

SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

ZAVRŠNI RAD

**IZRADA PORTALA O PROGRAMSKOM JEZIKU PYTHON  
U SUSTAVU ZA UREĐIVANJE WEB SADRŽAJA**

**Student:**

Ivona Banjan

**Mentor:**

doc.dr.sc. Ani Grubišić

Split, rujan 2015.

## Sadržaj

1. Uvod .....	1
2. Sustav za upravljanje sadržajem .....	2
2.1. WordPress.....	3
3. Postavljanje okruženja za oblikovanje sadržaja.....	5
3.1. Xampp – web server .....	5
3.2. Instalacija WordPressa .....	6
4. Oblikovanje sadržaja temeljeno na predlošcima .....	8
4.1. Alati.....	8
4.1.1. HTML .....	8
4.1.2. CSS .....	10
4.1.3. PHP .....	12
4.2. Oblikovanje izgleda sjedišta .....	13
4.3. Datoteke .....	15
4.3.1. header.php .....	17
4.3.2. index.php.....	18
4.3.3. footer.php .....	19
4.3.4. sidebar.php.....	19
4.3.5. page.php.....	19
4.3.6. single.php .....	20
4.3.7. functions.php .....	20
4.3.8. style.css .....	21
4.4. Postavljanje teme .....	21
5. Portal o Pythonu oblikovan primjenom predložaka .....	22
5.1. Sadržaj portala .....	22
5.1.1. Prvo upoznavanje s Pythonom.....	22
5.1.2. Priprema i izvođenje programa.....	23
5.1.3. Struktura programa u Pythonu .....	24
5.1.4. Funkcije, programi i moduli.....	25
5.1.5. Zadatci .....	26
5.2. Izgled portala .....	<b>Error! Bookmark not defined.</b>
6. Zaključak.....	30
7. Literatura.....	31

## 1. Uvod

U današnje vrijeme protok informacija i podataka je jako velik i brz stoga je potreba za upravljanjem istima svakim danom sve veća. Informacije se najčešće nalaze na raznim web sjedištima na kojima možemo prikupljati nove podatke na dnevnoj, tjednoj, mjesečnoj ili nekoj drugoj bazi. Veliku revoluciju vezanu za upravljanje podacima donijeli su sustavi za upravljanje sadržajem koji omogućavaju brzo i jednostavno dodavanje, mijenjanje i brisanje sadržaja na web sjedištima te samim time i korisnicima veću interakciju i lakše snalaženje na istima. WordPress je najkorišteniji takav sustav koji je besplatan te ima otvoren kod, tj. korisnicima nudi mogućnost izmjene sustava prema vlastitim potrebama. Počeo je kao sustav za blog, no s vremenom je evoluirao i postao ono što je danas. Uz WordPress, na internetu je moguće pronaći na tisuće različitih tema koje mogu ili ne moraju zadovoljiti naše potrebe. Međutim, mnogi teže unikatnosti te rade ili naruče izradu vlastite teme koju neće imati nitko drugi.

U ovom radu će biti prikazan način izrade portala o Pythonu što uključuje i samu izradu teme za WordPress. Prije izrade portala, prikazano je postavljanje potrebnog okruženja, objašnjeni su alati potrebni za oblikovanje portala kao i princip izrade teme koja će biti korištena na portalu. Na samom kraju dana je podjela sadržaja portala, kratki opisi lekcija te izgled portala.

## 2. Sustav za upravljanje sadržajem

Povijest sustava za upravljanje sadržajem (CMS, eng. *content management system*) je stara kao i sama povijest sadržaja koja se otprilike podudara s poviješću ljudskog znanja (Anonymous, 2005). U vrijeme kada ljudi nisu znali čitati ni pisati, sva znanja su se prenosila usmenom predajom koja nije bila dovoljna da bi se sačuvale mnoge bitne informacije. S vremenom su nastala prva pisma, samim time i zapisi koji su bili ispisivani na raznim vrstama materijala poput kamena, drva, zatim pergamenta i papirusa uz koje vežemo postojanje svitka koji je nekada predstavljao najčešći oblik pisane informacije. Od tih vremena, sve do danas, ljudi su imali sve veću potrebu za organizacijom i pohranjivanjem različitih vrsta podataka, tako su nastajale knjižnice i arhive koje su zahtijevale mehaničku obradu informacija. Iz današnjeg aspekta, sustavi poput knjižnica djeluju jako spori jer razvojem računarstva te osnivanjem i naglim razvojem interneta, pristup i obrada informacija se u potpunosti promijenila. Brzina kretanja istih postaje neusporedivo veća od same brzine fizičkog kretanja što uvelike utječe na poboljšanje sustava za upravljanje sadržajem koji nam u odnosu na prijašnje sustave donose sedam znatno unaprijeđenih karakteristika (Gibbons, 1981):

- Skupljanje podataka
- Pohrana informacija
- Organizacija informacija
- Izračun
- Komunikacija
- Presentacija podataka
- Kontrola

Na samom početku interneta, oblikovale su se isključivo statičke web stranice, za čiju je izmjenu sadržaja bila potrebna osoba koja je upoznata sa prezentacijskim jezikom za izradu istih (HTML, eng. *HyperText Markup Language*) te postavljanjem potrebnih dokumenata na internet. Razvijanjem novih programskih jezika, nastajala su nova, bolja rješenja koja zovemo sustavima za upravljanje web sadržajem (WCMS, eng. *web content management system*). Zahvaljujući navedenim rješenjima, danas korisnici mogu jednostavno dodavati, uređivati ili

brisati sadržaj na svojim web stranicama, a da im pritom nije potrebno tehničko znanje vezano za izradu istih (Vlahović, 2004). Brzim razvojem i korisniku jednostavnim, a naprednim funkcijama WCMS je postao sve neizostavniji dio web stranica. Tome su doprinijeli besplatni sustavi otvorenog koda (eng. *open source*) koje je moguće mijenjati i prilagođavati vlastitim potrebama. Neki od poznatijih su:

- Wordpress
- Joomla!
- Drupal
- Magento

Wordpress je najpopularniji i koristi se u 24.5% svih web stranica (W3Techs, 2015), stoga će u ovom radu biti prikazan postupak instalacije istog kao i izrada i implementacija teme (eng. *theme*) u Wordpress.

## 2.1. WordPress

Kao što je već navedeno, WordPress je besplatni *open source* WCMS, napravljen u PHP-u, programskom jeziku namijenjenom za izradu dinamičkih web stranica te koristeći MySQL bazu podataka. Na samim počecima WordPress je korišten isključivo kao blog sustav sa grafičkim korisničkim sučeljem koji je naknadno evoluirao u CMS. No danas WordPress nudi dvije mogućnosti:

- (i) registraciju na [www.wordpress.com](http://www.wordpress.com) te izradu poddomene (eng. *subdomain*) na kojoj će se nalaziti korisnikov blog ili jednostavna web stranica,
- (ii) preuzimanje (eng. *download*) skripte sa stranice [www.wordpress.org](http://www.wordpress.org) te instalacija iste na udomljeni (eng. *hosting*) prostor na internetu s vlastitom, otkupljenom domenom.

Na prvi pogled sustavi izgledaju veoma slično, no tijekom rada u svakom ponaosob nailazimo na bitne razlike. Naime, preuzimanjem i instalacijom skripte na vlastiti prostor na internetu, pružaju nam se bitno veće mogućnosti već korištenjem servisa na navedenim .com stranicama. Najbitnija prednost je u mogućnosti instalacije na tisuće različitih dodataka (eng. *plugin*) koji proširuju i povećavaju funkcionalnost WordPressa te korisniku omogućavaju da

kombinacijom istih stvore jedinstvene web stranice prema vlastitoj zamisli. Osim dodataka, korisnik ima opciju odabira teme koja omogućava da korisnikove web stranice izgledaju baš onako kako on to želi. Teme i dodatci mogu biti besplatni ili komercijalni, ali isto tako, osoba sa znanjem jezika za izradu web stranica, ima mogućnost izraditi iste prema vlastitim potrebama (WordPress, n.d.).

