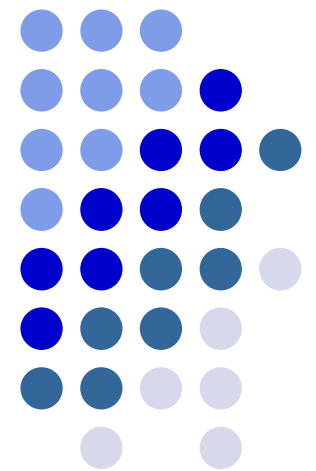
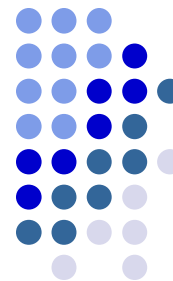


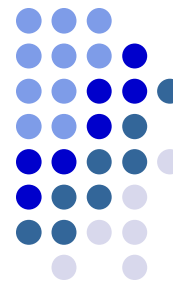
Kaskadni opisi stilova





Kaskadni opisi stilova - CSS

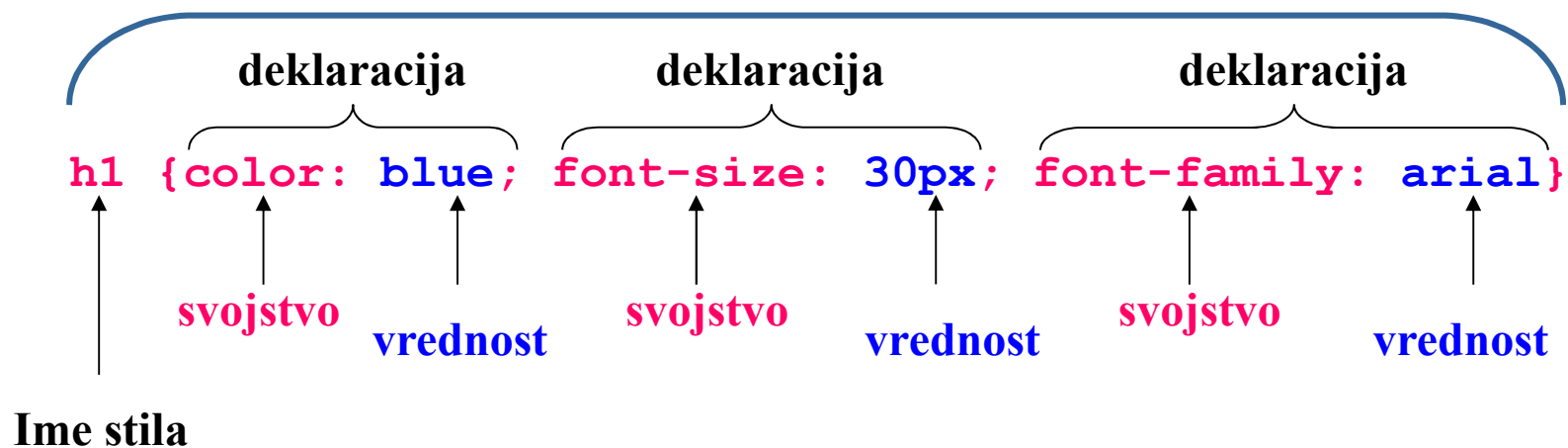
- ❑ **Kaskadni opisi stilova CSS - Cascade Style Sheets** – skup odrednica (zvanih pravila) pomoću kojih se u potpunosti upravlja rasporedom i izgledom stranice
 - ❑ Pomoću kaskadnih stilova odvaja se struktura od izgleda Web strane
 - ❑ Opisi stilova omogućavaju pisanje naredbi za izgled i formatiranje u zaglavlju Web stranice ili u spoljašnjoj datoteci, van HTML koda kojim se zadaje sadržaj stranice
 - ❑ Nasuprot upotrebi tabela za raspoređivanje elemenata stranice, CSS pravila su jednostavni tekstualni iskazi, pomoću kojih istovremeno može da se raspoređuje i formatira jedan ili više elemenata
-
-

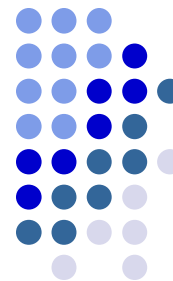


Kreiranje opisa stilova

Opisi stilova se sastoje od jednog ili više pravila

CSS pravilo:

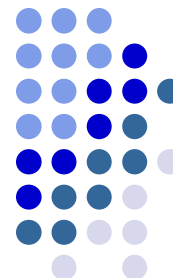




Kreiranje opisa stilova

- ❑ Svaka HTML oznaka može se koristiti kao ime stila, a jedno pravilo može imati više imena stila
- ❑ Na primer:

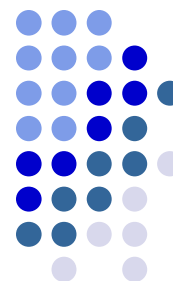
```
h2, p, ul {color: green; font-family: arial}
```
- ❑ Ovo pravilo čitaču govori da prikaže naslov drugog nivoa, pasus i neuređenu listu teksta zelenom bojom i fontom Arial.



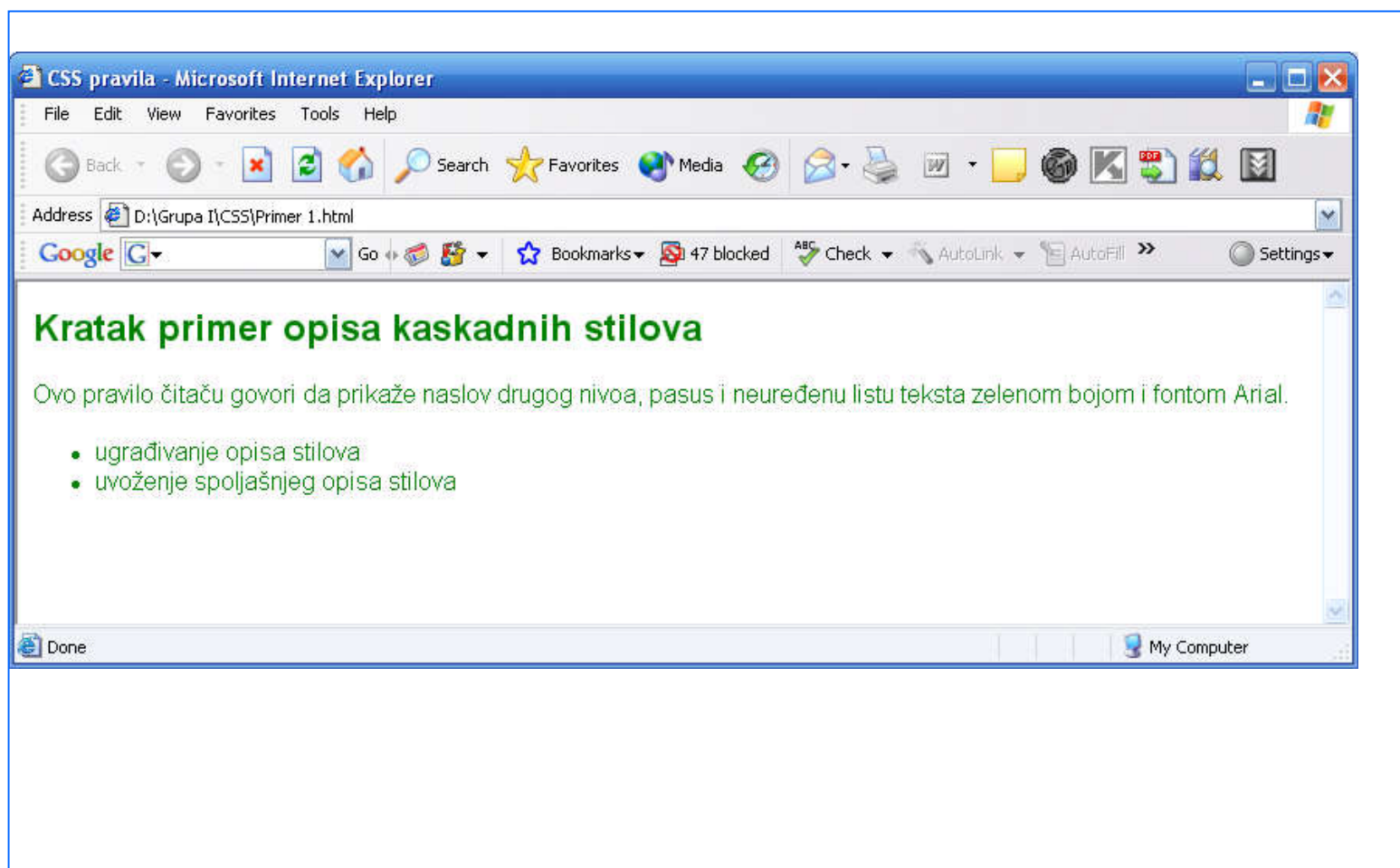
Primer 1.

