

# Računarske mreže i serversko vweb-programiranje



**JOVANA BRANKOV**

**108/2022**

**26.01.2026.**

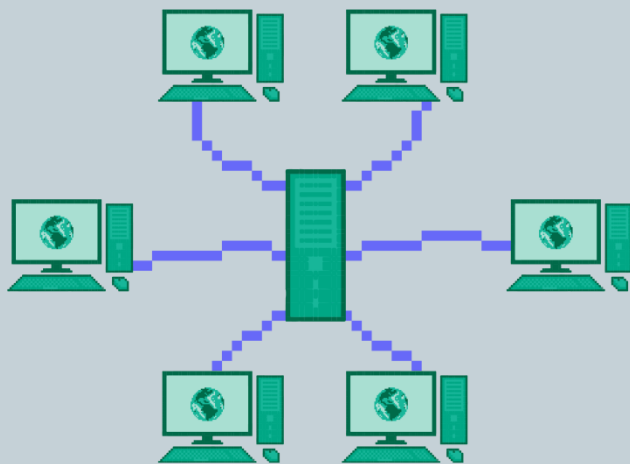
Metodika nastave računarstva i  
informatike

Prof. Sana Stojanović  
Đurđević

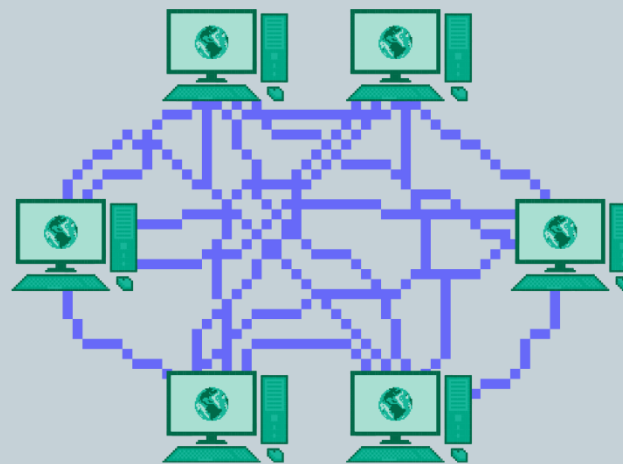
# Računarske mreže



- Računarska mreža je sistem koji se sastoji od skupa hardverskih uređaja međusobno povezanih komunikacionom opremom, snabdevenih odgovarajućim komunikacionim softverom, kojim se ostvaruje kontrola sistema tako da je omogućen prenos podataka između povezanih uređaja.
- Mreže se organizuju u model mreže **ravnopravnih računara** ili model **klijent-server**.



