



INTERNET TEHNOLOGIJE

Prof. dr Zoran Veličković, dipl. inž. el.

2022/23.

Prof. dr Zoran Veličković, dipl. inž. el.

Internet tehnologije



Bazične tehnologije Weba

(2)



Sadržaj

- ▶ **Internet kao skup servisa**
- ▶ **Internet kao hijerarhijsko skladište resursa**
 - ▶ Hijerarhija i adresiranje na Internetu
 - ▶ Web adrese - simboličko adresiranje
 - ▶ IP adresiranje – decimalna notacija
- ▶ **Bazne tehnologije Interneta**
 - ▶ **Klijent-server** arhitektura
 - ▶ **HTTP** protokol
- ▶ **Web čitači**
 - ▶ Izvorni kod Web stranice
 - ▶ **HTML** – jezik za opisivanje Web dokumenta
 - ▶ Osnovne **HTML** oznake
- ▶ **Struktura HTML dokumenta**
 - ▶ HTML oznake odeljaka
 - ▶ Kontejneri u HTML-u
 - ▶ Kodni raspored znakova u HTML-u

Internet kao skup servisa

➤ **Internet** je mreža računarskih mreža koja pruža širok spektar **proizvoda** i **usluga**.

➤ Internet omogućava ljudima da:

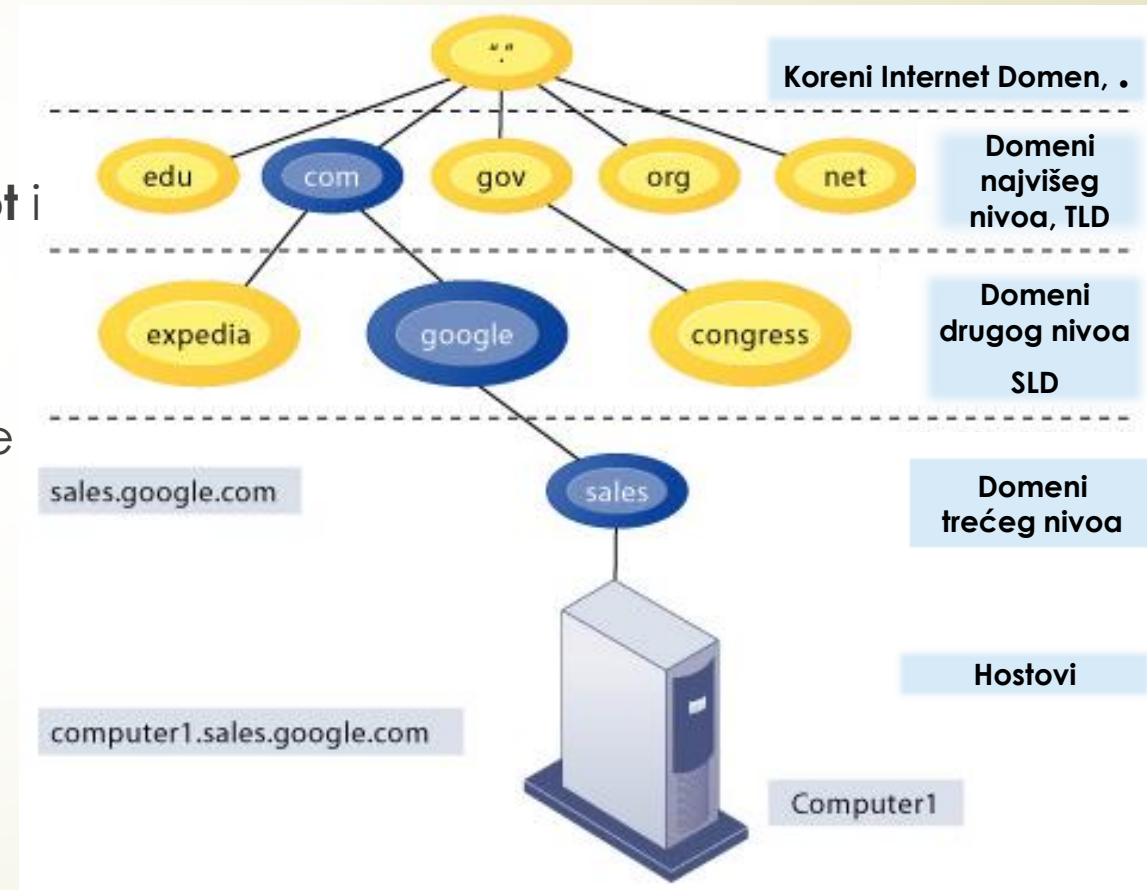
- kupuju,
- ostanu u kontaktu sa drugima,
- plaćaju račune,
- istražuju,
- uzimaju časove,
- uživaju u zabavi,
- planiraju putovanja,
- igraju igre,
- rade bilo gde ako imaju pristup internetu, ...



➤ Zajedničke Internet usluge uključuju: časkanje, e-mail, mape i uputstva, preuzimanja medija, audio i video striming u realnom vremenu, društvene medije, video konferencije ...

Internet kao hijerarhijsko skladište resursa

- ▶ Da bi se svaki imenovani resurs na Internetu efikasno **pronašao**, neophodna ga je **HIJERARHIJSKI** organizovati.
- ▶ Startna tačka u hijerarhiji Interneta se naziva **Root** i označava se tačkom(•).
- ▶ Internet je hijerarhijski organizovan, tako da se simbolički **TLD** (engl. **Top Level Domain**) nivo može podeliti na:
 - ▶ Organizacioni - **generički gTLD**
 - ▶ Geografski **ccTLD** (engl. **country code TLD**).
- ▶ Drugi nivo domena
- ▶ Treći nivo domena
- ▶ ...



Simboličko adresiranje na Internetu

- Pored **SIMBOLIČKIH IMENA** resursa na Internetu, koriste se i tzv. **IP** (engl. *Internet Protocol*) adrese.
- **Simboličke adrese** služe za pamćenje adrese resursa prilagođene **ČOVEKU**, dok se u komunikacionim protokolima koriste **IP adrese** za obeležavanje (pronalaženje) resursa na Internetu prilagođene računaru.
- Tako se za nalaženje željenog računara na Internetu mora prvo preslikati – mapirati **SIMBOLIČKA** adresua u **IP adresu**.
- Ovo preslikavanje obavlja **DNS-a**, (engl. *Domain Name System*) - **PROTOKOL** koji se u Internet komunikacijama **prvo poziva**.
- Za predstavljanje IP adrese u zaglavlju paketa se koristi **32 BITA** (verzija V4) koji se zapisuju na **specifičan način** koji nazivamo **DECIMALNA REPREZENTACIJA** IP adresa (primer: 128.4.70.9).
- Postoji i verzija 6 (V6) koja koristi 128 bita za IP adresiranje.