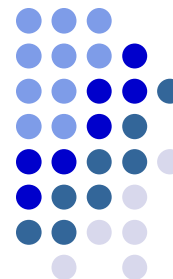
```
<html>
<head>
<title> CSS pravila </title>
<style type="text/css">
h2, p, ul {color: green; font-family: arial}
</style>
</head>
<body>
<h2> Kratak primer opisa kaskadnih stilova </h2>
<p> Ovo pravilo čitaču govori da prikaže naslov drugog nivoa, pasus i
neuređenu listu teksta zelenom bojom i fontom Arial. </p>
<ul> <li> ugrađivanje opisa stilova </li>
      <li> uvoženje spoljašnjeg opisa stilova </li> </ul>
</body>
</html>
```

Snimate dokument pod nazivom Primer1.html na D disk (D:\Grupa I\CSS\Primer1.html)



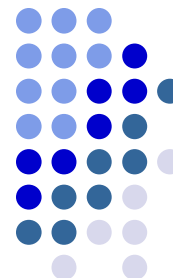
Primer 1.





Kreiranje opisa stilova

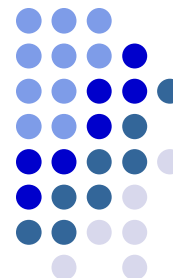
- ❑ Svaka HTML oznaka može biti ime stila. CSS pravila mogu se primenjivati na slike (koristeći *img* kao ime stila) ili tabele (*table* kao ime stila)
- ❑ Ime stila *table* čitaču govori da pravilo opisa stilova primeni na tabele na stranici
- ❑ U narednom primeru, čitač će napraviti tabelu sa:
 - ❖ zelenom pozadinom i
 - ❖ belim tekstom veličine 20 piksela,
 - ❖ odmaknutu od leve margine za 150 piksela i 200 piksela od vrha stranice



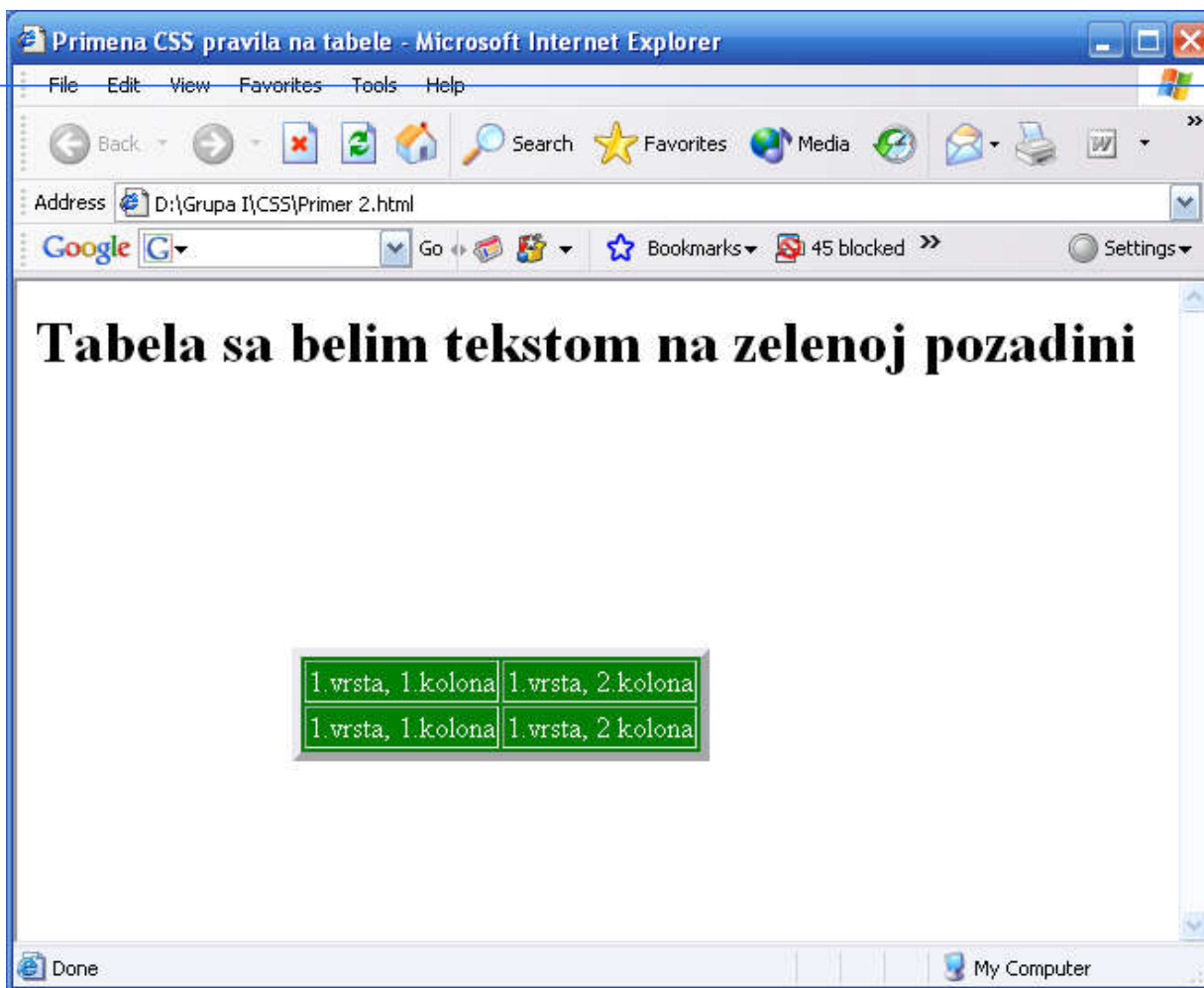
Primer 2.

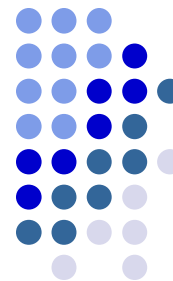
```
<html>
<head>
  <title> Primena CSS pravila na tabele </title>
  <style type="text/css">
    table {background:green; color: white; font-size: 20px;
    position:absolute; left:150px; top:200px}
  </style>
</head>
<body>
<h1> Tabela sa belim tekstom na zelenoj pozadini </h1>
<table border="5">
  <tr> <td> 1.vrsta, 1.kolona </td>
      <td> 1.vrsta, 2.kolona </td>
  <tr> <td> 1.vrsta, 1.kolona </td>
      <td> 1.vrsta, 2 kolona </td> </tr>
</table>
</body>
</html>
```

Snimate dokument pod nazivom Primer2.html na D disk (D:\Grupa I\CSS\Primer2.html)



Primer 2.





Nasleđivanje – Primer 3.

- U CSS terminologiji, element koji se nalazi unutar drugog je potomak, a spoljašnji element je roditelj.