CLIENT-SERVER NETWORK



P2P NETWORK

# Mrežni uređaji

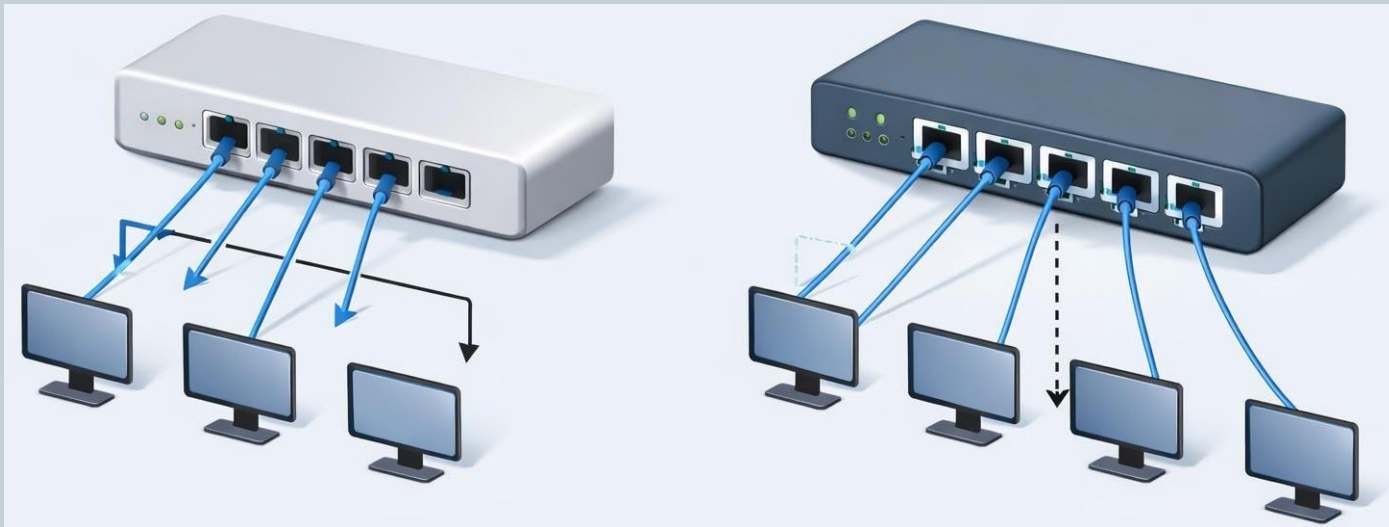
- **Mrežna kartica**  
(mrežni kontroler): -**žični** (UTP kabl),  
-**bežični** (radio-talasi, mikrotalasi,  
ili infracrveni zraci):
  - **bluetooth** (mala udaljenost),
  - **wifi** (lokalne mreže),
  - **satelitske veze i mobilne mreže** (veće područje).



# Mrežni uređaji



- Uređaji koji **posreduju u komunikaciji**: -pojačivači (repetitori),  
-habovi,  
-svičevi,  
-ruteri.
- **Komunikacioni kanali**: -upredene parice (UTP/STP kablovi),  
-koaksijalni kablovi,  
-optički kablovi.



# Adrese



- **Fizičke** (MAC) adrese (dodeljuje proizvođač),
- **IP** adrese (dodeljene statički ili dinamički): **-javne** (vidljive na internetu),  
**-privatne** (za lokalnu mrežu),

- **Domeni** i **URL**-ovi (u praksi veb-adrese).

<http://info.cern.ch/hypertext/WWW/TheProject>

- http -oznaka protokola koji se koristi za pristup resursu,
- info.cern.ch -naziv servera na kom se resurs nalazi,
- hypertext/WWW/TheProject.html-Putanja do resursa unutar tog servera,

Domeni su hijerarhijski organizovani, na primer: matf.bg.ac.rs

- rs-> nalazi se u Srbiji
- ac.rs-> deo je akademske mreže
- bg.ac.rs-> nalazi se na univerzitetu u Beogradu
- matf.bg.ac.rs-> nalazi se na Matematičkom fakultetu.

# Mrežni slojevi i protokoli



**Protokoli** predstavljaju skup pravila koja omogućavaju razmenu podataka između uređaja u mreži.

Internet je organizovan prema TCP/IP modelu koji se sastoji od četiri osnovna sloja: veznog, mrežnog, transportnog i aplikativnog.

**Vezni sloj** (sloj mrežnog interfejsa):

- protokol **IP** (za slanje poruka),
- protokol **ICMP** (za slanje kontrolnih poruka),
- protokol **ARP** (za razrešenje adresa),
- **ARP** (prevođenje IP adrese u MAC adresu),
- **RARP** (prevođenje MAC adrese u IP adresu).

