Benić

Front End Developer
@igorbenic

Godina: 2014

WPB3 - Wordpress na Bootstrapu 3.x

Kako napraviti svoj prvi Wordpress projekt koristeći Bootstrap Framework

Igor Benić

This book is for sale at <http://leanpub.com/wpb3>

This version was published on 2014-09-22



This is a [Leanpub](#) book. Leanpub empowers authors and publishers with the Lean Publishing process. [Lean Publishing](#) is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

©2014 Igor Benić

Tweet This Book!

Please help Igor Benić by spreading the word about this book on [Twitter](#)!

The suggested tweet for this book is:

Upravo sam kupio knjigu WordPress na Bootstrap-u 3.x!

The suggested hashtag for this book is [#wpb3](#).

Find out what other people are saying about the book by clicking on this link to search for this hashtag on Twitter:

<https://twitter.com/search?q=#wpb3>

Sadržaj

Predgovor	1
Upoznavanje sa Bootstrap Frameworkom	2
Korišćenje Bootstrap-a	2
Grid	2
Responzivne slike	4
CSS Komponente	4
Javascript	5
Prilagođen Bootstrap	7
Napredno prilagođavanje Bootstrapa	9
Raspakirani Bootstrap Izvorni kod	11
Korišćenje less-a	12
...	12
Upoznavanje WordPress-a	13
Koncept	13
Instalacija WordPress-a	14
Početne datoteke jedne teme	14
Razumijevanje hijerarhije datoteka	15
Glavna petlja - The Loop	16
Action i Filter funkcije	17

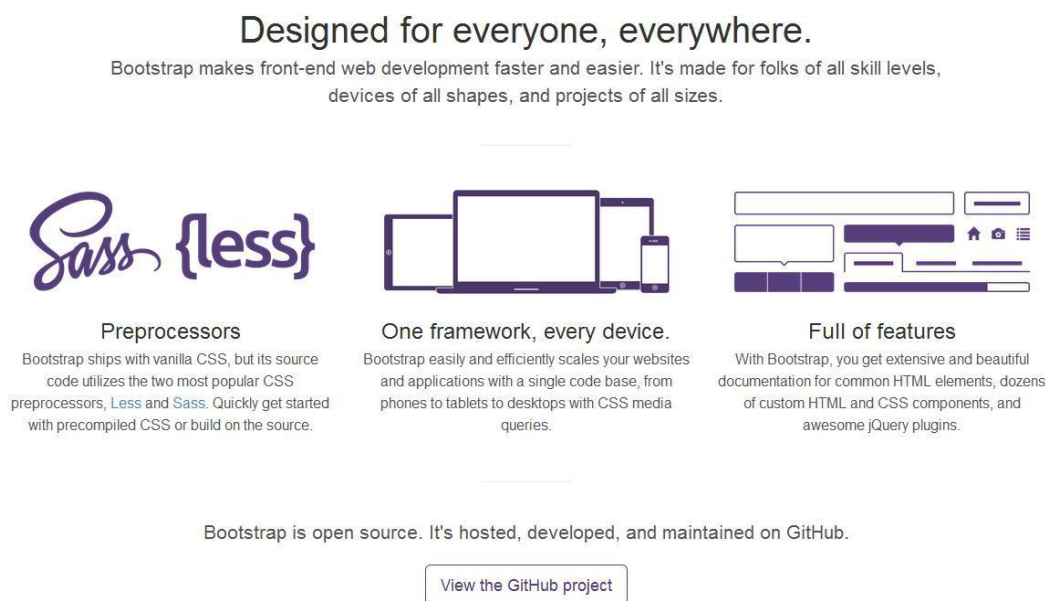
Predgovor

WordPress je jedan od vodećih CMS sustava u Blogging svijetu pa i šire. Danas se WordPress sve više koristi kod izrada raznih web aplikacija koje su daleko kompleksnije od jednog bloga za što je WordPress u početku bio namijenjen.

Ova knjiga namijenjena je za sve developere, a i dizajnere koji se žele okušati u WordPress-u i naučiti kako izraditi jedan mali projekt u WordPress-u. Knjiga WordPress na Bootstrapu 3.x može pomoći i iskusnijim developerima koji su već radili u WordPressu jer će uz izradu same WordPress teme biti korištene i neke druge korisne metode odnosno funkcije koje mijenjaju neke funkcije u samom CMS-u.