Klasifikacija IP V4 adresa

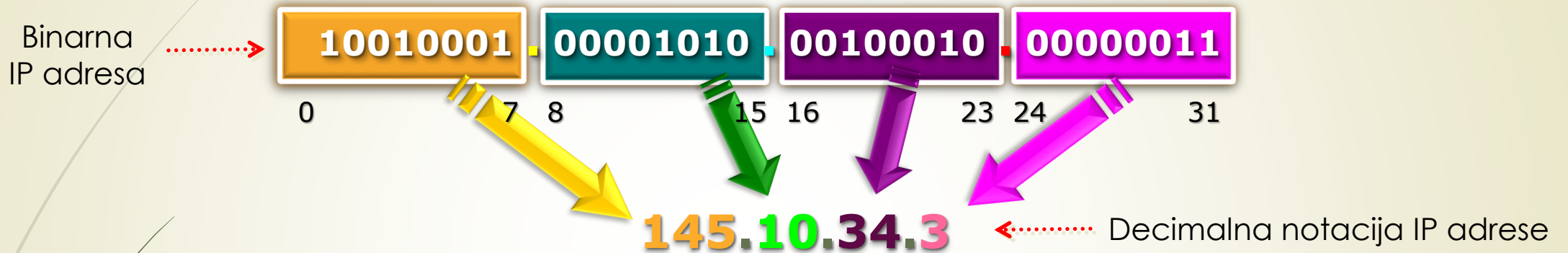


Biti koji određuju klasu IP adrese

- Od klase IP adrese zavise veličine **net-id** i **host-id** polja.
- Adrese klase **A** omogućavaju realizaciju **2^7** mreža (net-id) sa **2^{24}** računara (host-id).
- Adrese klase **B** omogućavaju realizaciju **2^{14}** mreža (net-id) sa **2^{16}** računara (host-id).

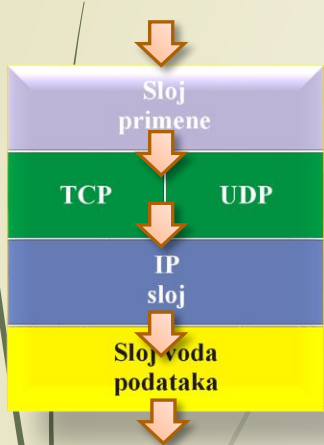


Decimalna notacija IP adrese



- U **DECIMALNOJ NOTACIJI** opseg dozvoljenih IP adresa je od 0.0.0.0 - 255.255.255.255.
- Koje je klase gore prikazane IP adresa?
- **DNS servis** obezbeđuje **MAPIRANJE** simboličkih imena resursa u odgovarajuće IP adrese (primer: *vtsnis.edu.rs* ⇒ 82.192.65.9).
- **DNS** je **prvi protokol** koji se poziva u pozadini Web čitača i nije direktno vidljiv korisniku.
- **DNS** servis je hijerarhijski **DECENTRALIZOVAN SISTEM** imenovanja za računare, usluge ili druge resurse povezane na Internet.

Struktura datagrama paketa

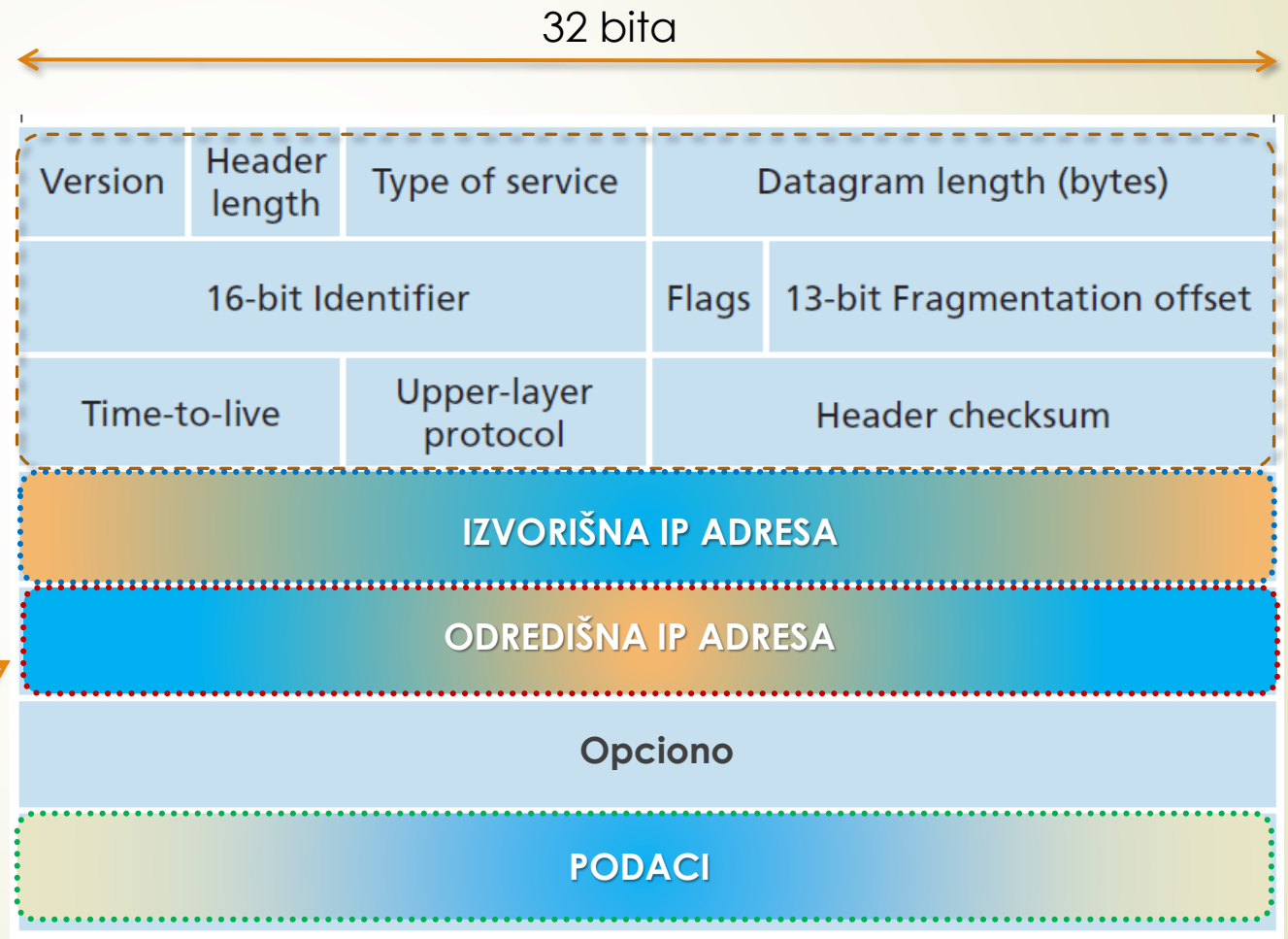


Zaglavlje **DATAGRAM** paketa

IZVORIŠNA IP adresa se koristi za usmeravanje paketa kroz mrežu do pošiljaoca

ODREDIŠNA IP adresa se koristi za usmeravanje paketa kroz mrežu do odredišta

Podaci generisani aplikacijom (engl. *payload*)