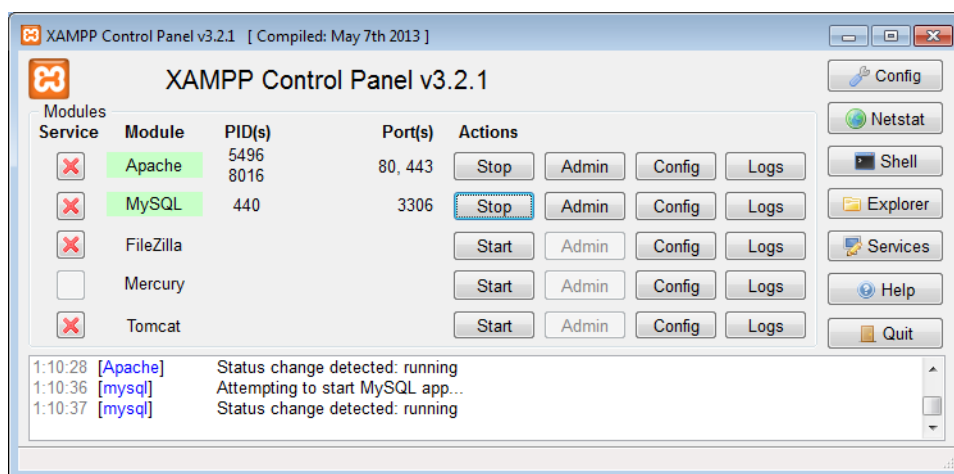
### 3. Postavljanje okruženja za oblikovanje sadržaja

Za instalaciju skripte skinute sa WordPressovih internet stranica potreban nam je web server. Iako se za postavljanje stranica na internet koriste *live* serveri, okruženje u kojem se stranice izrađuju, uglavnom je lokalno. Tako će biti i u ovom radu, stoga prije same instalacije skripte potrebno je instalirati lokalni web server. U ovom slučaju koristiti ćemo xampp.

#### 3.1. Xampp – web server

Xampp je besplatan *open source* web server čiji naziv je akronim od: *cross-platform*, Apache, MySQL, Php i Pearl. *Cross-platform* znači da se server može koristiti na različitim tipovima operacijskih sustava, dok su ostalo tehnologije koje instalacijom istoga možemo koristiti. Prilikom razvoja web stranica, korisnici se najčešće odlučuju na lokalno okruženje na kojem mogu raditi na svojim dokumentima i njihovim izmjenama, a da iste nitko drugi ne vidi dok ne budu konačne.

Sve inačice xampp-a i svi pripadajući dodatci mogu se preuzeti na stranicama [www.apachefriends.com](http://www.apachefriends.com) čiji je cilj promocija Apache web servera (Apache friends, n.d.). Instalacija se izvršava kao i kod svakog drugog programa, no nakon uspješne instalacije, za produktivan rad potrebno je otvoriti instalacijsku mapu (eng. *folder*) na vašem računalu te otvoriti datoteku `xampp-control.exe` (Slika 1) unutar koje je potrebno pokrenuti Apache i MySQL server.



Slika 1 - Prikaz otvorene xampp-control.exe datoteke s pokrenutim Apache i MySQL serverom

## 3.2. Instalacija WordPressa

Instalacija WordPressa na lokalni web server biti će prikazana u pet jednostavnih koraka koji obuhvaćaju preuzimanje skripte, stvaranje baze i korisnika, konfiguraciju `wp-config.php` datoteke te na samom kraju instalaciju WP-a.

### korak 1 – preuzimanje skripte

Prije same instalacije WordPressa, potrebno je otići na web stranice [www.wordpress.org](http://www.wordpress.org), preuzeti instalacijski paket koji dolazi u .zip formatu, te ga naposljetku raspakirati u htdocs mapu koja se nalazi u instalacijskoj mapi Xampp-a.

### korak 2 – kreiranje baze podataka

Nakon preuzimanja i raspakiravanja skripte, potrebno je kreirati bazu podataka koja se može napraviti tako da se otvori web preglednik, u adresnu traku (eng. *address bar*) se upiše localhost te se pritisne enter na tipkovnici, a pritom treba voditi računa da su prethodno uključeni Apache i MySQL serveri (slika 1). Zatim se odabire jezik<sup>1</sup> nakon čega se otvara stranica koja s lijeve strane ima narančasti stupac unutar kojeg je nužno pronaći i kliknuti na poveznicu (eng. *link*) „phpMyAdmin“. Klikom će se otvoriti nova stranica koja na vrhu ima vodoravni izbornik (eng. *menu*) na kojem je potrebno odabrati poveznicu „Databases“. Klikom na istu, otvoriti će se obrazac u kojem se kreira baza podataka. Nakon upisa imena i odabira opcije *collation*, klikom na botun (eng. *button*) „Create“, baza s odabranim imenom će biti stvorena.

### korak 3 – dodavanje korisnika

Nakon kreiranja baze podataka, potrebno je dodati korisnika (eng. *user*) koji će istom „upravljati“. Sa lijeve strane iste stranice na kojoj je napravljena baza, u listi kreiranih bazi potrebno je pronaći prethodno napravljenu te kliknuti na nju. Nakon toga, vodoravni izbornik koji smo u prošlom koraku spominjali se mijenja te samim time pojavljuje se poveznica „Privileges“ koju klikamo da bi odabirom poveznice „Add user“ dobili mogućnost dodavanja novog korisnika. Otvaranjem novog prozora, prikazuje se obrazac u koji treba upisati ime korisnika, odabrati localhost, napisati i potvrditi lozinku korisnika te na samom

---

<sup>1</sup> Tijekom izrade ovog rada odabran je engleski jezik.



kraju pritisnuti na botun „go“. Korisnik je uspješno dodan, dane su mu privilegije nad radom prethodno napravljene baze podataka, stoga možemo prijeći na sljedeći korak.

#### **korak 4 – postavljanje wp-config.php datoteke**

Neposredno prije instalacije, potrebno je konfigurirati `wp-config.php` datoteku koja se nalazi u mapi koju smo spremili na naše računalo prema uputama u koraku 1 ovog podpoglavlja. Datoteku je moguće promijeniti ručno, koristeći neki uređivač za tekst (npr. Notepad), no mi ćemo koristiti jednostavniji način. Za konfiguraciju navedene datoteke potrebno je otvoriti pretraživač u čiju adresnu traku treba upisati `localhost/folder`<sup>2</sup>. Ulaskom u stranicu prikazuje se tekst koji govori da prije instalacije treba popuniti stavke vezane za bazu podataka i korisnika koje smo prethodno kreirali. Klikom na „Započnimo“ pojavljuje se obrazac u koji treba upisati tražene podatke. Odabirom botuna „Pošalji“ konfigurira se `wp-config.php` datoteka nakon čega se može krenuti sa instalacijom WordPress skripte.

#### **korak 5 – instalacija skripte**

Nakon postave datoteke iz prethodnog koraka otvara se novi obrazac u kojem je potrebno upisati naziv web stranice, korisničko ime, lozinku, e-mail te na kraju postoji mogućnost odabira hoće li tražilicama biti dopušteno da indeksiraju web stranicu. Na samom kraju treba kliknuti na „Instaliraj WordPress“, čime je instalacija završena. Stoga, korisnik se još samo treba s prethodno unesenim podacima prijaviti se na stranici `localhost/folder/wp-admin`. Nakon prijave može se početi s uređivanjem stranice.

---

<sup>2</sup> Folder se mijenja prema imenu mape u kojoj se nalaze dokumenti.

## 4. Oblikovanje sadržaja temeljeno na predlošcima

WordPress tema je skup međusobno povezanih dokumenata koji predstavljaju funkcionalnost i dizajn web stranice napravljene koristeći istoimeni WCMS (WordPress, n.d.). Iako se na internetu može pronaći na tisuće besplatnih tema koje možemo koristiti na svom web sjedištu, svi težimo tome da imamo unikatan *site*, napravljen prema našim potrebama. Za izradu istoga potrebno je poznavanje alata koji se koriste za izradu web stranica te svoje znanje prenijeti u dobro strukturirane dokumente s čistim kodom bez greški.

### 4.1. Alati

Za izradu najjednostavnije web stranice potrebna su nam znanja HTML-a i CSS-a, dok za izradu teme u već navedenom sustavu za upravljanje sadržajem potrebna su nam i osnovna znanja PHP-a. U sljedećim podpoglavljima biti će riječ o navedenim jezicima. Također ću se dotaknuti osnova kodiranja u istima.