Upoznavanje sa Bootstrap Frameworkom

Bootstrap je Framework za bržu izradu web stranica pomoću njegovih unaprijed definiranih stilova, elemenata, komponenti te funkcionalnosti. Kako bi mogli izraditi WordPress na Bootstrap-u potrebno je najprije upoznati se sa samim Frameworkom. Verzija Bootstrap-a u trenutku pisanja ove knjige je 3.2.0. Ovaj Framework možete skinuti ili se s njime pobliže upoznati na njegovoj stranici <http://getbootstrap.com>¹.



slika

Korištenje Bootstrap-a

Bootstrap je veoma jednostavan za koristiti. Dovoljno je pozvati njegove datoteke i prizivati odnosno dodavati komponente koje su nam potrebne. Na takav način jednostavno je dodati par redaka sa strukturom koja sadrži klase koje Bootstrap koristi.

Grid

Grid je naša struktura stranice. Njime se definira naš layout u koji se zatim dodaju razne komponente Bootstrap-a. Početna struktura sastoji se od tri dijelova - *container*, *row* i *column*. Unutar *column*-a unosimo sadržaj.

¹<http://getbootstrap.com>

Klasična struktura jednog bloga je lijevi dio s glavnim sadržajem i desnim dijelom s nekim dodatnim sporednim informacijama. To bi se postiglo sa sljedećim kodom:

```
1 <div class="container">
2     <div class="row">
3         <div id="primary-content" class="col-sm-8">
4             Glavni sadržaj
5         </div>
6         <aside id="secondary-content" class="col-sm-4">
7             Sporedni sadržaj
8         </aside>
9     </div>
10 </div>
```

Jedan redak sastoji se od 12 stupaca. Ovakvom strukturom rekli smo da naš glavni sadržaj koristi 8/12 prostora retka, dok sporedni sadržaj koristi 4/12 prostora, što ukupno daje 12 stupaca.

Bootstrap je responzivan pa tako ima i svoje točke prekidanja (*eng. breakpoint*) na kojima se sadržaj lomi, presavija i sl. Te točke po *default*-u su **< 768px**, **>= 768**, **>= 992px** i **>= 1200px**. Ove točke prekidanja odnose se na širinu ekrana na kojemu se web prikazuje. Na način na koji je napravljen Bootstrap odnosno na način na koji su postavljene točke prekidanja, radi se s Mobile-First Frameworkom te zbog toga ne vidimo nigdje točke prekidanja koje gledaju na **< 481px** jer se smatra da se za to definira sadržaj u početku.

Upravo zbog tih točaka prekidanja postoje različite klase **xs**, **sm**, **md**, **lg**. Klasu **sm** možete vidjeti u gornjem primjeru. To znači da će sadržaj iznad **768px** biti prikazan upravo u ovakvoj strukturi, dok sadržaj ispod te širine, koji zahtjeva klasu **xs** za dodatnu definiciju strukture će biti jedan ispod drugog sa širinom namještenom na 12/12 odnosno svaki stupac biti će jednak širini 12 stupaca tj. na 100% širine jer nema definiranu strukturu za tu širinu. Isti učinak dobili bi kada bi napisali sljedeće:

```
1 <div class="container">
2     <div class="row">
3         <div id="primary-content" class="col-sm-8 col-xs-12">
4             Glavni sadržaj
5         </div>
6         <aside id="secondary-content" class="col-sm-4 col-xs-12">
7             Sporedni sadržaj
8         </aside>
9     </div>
10 </div>
```

Više o gridu možete saznati na [Bootstrap Grid²](http://getbootstrap.com/css/#grid).

²<http://getbootstrap.com/css/#grid>

Responzivne slike

Bootstrap 3 pruža jednostavan način za postizanje responzivnih slika. Dovoljno je za sliku dodatni klasu `.responsive-img` kako bi ona dobila širinu elementa u kojemu se nalazi te se skalirala tj. mijenjala dimenzija u ovisnosti o elementu u kojemu se nalazi.

```
1 
```

O jednostavnim CSS dijelovima koje omogućuje Bootstrap 3 više možete pronaći na [Bootstrap CSS³](#).