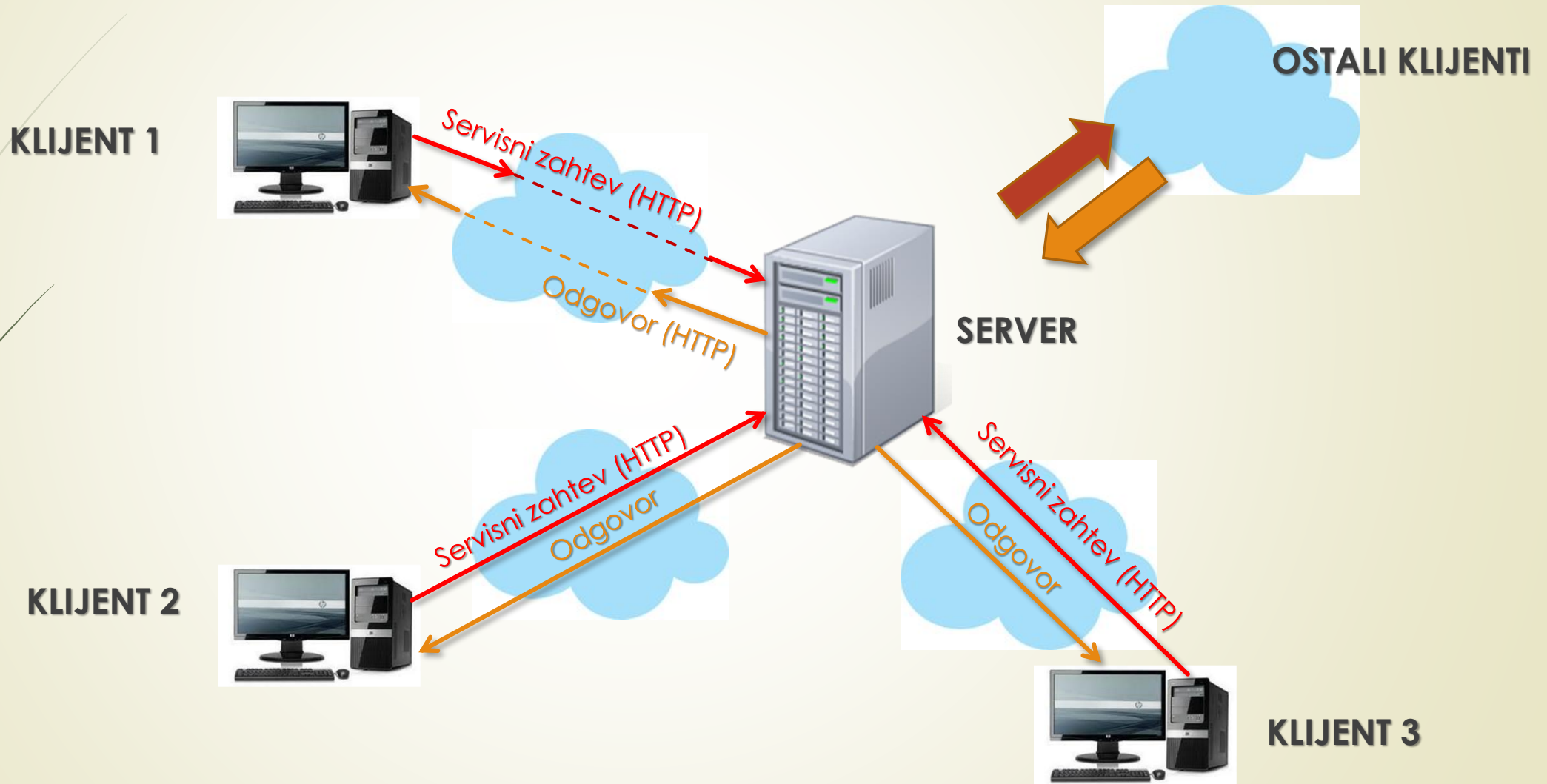
Bazne tehnologije Interneta

- ▶ Jedno od rešenja koje može rešiti deo pomenutih mrežnih problema je koncept **KLIJENT - SERVER** (engl. *Client - Server*) arhitekture.
- ▶ **KLIJENT - SERVER** arhitektura opisuje **MEĐUSOBNI ODNOS** između dva programa – entiteta (klijenta i servera).
- ▶ Jedan od programa, **KLIJENT**, postavlja servisni zahtev, a drugi program, **SERVER**, odgovara na taj zahtev.
- ▶ Iako se **KLIJENT - SERVER** koncept može realizovati u okviru jednog računara, za nas je mnogo značajnije što se ta ideja može sprovesti i u **računarskoj mreži**.
- ▶ U mreži, **KLIJENT - SERVER** model pruža pogodan način za povezivanje programa koji su locirani na **UDALJENIM RAČUNARIMA** širom Interneta.
- ▶ Većina mrežnih **POSLOVNIH APLIKACIJA** koje se danas dizajniraju koriste klijent - server model.

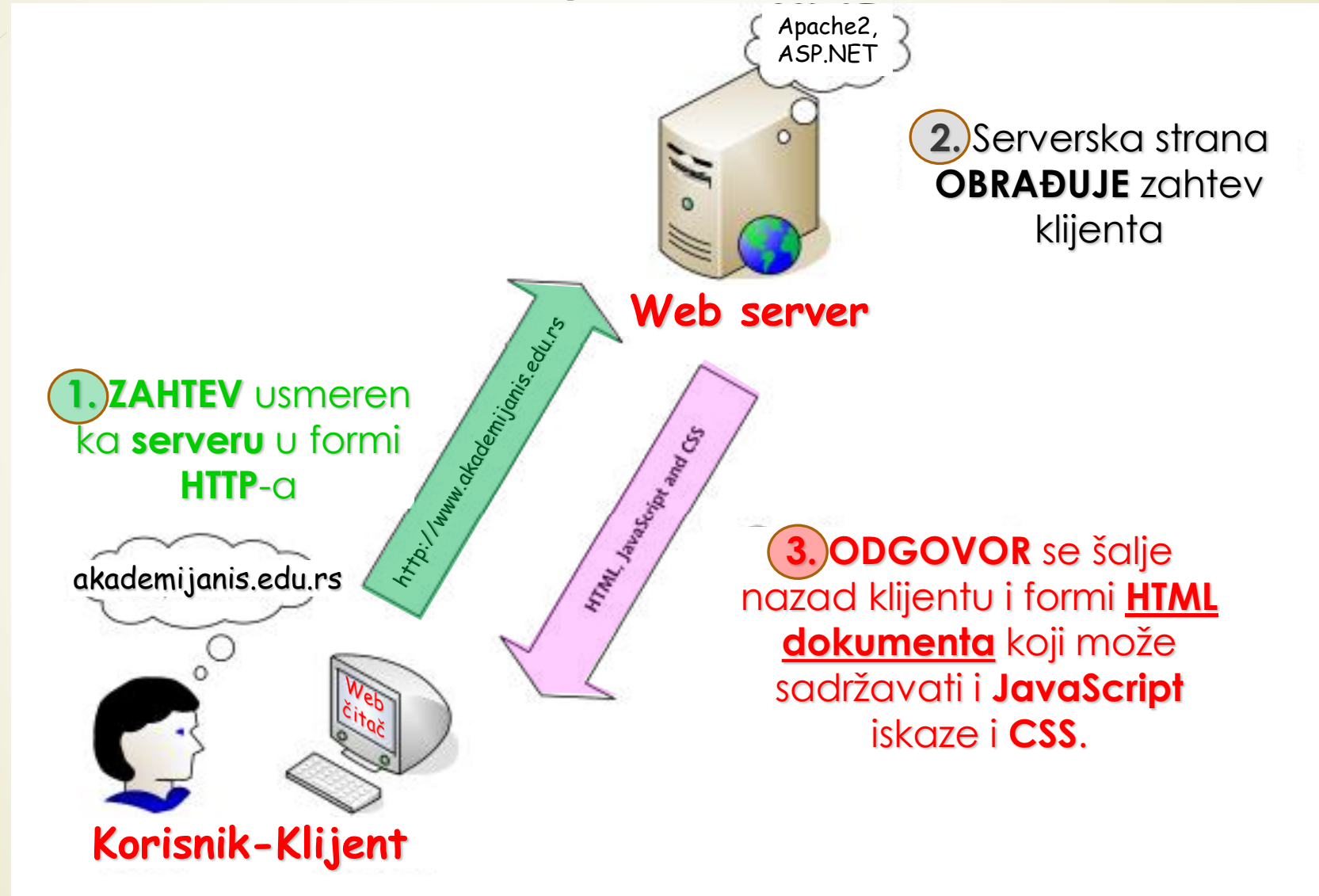
Klijent-Server tehnologija (1)