# Mrežni slojevi i protokoli



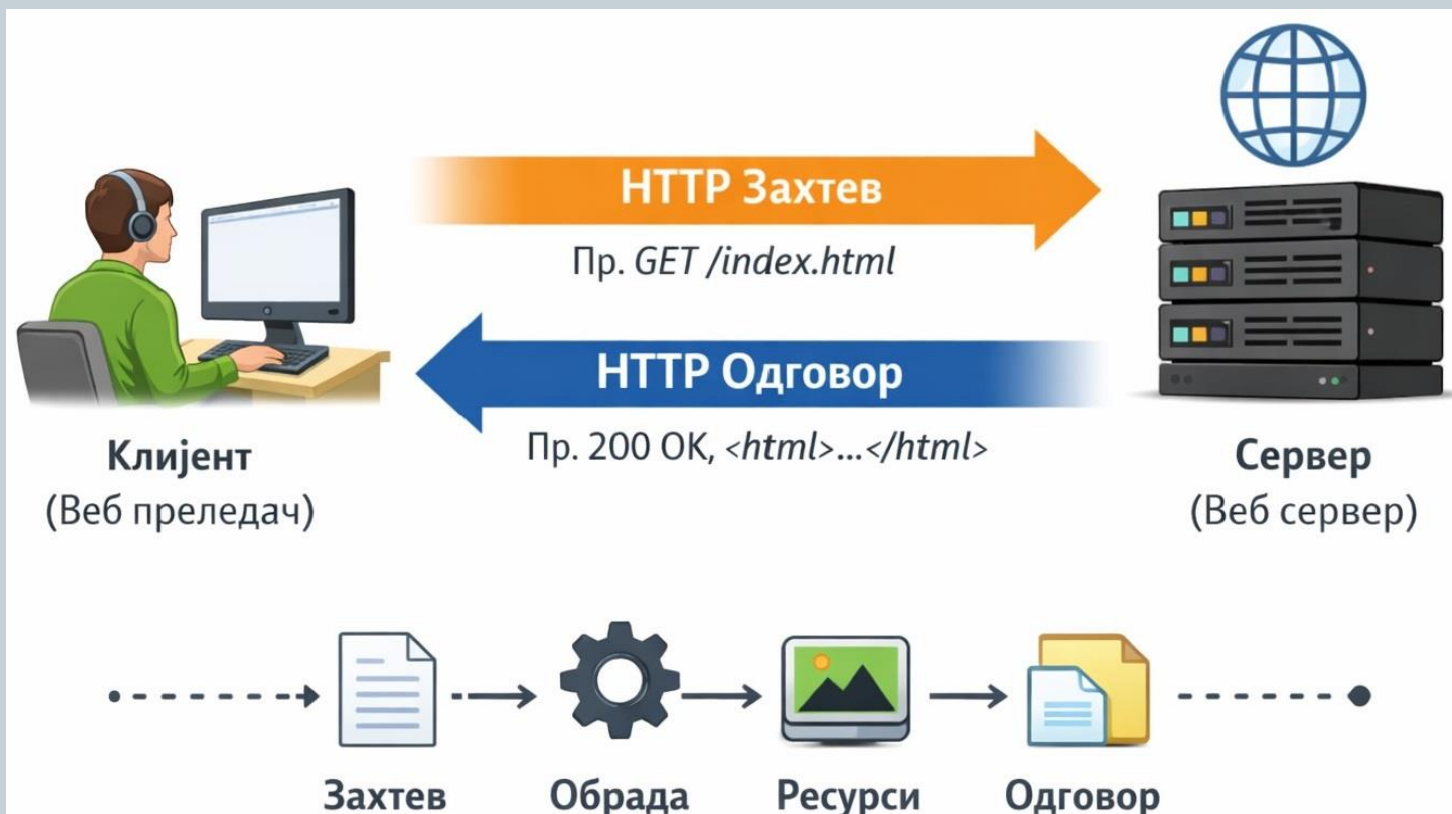
**Mrežni sloj**(internet sloj): •**adresiranje**,  
•**rutiranje**.

**Transportni sloj**: •**TCP** (komunikacija uz kontrolu grešaka),  
• **UDP** (brži, al manje pouzdan).

**Aplikativni sloj**: • **HTTP** i **HTTPS** (za preuzimanje sa veb),  
• **POP3** i **IMAP** (za preuzimanje elektronske pošte),  
• **SMTP** (za slanje elektronske pošte),  
• **ETP** (za prenos fajlova).

# Klijent-server

Veb funkcioniše po modelu klijent-server, gde klijent šalje zahteve, a server na njih odgovara.