#### 4.1.1. HTML

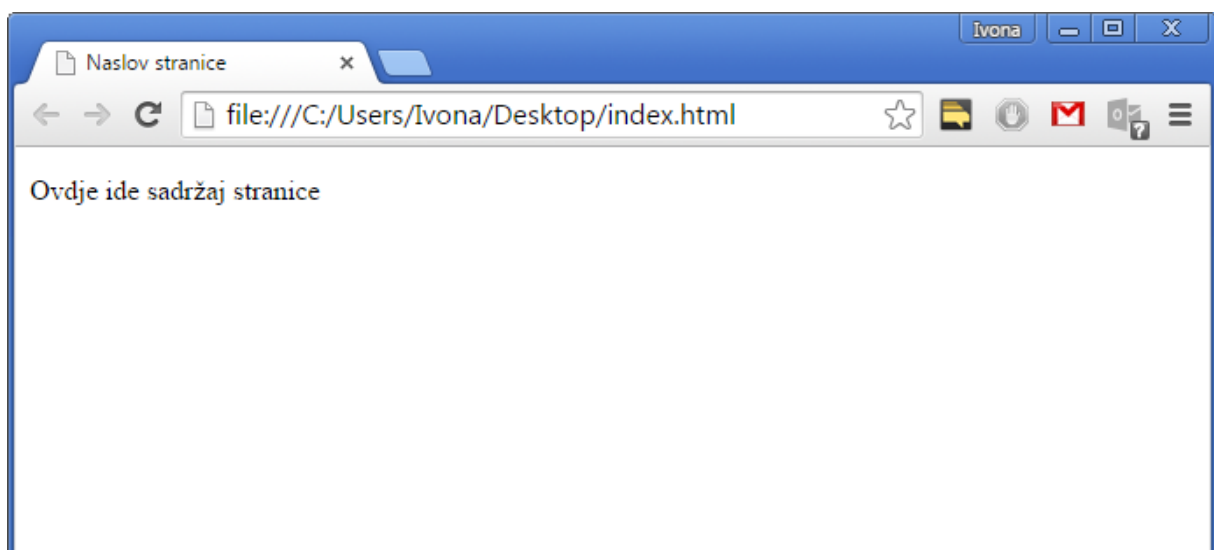
*HyperText Markup Language* tj. HTML je prezentacijski jezik za izradu web stranica koji se kao i sve što je počelo s *World Wide Web*-om godinama razvijao. U današnje vrijeme se sve češće koristi najnovija verzija HTML5 koja se pretjerano ne razlikuje od prethodnog XHTML-a, no ima neke bitne značajke koje su promijenjene u odnosu na prošle verzije. Za pisanje koda svih verzija HTML-a, pa tako i posljednje, postoji veliki broj alata koji nam pomažu u tom, no za izradu stranice je dovoljno imati i običan uređivač teksta poput Notepada i web preglednik s kojim se provjeri izgled stranice. HTML je jednostavan jezik čiji je osnovni element oznaka (eng. *tag*) koja opisuje kako će se iskodirana web stranica prikazivati na web pregledniku. Sve oznake su prethodno definirane i standardizirane te postoje određena pravila kako ih koristiti. Osnovnu strukturu web stranice i korištenje oznaka možemo proučiti promatrajući primjer 1. Iz priloženog možemo primijetiti da je svaka oznaka omeđena znakovima < i >. Isto tako, vidljivo je da oznake dolaze u paru, što najčešće i je slučaj. Možemo ih podijeliti na početnu (eng. *start tag*) i završnu oznaku (eng. *end tag*). Početna oznaka, kao šta sama riječ govori bi bila ona koja se pojavljuje prva, a završna oznaka je ona

koja nakon znaka < ima znak /. Ako za primjer uzmemo oznaku html, *start tag* bio bi <html>, a *end tag* </html>.

```
<!DOCTYPE html>
<html>
<head>
<title>Naslov stranice</title>
</head>
<body>
<p>Ovdje ide sadržaj stranice</p>
</body>
</html>
```

#### Primjer 1 – Najjednostavnija struktura HTML5 dokumenta

Svaki HTML dokument, na samom početku, prije <html> oznake treba imati deklaraciju. U primjeru 1, to je instrukcija <!DOCTYPE html> koja web pregledniku javlja da je dokument napisan u HTML5 verziji HTML-a (W3Schools, n.d.). Primjer 1 je jako jednostavan, stoga u pregledniku (slika 2) ne bi bilo razlike da je na početku dokumenta deklarirana neka druga verzija HTML-a. No, ukoliko bi dokument imao puno više elemenata, kao što stranice u stvari imaju, deklaracija bi imala veći značaj. Nakon navedene deklaracije slijedi oznaka <html> koja definira html dokument. Kao i većina ostalih oznaka, <html> obavezno mora imati i završnu </html> oznaku koja se nalazi na dnu svakog HTML dokumenta. Unutar te dvije oznake nalaze se glava i tijelo dokumenta, sljedeće najbitniji elementi unutar HTML-a. Glava se označava s <head>, a tijelo s <body> i također imaju svoje pripadajuće završne oznake.



Slika 2 - Kod iz primjera 1 prikazan u web pregledniku

Unutar glave se obično nalazi naslov stranice koji se prikazuje unutar *title bar*-a, zatim metapodaci koji opisuju stranicu, sadrže ključne riječi itd. koji se ne prikazuju u pregledniku, zatim razne skripte ili poveznice na dokumente.

S druge strane, unutar tijela se nalaze svi elementi koje korisnik vidi pregledavajući web stranicu. Broj oznaka koje se mogu pojaviti unutar tijela je velik, stoga se kombinacijom istih može napraviti stranica čija je kombinacija elemenata ograničena samo maštom autora, a da se pritom pazi, ukoliko nakon jedne početne oznake stavimo drugu, isto tako početnu, potrebno je prvo zatvoriti drugu oznaku, pa tek onda prvu. Što se tiče HTML elemenata, ovisno o prikazu možemo ih podijeliti u dvije skupine: *block* i *inline*. Elementi koji se uvijek prikazuju u novom redu i zauzimaju cijelu širinu stranice spadaju pod *block* elemente, dok oni koji su *inline* ne zauzimaju punu širinu stranice, već onoliko koliko je elementu potrebno, a isto tako, svaki novi *inline* element koji se doda u stranicu prikazati će se u istom redu kao i prethodni, sve dok ne upotpune širinu stranice.

Iako su se stranice prije uređivale koristeći HTML oznake i njihove attribute, danas to nije slučaj. Uz HTML5 dolazi i CSS3, najnovija verzija CSS-a za koji je neophodna mogućnost pristupa HTML elementama. Pristup istima se može vršiti preko klasa ili identifikatora. Identifikator je atribut koji se dodaje unutar početne oznake te ga označavamo s `id` koji zatim izjednačavamo s vrijednosti tj. nazivom elementa kojeg želimo dodijeliti. Jedan naziv identifikatora se može koristiti isključivo na jednom mjestu, dok klase koje se definiraju unutar oznake isto kao i identifikatori samo s atributom `class`, mogu se upotrebljavati na neograničenom broju elemenata.

#### **4.1.2. CSS**

Cascading Style Sheets ili popularnije CSS je stilski jezik koji se koristi za dizajniranje web stranica, tj. za definiranje izgleda elemenata stvorenih u HTML dokumentu. Najnoviji standard istog je CSS3 koji je donio brojne mogućnosti u odnosu na prethodnu verziju. No, one nisu bitne za razumijevanje CSS-a kao jezika za stiliziranje web stranica jer su najbitnije stavke ostale nepromijenjene. Naime, ključ razumijevanja stiliziranja HTML dokumenata pomoću CSS-a je u tom da oko svakog HTML elementa zamislimo nevidljivi okvir koji, ovisno o korištenim oznakama, može biti *inline* ili *block* (Duckett, 2011). CSS daje mogućnost

kontrola izgleda svakog od tih okvira, kao i svih elemenata unutar njega, a mogu se definirati na tri načina (Maleš & Mladenović, 2007): u posebnoj datoteci sa .css ekstenzijom (primjer 2), unutar HTML datoteke koristeći oznaku `<style>` unutar glave HTML dokumenta (primjer 3) ili kao dio HTML oznake koju želimo stilizirati (primjer 4).

```
p{
  color: red;
}
```

Primjer 2 - izgled tekstualnog dokumenta spremljenog sa .css ekstenzijom

```
<style type="text/css">
p{
  color: red;
}
</style>
```

Primjer 3 - izgled dijela HTML dokumenta koji se stavlja unutar `<head>` oznake

Prva dva načina stiliziranja (primjer 2 i 3) su jako slična, razlika je u tom što se u drugom navedenom načinu instrukcije pišu unutar HTML dokumenta i ne mogu se primjenjivati na niti jedan drugi dokument, dok je to moguće korištenjem prvog načina stiliziranja. Inače, svaka instrukcija se sastoji od dva dijela, selektora koji određuje koji element želimo uređivati i deklaracijskog bloka unutar kojeg deklariramo kako će određeni element izgledati. Oznaka `p` u primjerima 2 i 3 predstavlja selektor, dok ostatak koda spada pod deklaracijski blok. Nadalje, navedeni blok ima svoje deklaracije koje se također sastoje od dva dijela, atributa i vrijednosti koji su odvojeni dvotočkom te obavezno završavaju sa znakom točke-zareza kao označavanjem kraja deklaracije (Castro & Hyslop, 2012). Atribut jedine deklaracije unutar ovih primjera je `color`, dok je `red` vrijednost atributa. Ista deklaracija je upotrebljena i u primjeru 4, tj. trećem načinu uređivanja izgleda nekog HTML dokumenta. Kod tog tipa uređivanja nije potrebno pisanje selektora, već unutar oznake elementa koji želimo uređivati dodajemo HTML atribut `style` čije vrijednosti postaju deklaracije kojima određujemo izgled tog elementa, u ovom slučaju boja paragrafa je postavljena u crveno.

```
<p style="color:red;">Ovdje ide sadržaj stranice</p>
```