- ▶ U običajenom **KLIJENT - SERVER** modelu, **SERVER** očekuje klijentske zahteve.
- ▶ Najčešće, **VIŠE KLIJENATA** deli usluge **JEDNOG SERVERA**.
- ▶ Klijentski i serverski programi su samo deo **ŠIRE** Web aplikacije.
- ▶ Ispostavljanje **ZAHTEVA** serveru se realizuje **HTTP** (engl. *Hyper Text Transport Protocol*) **PROTOKOLOM**, te se i server po njemu naziva **HTTP SERVER**.
- ▶ U odnosu na Internet, **WEB čitač** je **KLIJENTSKI PROGRAM** koji zahteva usluge (najčešće slanje Web stranice ili datoteke) iz Web servera drugom računaru - negde na Internetu.
- ▶ Takođe, sa računara na kome je instalirani **TCP/IP** stek komunikacionih protokla mogu se zahtevati fajlovi od **FTP** (engl. *File Transfer Protocol*) servera.
- ▶ O osnovnim komunikacionim protokolima Interneta bilo je reči na 1. predavanju.

Klijent-Server tehnologija (2)



HTTP zahtev/odgovor



Metode HTTP protokola (1)

- Klijenti i serveri za komunikaciju koriste **POSEBNE PORUKE - METODE** koje su definisane **HTTP protokolom**.
- **HTTP** specifikacijom je definisan **SKUP METODA** koje se mogu naći u HTTP zahtevu.
- **HTTP METODE** se navode na **početku** startne linije HTTP zahteva.
- HTTP Metode nalažu serveru **KOJI POSAO** se od njega očekuje da obavi.
- Sve **HTTP metode PROTOKOLA** koje stoje na raspolaganju klijentu su:
 - **GET**
 - **PUT**
 - **DELETE**
 - **POST**
 - **HEAD**
 - **TRACE**
 - **OPTIONS**
 - **CONNECT**
 - **PATCH**

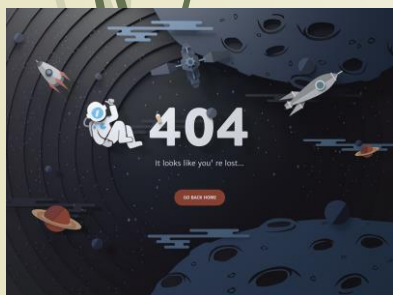
Metode HTTP protokola (2)

HTTP metod	OBJAŠNENJE
GET	Pošalji imenovani resurs od servera ka klijentu.
PUT	Smesti podatke sa klijenta u imenovani sereverski resurs.
DELETE	Obriši imenovani resurs servera.
POST	Slanje klijentskih podataka u serversku gateway aplikaciju.
HEAD	Pošalji samo HTTP zaglavlja u odgovoru na imenovani resurs.
TRACE	Vrati primljeni zahtev kako bi se uočile izmene ili dodaci koje su načinili posrednički servisi.
OPTIONS	Vrati HTTP metode koje server podržava za navedeni URL.
CONNECT	Konvertuje nešifrovani HTTP zahtev na SSL-šifrovanu komunikaciju.
PATCH	Primeni parcijalne metode modifikacije resursa

HTTP odgovori

- Na **HTTP zahtev**, server odgovara **HTTP odgovorom**.
- Da bi detaljnije objasnio svoj status, SERVER **KODIRA SVOJE ODGOVORE** na način kako je to dato u tabeli.
- Odgovori su svrstani u nekoliko grupa (**2xx, 3xx, 4xx, 5xx**).
- Najčešće korišćeni kodni statusi:

HTTP status - kod	OBJAŠNENJE
200	OK. Dokument vraćen korektno.
302	Redirekcija. Idi na neku drugu lokaciju da dobiješ traženi resurs.
404	Nije pronađeno. Ne može se pronaći traženi resurs.
505	HTTP verzija nije podržana.



Struktura HTTP metode **GET** - PuTTY

Klijent:
HTTP ZAHTEV

```
GET /tools.html HTTP/1.1  
User-agent: Mozilla/4.75 [en]  
Host: www.joes-hardware.com  
Accept: text/html, image/gif, image/jpeg  
Accept-language: en
```

Startna linija **HTTP ZAHTEVA** (GET metoda)

Zaglavlje servisnog zahteva



1

Server:
HTTP ODGOVOR

```
HTTP/1.1 200 OK  
Date: Sun, 02 Jun 2013 12:40:15 GMT  
Server: Apache/2.2.22 (Unix) DAV/2 FrontPage/5.0.2.2635 mod_ssl/2.2.22 OpenSSL/1  
.0.1c  
Last-Modified: Fri, 12 Jul 2002 07:50:17 GMT  
ETag: "146deb7-1b1-3a58f649c4040"  
Accept-Ranges: bytes  
Content-Length: 433  
Content-Type: text/html
```

Startna linija **HTTP ODGOVORA**

Zaglavlje servisnog odgovora



2

Odgovor Web servera:
Web stranica

```
<HTML>  
  
  <HEAD><TITLE>Joe's Tools</TITLE></HEAD>  
  
  <BODY>  
  
    <H1>Tools Page</H1>  
    .....
```