CSS Komponente

Uz jednostavne CSS klase koje omogućuju lakše slaganje Bootstrap pruža i dodatne kompleksnije CSS komponente koje mogu poslužiti za što brži razvoj nekakvih modula kao što je npr. navigacija i još uz to, responzivna navigacija koja se na manjim ekranima otvara na klik. Sve što je potrebno je kopirati i zalijepiti sljedeći kod:

```
1 <nav class="navbar navbar-default" role="navigation">
2   <div class="container-fluid">
3     <!-- Brand and toggle get grouped for better mobile display -->
4     <div class="navbar-header">
5       <button type="button" class="navbar-toggle" data-toggle="collapse" data\
6 -target="#bs-example-navbar-collapse-1">
7         <span class="sr-only">Toggle navigation</span>
8         <span class="icon-bar"></span>
9         <span class="icon-bar"></span>
10        <span class="icon-bar"></span>
11      </button>
12      <a class="navbar-brand" href="#">Logo Tekst ili Slika</a>
13    </div>
14
15    <!-- Collect the nav links, forms, and other content for toggling -->
16    <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
17      <ul class="nav navbar-nav">
18        <li class="active"><a href="#">Link</a></li>
19        <li><a href="#">Još jedan Link</a></li>
20        <li class="dropdown">
21          <a href="#" class="dropdown-toggle" data-toggle="dropdown">Padajući \
22 izbornik <span class="caret"></span></a>
23          <ul class="dropdown-menu" role="menu">
24            <li><a href="#">Jedan link</a></li>
25            <li><a href="#">Drugi link</a></li>
26            <li><a href="#">Treći Link</a></li>
```

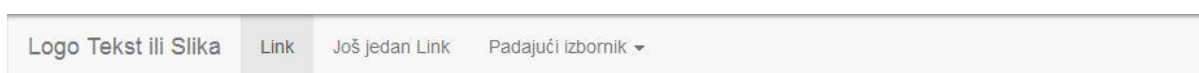
³<http://getbootstrap.com/css/>

```

27         <li class="divider"></li>
28         <li><a href="#">Odijeljeni Link</a></li>
29         <li class="divider"></li>
30         <li><a href="#">Odijeljeni Link No. 2</a></li>
31     </ul>
32 </li>
33 </ul>
34 </div><!-- /.navbar-collapse -->
35 </div><!-- /.container-fluid -->
36 </nav>

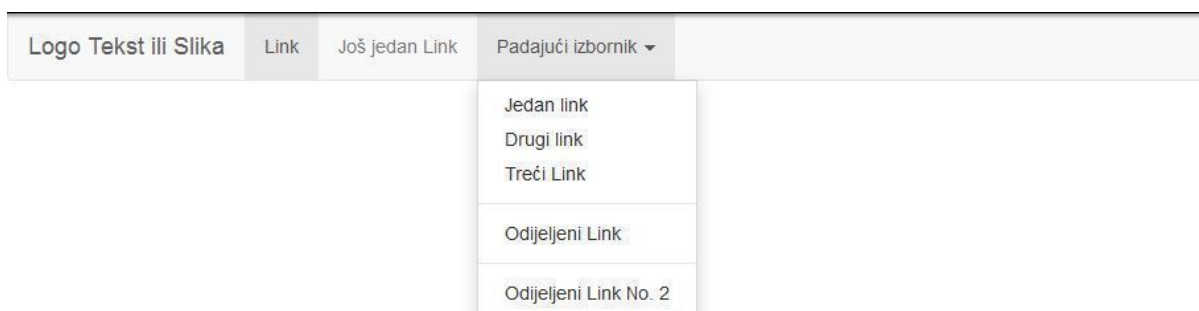
```

Uz ovakav kod dobiti ćemo meni s padajućim izbornikom koji se otvara na klik mišem ili prstom ako koristimo mobitel ili tablet.



Bootstrap Navigacija

Ista navigacija s otvorenim padajućim izbornikom:



Bootstrap Navigacija i padajući zbornik

Kako bi ovaj padajući zbornik mogao funkcionirati kako spada potrebno je koristiti javascript pluginove koje pruža Bootstrap 3. Plugin koji se ovdje koristi je tzv. collapse.

Sve ostale CSS Komponente možete pronaći na [Bootstrap CSS Komponente](#)⁴.

Javascript

Kao što je prikazano u prethodnom poglavlju s navigacijom, Bootstrap dolazi zapakiran i s Javascript pluginovima koji daju dodatne funkcionalnosti. Na takav način možemo lako pozvati neke dijelove s CSS klasama i HTML atributima koje izvršavaju dodatne funkcionalnosti jer su usko povezane s pluginovima, kao što je npr. [carousel plugin](#)⁵.