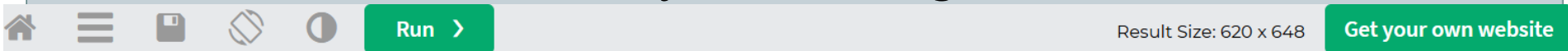




# Statičke veb-strane HTML/CSS



Za izradu statičkih veb strana koriste se osnovne veb-tehnologije- HTML i CSS. **HTML** je jezik za obeležavanje koji definiše strukturu i sadržaj veb-strane, dok se **CSS** koristi za njen vizuelni izgled.



```
<!doctype html>
<html>
<head>
<title>Мрежни слојеви и протоколи</title>
</head>
<body>
```

```
<h1>МРЕЖНИ СЛОЈЕВИ И ПРОТОКОЛИ</h1>
```

```
<p>
Рачунарске мреже су веома сложене. И у најједноставнијим мрежама
учествује
широк дијапазон хардвера и софтвера, а ствари се само још усложњавају код
великих мрежа каква је интернет. Као што је обично случај у рачунарству,
сложеност система решава се разлагањем на делове и поделом задужења
између
њих. Слично оперативним системима који врше апстракцију хардвера и
програмерима дају интерфејс за једноставније писање апликативних
програма,
и рачунарске мреже користе <b>слојевитост</b> (енгл. <i>layering</i>) и
јасно
дефинисане <b>протоколе</b> (енгл. <i>protocol</i>) комуникације.
</p>
```

```
<p>
Комуникација између виших слојева остварује се тако што се достава порука
препушта нижим слојевима.
</p>
```

```
</body>
</html>
```

Result Size: 620 x 648

Get your own website

## МРЕЖНИ СЛОЈЕВИ И ПРОТОКОЛИ

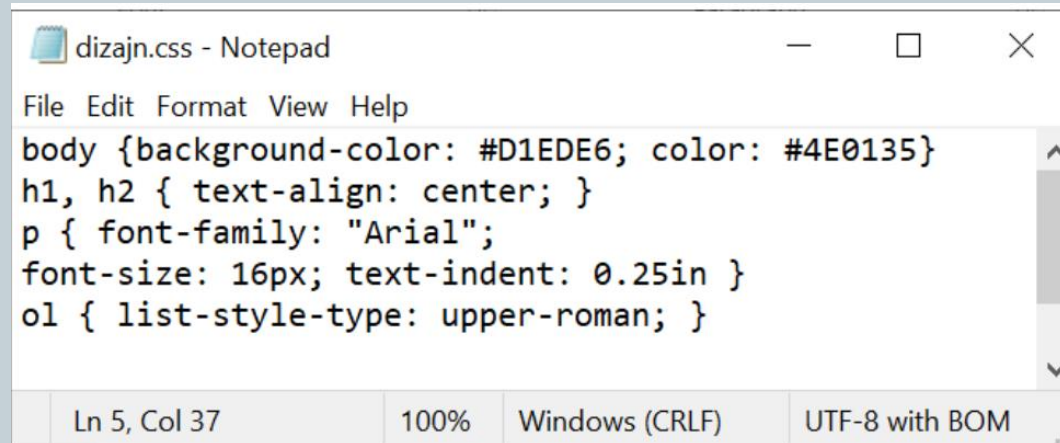
Рачунарске мреже су веома сложене. И у најједноставнијим мрежама учествује широк дијапазон хардвера и софтвера, а ствари се само још усложњавају код великих мрежа каква је интернет. Као што је обично случај у рачунарству, сложеност система решава се разлагањем на делове и поделом задужења између њих. Слично оперативним системима који врше апстракцију хардвера и програмерима дају интерфејс за једноставније писање апликативних програма, и рачунарске мреже користе **слојевитост** (енгл. *layering*) и јасно дефинисане **протоколе** (енгл. *protocol*) комуникације.

Комуникација између виших слојева остварује се тако што се достава порука препушта нижим слојевима.