- Na primer:

```
<html>
```

```
<head>
```

```
  <title> Nasleđivanje </title>
```

```
  <style type="text/css">
```

```
    p {color:blue}
```

```
    b {color:red}
```

```
  </style>
```

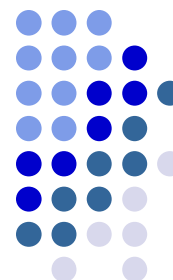
```
</head>
```

```
<body>
```

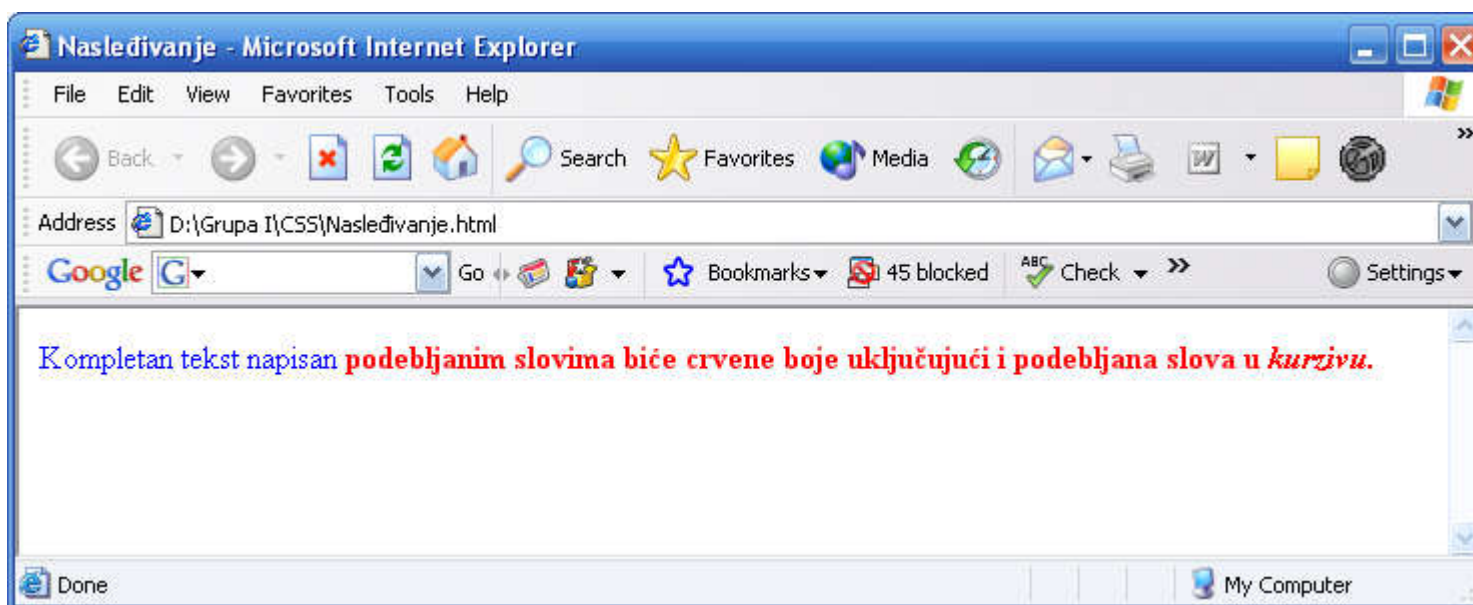
```
  <p> Kompletan tekst napisan <b>podebljanim slovima biće crvene boje  
  uključujući i podebljana slova u <i> kurzivu </i></b> </p>
```

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

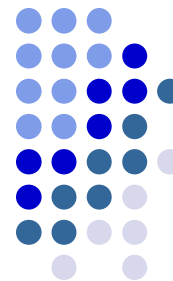
Snimite dokument pod nazivom Primer3.html na D disk (D:\Grupa I\CSS\Primer3.html)



Nasleđivanje – Primer 3.



Povezivanje sa spoljašnjim opisima stilova



- ❑ Spoljašnji opisi stilova su idealni kada se isti skup pravila želi primeniti na elemente više stranica
- ❑ Ukoliko se spoljašnji opisi stilova povežu sa svim stranicama Web lokacije, menjanjem CSS pravila u samo jednoj datoteci promeniće se izgled elemenata na celoj lokaciji
- ❑ U HTML dokumentima lokacije navodi se datoteka s opisom stilova u okviru oznake **<link>**
- ❑ Oznaka **link** postavlja se u zaglavlje Web stranice:

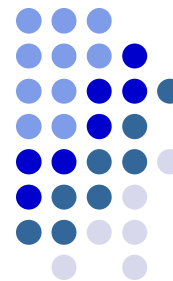
```
<head>
```

```
  <title> Linked Cascading Style Sheets (CSS) </title>
```

```
  <link rel=stylesheet href="Styles.css" type="text/css">
```

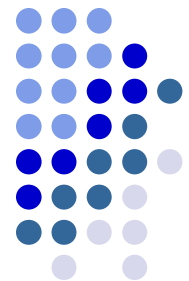
```
</head>
```

Povezivanje sa spoljašnjim opisima stilova



```
<link rel="stylesheet" href="Styles.css" type="text/css">
```

- ❑ Oznaka **<link>** govori čitaču da primeni CSS pravila formatiranja iz spoljašnje datoteke (**Styles.css**)
- ❑ Atribut **rel** ukazuje na tip veze; **rel="stylesheet"** obaveštava čitač da dokument na koji se odnosi veza jeste strana sa stilovima.
- ❑ Atribut **type** prepoznaje datoteku kao tekstualnu datoteku s CSS podacima
- ❑ Atribut **href** čitaču govori ime i ako je neophodno putanju do datoteke s definicijama opisa stilova koju treba da otvori
- ❑ Ako čitač ne može da pronađe spoljašnju datoteku (naznačenu atributom **href**) on će koristiti svoje podrazumevane parametre



Povezivanje sa spoljašnjim opisima stilova

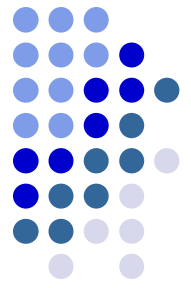
- ❑ U spoljašnjoj datoteci opisa stilova (npr. **Styles.css**) pravila se definišu na sledeći način:

```
h1 {color:blue; font-size:40px; font-family:helvetica}
```

```
p {color:white; background:green; font-family:verdana; text-indent:2cm}
```

```
body {background:#add8e6; color:maroon}
```

- ❑ U spoljašnjim opisima nema HTML oznaka jer spoljašnja datoteka opisa stilova treba da sadrži samo CSS pravila
- ❑ Kreirati datoteku Styles.css na D disku (D:\CSS\Styles.css), koja će sadržati gore navedena pravila



Zadatak

- ❑ Kreirati datoteku **Styles.html** na D disku (D:\CSS\Styles.html) čiji će sadržaj biti naslov prvog nivoa i jedan pasus.

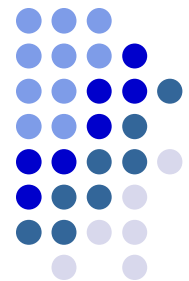
- ❖ Tekst naslova 1. nivoa:

Povezivanje sa spoljašnjim opisima stilova

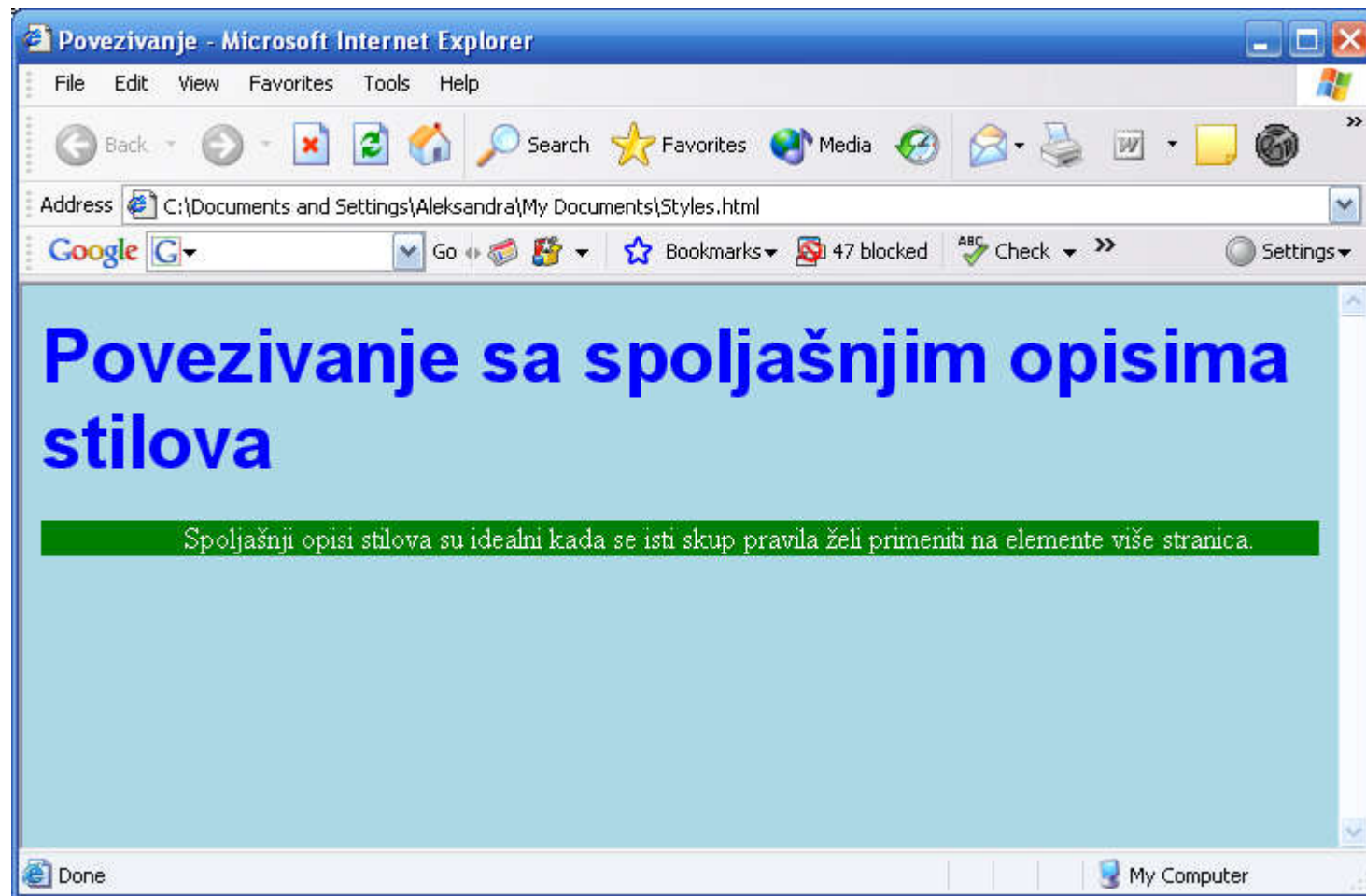
- ❖ Pasus sadrži sledeći tekst:

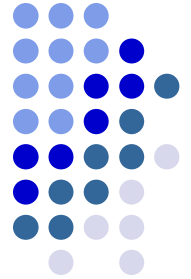
Spoljašnji opisi stilova su idealni kada se isti skup pravila želi primeniti na elemente više stranica

- ❑ Povezati realizovanu spoljašnju datoteku opisa stilova **Styles.css** sa realizovanim HTML dokumentom **Styles.html**.
-
-



Zadatak





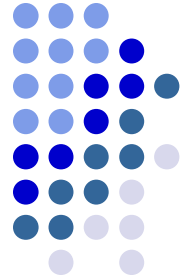
Styles.css

```
h1 {color:blue; font-size:40px; font-family:  
helvetica}
```

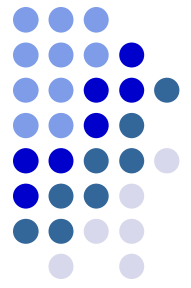
```
p {color:white; background:green; font-  
family:verdana; text-indent:2cm}
```

```
body {background:#add8e6; color:maroon}
```

Styles.html



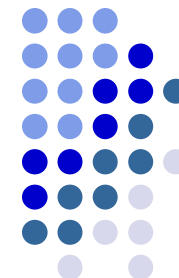
```
<html>
<head>
<title> Povezivanje </title>
<link rel="stylesheet" href="Styles.css" type="text/css">
</head>
<body>
<h1> Povezivanje sa spoljašnjim opisima stilova </h1>
<p> Spoljašnji opisi stilova su idealni kada se isti
    skup pravila želi primeniti na elemente više stranica.
</p>
</body>
</html>
```



Zadavanje stilova unutar HTML oznaka

- ❑ Skup CSS pravila formatiranja može se primeniti **na jednu** Web stranicu – ugrađivanjem opisa stilova
- ❑ Skup CSS pravila formatiranja može se primeniti **na sve** stranice Web lokacije – uvoženjem ili povezivanjem spoljašnje datoteke opisa stilova sa svakim HTML dokumentom lokacije
- ❑ Pravilo formatiranja može se primeniti **samo na određeni element** Web stranice (npr. jedan pasus ili naslov), a ne na sve elemente istog tipa
 - ❖ Za zadavanje CSS pravila koje se odnosi na samo jednu oznaku koristi se atribut *style* u HTML oznaci
 - ❖ `<imeoznake style="CSS deklaracije">`

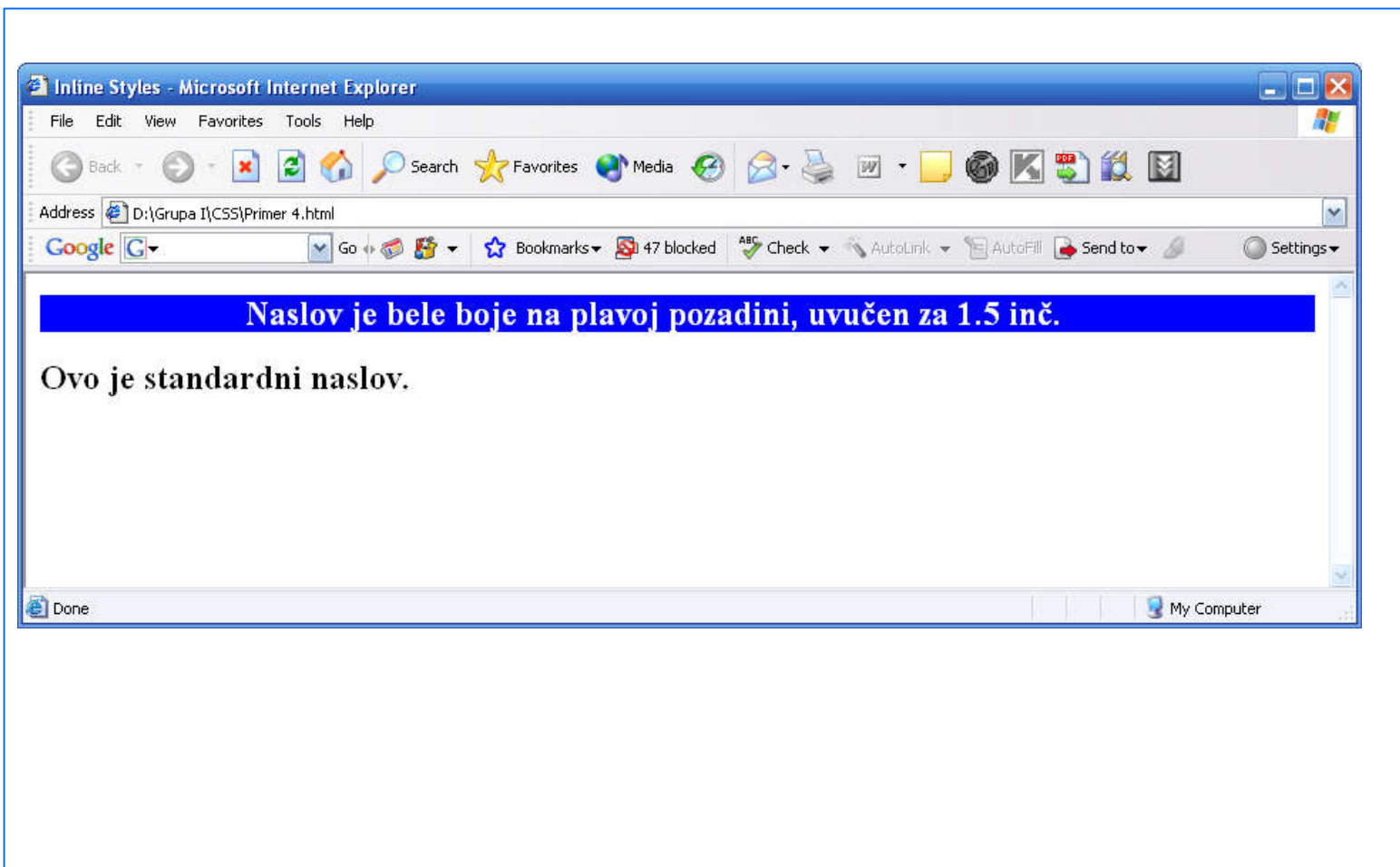
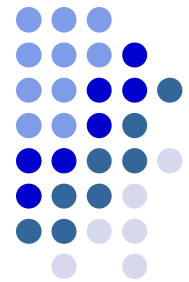
Primer 4.