⁴<http://getbootstrap.com/components>

⁵<http://getbootstrap.com/javascript/#carousel>

Dovoljno je uzeti njihov kod iz primjera, prilagoditi ga malo i dobiti ćemo funkcionalan slider.

```

1  <div id="carousel-example-generic" class="carousel slide" data-ride="carousel\
2  ">
3    <!-- Indicators -->
4    <ol class="carousel-indicators">
5      <li data-target="#carousel-example-generic" data-slide-to="0" class="acti\
6  ve"></li>
7      <li data-target="#carousel-example-generic" data-slide-to="1"></li>
8      <li data-target="#carousel-example-generic" data-slide-to="2"></li>
9    </ol>
10
11   <!-- Wrapper for slides -->
12   <div class="carousel-inner">
13     <div class="item active">
14
15       <h1>Ovo je slider</h1>
16       <div class="carousel-caption">
17         ...sdasdsadada
18
19       <br/>
20       sadijasidjaidsjaidjas
21     </div>
22   </div>
23   <div class="item ">
24
25     <h1>Ovo je slider</h1>
26     <div class="carousel-caption">
27       ...sdasdsadada
28
29     <br/>
30     sadijasidjaidsjaidjas
31   </div>
32 </div>
33
34   <!-- Controls -->
35   <a class="left carousel-control" href="#carousel-example-generic" role="but\
36  ton" data-slide="prev">
37     <span class="glyphicon glyphicon-chevron-left"></span>
38   </a>
39   <a class="right carousel-control" href="#carousel-example-generic" role="bu\
40  tton" data-slide="next">
41     <span class="glyphicon glyphicon-chevron-right"></span>
42   </a>
43 </div>

```

Sada to izgleda kao ovo:



Carousel Slider

Uz takve pluginove moguće je nadodati dosta zanimljivih i korisnih funkcionalnosti. O njima možete više pročitati na [Bootstrap Javascript](#)⁶.

Prilagođen Bootstrap

Bootstrap na svojoj web stranici daje mogućnost prilagođavanja Bootstrap-a svojim željama i potrebama. Dakle, moguće je namještati razne varijable od boja do veličina fonta kako bi si kasnije olakšati izradu tema na Bootstrap-u. To je moguće na njihovoj stranici: <http://getbootstrap.com/customize/>⁷.

Ako poznajemo potrebe našeg projekta moguće je ovdje postaviti sve potrebne postavke te odabrati samo one Javascript pluginove koji su nam potrebni u tome projektu kako ne bi uzimali komponente koje nam nisu potrebne i tako podizali veličinu našeg projekta bez razloga.

Recimo na primjer da sve što nam treba od Bootstrap-a je njegov Grid i nešto responzivnih css klasa. Dovoljno je kliknuti na Toggle All i odabrati samo **Grid** i **Responsive utilities** kao na slici ispod.

⁶<http://getbootstrap.com/javascript>

⁷<http://getbootstrap.com/customize/>

Less files

[Toggle all](#)

Choose which Less files to compile into your custom build of Bootstrap. Not sure which files to use? Read through the [CSS](#) and [Components](#) pages in the docs.

Common CSS

- ☐ Print media styles
- ☐ Typography
- ☐ Code
- ☒ Grid system
- ☐ Tables
- ☐ Forms
- ☐ Buttons
- ☒ Responsive utilities

Components

- ☐ Glyphicons
- ☐ Button groups
- ☐ Input groups
- ☐ Navs
- ☐ Navbar
- ☐ Breadcrumbs
- ☐ Pagination
- ☐ Pager
- ☐ Labels
- ☐ Badges
- ☐ Jumbotron
- ☐ Thumbnails
- ☐ Alerts
- ☐ Progress bars
- ☐ Media items
- ☐ List groups
- ☐ Panels

JavaScript components

- ☐ Component animations (for JS)
- ☐ Dropdowns
- ☐ Tooltips
- ☐ Popovers
- ☐ Modals
- ☐ Carousel

Prilagođeni Bootstrap

Ako su nam potrebne svi ili pojedini jQuery plugini malo ispod ovih **Less datoteka** potrebno je na isti način odabrati željene jQuery pluginove ili ostaviti kako je.

Nakon toga možemo prilagoditi **Grid** našim potrebama mijenjajući vrijednosti varijabla koje se odnose na Grid.

Grid system

Define your custom responsive grid.

@grid-columns

12

Number of columns in the grid.

@grid-gutter-width

30px

Padding between columns. Gets divided in half for the left and right.