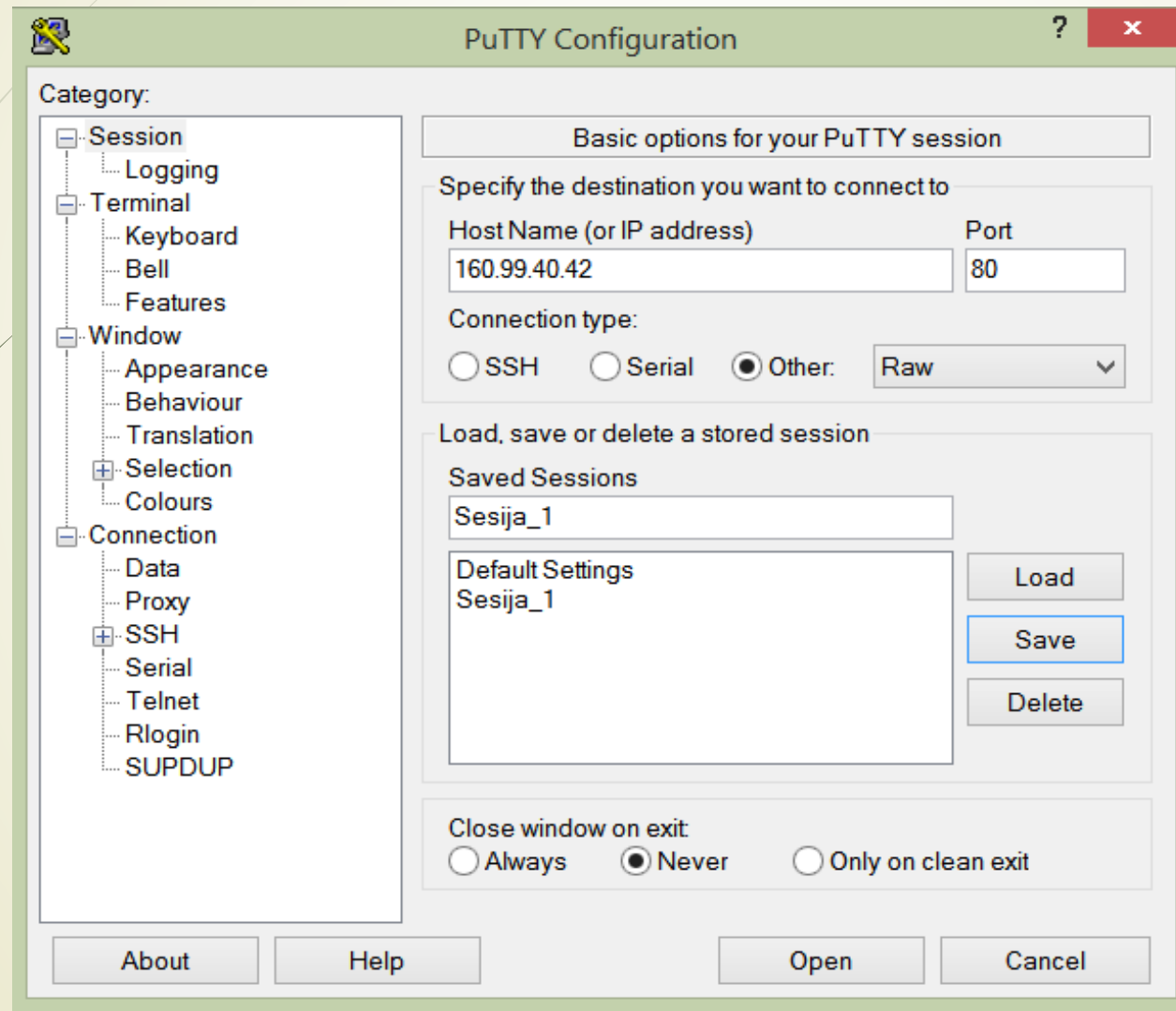
HTML kod + JavaScrip kod + CSS kod

<H1>Tools Page</H1>



3

Programski paket PuTTY



- ▶ Prva Lab. Vežba će biti realizovana primenom ovog programskog paketa.
- ▶ Biće testirane sve metode HTTP protokola.

HTTP i Web čitači

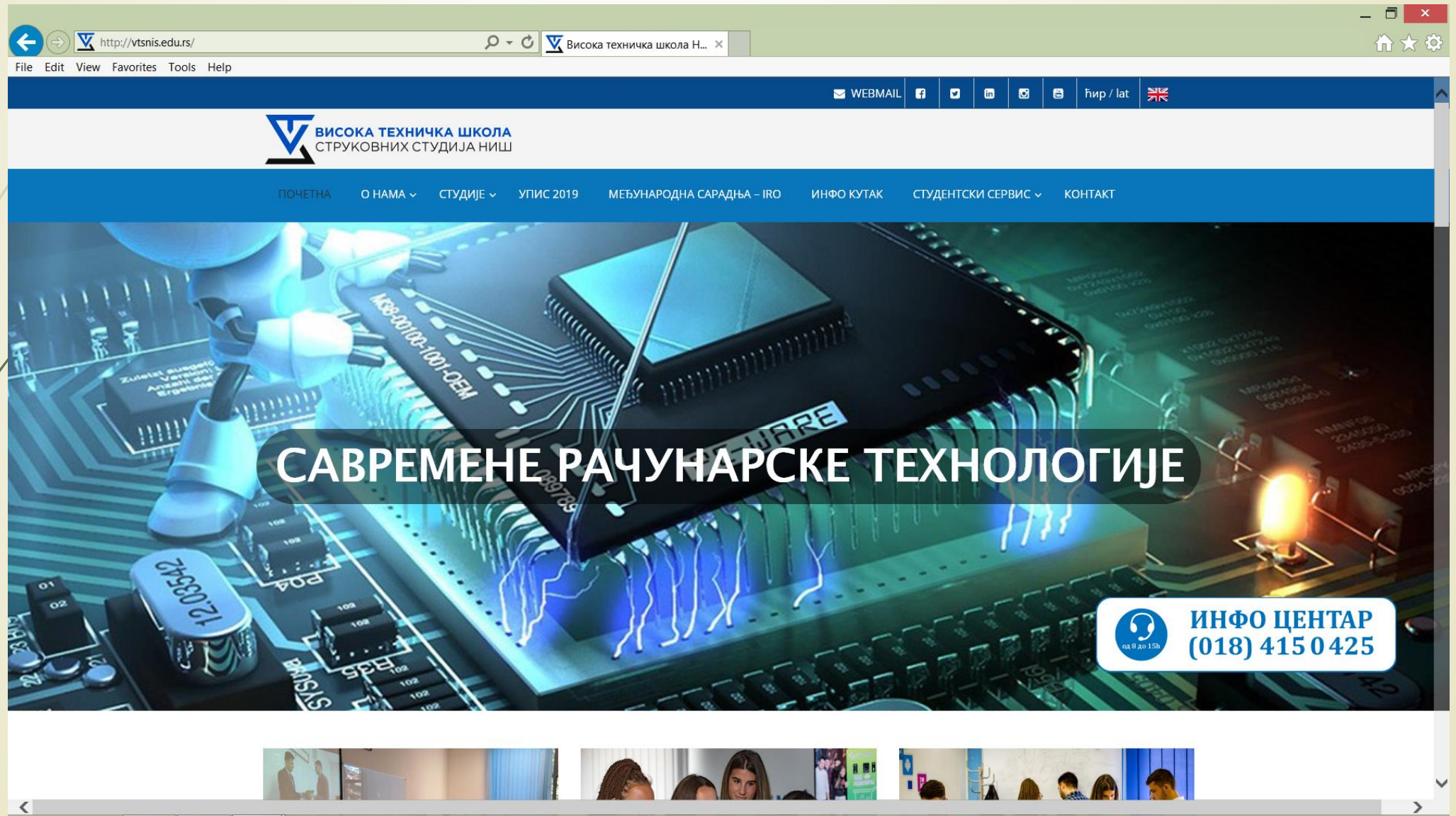
- ▶ Čitač Weba prevodi - **renderuje** odgovor klijenta – vraćenu **HTML** stranicu i prikazuje je u prozoru Web čitača kao:
 - ▶ Tekst,
 - ▶ Slike,
 - ▶ Neki drugi multimedijalni oblik (audio, video, gif animacije, VR, ...).
- ▶ Već smo predočili da **WEB SERVERI** i **WEB ČITAČI** najčešće komuniciraju korišćenjem protokola za PRENOS **HIPERTEKSTA** – **HTTP**-om, mada su danas mogući i drugi načini (SOAP, RESTful Web servisi, Web sockets, ...)
- ▶ Sa druge strane, za **PRENOS DATOTEKA** sa jednog računara na drugi, preko Interneta, koristi se - **FTP** protokol, za razmenu **ELEKTRONSKE POŠTE - SMTP** a za rad na UDALJENOM računaru - **TELNET**.
- ▶ Već znate, za prevođenje naziva hostova u IP adrese koristi se standardni protokol za podršku **DNS** (engl. *Domain Name System*), <https://www.cloudflare.com/learning/dns/what-is-dns/>.