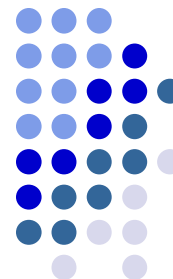


```
<html>
  <head>
    <title> Inline Styles </title>
  </head>
  <body>
    <h2 style="text-indent:1.5 in; background:blue;
    color:white">
    Naslov je bele boje na plavoj pozadini, uvučen za 1.5
    inč </h2>
    <h2> Ovo je standardni naslov </h2>
  </body>
</html>
```

Snimite dokument pod nazivom Primer4.html na D disk (D:\Grupa I\CSS\Primer4.html)

Primer 4.





Primena više CSS pravila na jedno ime stila

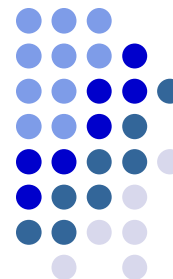
- CSS klasi može se zadati bilo koje ime – između imena stila (za HTML oznaku) i imena klase u svakoj definiciji CSS pravila potrebno je staviti tačku

```
p.critical {color:red; font-size:18pt; font-weight:bold}
```

- Primena formatiranja klase na element – postavljanjem atributa **class** u HTML oznaku elementa na sledeći način:

```
<p class="critical">
```

↑
ImeKlase

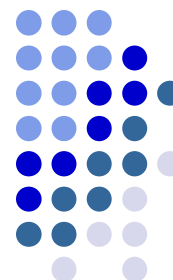


Primer 5.

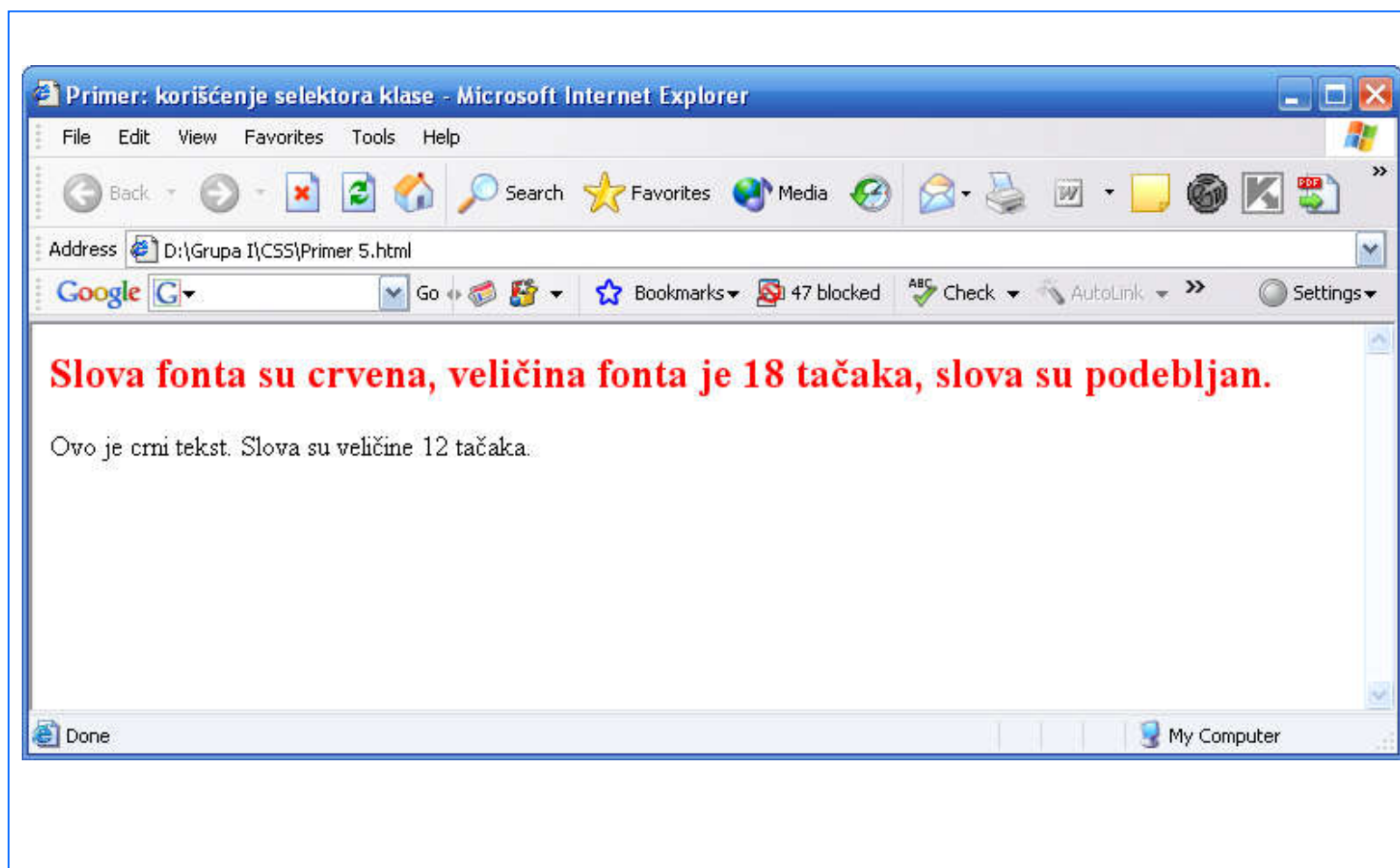
```
<html>
  <head>
    <title> Primer: korišćenje selektora klase </title>
    <style type="text/css">
      <!--
        p.critical {color:red; font-size:18pt; font-weight:bold}
        p.normal {color:black; font-size:12pt}

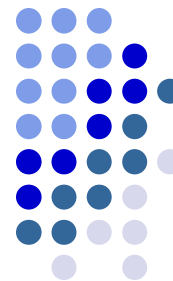
      -->
    </style>
  </head>
  <body>
    <p class="critical"> Slova fonta su crvena, veličina fonta je 18
    tačaka, slova su podebljana </p>
    <p class="normal"> Ovo je crni tekst. Slova su veličine 12
    tačaka.</p>
  </body>
</html>
```

Snimite dokument pod nazivom Primer5.html na D disk (D:\Grupa I\CSS\Primer5.html)



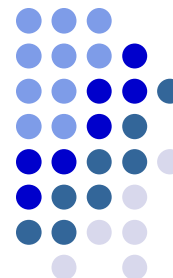
Primer 5.





Definicije CSS klasa

- ❑ Ako je klasa pridružena određenom imenu stila – može se koristiti za određivanje izgleda samo elementa na koji utiče HTML oznaka povezana s tim imenom stila
 - ❑ Definicije CSS klasa mogu se učiniti fleksibilnijim ako se izostavi ime stila iz deklaracije klase
 - ❑ Ako se CSS klasa definiše bez imena stila može se koristiti za određivanje stila svakog elementa stranice
-
-

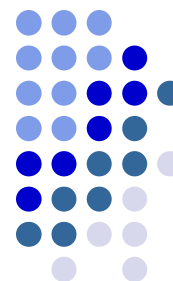


Primer 6.