Primjer 4 – izgled oznake `<p>` nakon dodavanja stila

U navedenim primjerima navedeno je direktno uređivanje jedne vrste oznake koje se primjenjuje na cijeli dokument, no to je nezgodno kada želimo urediti određeni element koji može ili ne mora izgledati isto kao i drugi elementi s istom oznakom. Tada se koriste klase i

identifikatori. Označavanje klase unutar CSS-a se izvršava tako da se umjesto imena oznake kao selektora napiše točka te odmah nakon nje ime klase elementa kojeg želimo uređivati. Definiranje instrukcije za uređivanje nekog elementa preko identifikatora se vrši tako da se napiše skala (eng. *hashtag*) te ime id-a odmah nakon nje. Jedan element može imati definiran i identifikator i klasu, no kod stiliziranja elemenata identifikator ima veći prioritet, što znači da bez obzira jesmo li napisali prvu instrukciju koristeći klasu, identifikator ili oznaku, prikazati će se one vrijednosti koje su deklarirane unutar ID-a. Identifikator po važnosti slijedi klasa i tek na kraju dolaze instrukcije deklarirane direktno preko imena oznaka. No, ukoliko ipak želimo promijeniti stil nekog elementa kojeg smo već definirali u CSS-u preko ID-a, a da pritom ne želimo mijenjati vrijednosti navedene unutar id selektora, to možemo napraviti koristeći princip stiliziranja prikazan u primjeru 4.

Kao i kod kodiranja HTML dokumenta, pisajući stilove za iste, poželjno je ostavljati komentare kojima kratko objašnjavamo instrukcije, tj. na što se one odnose. Oni nam pomažu kod naknadnog uređivanja .css dokumenata da bi se lakše i brže snašli. Mogu se pisati u jednoj ili više linija, a pretraživači ih ignoriraju. Kombinacija znakova za otvaranje komentara je `/*` tj. kosa crta i zvjezdica. Nakon upisa navedenih znakova upisuje se komentar te ga zatvaramo istim, obrnuto utipkanim znakovima (primjer 5).

```
/* Ovo je komentar */  
  
/* Ovo je komentar  
   napisan u  
   više redova */
```

Primjer 5 – Izgled komentara napisanih unutar .css dokumenta

### 4.1.3. PHP

PHP je *open source* skriptni jezik nastao 1994. godine koji se koristi kod izrade dinamičnih i interaktivnih web stranica. Besplatan je, široko rasprostranjen i jedan je od najkorištenijih jezika za izradu web aplikacija. Korišten je kod izrade mnogih sustava za upravljanje sadržajem, pa tako i kod WordPress-a i samim time olakšao dodavanje, brisanje i uređivanje sadržaja unutar web sjedišta. PHP kod se može pisati u bilo kojem uređivaču teksta, kod pisanja naredbi<sup>3</sup> nije osjetljiv na velika i mala slova (eng. *case-sensitive*), a napisane

---

<sup>3</sup> Kod pisanja imena varijabli bitno je jesu li napisana velikim ili malim slovima

dokumente je potrebno spremi s ekstenzijom .php. Osim PHP-a, navedeni dokumenti mogu sadržavati kodove HTML-a, CSS-a i JavaScript-a. Za razliku od navedena tri jezika, PHP se izvršava na serveru te se rezultat u obliku HTML-a šalje *web* pretraživaču (W3Schools, n.d.). No, kao i ostali programski jezici ima svoja pravila, a prvo koje ćemo spomenuti je i najosnovnije: svako kodiranje u PHP-u mora započeti s `<?php` i završiti s `?>`, a između se pišu instrukcije koje kao i u velikom broju programskih jezika završavaju znakom točkazareza. Oznake početka i završetka PHP koda su znak interpreteru na serveru da se kod između oznaka treba izvršiti. Za ispis teksta, među ostalim koristimo naredbu `echo` (primjer 6).

```
<!DOCTYPE html>
<html>
<head>
<title>PHP stranica</title>
</head>
<body>

<?php
echo "Tekst napisan koristeći PHP";
?>

</body>
</html>
```

Primjer 6 – kod web stranice čiji je sadržaj napisan preko PHP-a

Kao i kod većine programskih i skriptnih jezika, kod PHP-a također susrećemo tipove varijabli `string`, `int`, `float`, `char`, `array` koje nije potrebno posebno deklarirati već se kod same inicijalizacije tip varijable automatski određuje. Prilikom inicijalizacije varijable, prije imena iste se obavezno stavlja znak dolara dok samo ime mora početi slovom ili znakom donje crtice. Također, PHP ima predefinirane varijable kao i već gotove funkcije, a sintaksa jezika bazirana je na onoj C-a i Pearl-a.

## 4.2. Oblikovanje izgleda sjedišta

Prije kodiranja teme, preporučljivo je odrediti funkcionalnosti web sjedišta koje radimo, napraviti njegovu strukturu sadržaja, odrediti ciljanu grupu posjetitelja te na račun dobivenih informacija krenuti na posao. Svako sjedište ima ciljane korisnike te se njegov sadržaj i dizajn mora prilagoditi onima za koje se očekuje da će posjećivati *site*, pa tako i portal o Pythonu, čiju izradu u ovom radu opisujemo, će biti *web* sjedište predviđeno za osobe koje žele naučiti

osnove programiranja u programskom jeziku Python. Prije samog početka kodiranja poželjno je napraviti *site map* tj. dijagram (slika 3) na kojem je prikazana struktura stranica na sjedištu te grupiranje istih, ukoliko postoji.



Slika 3 - *site map* portala o Pythonu

*Site map* obično počinje sa stranicom „Naslovnica“ (eng. *homepage*) koja se dijeli na odjeljke (eng. *sections*) koji mogu biti naslovna stranica nekih drugih odjeljaka (Duckett, 2011). Odjeljci naslovne stranice portala o Pythonu biti će: okruženje, lekcije, zadatci te impressum. Nakon što se napravi *site map*, poželjno je krenuti u izradu *wireframe*-a tj. jednostavne skice ključnih informacija koje trebaju ići na svaku stranicu. Na to se odnose položaj svih HTML elemenata koji se očekuju na stranicama (slika 4), dok odabir boja, pozadina i fontova određujemo naknadno. Isto tako, bitno je paziti na vizualnu strukturu stranice. Primarna navigacija uvijek treba biti prisutna i intuitivna kao što je naslovna stranica ujedno i najposjećenija te se na njoj trebaju nalaziti ključne informacije vezane za sjedište. Kod ovog portala primarna te uvijek prisutna navigacija se nalazi u zaglavlju stranice, desno od logotipa portala, a odmah ispod nailazimo na *slideshow* koji vizualno prikazuje da se radi o portalu za učenje i programiranje. Strukturu sadržaja možemo usporediti s onom u novinama. Svaki tekst ili lekcija imaju svoj naslov, a ukoliko je sadržaj duži, zbog lakšeg snalaženja i lakšeg pregledavanja koriste se i podnaslovi. U novinama kao i na drugim portalima često nailazimo na slike koje opisuju ili nadopunjuju tekstualne sadržaje, no na ovom sjedištu će tekst češće pratiti određeni kodovi za lakše svladavanje gradiva. Sadržaj naslovne stranice, baš kao i kod naslovnih stranica novina, je sažet te jasno govori o čemu se radi u ostalim dijelovima portala. Dizajn sjedišta je čist, jednostavan i bez detalja koji bi ometali korisnika prilikom dužeg pregledavanja stranica. Prije prilagođavanja istog za WordPress, iskodirana je prvo web stranica `index.html` s popratnim stilovima definiranim unutar posebnog `style.css` dokumenta.





Slika 4 - wireframe naslovne stranice portala

### 4.3. Datoteke

Kada bi portal radili koristeći isključivo HTML i CSS, isti bi se sastojao od mnoštva HTML dokumenata u kojima se redovito ponavlja ista struktura oznaka. No, kada u igru uđe WordPress i PHP zajedno s njim, princip izrade stranice je znatno drugačiji i što je najbitnije,

za krajnjeg korisnika znatno jednostavniji za korištenje. Prilikom izrade teme, programer treba proučiti pravila izrade tema jer WordPress već ima mnoštvo definiranih funkcija te predložaka koji se mogu, a neki i moraju iskoristiti te na kraju vizualno prilagoditi prema vlastitoj želji. Naravno, nije neophodno koristiti isključivo WordPress-ove predloške već je moguće kreirati i koristiti svoje.