@grid-float-breakpoint

@screen-sm-min

Point at which the navbar becomes uncollapsed.

@grid-float-breakpoint-max

(@grid-float-breakpoint - 1)

Point at which the navbar begins collapsing.

Container sizes

Define the maximum width of `.container` for different screen sizes.

@container-tablet

((720px + @grid-gutter-width))

@container-sm

@container-tablet

For `@screen-sm-min` and up.

@container-desktop

((940px + @grid-gutter-width))

@container-md

@container-desktop

For `@screen-md-min` and up.

@container-large-desktop

((1140px + @grid-gutter-width))

@container-lg

@container-large-desktop

For `@screen-lg-min` and up.

Prilagođeni Bootstrap

Nakon što smo promijenili sve što je trebalo promijeniti treba pritisnuti na **Compile and Download** i dobiti ćemo u .zip formatu datoteku koju je potrebno skinuti te koja sadrži naše .css datoteke i .js datoteke.

Nakon toga potrebno je polinkati našu web stranicu s tim datotekama i možemo je jednostavno razvijati na Bootstrap 3 Framework-u.

Napredno prilagođavanje Bootstrapa

Uz njihov alat za prilagođavanje Bootstrap-a, moguće je skinuti Bootstrap u izvornom formatu odnosno u .less formatu. ### Što je Less? Less je pretprocesor za CSS koji omogućava jednostavnije pisanje CSS-a koristeći grananja, varijable i ostale razne pogodnosti kako bi se ubrzalo pisanje CSS-a. Nakon toga kompajlira se takva datoteka u .css format koja može biti i kompresirana na najmanju moguću veličinu. Time dobivamo brži i lako održivi CSS.

Mali primjer Less-a

.less datoteka

```
1  @crvena_boja: red;
2
3  a { color:black;
4
5    &:hover {
6        color: @crvena_boja;
7    }
8
9  }
10
11  .lista {
12      padding:0;
13
14      li {
15          float:left;
16          display:block;
17      }
18  }
```

.css datoteka

```
1  a { color:black; }
2
3  a:hover { color: red; }
4
5  .lista { padding:0; }
6  .lista li { float:left;          display:block;          }
```

Može se primijetiti da se grananjem i korištenjem varijabli uštedjelo na pisanju ponavljajućih klasa ili boja i sl.

Uz Less, Bootstrap omogućuje korištenje Sass-a koji je također jedan od pretprocesora CSS-a, ali Less je njihov izvorni jezik pa ću se držati toga u ovoj knjizi.

Kako bi dobili cijeli Bootstrap 3 u izvornom obliku sa *less* datotekama potrebno je odabrati **Download Source**.

Download

Bootstrap (currently v3.2.0) has a few easy ways to quickly get started, each one appealing to a different skill level and use case. Read through to see what suits your particular needs.

Bootstrap

Compiled and minified CSS, JavaScript, and fonts. No docs or original source files are included.

[Download Bootstrap](#)

Source code

Source Less, JavaScript, and font files, along with our docs. **Requires a Less compiler and some setup.**

[Download source](#)

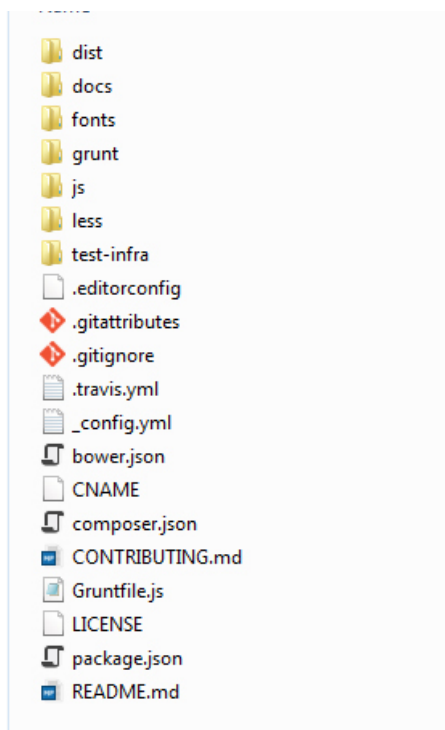
Sass

[Bootstrap ported from Less to Sass](#) for easy inclusion in Rails, Compass, or Sass-only projects.