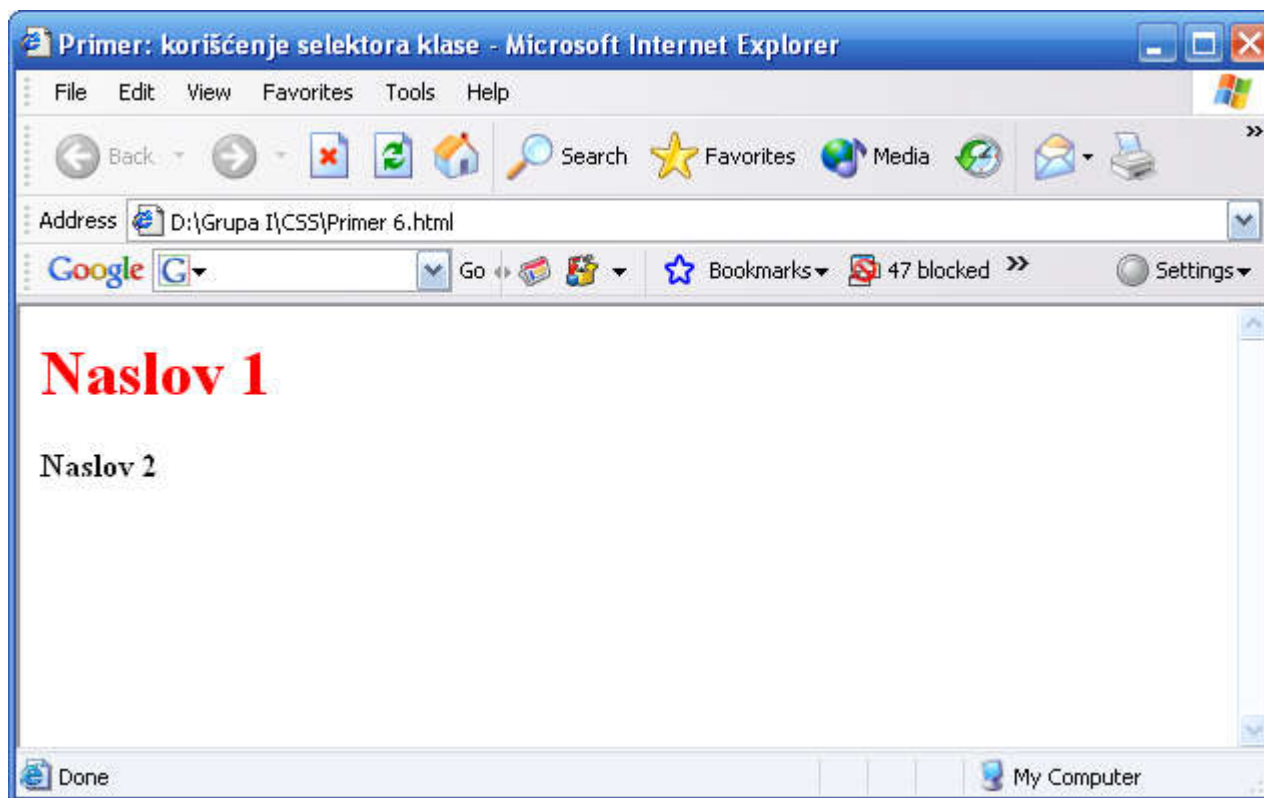
```
<html>
  <head>
    <title> Primer: korišćenje selektora klase </title>
    <style type="text/css">
      <!--
        .critical {color:red; font-size:18pt; font-weight:bold}
        .normal {color:black; font-size:12pt}

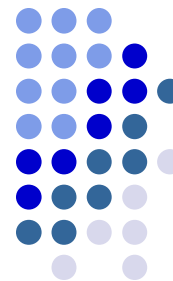
      -->
    </style>
  </head>
  <body>
    <h1 class="critical"> Naslov 1 </h1>
    <h2 class="normal"> Naslov 2 </h2>
  </body>
</html>
```

Snimite dokument pod nazivom Primer6.html na D disk (D:\Grupa I\CSS\Primer6.html)



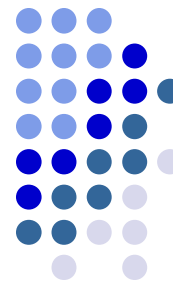
Primer 6.





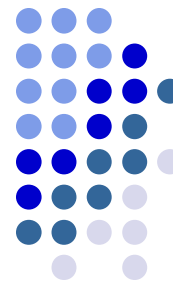
Svojstva fonta – font-family

- ❑ Vrednost svojstva **font-family** između vitičastih zagrada ({})
deo je CSS deklaracije izgleda koja čitaču govori koji font da
upotrebi
- ❑ Primer:
 - ❖ `p {font-family:helvetica}`
 - ❖ Čitač tekst svih pasusa prikazuje fontom Helvetica
- ❑ Moguće je zadati **rezervne fontove**, za slučaj da primarni nisu
instalirani na računaru posetioca
 - ❖ `p {font-family:helvetica, times new roman,
courier}`



Svojstva fonta - font-size

- ❑ CSS svojstvo **font-size** pruža mogućnost zadavanja veličine teksta Web stranice u tačkama:
`p {font-size:12pt}`
- ❑ CSS svojstvo **font-size** pruža mogućnost zadavanja veličine teksta Web stranice u pikselima:
`p {font-size:10px}`
- ❑ CSS svojstvo **font-size** pruža mogućnost zadavanja veličine teksta Web stranice brojem tačaka elementa roditelja:
`cite {font-size:1.5 em}`
`p {font-size:12pt}`
`<p>` Ceo tekst koji se nalazi između početne i završne oznake pasusa je deo elementa pasusa. `<cite>` Ovaj element citat je potomak pasusa čija ga početna i završna oznaka okružuju. `</cite>` Kada stavljate jedan HTML element u drugi, unutrašnji element je potomak, a spoljašnji roditelj. `</p>`



Izgled slova i reči, prored

- ❑ Svojstvo letter-spacing omogućava upravljanje razmakom između slova:

```
p {letter-spacing: 5px}
```

- ❑ Svojstvo word-spacing upravlja razmakom između reči:

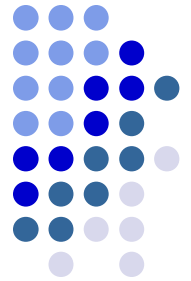
```
p {word-spacing: 5px}
```

- ❑ Svojstvo line-height upravlja proredom (podrazumevana vrednost ovog svojstva je 1 em):

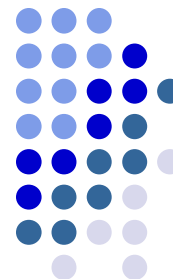
```
p {line-height: 12pt}
```

- ❑ Svojstva za upravljanje izgledom slova, reči i proreda mogu se zadati u bilo kojim CSS jedinicama mere (pt, px, in, cm, itd)

Uvlačenje pasusa i upravljanje razmakom između elemenata



- ❑ Uvlačenje prvog reda pasusa moguće je korišćenjem pravila
`p {text-indent: 2em}`
 - ❖ Iako se mogu koristiti sve jedinice mere (pt, px, in, cm) najbolje je koristiti emove, jer se emovi menjaju u skladu sa promenom veličine fonta
- ❑ CSS svojstva margin-top i margin-bottom koriste se za upravljanje prostorom iznad, odnosno ispod elementa
- ❑ Na primer, za dva prazna reda ispod pasusa u opisu stilova može se napisati ime stila:
`p {margin-bottom: 2em}`



Primer 7.

Svojstva:
font-family
font-size
color
letter-spacing
word-spacing
line-height
margin-bottom
text-indent

h1: Arial Black, zelene boje

p: Courier New, crvene boje, 12pt

b: plave boje, 1.5em

i: crne boje, 20px

10px

1 cm

1 cm

5px

2em

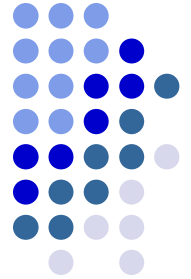
2em

1 cm

Prva linija prvog pasusa
Druga linija prvog pasusa.

Prva linija drugog pasusa.
Druga linija drugog pasusa.

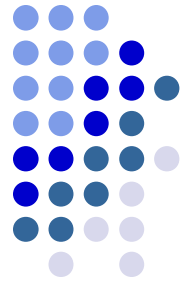
Done My Computer



Primer 7.

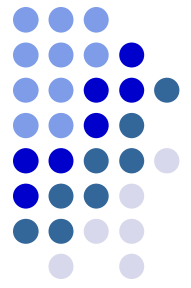
```
<html><head>
  <title> Svojstva fonta i pasusa </title>
  <style type="text/css">
<!--
h1{color:green; font-family:Arial Black; letter-spacing:10px}
p {margin-bottom:2em; font-family:Courier New; color:red; font-
  size:12pt;text-indent:1cm}
b {font-size:1.5em; color:blue; word-spacing:5px}
i {font-size:20px; color:black}
-->
  </style></head>
  <body>
    <h1> Naslov </h1>
    <p>Prva linija prvog pasusa <br>
      Druga linija prvog pasusa.
    <p style="line-height:2em">
      <b>Prva linija drugog pasusa.</b> <br>
      Druga linija drugog <i>pasusa.</i> </p>
  </body></html>
```

Snimite dokument pod nazivom Primer7.html na D disk (D:\Grupa I\CSS\Primer7.html)



Raspored elemenata na stranici

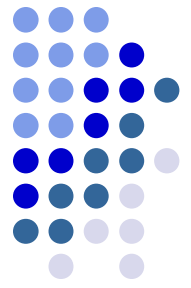
- ❑ CSS svojstvo **position** zadaje poziciju elementa na stranici
- ❑ **Apsolutno** raspoređivanje omogućava postavljanje objekta u odnosu na levu ivicu i vrh stranice
- ❑ **Relativno** raspoređivanje omogućava određivanje mesta elementa u odnosu na to gde bi se inače pojavio
- ❑ Korišćenjem apsolutnog raspoređivanja jedan elemenat može se postaviti preko drugog