Na internetu se može pronaći veliki broj različitih tema za WordPress, no kada bi promatrali dokumente koji ih sačinjavaju, primjetili bi da se kroz svaku temu provlači nekolicina istih. Naime, kao što je već napomenuto postoje određeni predlošci tj. dokumenti koje WordPress prepoznaje, neki od njih su neophodni da bi tema uopće bila prepoznata od strane WordPress-a, a to su `index.php`<sup>4</sup> i `style.css`, dok također postoji i niz onih koji se često koriste, ali nije neophodno da se nalaze u mapi teme. U praksi ne postoje teme napravljene od samo spomenuta dva dokumenta jer onda izrada stranica preko WordPressa nebi imala neku svrhu, stoga se najčešće navedeni `index.php` dijeli na više njih: `single.php`, `page.php`, `comments.php`, `search.php`, `footer.php`, `sidebar.php`, `404.php`, `archive.php`, s tim da se ne mora koristiti isključivo takva podjela te se mogu koristiti samo neki od navedenih predložaka. WordPress-ov sustav funkcionira tako da se otvaranjem web sjedišta poziva jedna datoteka, koja poziva ostale datoteke, koje također mogu pozivati neke druge datoteke itd. U svakom slučaju, za WordPress temu možemo reći da je skup međusobno povezanih dokumenata koji sami za sebe nemaju svrhu, ali logičnim povezivanjem slažu izvrsnu cjelinu i funkcionalno web sjedište.

U nastavku će biti opisano korištenje i funkcionalnost svake datoteke zasebno, kao i način povezivanja sa ostalim datotekama. Za početak će `index.php` datoteka biti podijeljena na tri dijela. Prvi dio koda, onaj u kojem se nalazi glava dokumenta te zaglavlje web sjedišta nalaziti će se unutar datoteke `head.php`. Dio koda koji se odnosi na podnožje stranice biti će spremljen unutar `footer.php` datoteke, dok će onaj kod koji je bio između kodova navedena dva dokumenta ostati unutar `index.php`-a.

---

<sup>4</sup> Razlika između datoteka `index.php` i `index.html` je u tom što se unutar `.php` datoteka osim HTML oznaka nalaze i PHP kodovi koji stranicu povezuju sa njenim CMS-om

### 4.3.1. header.php

Header.php datoteka je ona koja se prva poziva otvaranjem bilo kojeg tipa stranice na nekom web sjedištu. Njen sadržaj počinje deklaracijom HTML dokumenta, a završava početnom oznakom <div> unutar koje će se nalaziti sadržaj stranice. Unutar oznake <head> nalaze se meta oznake. Neke poput imena autora teme i označavanja skupa znakova definiramo odmah unutar ovog dokumenta dok one meta oznake poput opisa (eng. *description*) stranice i ključnih riječi (eng. *key words*), za koje bi bilo dobro da se mijenjaju klikanjem na različite poveznice, definiramo dodavanjem novih stranica na web sjedištu. Za to je najbolje koristiti neki već gotovi dodatak (eng. *plugin*) za WordPress. Također, unutar glave dokumenta se nalazi i oznaka za naslov koji se prikazuje na vrhu pretraživača. On se kao i *description* meta oznake prikazuje prilikom pretraživanja na internetu, stoga je preporučljivo da se naslov mijenja istodobno kad i korisnik otvori novu stranicu našeg web sjedišta. To možemo riješiti preko par linija PHP koda (primjer 7) kojima pomoću if-else naredbi jednostavno pretraživaču zadajemo na koji način da prikaže naslov. Naravno, ovo je samo jedan od načina kako izvesti da se naslov na pretraživaču mijenja ovisno o prikazanoj stranici.

```
<title>
<?php

if (!(is_404() && (is_single() || (is_page())))) {
    wp_title(''); echo ' - ';
}
elseif (is_404()) {
    echo 'Not Found - ';
}
if (is_home()) {
    bloginfo('name'); echo ' - '; bloginfo('description');
}
else {
    bloginfo('name');
}
?>
</title>
```

Primjer 7 – PHP kod koji omogućava da se naslov mijenja paralelno sa promjenom stranice web sjedišta

Nakon definiranih meta oznaka i određivanja naslova stranice, neophodno je povezati stranicu sa `style.css` datotekom, te eventualno s nekim drugim skriptnim datotekama. Isto tako, preporučljivo je unutar oznake <head> dodati PHP funkciju `wp_head()`, koja je

potrebna radi prikaza određenih dodataka. Zatim unutar datoteke `header.php` otvaramo i oznaku `<body>` te kreiramo elemente koji se nalaze u zaglavlju stranice (Slika 4). U slučaju portala o Pythonu, unutar zaglavlja će se nalaziti logo portala, navigacija i *slider* koji će se prikazivati samo na početnoj stranici ovog sjedišta. Problem postavljanja *slider*-a na samo jednu stranicu sjedišta riješiti ćemo preko jednostavnog if upita prikazanog u primjeru 8. Prilikom otvaranja oznake `body` preporučljivo je navesti klasu, ali ne na tipičan način, direktnim upisom imena klase nego preko PHP-a, dodavajući funkciju `body_class()` koja omogućava da se ime klase mijenja ovisno o načinu prikaza stranica. Na taj način dizajner ima veću slobodu kao i veće mogućnosti prilikom pisanja stilova.

```
<?php
if (is_front_page()) {
    wd_slider(1);
}
?>
```

Primjer 8 – prikaz koda kojim se određuje da se *slider* prikazuje isključivo na naslovnoj stranici portala o Pythonu

### 4.3.2. index.php

Datoteka `index.php`, kako smo već napomenuli je jedna od dvije osnovne datoteke koje su potrebne za rad teme u sklopu Wordpress-a iako to najčešće nije slučaj, već se `index.php` dijeli na više dijelova. Osnovna podjela je na `header.php`, `index.php` i `footer.php`. Sadržaj datoteke `header.php` smo već objasnili, stoga možemo primjetiti da se unutar nje nalazi cijelo zaglavlje i početna oznaka za sadržaj dokumenta. Takav dokument sam za sebe nema funkciju, tako da se nastavak istog nalazi unutar datoteke `index.php` unutar koje se nalazi sadržaj koji se mijenja svakim klikom na poveznicu, te na samom kraju dolazi sadržaj datoteke `footer.php` koju ćemo naknadno opisati. Navedena tri dokumenta se spoje tako da se na početku `index.php` datoteke doda PHP funkcija `get_header()`, a na kraju `get_footer()` te se tako sadržaji datoteka `header.php` i `footer.php` dodaju na vrh odnosno na dno `index.php` dokumenta. Bitno je napomenuti da pretraživač otvara `index.php` tek u slučaju da unutar teme ne postoji dokument koji se zove `home.php`, `single.php`, `page.php`, `archive.php` itd. koji se obično otvaraju, ovisno o načinu prikaza stranice (WordPress, n.d.).

### 4.3.3. footer.php

Dokument u kojem se nalazi podnožje web sjedišta zove se `footer.php` koji dolazi kao nastavak na `index.php`. Podnožje sjedišta sadrži jako mali dio koda te se njegov sadržaj uglavnom ne mijenja. No, preporučeno je da ima funkciju `wp_footer()`, kao što `header.php` ima `wp_head()`, koja sama za sebe ne radi ništa, ali služi da bi WordPress-u rekla gdje se nalazi podnožje stranice zbog eventualnog dodavanja raznih skripti unutar iste. Ukoliko postoji potreba, unutar *footera* se može omogućiti dodavanje različitih elemenata (eng. *widget*) preko kojih se ispunjava sadržaj *footera*, no u slučaju ovog portala za to nema potrebe, stoga nećemo detaljnije objašnjavati način modificiranja podnožja stranice. Isto tako, osim podnožja stranice, u ovom dokumentu se nalazi završna HTML oznaka za `div` u kojem se nalaze svi elementi stranice te ona koja označava kraj HTML-a.

### 4.3.4. sidebar.php

Rubni trak (eng. *sidebar*) je element stranice koji može ili ne mora postojati. Konkretno, na sjedištu koje obrađujemo *sidebar* se ne nalazi na početnoj stranici istoga, dok je na svim ostalim prisutan sa desne strane od sadržaja. Preko njega korisnik može lakše pristupiti nastavku lekcija koje pregledaje ili može direktno provjeriti znanje onih sadržaja koje je proučio. Također, može se naići neka zanimljiva informacija koja ne piše u literaturi. Datoteka `sidebar.php` sadrži samo onaj dio koda koji prikazuje sidebar na stranici. Ako želimo da isti bude dinamičan te da korisnik preko sustava za upravljanje sadržajem može mijenjati sadržaj istog, potrebno je kreirati datoteku `functions.php` te unutar nje kreirati sidebar u kojeg će se moći dodavati razni *widgeti*. Datoteka `sidebar.php` poziva se jednostavnom funkcijom `get_sidebar()` koju definiramo na području HTML koda unutar kojeg želimo smjestiti istu.