[Download Sass](#)

Raspakirani Bootstrap Izvorni kod

Ako ste skinuli tj. preuzeli Bootstrap izvorni kod i raspakirali njihovu .zip datoteku, dobili ste nešto slično ovome:



Izvorna datoteka

Ako otvorite prvi direktorij kao na slici **dist** vidjet ćete isto ono što bi dobili ako kompajlirate bootstrap preko njihovog alata na stranici (*css*, *font* i *js* direktorije) uz još neke dodatne datoteke koje Vam ne služe ako ne koristite **Grunt** ili nešto slično. Kako je ova knjiga zamišljena kao priručnik za početnike ili one koji se već bave ovom strukom, preskočit ćemo alate kao **Grunt** i fokusirati se na što jednostavniji pristup kako bi mogli obavljati ono najbitnije, a to je praviti WordPress temu iz Bootstrap-a.

Ako ponovno bacimo pogled na početni direktorij osvrnimo se na ono najbitnije u ovome poglavlju, a to je **less** direktorij. Kada bi uzeli taj direktorij uz **fonts** i **js** dobili bi pravi Bootstrap-ov nekompilirani kod jer u **less** direktoriju nalaze se razne *less* datoteke dok se u **js** direktoriju nalaze razne *js* datoteke. Sve te datoteke u sebi sadrži neki dio *css*-a odnosno *js*-a koji su potrebni bi Bootstrap bio Bootstrap.

Korištenje less-a

Kako je *less* preprocesor *CSS*-a potrebno je koristiti dodatne stvari kako bi mogli iz *less*-a dobiti *css*. Iako je to moguće i preko obične javascript datoteke koja čita *less* datoteke i pretvara to u *css* radi produkcije bolje je odmah raditi pretvorbu *less*-a u *css* i to prikazati na webu bez dodatnih komplikacija.

Kako to napraviti? Dosta jednostavno, mogu se koristiti alati poput [Grunt-a](http://gruntjs.com/)⁸ koji rade preko terminala odnosno komande linije (**command prompt**) ili možete koristiti GUI (eng. *Graphic User Interface*) alate koji obavljaju istu stvar kao Grunt. Jedan takav alat je [Prepros](http://alphapixels.com/prepros/)⁹.

...

⁸<http://gruntjs.com/>

⁹<http://alphapixels.com/prepros/>

Upoznavanje WordPress-a

WordPress je CMS koji je u početku bio zamišljen kao blog platforma. Nakon nekog vremena počeo se koristiti kao platforma za web stranice različitog tipa. Zbog njegove jednostavne uporabe mnogi ga uzimaju u obzir kada započinju svoje web projekte pa čak i ljudi koji nisu developeri već su upoznati s WordPress-om te klijenti znaju sami tražiti da se njihova web stranica radi u WordPress-u. U ovome poglavlju upoznati ćemo se s WordPress-om kao temeljem za izradu web projekta.

Koncept

WordPress kao CMS koncipiran je u nekoliko segmenata. Sastoji se od članaka, stranica, komentara te uz cijeli taj sadržaj omogućava korištenje raznih tema i nadogradnju samog CMS-a pomoću njegovih dodataka.

Članci - su zapisi poput nekih novosti i sl. koji se zatim listaju, često, na početnoj stranici. Moguće ih je kategorizirati pomoću kategorija i tagova.

Stranice - su zapisi slični člancima ali oni se smatraju kao obične stranice koje se dodaju i u navigaciji WordPressa. Bez dodatne konfiguracije stranice su automatski dodane u navigaciji i tako predstavljaju stranice poput "O nama", "Usluge" i sl. koje su dosta česte statične stranice.

Komentari - su zapisi koje prave korisnici WordPress stranice. Komentare mogu pisati i korisnici koji nisu registrirani na stranici ako je tako omogućeno.

Teme - su različiti predlošci dizajna koji se mogu skinuti i instalirati putem WordPress administracije ili direktno preko sistema direktorija. Sve instalirane teme moguće je pregledati preko WordPress administracije i uključivati po želji. Teme se nalaze u folderu **wp-content/themes/**.

Dodaci - služe za proširivanje funkcija same WordPress stranice. Danas postoje različiti dodaci za svakakve potrebe. Dodaci se nalaze u folderu **wp-content/plugins/**.

Instalacija WordPress-a

Stranica za download WordPressa

WordPress se može skinuti sa stranice wordpress.org¹⁰. Ako niste zadovoljni engleskom verzijom možete skinuti WordPress za neki drugi jezik. Klikom na download dovesti će vas na stranicu za skidanje WordPress-a u .zip ili .tar.gz formatu.