# Dizajn veb-strane



Veb dizajn predstavlja proces oblikovanja vizuelnog izgleda veb-stranice i načina na koji su njeni elementi raspoređeni.



```
dizajn.css - Notepad
File Edit Format View Help
body {background-color: #D1EDE6; color: #4E0135}
h1, h2 { text-align: center; }
p { font-family: "Arial";
font-size: 16px; text-indent: 0.25in }
ol { list-style-type: upper-roman; }
```

```
<!doctype html>
<html>
  <head>
    <title>Мрежни слојеви и протоколи</title>
    <link rel="stylesheet" href="dizajn.css">
  </head>
  <body>
    . . .
```

# Kreiranje veb-sajta



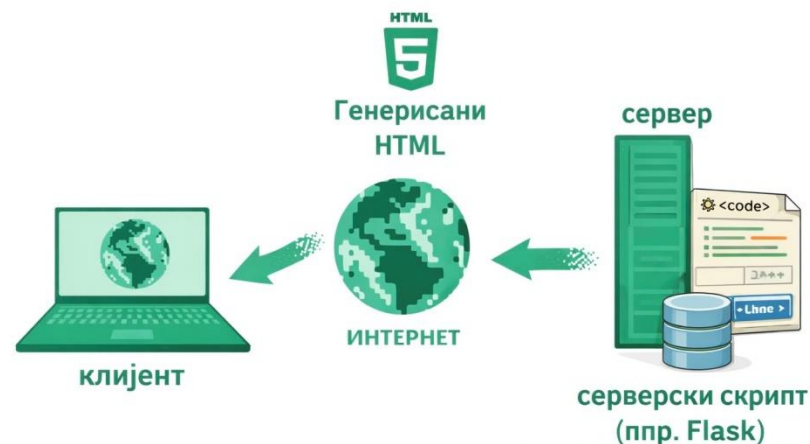
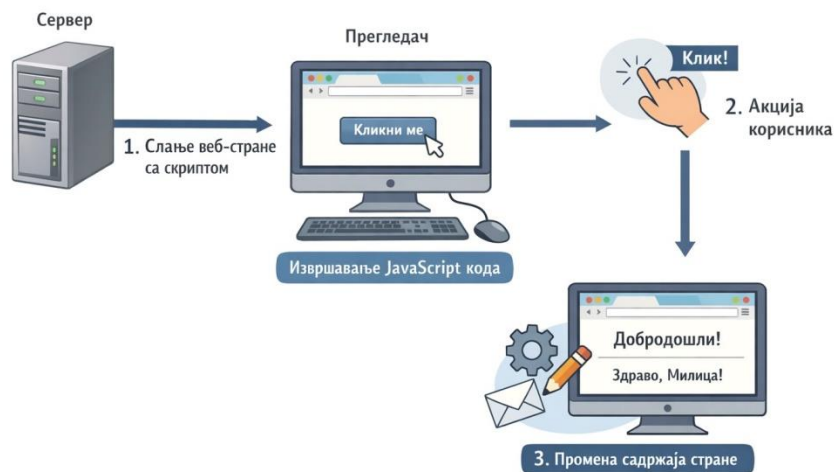
Veb-sajt se najčešće sastoji od više međusobno povezanih veb-stranica koje su smeštene u zajednički folder. Jedna od tih stranica ima posebnu ulogu i predstavlja početnu stranicu sajta.

```
<a href="podaci.html">Osnovni podaci</a>  
<a href="više.html">Nešto više o meni</a>  
<a href="zanimljivo.html"> Zanimljivo </a>
```

Da bi sve stranice imale ujednačen izgled, koristi se jedan zajednički CSS dokument koji definiše dizajn celog sajta.

Slike koje se koriste na veb-sajtu poželjno je organizovati u poseban podfolder. Prilikom dodavanja slika u HTML dokumente, neophodno je navesti i putanju do foldera u kome se one nalaze, kako bi se pravilno prikazale u internet pregledaču.