Web čitači

- ▶ Web čitači (engl. *browsers*) ili Internet pretraživači su **OPŠTE KLIJENTSKE APLIKACIJE** koje se koriste za **PREUZIMANJE** i **PRIKAZIVANJE** resursa na Internetu.
- ▶ **HIPERLINKOVI** koji su prisutni u Internet resursima omogućavaju korisnicima da se lako kreću kroz **NIZ POVEZANIH** Web resursa.
- ▶ **Tim Berners-Li** je razvio prvi Web čitač – **Nexus** u decembru 1990., a 1991. je prezentovan naučnoj zajednici.
- ▶ Evo razloga za njegov nastanak kako ih je video Tim Berners-Li:
 - ▶ “I found it frustrating that in those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you had to learn a different program on each computer.”



Izgleđ renderovane Web stranice vtsnis.edu.rs



Jezik za struktuiranje Web stranica - HTML

- **HTML** – (engl. *HyperText Markup Language*), je **RAČUNARSKI JEZIK** koji služi za **OBELEŽAVANJE HIPERTEKSTA** i predstavlja osnovu **World Wide Weba (WWW)**.
- **HTML** predstavlja **ALAT** pomoću kojeg se **OPISUJU – STRUKTUIRAJU** Web stranice – Web sajtovi.
- **ČITAČI Weba** (engl. *Web browsers*) su programi koji **prevode HTML** kod (kaže se još i renderuju) pročitane dokumenta sa Web stranice i **prikazuju** njen sadržaj na **KLIJENTU** onako kako je dizajner Web stranice zamislio.
- Za pristup nekoj Web lokaciji korisnik se **MORA POVEZATI** sa Internetom (ili na LAN) i **POKRENUTI klijentsku aplikaciju** - ČITAC WEBA.
- Već je pokazano, na **zahtev korisnika** - Web server šalje traženu Web stranicu u formi **HTML (koda)** - fajla na računar posetioca (klijenta).

HTML oznake

- Web stranica je **TEKSTUALNA DATOTEKA** (ASCII fajl) sa **NAREDBAMA** u obliku **HTML OZNAKA** (engl. *tag*) i **ATRIBUTA**.
- Kada čitač prikaže sadržaj stranice, može se videti **HTML** (i drugi) **kod** kojim je ta stranica opisana (Chrome: *Ctrl+U*).
- Svaka oznaka u HTML-u ima svoju **POČETNU** (<) i **ZAVRŠNU** (>) signaturu.
- Iza početne signature sledi **NAZIV OZNAKE** i opciono **SKUP ATRIBUTA**.
- Svaka **početna** oznaku ima odgovarajuću **UPARENU završnu** oznaku:

<naziv oznake [atributi]>

a završne

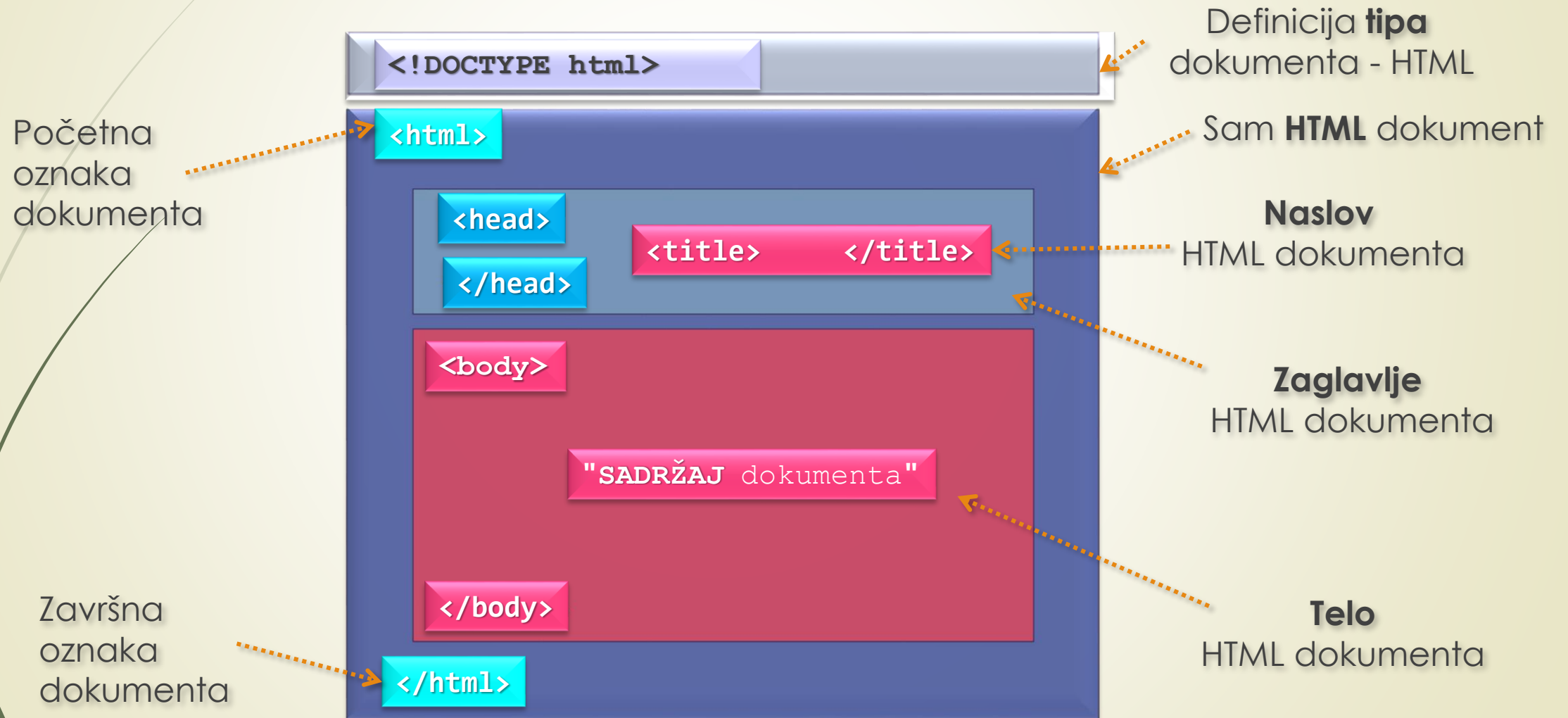
</naziv oznake >

- U sledećoj tabeli su prikazane **OSNOVNE OZNAKE** za formiranje **STRUKTURE** HTML dokumenta - **oznake odeljaka**.