### 4.3.5. page.php

Prilikom pregledavanja portala, prikaz stranice može biti različit. Može se prikazivati jedan post, svi objavljeni postovi ili svaka stranica zasebno. Za svaki od prikaza preporučljivo je kreirati predložak koji će kao i `index.php` biti povezan s zaglavljem i podnožjem stranice. Ako želimo napraviti predložak za prikaz stranica, koristiti ćemo predložak `page.php`, iz čega se može zaključiti da će se prilikom otvaranja svake stranice zasebno pozvati datoteka

`page.php` te u skladu s definiranim stilovima i rasporedom elemenata definiranim u istoj, stranica će se prikazivati.

#### 4.3.6. `single.php`

Ukoliko se nalazimo na stranici s postovima, prilikom klika na jedan od njih pokreće se datoteka `single.php` koja omogućava prikaz samo jednog posta. Isto tako, ukoliko ispod posta postoje komentari, isti se prikazuju na stranici te korisnik ima mogućnost komentiranja. Kao i kod ostalih predložaka i ovaj tip prikaza stranica može imati definirane vlastite stilove.

#### 4.3.7. `functions.php`

Datoteka `functions.php` omogućuje korisnicima mijenjanje i prilagođavanje WordPressa svojim potrebama te sadržava funkcije kojima ga nadograđujemo. Prilikom izrade portala, ova datoteka je korištena samo za izradu rubnog traka te omogućava dodavanje *widgeta* unutar njega. Funkcija iz primjera 9 kreira rubnu traku imena Sidebar, unutar koje će se dodavati widgeti.

```
<?php
function pypmf_widgets_init() {
    register_sidebar( array(
        'name'           => esc_html__( 'Sidebar', 'pypmf' ),
        'id'             => 'sidebar-1',
        'description'    => '',
        'before_widget' => '<aside id="%1$s" class="widget %2$s">',
        'after_widget'  => '</aside>',
        'before_title'  => '<h2 class="widget-title">',
        'after_title'   => '</h2>',
    ) );
}
add_action( 'widgets_init', 'pypmf_widgets_init' );
?>
```

Primjer 9 – funkcija koja prikazuje registraciju *sidebara* unutar WordPress-a

Njihovim dodavanjem preko CMS-a, isti će se prikazati na stranici tako da će svaki widget biti smješten unutar oznake HTML oznake `<aside>`, dok će naslov biti označen kao `<h2>`. Njima, kao i svim ostalim HTML oznakama možemo definirati stilove te se tako pobrinuti za njihov izgled.

#### 4.3.8. style.css

Kao što je već navedeno, `style.css` je jedna od dvije osnovne datoteke potrebne za WordPress temu. Ukoliko ova datoteka ne bi postojala, WordPress-ov sustav ne bi prepoznao folder u kojem se nalaze ostale datoteke od kojih je napravljen predložak. Iako se *file* `style.css` uvelike koristi, to nije neophodno, no datoteka obavezno mora postojati. Sustav će prepoznati temu i ako `style.css` nije pozvan unutar glave HTML koda. Na samom početku njenog sadržaja moraju se nalaziti detalji o temi napisani u obliku komentara. WordPress ne dopušta postojanje dviju tema sa istim detaljima opisanim na vrhu `style.css` dokumenta (primjer 10). U ostatku dokumenta definirani su stilovi za sjedište. Počevši od resetiranja preddefiniranih stilova koji postoje u svakom pregledniku pa do postavljanja pozadina, definiranja fonta, njegove veličine, boje i svega ostalog što stranicu čini estetičnom.

```
/*
Theme Name: Pypmf
Theme URI: localhost
Author: Ivona
Author URI: localhost
Description: Site about Python programming language
Version: 1.0
License: GNU General Public License v2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html
Tags: blue, white, light, one-column, two-columns, right-sidebar
Text Domain: pypmf
This theme, like WordPress, is licensed under the GPL.
Use it to make something cool, have fun, and share what you've learned
with others.
*/
```

Primjer 10 – komentar unutar `style.css` datoteke unutar kojeg su navedeni potrebni detalji o temi

#### 4.4. Postavljanje teme

Postavljanje teme na instalirani WordPress-ov sustav je veoma jednostavno. Naime, potrebno je mapu unutar kojeg se nalaze svi predlošci i datoteke teme kopirati ili premjestiti u mapu „themes“ koji se nalazi unutar mape „wp-content“, a koja je dio instalacijskog foldera navedenog CMS-a. Zatim je neophodno otvoriti pretraživač te ući u administracijsku ploču, odabrati poveznicu „Izgled“ te naposljetku aktivirati željenu temu.

## 5. Portal o Pythonu oblikovan primjenom predložaka

Portal o Pythonu namijenjen je osobama koje se žele upoznati s programiranjem koristeći programski jezik Python. Sadržaj stranice podijeljen je na pet glavnih odjeljaka odvojenih uočljivim poveznicama na glavnoj navigacijskoj traci. Prvi odjeljak je ujedno i naslovna stranica na kojoj se mogu pročitati zanimljivosti o ovom programskom jeziku, pronaći citat vezan za programiranje te slike koje vizualiziraju sadržaj stranice. U drugom dijelu nailazimo na kratke i vrlo jasne upute o postavljanju okruženja za Python dok su treći i četvrti dio rezervirani za učenje i provjeravanje stečenog znanja. Peti i zadnji odjeljak rezerviran je za podatke vezane za sadržaj portala.

### 5.1. Sadržaj portala

Najveću važnost ovog portala dodjeljujemo odjeljcima „Lekcije“ i „Zadatci“ unutar kojih se nalaze dijelovi knjige „Rješavanje problema programiranjem u Pythonu“ (Budin et al., 2012). Za učenje Pythona preko ovog portala odabrana su poglavlja 3, 4, 5 i 6 iz istoimene knjige, a u njima se nalaze osnovne i najbitnije informacije potrebne budućem programeru. U nastavku ovog rada možete vidjeti koji sadržaji se nalaze na portalu.

#### 5.1.1. Prvo upoznavanje s Pythonom

Prvo upoznavanje s Pythonom naslov je trećeg poglavlja prethodno spomenute knjige, a ujedno i prva cjelina koja će biti obrađena na portalu. U ovom poglavlju mladi programer dolazi u doticaj s jednostavnim tipovima podataka te uči razliku između istih. Nakon kratkog uvoda je objašnjen svaki tip podataka u posebnom podpoglavlju. Prvi tip s kojim će se učenik sresti su cijeli brojevi koje u Pythonu nazivamo int. Prikazani su primjeri ispisa podataka, načini unošenja brojeva u binarnom i heksadekadskom sustavu koje Python ispisuje u dekadskom obliku, kao i mogućnosti obrnutog pretvaranja za što je potrebno korištenje ugrađenih funkcija. Nakon savladavanja cijelih brojeva, u sljedećoj lekciji riječ je o racionalnim brojevima. Prilikom pisanja istih koristimo decimalnu točku zahvaljujući kojoj Python prepoznaje tip podataka float. Korisnik portala ima priliku naučiti načine pisanja navedenih brojeva, s naglaskom na ispis brojeva manjih od 0.0001 ili većih od  $10^{15}$ . Također, kroz primjere će biti uočljivo da treba oprezno koristiti brojeve s pomičnom točkom. U



nastavku lekcije riječ je o Booleovom ili logičkom tipu podataka kod kojeg se svaka tvrdnja ocjenjuje kao istina ili laž. Ne postoji ni jedan drugi sud koji je dopušten koristiti ga u logici, kao ni mogućnost istodobnog korištenja istine i laži. U Pythonu se taj tip naziva `bool` i isto tako može poprimiti samo dvije vrijednosti: `True` i `False`. Pisanje vrijednosti tipa `bool` je osjetljiv na velika i mala slova. Isto tako, Python ima ugrađene funkcije pomoću kojih se može ustanoviti cjelobrojna vrijednost neke logičke vrijednosti, i obrnuto. Zorniji prikaz i lakše razumijevanje ovog, kao i svih ostalih tipova podataka mogu se vidjeti na primjerima postavljenim uz lekciju na portalu. Posljednji tip koji obrađen u ovoj lekciji je onaj vezan za tekstualne sadržaje, a zove se znakovni niz ili *string*. U Pythonu je taj tip nazvan `str`, a njegova vrijednost za razliku od brojeva, obilježava se jednostrukim ili dvostrukim navodnicima na početku i na kraju napisanog niza znakova. Isto tako, prikazan je način korištenja funkcije `print()` kao i popis posebnih znakova za oblikovanje teksta pomoću kojih programer ima mogućnost znakovne nizove ispisivati u više redova, koristiti tabulator i slično.