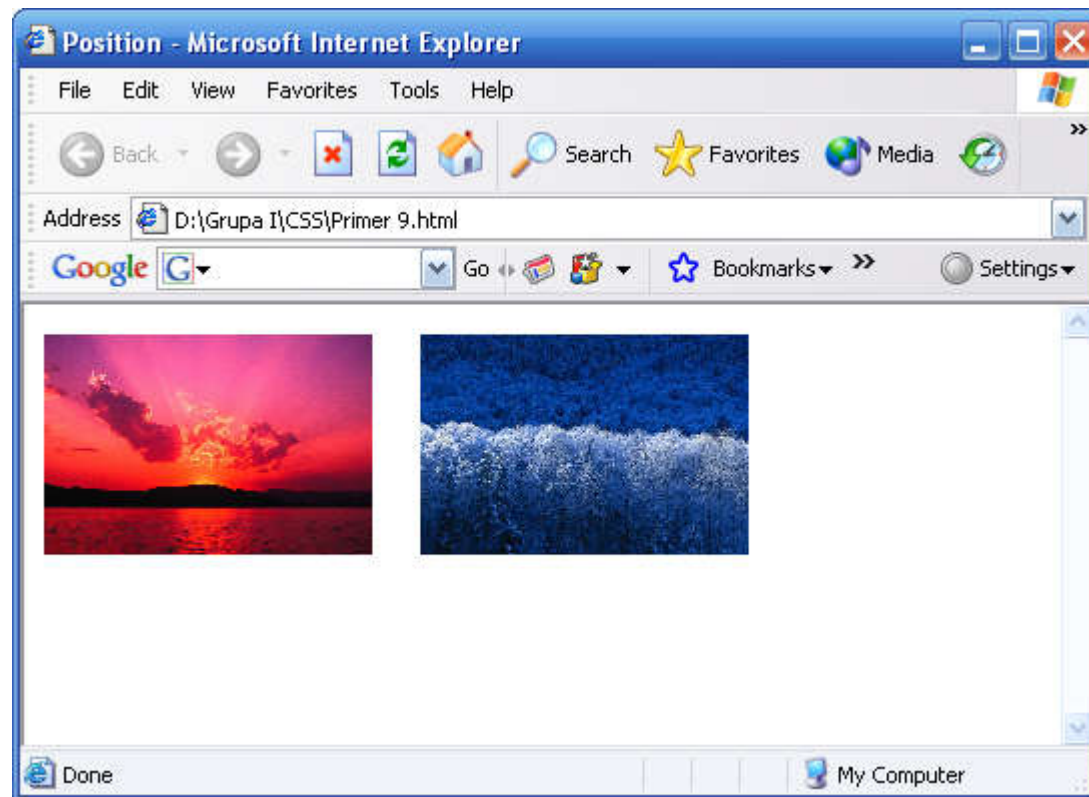
Primer 9.

```
<html>
  <head><title> Relative position </title></head>
<body>
  
  
</body></html>
```

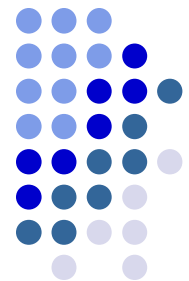
Snimite dokument pod nazivom Primer9.html na D disk (D:\Grupa I\CSS\Primer9.html)



Primer 9.



Primer 10.



```
<html>

  <head><title> Absolute position </title></head>

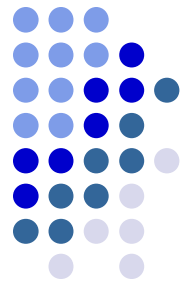
<body>

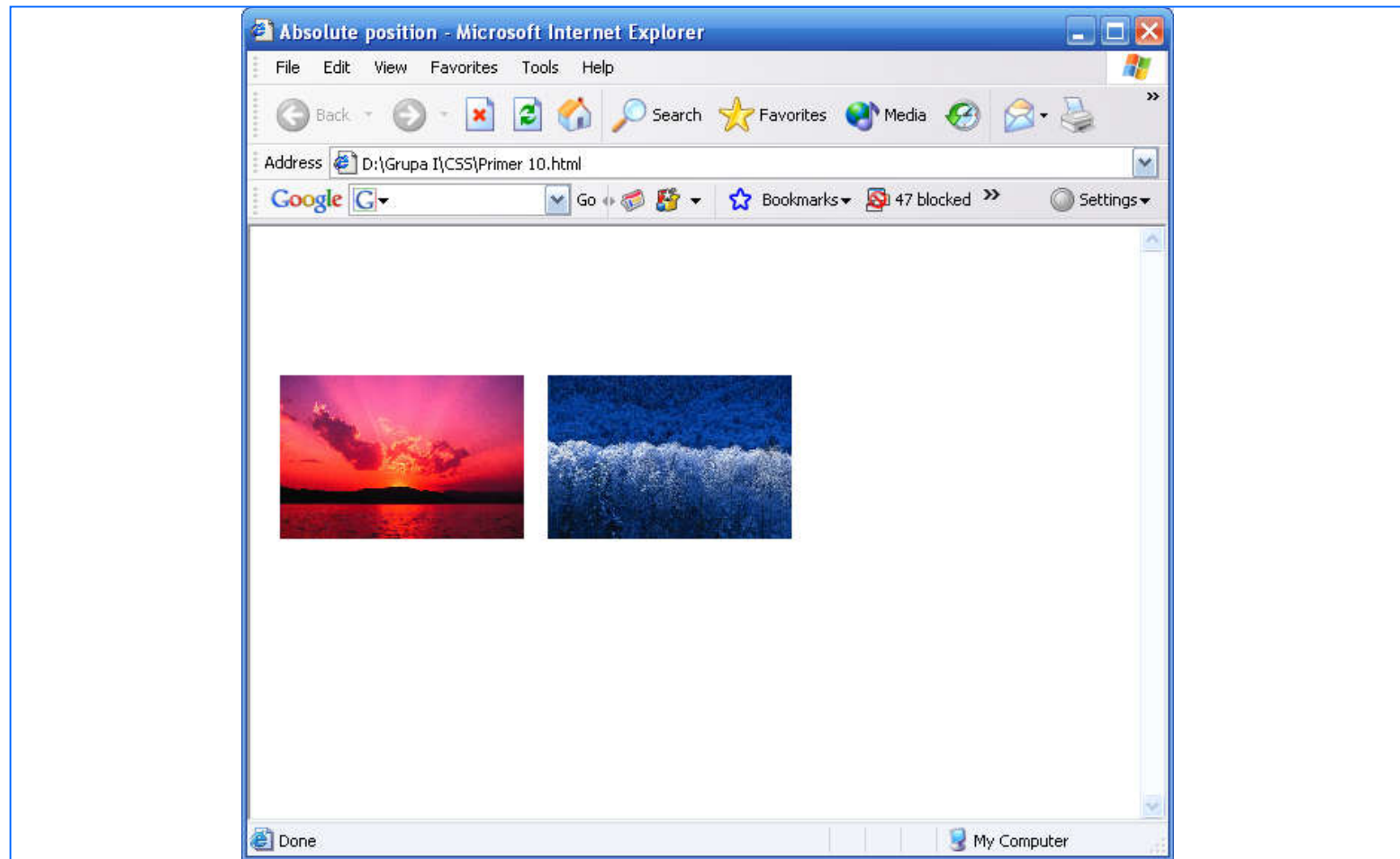
  

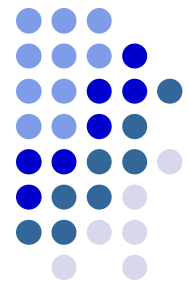
</body></html>
```

Snimite dokument pod nazivom Primer10.html na D disk (D:\Grupa I\CSS\Primer10.html)



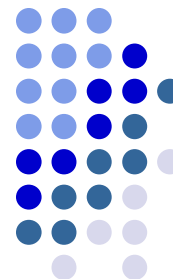
Primer 10.