Bazne HTML oznake odeljaka

OZNAKA ODELJKA	OBJAŠNENJE
<code><html> </html></code>	Parovih oznaka se pojavljuju na početku i na kraju HTML dokumenta, a između njih se nalaze sve oznake za definiciju izgleda Web stranice.
<code><head></head></code>	Par ovih oznaka predstavlja početnu i završnu oznaku zaglavlja i slede neposredno iza početnih <code><html></code> oznaka. Oznake u zaglavlju Web stranice se koriste za informacije o imenu autora i datuma postavljanja stranice, URL adrese resursa i sl.
<code><title> </title></code>	Naslov stranice se prikazuje na korisnikovom ekranu (u vrhu prouora Web čitača), i treba ga upisati između oznaka <code><title></code> i <code></title></code> .
<code><body> </body></code>	Početna i završna oznaka tela stranice. Ove oznake postavljaju se odmah iza zaglavlja Web stranice. Tekst u telu oznake definiše čitaču šta i na koji način da prikaže na ekranu posetioca Web stranice.

Primena HTML oznaka odeljaka



Strukturiranje HTML dokumenta

- ▶ **OZNAKE ODELJAKA** se koriste za sređivanje različitih delova HTML koda čime se definiše struktura (u prethodnim verzijama i izgled) stranice.
- ▶ Primer upotrebe **uparenih oznaka** odeljaka:

```
<html>  
  <head>  
    <title> Naslov Web stranice </title>  
  </head>  
  <body>  
    "Tekstualni sadržaj i HTML oznake tela stranice"  
  </body>  
</html>
```

- ▶ Za razliku od uparenih oznaka **prazne oznake** ne zahtevaju završnu signaturu (prelom reda **
**).

HTML oznaka pasusa

- Naziv HTML oznake čitaču definiše njenu **NAMENU**, a **ATRIBUTI** (koji su opcioni) iza naziva oznake daju Web čitaču **DODATNE INFORMACIJE** za interpretaciju te naredbe.
- Postoje **SPECIJALIZOVANI** (kreirani za odgovarajuću HTML oznaku) i **GLOBALNI** atributi kao što su **class**, **id**, **style**, ... i mogu se naći u **SVIM** HTML oznakama.
- Oznake za **POČETAK** i **KRAJ PASUSA** su **<p>** odnosno **</p>** i primenjuju se na **deo teksta**.
- U sledećem primeru, oznaka **<p>** nalaže Web čitaču da prikaže celokupan tekst **DO** oznake **</p>** primenjujući **PODRAZUMEVANA PRAVILA** za formatiranja pasusa:

<p> Tekst između početnih i krajnjih oznaka. </p>

- Za razliku od UPARENIH oznaka, **PRAZNE OZNAKE** ne zahtevaju završnu signaturu (primer: prelom reda **</br>**).
- Iako završna asignacija nije neophodna, ona se formalno postavlja : **</br>**

Kodni raspored znakova u HTML-u

- ▶ HTML oznaka **<meta>** se do pojave **HTML5** koristila za definisanje **opštih informacija** koje se **NE PRIKAZUJU** na ekranu Web čitača.
- ▶ Novine vezane za **KODNI RASPORED** znakova u čitaču, definisan je upravo HTML oznakom **<meta>**.
- ▶ Oznaka **<meta>** određuje **KAKO** će tekst biti prikazan na ekranu.

```
<html lang = "en">
  <head>
    <meta charset = "utf-8" />
  </head>
  <body>
    ...
  </body>
</html>
```

Postavlja se u **head** sekciji

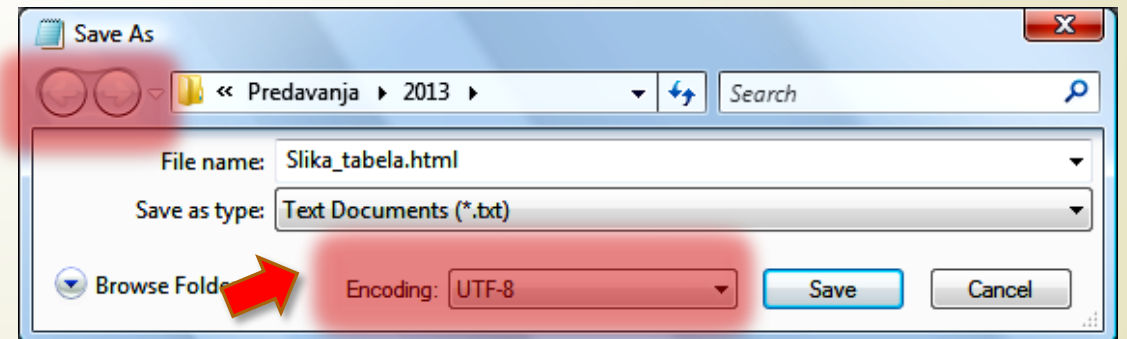
Atributi oznake **html** za korišćeni **jezik** i
oznake **meta** za **kodni raspored**

Karakteristi srpske latinice

- Da bi se prikazala slova srpske latinice na Web stranici, treba odabrati odgovarajući kodni raspored u **META OZNACI** sa odgovarajućom vrednošću atributa **charset (UTF-8)**.

```
<html>
  <head>
    <title>Srpska latinična slova</title>
    <meta http-equiv="content-type" content="text/html"; charset="utf-8" />
  </head>
  ...
</html>
```

- Takođe, značajno je da se izvorni kod HTML stranice **ZAPAMTI** u odgovarajućem kodnom rasporedu (**UTF-8**).



HTML i formatiranje teksta (1)

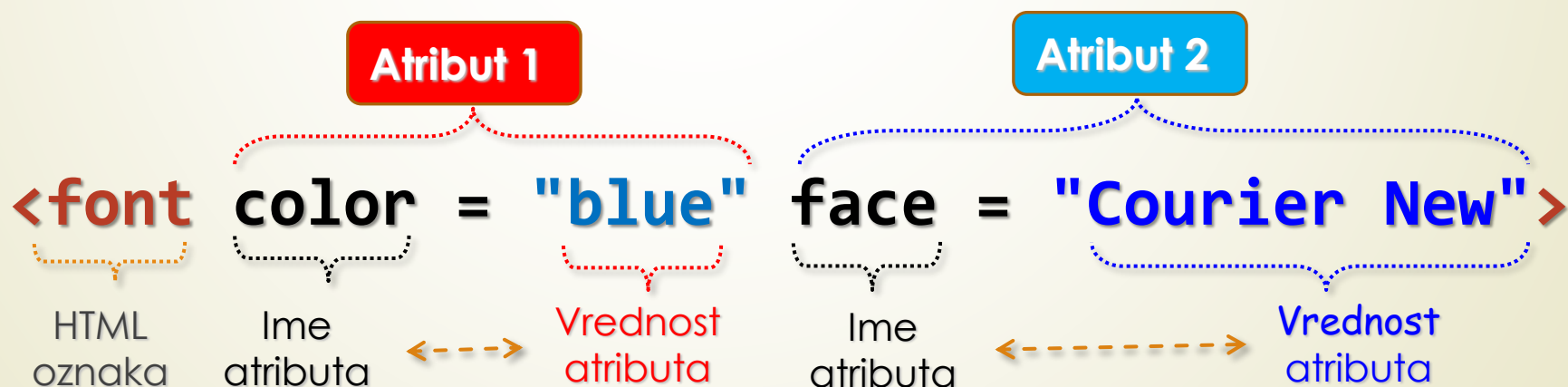
- Prelom reda **
, je primer **PRAZNE OZNAKE koja nalaže da se pređe u naredni red, pa tek onda sve ostalo.
- Novim standardom **HTML5** je uvedena **<wbr>** oznaka kojom se predlaže prelom teksta na odgovarajućem mestu ako je to potrebno.
- Iako su im pridodate i nove namene, **FORMATIRANJE TEKSTA** se obavlja unošenjem početne **oznake formata** na sam početak teksta:
 - ** ** Tekst koji se nalazi između ovih oznaka biće prikazan **podebljanim fontom**.
 - **<i> </i>** Tekst koji se nalazi između ovih oznaka biće prikazan **kurzivom** (**<cite> </cite>**)
 - **<u> </u>** Tekst koji se nalazi između ovih oznaka biće **podvučen**.
 - ** ** Tekst koji se nalazi između ovih oznaka biće **precrtan**.
 - **<code> </code>** Tekst koji se nalazi između ovih oznaka biće prikazan kao prog. kod.
 - **<ins> </ins>** Tekst koji se insertuje u dokument ...