# Veb-stranice



## Активне стране (клијентски скриптови)

Клијентске вeb-технологije се извршавају у ин тernet прегледачу корисника.

Најпознатији језик за ову намену је JavaScript. Он омогућава да вeb-странице постану интерактивне, односно да реагују на акције корисника, као што су кликови мишем, унос текста или избор одређених опција.

## Динамичке стране (serverski skriptovi)

Serverske veb-tehnologije се извршавају на серверу. Serverski skriptovi, napisani na jezicima kao što su PHP, Python ili ASP.NET, обрађују податке, најчешће из база података, и на основу тога генеришу HTML страницу која се шаље клијенту.

# HTTP protokol



Razmena podataka između klijenta (veb-pregledača) i servera na internetu obavlja se pomoću protokola HTTP ili njegove sigurnije varijante HTTPS. Osnovna razlika između ova dva protokola je u tome što HTTPS koristi šifrovanje podataka, čime se obezbeđuje zaštita informacija tokom njihovog prenosa kroz mrežu.

<http://www.moja-prodavnica.com/site/products?page=3>

- *http* –oznaka protokola koji se koristi za komunikaciju ,
- *www.moja-prodavnica.com* –naziv veb-servera,
- */site/products?* –putanja do veb-strane na serveru,
- *page=3* -dodatni parametri.

# Flask biblioteka



**Flask** (<https://flask.palletsprojects.com/>) je biblioteka programskog jezika Python koja se koristi za kreiranje veb-aplikacija. Pored Flask-a, u Python-u postoje i druge biblioteke za razvoj veb-aplikacija, jedna od njih je **Django** (<https://www.djangoproject.com/>).

Kako se instalira biblioteka Flask?

Kako da instaliramo jezik Python?

# Prva Flask veb-aplikacija



Svaka veb-aplikacija se sastoji od više fajlova, pa je uobičajena praksa da se oni organizuju u poseban folder. U ovom primeru kreira se folder pod nazivom *hello*, u kome se nalazi glavni programski fajl *app.py*. U njemu se definiše Flask aplikacija i jedna funkcija koja vraća tekst koji će biti prikazan u pregledaču.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def index():
    return "Zdravo"
```



# Rutiranje



**Ruta** predstavlja vezu između URL putanje i funkcije koja će obraditi zahtev. To se postiže upotrebom dekoratora `@app.route(...)`, koji se postavlja iznad definicije funkcije.

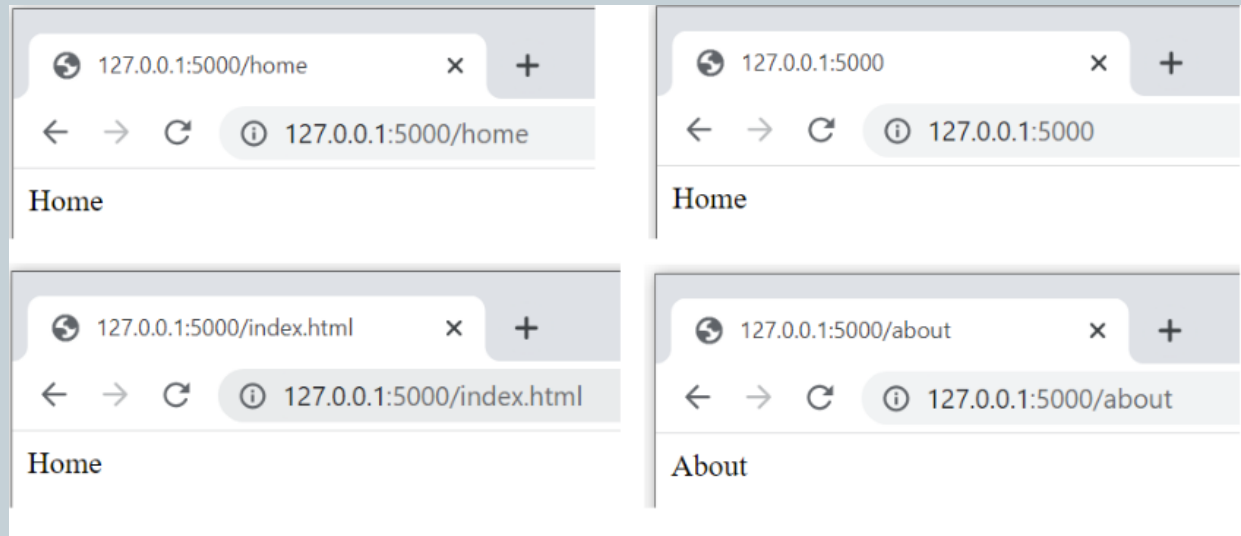
**Dekorator** je mehanizam u programiranju koji omogućava da se funkciji ili metodi doda dodatna funkcionalnost bez izmene njenog izvornog koda. Funkcija vraća tekst ili drugi sadržaj koji se zatim prikazuje u pregledaču.

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
@app.route("/index.html")
@app.route("/home")
def home():
    return "Home"

@app.route("/about")
def about():
    return "About"
```