Sam proces konfiguriranja WordPress-a na vašem lokalnom serveru biti će objašnjeno kasnije u knjizi kada se bude radila WordPress tema.

Početne datoteke jedne teme

Kako bi izradili WordPress temu potrebno je razumjeti strukturu jedne teme. Potrebne datoteke za jednu temu kako bi funkcionirala jesu:

- style.css - sadrži sva css pravila koja definiraju izgled teme
- index.php - početna datoteka koja se uvijek prvo gleda ako drugačije nije postavljeno. Često se koristi za početnu stranicu gdje se listaju svi članci.

Obično uz ove datoteke imamo i sljedeće:

- header.php - sadrži dijelove teme koji se ponavljaju kao npr. navigacija, logo i sl. na vrhu teme

¹⁰<http://www.wordpress.org>

- footer.php - sadrži dijelove teme koji se ponavljaju u podnožju teme
- sidebar.php - sadrži strukturu sidebar-a koji se koristi u temi, najčešće poziva dinamični sidebar
- comments.php - sadrži strukturu komentara koji se dodaju svakom članku ili stranici gdje su omogućeni komentari

Uz navedene datoteke dobro je imati i sljedeće:

- single.php - prikazuje samo jedan članak kojeg se gleda
- page.php - prikazuje sadržaj jedne stranice
- archive.php - prikazuje listu članaka, često sadrži istu strukturu kao i index.php. Ovo se koristi pri listanju članaka iz pojedine kategorije, taga i sl.
- functions.php - sadrži razne opcije i funkcije vezane uz temu, tu se mogu definirati novi post type-ovi, widgeti i sl.
- 404.php - ova datoteka se koristi u slučaju da stranice ili članak nije pronađen.

Ako je potrebno imati posebnu početnu stranicu za WordPress temu koristi se jedna od sljedećih datoteka:

- home.php
- front-page.php

Razumijevanje hijerarhije datoteka

Moguće je koristiti još datoteka kako bi si olakšali održavanje. Cijelu hijerarhiju datoteka moguće je pogledati na [Chip Bennett Template Hijerarhija](http://www.chipbennett.net/themes/template-hierarchy/)¹¹. Ovdje neću objasniti svaku pojedinu ali iz nekoliko primjera biti će dovoljno jasno da se može čitati dijagram na gornjoj slici.

Arhiva

Arhiva članaka je jedan od najosnovnijih načina prikazivanja članaka u WordPress-u. Prilikom pristupanja nekoj arhivi, za primjer uzet ćemo kategoriju Auti (slug od kategorije je auti, id od kategorije je 2), WordPress čini sljedeće:

- Prepozna je da se radi o arhivi
- Prepozna je da se radi o arhivi kategorije
- Pregledava folder od teme da li postoji datoteka category-*auti*.php, ako postoji koristit će nju
- Ukoliko nema prethodno navedene datoteke, pregledava folder da li postoji category-2.php, ako postoji koristiti će nju
- Ukoliko nema category-2.php, pregledava se folder za category.php datotekom
- Ukoliko nema ni category.php, pregledava se folder za archive.php

¹¹<http://www.chipbennett.net/themes/template-hierarchy/>

- Ukoliko nema archive.php koristi se index.php

Članak

Članak je prikaz jednog članka koji se prikazuje na stranici jer smo kliknuli na njegov naslov ili neki link koji nas je doveo do njega. Taj članak se zatim preko njegovog URL-a ili ID-a prikazuje na stranici. WordPress radi sljedeće:

- Prepoznaje da se radi o stranici koja prikazuje samo sadržaj jednog posta ili stranice
- Prepoznaje da se radi o stranici koja prikazuje samo sadržaj jednog posta
- Pregledava koji tip članka
- Ukoliko je članak tipa "Post", pregledava se folder za datotekom single-post.php
- Ukoliko je članak nekog našeg tipa koji je naknadno dodan, što je poznato kao "Custom Post Type", pregledava se folder za datotekom single-auto.php ukoliko bi naš custom post type bio auto.
- Ukoliko je članak nekog priloga kao npr. uploadane slike tada se pregledava folder za datotekom image.php ukoliko je to datoteka tipa image, zatim ako je nema gleda za png.php ako je slika u formatu .png, a nakon toga gleda ako postoji image_png.php datoteka. Ukoliko nema niti jedne od tih datoteka gleda attachment.php
- Ukoliko nije pronađena neka od spomenutih datoteka pregledava se folder za datotekom single.php
- Ukoliko nema ni single.php, koristi se index.php