Nakon savladavanja osnovnih tipova podataka, njihovog upisa i ispisa, korisnik je spreman naučiti kako vršiti aritmetičke operacije nad tipovima `int` i `float`. Na samom početku lekcije prikazan je popis postojećih aritmetičkih operatora koji obavljaju operacije nad jedan ili dva operanda. Isto tako, navedeni su primjeri za svaki tip operatora, specifičnosti nekih od njih kao i red prvenstva. U nastavku je riječ o varijablama tj. imenima kojima pridružujemo vrijednost koju naknadno možemo upotrebljavati. Njihovim korištenjem je riješen problem pamćenja vrijednosti te je isti opisan i prikazan u primjerima koristeći različite aritmetičke operacije i ugrađenu funkciju `divmod()` koja vraća cjelobrojni količnik i ostatak.

### 5.1.2. Priprema i izvođenje programa

Sadržaji portala koji se opisuju u prethodnom poglavlju uključuju osnovne tipove podataka i obradu istih koliko je to za sada moguće, no vježbajući uz primjere postavljene na portalu možemo zaključiti da zatvaranjem sučelja Pythona, napisane instrukcije nestaju te ih ne možemo vratiti. Kako bismo sačuvali napisane programe potreban nam je IDLE – uređivački dio sučelja unutar kojega se piše program, a napisani kod se potom može spremati na disk te po potrebi mijenjati. Osim načina otvaranja uređivačkog dijela sučelja, na portalu se može naučiti da ekstenzija datoteka u kojima se nalaze programi napisani u Pythonu glasi `.py`. U ovom poglavlju se može naučiti kako pokreniti program, jer za razliku od rada koji je

prakticiran u prethodnom poglavlju, sada se program ne izvršava sve dok se za to ne da naredba. Isto tako, prikazan je način komentiranja u ovom programskom jeziku što je, kako smo već spomenuli kod drugih jezika prilikom izrade portala, jako korisno kada nakon dugo vremena želimo nešto promijeniti unutar koda. Što se tiče pokretanja programa, isti se otvara u interaktivnom sučelju u kojem se program može testirati, a testni podatci se ne mogu spremiti. U nastavku ovog poglavlja je prikazano korištenje metode `input()` koja od korisnika programa traži nekakav unos koji se može spremiti u varijablu te naknadno koristiti kao i sve ostale vrijednosti drugih varijabli. Osim što je objašnjen i na primjerima pokazan način formatiranja tekstualnih sadržaja, jedno cijelo podpoglavlje posvećeno je osnovnim pravilima pisanja programa. Upoznajemo se s ključnim riječima koje se ne mogu koristiti kao imena varijabli, zatim su dani primjeri ispravnog i neispravnog pisanja istih kao i preporuka na koji način odabrati njihova imena. Prikazan je opći oblik načina pridruživanja u Pythonu, kao i višestruko pridruživanje korištenjem jedne naredbe, a izvršavanje aritmetičkih operacija u Pythonu nerijetko rezultira mijenjanjem tipa varijable. Isto tako, na portalu je objašnjen način zamjene vrijednosti varijabli, a isti je prikazan na vizualnom primjeru zamjene ribica u dva akvarija. Rekapitulacija svega naučenog u prve dvije cjeline dana je preko programa koji unesene sekunde pretvara u dane, sate i minute.

### 5.1.3. Struktura programa u Pythonu

Savladavajući elemente spominjane u prethodnim poglavljima, učenik je spreman napisati jednostavan sekvencijalni program koji korisniku istoga ne dozvoljava pogrešku. Korisnici računala veliki broj puta u programe unesu krive vrijednosti, stoga programer treba predvidjeti takve situacije da ne bi dolazilo do pucanja programa ili nenadanog završavanja istog. Takve situacije ćemo, za početak rješavati donošenjem odluka u Pythonu. Prije nego mladi programer bude spreman napisati program prilikom čijeg pokretanja će se obavljati blok naredbi isključivo ako je ispunjen traženi uvjet, treba se upoznati s relacijskim operatorima te njihovim značenjem. U nastavku lekcije upoznajemo se sa modulima, tj. zbirka funkcija koje kao i prethodno spominjane funkcije možemo koristiti u našem programu. No, prilikom pozivanja je neophodno upotrijebiti naredbu `import` kojom prvo najavimo funkciju, a naknadno je pozivamo unutar programa. Donošenjem odluka tijekom pisanja koda možemo napisati programe koji mogu rješavati mnoge probleme, no

programerima je jako korisno kada mogu ponavljati blokove naredbi stoga ćemo se kratko upoznati s petljama for i while te na primjerima vidjeti kako funkcioniraju i za što su korisne. Nakon što je sve navedeno savladano, učenik je spreman za upoznavanje s funkcijama i programima te načinom na koji naš program postaje modul.

#### 5.1.4. Funkcije, programi i moduli

Složene probleme je veoma praktično podijeliti na manje dijelove, tj. podprobleme koje u programiranju možemo napisati pomoću funkcija. One predstavljaju dijelove programa koji obavljaju složenije operacije te nam vraćaju neku vrijednost. Možemo ih koristiti n puta te na taj način se nepotrebno rješavamo uzastopnog ponavljanja istih linija koda te smanjujemo veličinu svojih dokumenata. U prethodna tri poglavlja, imali smo prilike sreći sa nizom jako korisnih ugrađenih funkcija koje predstavljaju samo mali dio onoga što Python u svojim modulima sadržava. Neke nove ćemo naučiti i na početku ovog poglavlja. Isto tako, imamo priliku upoznati n-torku, još jedan tip podataka u Pythonu. Postepeno ćemo se upoznavati s modulima, kojih službeno ima na stotine, a mi ćemo koristiti one koji su nam za početak zanimljivi.

„Osnovna svrha programiranja je pripremanje računala za rješavanje problema koji potiču iz raznih područja ljudskog djelovanja. Programi su svojevrsno ostvarenje algoritama koji rješavaju te probleme. S druge strane, programi imaju i svoju estetsku vrijednost. Za neke programe možemo reći da su dobro osmišljeni i uredno oblikovani te da ih je ugodno čitati. S druge strane, programi mogu biti neuredno i neujednačeno napisani tako da ih je teško razumjeti.“ (Budin et al., 2012). Iako kod malih programa kakve mi trenutno pišemo nije toliko bitno uvažavati sve preporuke za stil programiranja, preporučljivo je odmah steći naviku pisanja pravilnih i urednih linija koda kako bi nam poslije bilo lakše. U izradi preporuka sudjelovao je i sami autor Pythona, a neke spominjanih su:

- uvlačenje bloka naredbi za četiri znakovna mjesta,
- duljina retka ograničena na 80 znakova,
- prazni redovi unutar programa te razmaci u izrazima i naredbama,
- dogovor o pisanju imena,
- olakšavanje pisanja programa koji nudi uređivački prozor IDLE,

koje su na portalu detaljno objašnjene.

Nakon savladavanja pravilnog pisanja koda, dolazimo u doticaj s pisanjem vlastitih funkcija, načinom definiranja istih unutar programa, kako u njih prenositi jedan ili više parametara itd. Isto tako, imamo priliku vidjeti razne primjere funkcija kako bi lakše savladali pisanje istih. Nakon svega do sad naučenog ostaje samo objasniti kako da naš program postaje modul i kako ga naknadno koristiti.

### **5.1.5. Zadatci**

Sadržaj portala namijenjen je svim onima koji žele naučiti osnove programiranja u Pythonu. Prilagođen je početnicima te ih detaljno, objašnjavajući osnovne elemente uvodi u svijet programiranja. Nakon savladavanja svih lekcija i popratnog vježbanja prema primjerima prikazanim na portalu, korisnik portala je u mogućnosti na kraju svake cjeline provjeriti svoje znanje. Zadatci su koncipirani prema onima koji se nalaze na kraju svakog poglavlja u već spomenutoj knjizi o Pythonu. No, razlika je u tom što na portalu svako pitanje ima više odgovora između kojih korisnik bira onaj koji je točan. Nakon što korisnik odgovori na sva pitanja, šalje potvrdu te nakon toga dobiva konačan rezultat ispitnog kviza. Na taj način, korisnik odma može provjeriti svoje znanje i uočiti koji sadržaji nisu dobro usvojeni te ih je potrebno ponoviti.

## **5.2. Pypmf - portal o Pythonu**

Nakon što smo portal izradili na lokalnom računalu, došlo je vrijeme za premještanje istog na internet. Kako WordPress nudi svoj prostor samo za korisnike koji upotrebljavaju ponuđene teme, tj. ne nudi mogućnost postavljanja vlastitih, za početak je portal postavljen na drugi server i koristi besplatnu domenu `pypmf.bugs3.com`. Kada mislimo na postavljanje portala na internet, to uključuje prijenos cijele WordPress skripte, zajedno sa temom i pripadajućim dodatcima kao i transfer baze u kojoj se nalazi sav sadržaj sjedišta.