# Šabloni



Direktno pisanje većih HTML sadržaja unutar programskog koda u vidu tekstualnih niski brzo postaje nepregledno i otežava održavanje aplikacije. Zbog toga savremeni veb-okviri, uključujući Flask, omogućavaju izdvajanje HTML koda u posebne datoteke koje se nazivaju **šabloni**.

Flask koristi biblioteku **Jinja** (<https://jinja.palletsprojects.com/>), koja omogućava rad sa šablonima.

```
.
├── app.py
└── templates
    └── index.html
```

This PC > Documents > 01\_hello >

Name

- \_\_pycache\_\_
- templates
- app.py

app.py - Notepad

File Edit Format View Help

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")
```

index.html - Notepad

File Edit Format View Help

```
<!DOCTYPE html>
<html>
  <head>
    <title>Zdravo</title>
  </head>
  <body>
    <h1>Zdravo</h1>
  </body>
</html>
```

# Statičke datoteke i linkovi



Veb-stranica se ne sastoji samo od HTML koda, već i od različitih pratećih datoteka kao što su CSS stilovi, JavaScript datoteke, slike, video i audio-zapisi. Iako se kod dinamičkih veb-aplikacija HTML sadržaj često generiše na serveru, ove prateće datoteke su najčešće statičke i klijentu se šalju u nepromenjenom obliku.

U Flask veb-aplikacijama postoji jasno definisana struktura direktorijuma. HTML šabloni se smeštaju u direktorijum *templates*, dok se sve statičke datoteke (CSS, slike, JavaScript) nalaze u direktorijumu *static*.

```
.
├── app.py
├── templates
│   └── index.html
└── static
    ├── style.css
    └── logo.png
```

# Funkcija `url_for`



Prilikom korišćenja statičkih datoteka u HTML šablonima, nije preporučljivo ručno navoditi relativne ili apsolutne putanje, jer se putanja veb-stranice može menjati ili isti šablon može biti korišćen na više različitih URL-ova. Zbog toga Flask pruža funkciju ***url\_for***, koja automatski generiše ispravnu putanju u trenutku prikaza stranice.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Uključivanje statičkih datoteka</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')
  }}" />
  </head>
  <body>
    <h1>Zdravo</h1>

    
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <head>
    <title>Uključivanje statičkih datoteka</title>
    <meta charset="utf-8" />
    <link rel="stylesheet" href="/static/style.css" />
  </head>
  <body>
    <h1>Zdravo</h1>

    
  </body>
</html>
```



# Literatura



- <https://petlja.org/sr-Latn-RS/kurs/18657/20/6544>
- [https://sr.wikipedia.org/sr-ec/Ra%C4%8Dunarska mre%C5%BEa](https://sr.wikipedia.org/sr-ec/Ra%C4%8Dunarska_mre%C5%BEa)
- <https://www.w3schools.com/html/default.asp>