Prateći dijagram koji je gore naveden tema se može organizirati u hijerarhiju datoteka koja je lako održiva te je jednostavno shvatiti koja datoteka se odnosi na koji dio teme. Iz gore navedenih primjera može se primijetiti kako sve uvijek vodi na index.php ukoliko tražene datoteke ne postoje. Ako neke promjene nisu pretjerano velike mogu se napraviti i u index.php datoteci kao što je npr. naslov archive za kategoriju.

```
1 //ovo je unutar petlje the_loop()
2 if(is_category()){ single_cat_title("Članci iz kategorije: ", true); }
```

Glavna petlja - The Loop

Glavna petlja ili The Loop je PHP kod koji se koristi za listanje sadržaja članka, stranica i sl. Ova petlja prikazuje svaki članak ili stranicu koji su dobiveni kriterijima postavljenim u glavnom query-u koji pristupa bazi. Taj query se npr. mijenja i ovisno o kategoriji kada se poziva category-auto.php.

Svaki HTML i PHP kod unutar petlje prikazat će se odnosno izvršiti za svaki članak unutar petlje. Koristeći razne **Template Tagove** možemo prikazati različite informacije. Jedne od osnovnih **tagova** kod prikaza informacija za članak unutar petlje jesu:

- the_title() - prikazuje naslov članka

- `the_content()` - prikazuje sadržaj članka, ukoliko nije stranica pojedinog članka onda se prikazuje skraćeni tekst
- `the_category()` - prikazuju se kategorija članka
- `the_date()` - prikazuje se datum članka
- `the_permalink()` - daje cijeli link na članak, ovaj dio se dodaje npr. u atribut **href**

Razni **tagovi** mogu se pronaći na [WordPress Codex-u](#)¹².

Osnovna struktura petlje

```
1 <?php
2 if( have_posts() ) {
3     while( have_posts() ) {
4         the_post();
5         //Ostali sadržaj
6     }
7 }
8 ?>
```

Action i Filter funkcije

Action i Filter funkcije služe za nadograđivanje naše teme ili admin-a. Njima se može određivati da se u određeno vrijeme ili na određenom mjestu neka funkcija izvrši. Action funkcije se izvršavaju kada se nešto dogodi odnosno izvrši kao što je npr. publish članka ili prikazivanje admin stranice. Filter funkcije su funkcije koje se izvršavaju kod podataka kao što je mijenjanje nekih podataka ili filtriranje istih prije nego se spremaju u bazu ili ispisuju korisniku stranice.

Action funkcije

Kako bi pozvali neku funkciju uz neku Action funkciju koristi se

```
1 add_action("ime_actiona", "ime_funkcije", prioritet, broj_parametara );
```

- `ime_actiona` predstavlja ime actiona pri kojem se poziva funkcija
- `ime_funkcije` predstavlja ime funkcije koja se poziva kada se dogodi odgovarajući action
- `prioritet` predstavlja broj, ukoliko ga ne unesemo postavlja se na 10. Što je manji broj to se prije izvršava dodana funkcija
- `broj_parametara` predstavlja broj parametara koji dolaze uz funkciju, npr. `function update_post($a,$b)` - tu imamo broj 2 jer se proslijeđuju dva parametra

Mjesto pozivanja te action funkcije određuje se sa

¹²http://codex.wordpress.org/Template_Tags

```
1 do_action("ime_actiona");
```

Filter funkcije

Za pozivanje Filter funkcije koristi se slični poziv kao i za Action funkcije gdje parametri predstavljaju istu stvar kao i pozivanje Action funkcija:

```
1 add_filter("ime_filtera", "ime_funkcije", prioritet, broj_parametra);
```

Mjesto pozivanja filter funkcije određuje se sa

```
1 do_filter("ime_filtera", $vrijednost, $varijabla1, $varijabla2, ....);
```

- \$vrijednost predstavlja vrijednost koja se šalje u filter i koja se može mijenjati u filteru
- \$varijabla1, \$varijabla2 predstavljaju parametre koji se šalju skupa s vrijednošću, moguće je dodavati još varijabla

Brisanje povezanih funkcija na Action i Filter funkcije

Brisanje povezanih funkcija može se izvršiti sa naredbama

```
1 remove_filter("ime_filtra", "ime_funkcije_za_brisanje");  
2 remove_action("ime_actiona", "ime_funkcije_za_brisanje");
```