Sadržaj koji se nalazi na portalu je objašnjen u prethodnim poglavljima, stoga će u ovom biti riječ o elementima korištenim prilikom izrade teme. Dizajn sjedišta je jednostavan te korisniku omogućuje dugo korištenje istoga bez imalo naprezanja koje bi postojalo da smo recimo koristili isključivo jarke boje.

Naime, početna stranica portala (Slika 5), osim logotipa i navigacije koji se cijelo vrijeme nalaze na stranici, sadrži i niz slika koje se dinamično izmjenjuju u određenom vremenskom periodu. Gledajući stranicu prema dole, nailazimo na tri slike-poveznice koje vizualno prikazuju sadržaj koji se nalazi klikom na njih, dok se opis nalazi ispod njih. U nastavku stranice se može vidjeti kratak opis programskog jezika Python, koji je glavna tema ovog portala, kao i citat vezan za programiranje.



Slika 5 - Naslovna stranica portala

Stranice „Okruženje“ i „Lekcije“ i svi njihovi popratni postovi imaju isti dizajn (Slika 6). S lijeve strane nalazi se sadržaj predviđen za učenje, s desne strane je rubna traka, dok se na ostatku stranice nalazi onaj dio koji se ponavlja kroz cijelo sjedište. Unutar rubne trake nalazi se popis lekcija koje su ujedno i poveznice koje omogućavaju jednostavan prelazak s jedne na drugu, a isto tako mogu se nalaziti i neke zanimljive činjenice vezane za Python. Ostatak stranice je onaj dio koji se ponavlja kroz cijelo sjedište.

The screenshot shows the website 'pypmf' with a navigation bar containing 'NASLOVNICA', 'OKRUŽENJE', 'LEKCIJE', 'ZADATCI', and 'IMPRESSUM'. The main content area is titled '2.1. Formatirani ispis'. The text explains the use of the print() function with format() for precise output. It includes a code block showing how to format integers and floats. A list of bullet points explains the variables used in the code. Below, another code block shows how to format output with specific data types (d, f, s) and a 'pre' tag for preformatted text. On the right side, there is a 'Lekcije' (Lessons) sidebar with a list of lesson titles and numbers, such as '1. Jednostavni tipovi podataka', '1.1. Cijeli brojevi', etc. At the bottom of the page, there is a quote by Larry Niven: 'That's the thing about people who think they hate computers. What they really hate is lousy programmers.' and a copyright notice for Ivona Barjan, PMFST.

Slika 6 - prikaz lekcije na portalu

Dizajn stranica „Zadatci“ i „Impressum“ je jednak kao i prethodne dve navedene. No, unutar odjeljka „Zadatci“ (slika 7) s lijeve strane se nalaze naslovi četiri cjeline čiji su sadržaji obrađeni na portalu. Svaki od naslova ima svoj botun „Start quiz“ kojim otvaramo set pitanja vezanih za sadržaje cjeline ispod koje se spomenuti botun nalazi. Nakon što se odgovori na prvo pitanje, za otvaranje sljedećeg je potrebno kliknuti na botun „Next“. Isti postupak se ponavlja sve dok ne budu odgovorena sva pitanja. Na završetku kviza prikazan nam je postignuti rezultat, mogu se vidjeti rješenja ili pokrenuti kviz iznova.

**Zadatci**

**I - Upoznavanje s Pythonom**  
Question 2 of 5  
**2. Question**  
Koje su osnovne logičke vrijednosti?

True i Talse  
 True i False  
 Frue i False  
 Frue I Talse

**Next**

**II - Priprema i izvođenje programa**  
Znanje iz druge cjeline možete provjeriti unutar upitnika Priprema i izvođenje programa

**Start quiz**

**III - Struktura programa u Pythonu**  
U ovom upitniku možete provjeriti znanje cjeline Struktura programa u Pythonu

**Start quiz**

**IV - Funkcije programi i moduli**  
Funkcije programi i moduli je zadnja cjelina na ovom portalu. Ovdje možete provjeriti svoje znanje.

**Start quiz**

**Lekcije**

1. Jednostavni tipovi podataka
- 1.1. Cijeli brojevi
- 1.2. Brojevi s pomičnom točkom – zapisivanje realnih brojeva
- 1.3. Logički ili Booleov tip podataka
- 1.4. Znakovni niz – string
- 1.5. Python kao veliki kalkulator – aritmetički izrazi
2. Interaktivni i uređivački dio sučelja IDLE
- 2.1. Formatirani ispis
3. Neka osnovna pravila pisanja programa
- 3.1. Pravila za pisanje imena
- 3.2. Naredbe pridruživanja
- 3.3. Model pohranjivanja vrijednosti u spremniku
- 3.4. Naredbe višestrukog pridruživanja
- 3.5. Zamjena vrijednosti varijabil, tip n-torka
4. Struktura programa u Pythonu
- 4.1. Donošenje odluka i grananja u programima
- 4.2. Moduli
- 4.3. Ponavljanje blokova naredbi
5. Funkcije, programi i moduli
- 5.1. Poželjni izgled programa – stil programiranja
- 5.2. Funkcije
- 5.3. Program postaje modul

That's the thing about people who think they hate computers. What they really hate is lousy programmers.  
- Larry Niven -

© Ivona Banjan, PMFST

Slika 7 - odjeljak stranice u kojem se može provjeriti stečeno znanje

## 6. Zaključak

U vremenima kada su se izrađivale statičke web stranice, održavanje istih je bio dug i s današnjeg gledišta bespotreban proces. Razvojem bazi podataka i programa koji omogućavaju izradu dinamičkih web stranica dovelo je i do razvoja raznih sustava za upravljanje sadržajem. Jedan od njih je i WordPress koji je ujedno i najkorišteniji takav sustav na svijetu. Osim što na njegovom razvoju radi veliki broj ljudi, besplatan je i ima niz raznih dodataka i tema koji se mogu aplicirati na stranicu. No, sustav je otvorenog koda stoga nije neophodno koristiti isključivo gotove stvari već ga svaki programer može prilagoditi svojim ili klijentovim potrebama. To je uvelike korisno, najčešće zbog unikatnosti dizajna i posjedovanja određenih elemenata unutar sustava koje neke estetično prihvatljive teme nemaju.

Prilikom izrade teme dovoljno je poznavati HTML, CSS te imati osnovna znanja iz PHP-a. Interaktivnost stranice možemo povećati dodavanjem ili izradom dodataka kao i nadogradnjom vlastite teme. Važno je znati razlučiti kada je bolja opcija instalirati dodatak, a kada nadograđivati vlastiti kod.



## 7. Literatura

- [1] Apache friends, *What is XAMPP?*, [Internet], <raspoloživo na: <https://www.apachefriends.org/>>, [pristupljeno 2.kolovoza, 2015]
- [2] Budin et al., (2012) *Rješavanje problema programiranjem u Pythonu*, Element, Zagreb
- [3] Castro, E., Hyslop, B., (2012) *HTML5 and CSS3*, Peachpit Press, Berkeley, pp.13-16
- [4] CMS Wiki, *History of CMS*, Srpanj 2005, [Internet], <raspoloživo na: <http://www.cmswiki.com/tiki-index.php?page=HistoryOfCMS>>, [pristupljeno 16.srpnja, 2015].
- [5] Coyier, C., Starr, J., (2015) *Digging into WordPress*, pp.51-84
- [6] Duckett, J., (2011) *HTML & CSS design and build websites*, John Wiley & Sons, Inc., Indianapolis, pp. 229-230, 461-464
- [7] Gibbons, J.H., (1981) *Computer-Based National Information Systems: Technology and Public Policy Issues*, pp. 8-9
- [8] Maleš, L., Mladenović, S., (2007) *Osnove programiranja*, Filozofski fakultet, Split, pp. 56
- [9] PHP, *History of PHP*, [Internet], <raspoloživo na: <https://secure.php.net/manual/en/history.php.php>>, [pristupljeno 23.kolovoza, 2015]
- [10] Vlahović, V., (2004) *Sustavi za upravljanje sadržajem*, FER, Zagreb, pp.3
- [11] Wordpress, *Theme development*, [Internet], <raspoloživo na: [https://codex.wordpress.org/Theme\\_Development](https://codex.wordpress.org/Theme_Development)>, [pristupljeno 15.kolovoza, 2015]
- [12] W3Techs, *Usage of content management systems for websites*, Srpanj 2015, [Internet], <raspoloživo na: [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)>, [pristupljeno 23.srpnja, 2015]
- [13] W3Schools, *HTML, CSS, PHP*, [Internet], <raspoloživo na: <http://www.w3schools.com>>, [pristupljeno 18. kolovoza, 2015]