HTML i formatiranje teksta (2)

- ▶ Uparena HTML oznaka za rad sa **FONTOVIMA** je:

**** i ****

- ▶ Iako se ova oznaka smatra zastarelom, ona je dobar primer prikaza upotrebe **atributa** i još vek se može naći u **in-line** primenama.
- ▶ Osnovni **atributi** ove oznake su "**color**" i "**face**".



HTML i formatiranje teksta (3)

- Dodavanje **ISTOG SKUPA ATRIBUTA** većem broju SUSEDNIH ELEMENATA stranice može se obaviti postavljanjem željenih elemenata između oznaka `<div>` `</div>` ili `` ``.
- Oznaka `<div>` nema značaja za korisnika, ona je bitna za Web čitače!
- Dodeljivanjem atributa "**align**" (poravnanje) između početne oznake `<div>` i njene uparene završne oznake `</div>` poravnavaju se **SVI ELEMENTI** između ovih oznaka.
- Primer dodavanja atributa skupu oznaka:

```
<div align = "center">
```

```
<p> Ovaj tekst se nalazi IZNAD slike.</p>
```

```

```

```
<p> Ovaj tekst se nalazi ISPOD slike.</p>
```

```
</div>
```

Kontejner načinjen
div oznakom



Kontejneri u HTML-u (1)

- ▶ HTML element **<div>** je u sledećem primeru iskorišćen kao **KONTEJNER** koji grupiše sliku i dva paragrafa čiji je izgled formatiran **CSS PRAVLIMA** čije je ime „**listing**“.
- ▶ Kaskadni opisi stilova **CSS** (engl. *Cascade Style Sheat*) formiraju klase koje se koriste za **opis izgleda stranice** i nevezani su za sadržaj (o njima više kasnije).

```
<div class="listing">
```

```

```

```
<p> <cite> Praktikum Internet Laboratorijskih vežbi</cite>, Z.  
Veličković, S. Stošović </p>
```

```
<p> Primeri dobre prakse iz osnova HTML-a. </p>
```

```
</div>
```

Završna asignacija za
praznu oznaku (img)

Kontejneri u HTML-u (2)

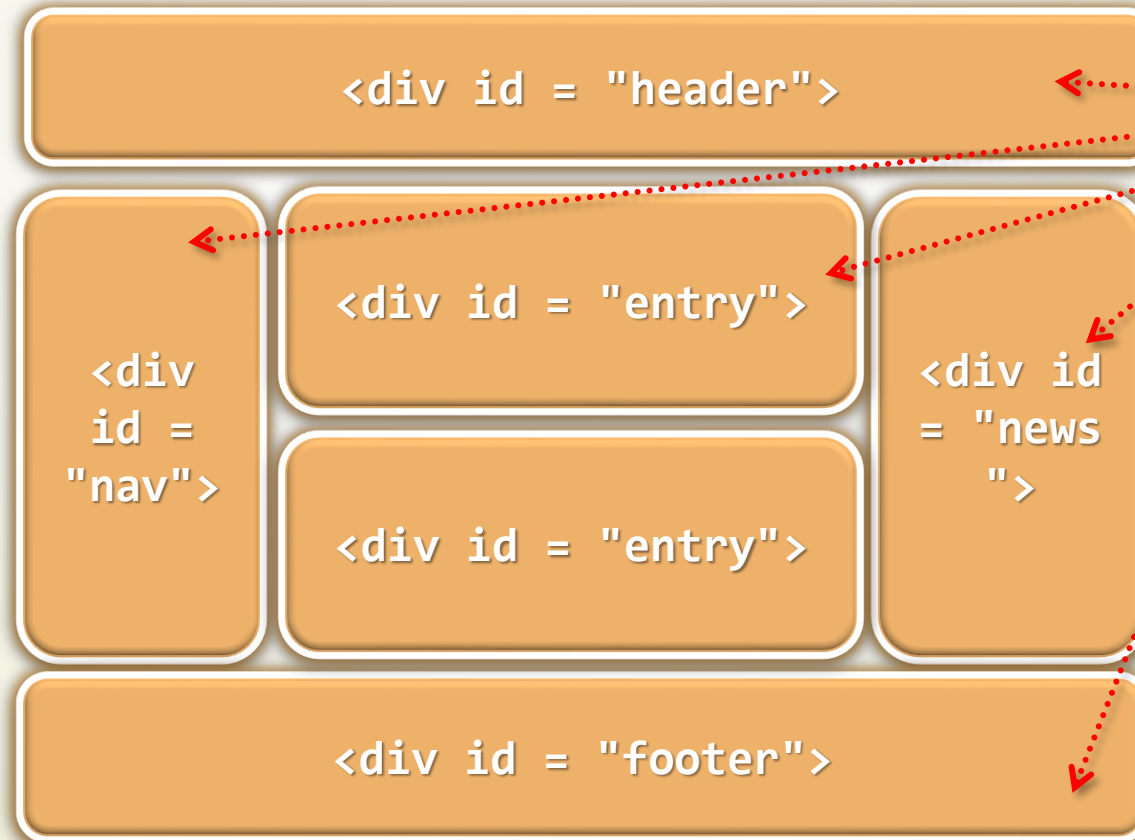
- ▶ Još jedan način korišćenja oznake `<div>` za **DELJENJE STRANICE U SEKCIJE**.
- ▶ U primeru se ovako definisana sekcija referencira identifikatorom koji koristi atribut `id` „`news`“.
- ▶ **JavaScript** i **CSS SELEKTORI** će "`news`" kontejner prepoznati putem njegove identifikacije koja je naznačena ATRIBUTOM "`id`" i potom izvršiti zahtevanu radnju (ili kod koji mu je namenjen).

Atribut `news` služi za mapiranje odgovarajućeg `div` elemenata

```
<div id = "news">  
  <p> Radili smo na projektu ...</p>  
  <p> Poteškoće u realizaciji ... </p>  
</div>
```


<div> oznaka za strukturiranje stranice

- ▶ Struktura HTML dokumenta bazirana na **<div>** oznaci i **"id"** atributu.



Kontejneri formirani oznakama **"div"** i atributom **"id"**