Poravnanje teksta Web stranice

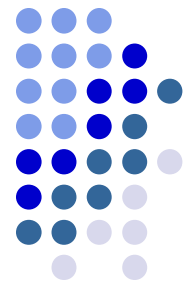
- ❑ CSS mogućnosti za poravnanje teksta postavljaju se u opisu stila, definisanjem nepridružene klase pravila:
 - ❖ `.left {text-align:left}`
 - ❖ `.right {text-align:right}`
 - ❖ `.center{text-align:center}`
 - ❖ `.justify{text-align:justify}`
- ❑ Pomoću atributa **class** u HTML oznakama može se poravnavati tekst u okviru bilo kog elementa koji može sadržati tekst (tekst tela stranice, ćelija tabele, samo određenih ćelija ili određenih redova tabele...)
- ❑ Na primer, poravnanje uz obe ivice celog teksta:
 - ❖ `<body class="justify">`



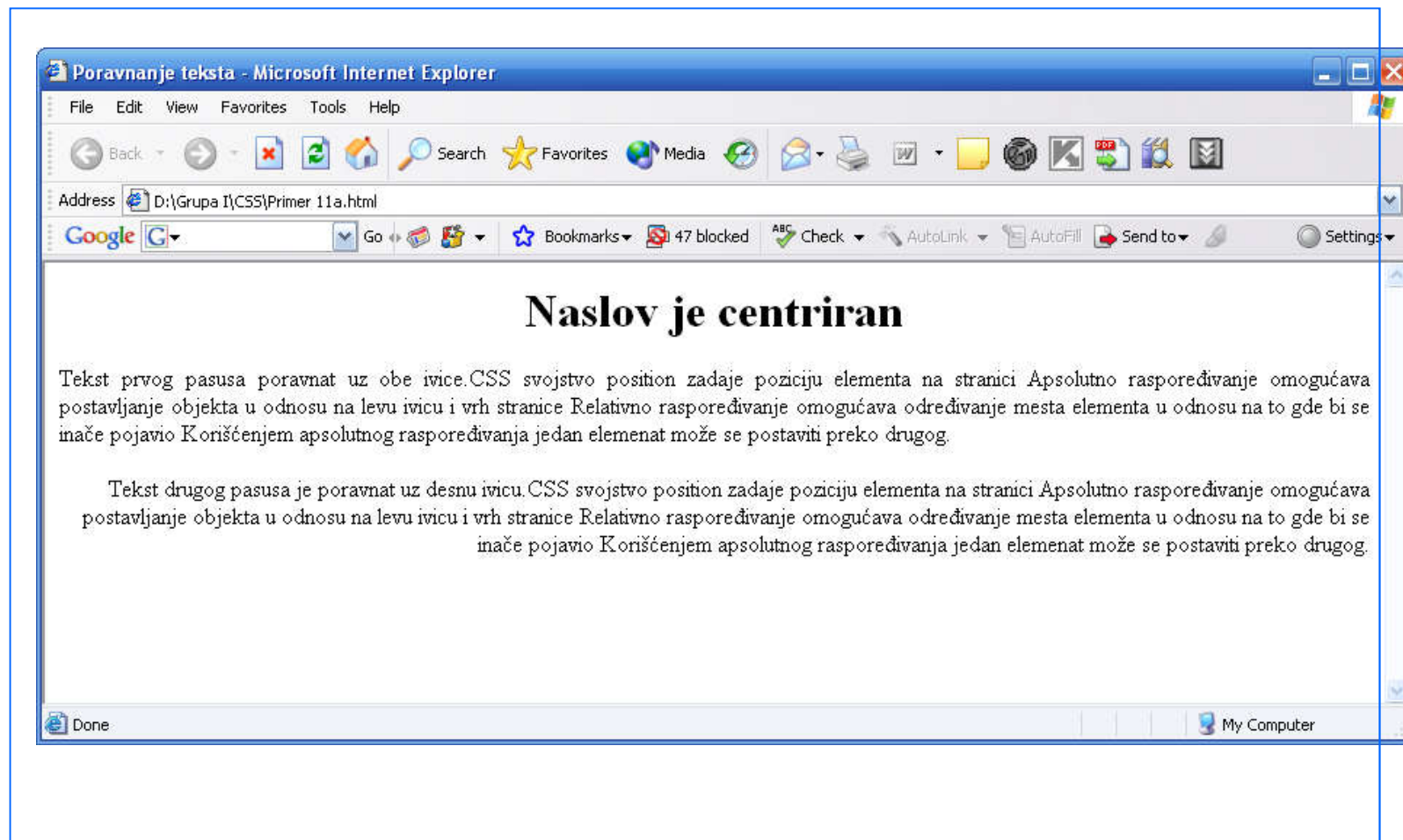
Primer 11.

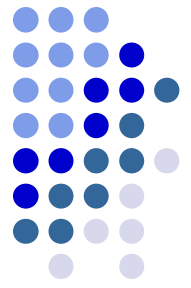
```
<html>
  <head>
    <title> Poravnanje teksta </title>
    <style type="text/css">
<!--
      .left {text-align:left}
      .right {text-align:right}
      .center{text-align:center}
      .justify{text-align:justify}
-->
    </style>
  </head>
  <body>
    <h1 class="center"> Naslov je centriran </h1>
    <p class="justify"> Tekst prvog pasusa poravnat uz obe ivice.
    </p>
    <p class="right"> Tekst drugog pasusa je poravnat uz desnu ivicu.
    </p>
  </body>
</html>
```

Snimite dokument pod nazivom Primer11.html na D disk (D:\Grupa I\CSS\Primer11.html)



Primer 11.

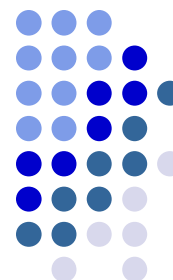




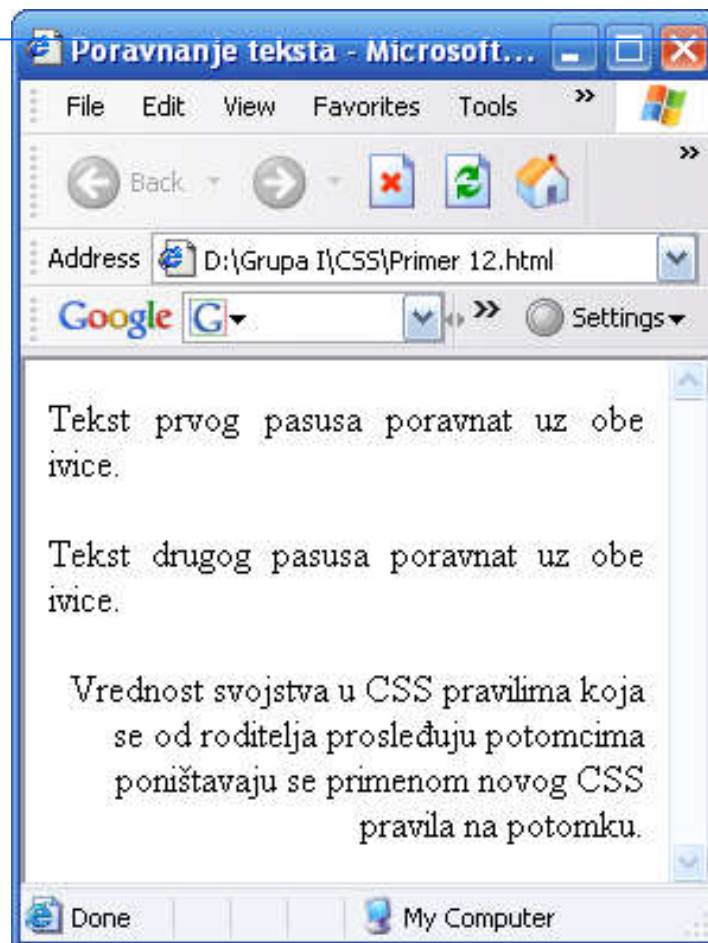
Primer 12.

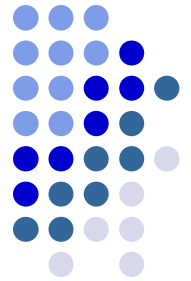
```
<html>
  <head>
    <title> Primer: poravnanje teksta </title>
    <style type="text/css">
      <!--
      .left {text-align:left}
      .right {text-align:right}
      .center{text-align:center}
      .justify{text-align:justify}
      -->
    </style>
  </head>
  <body class="justify">
    <p> Tekst prvog pasusa poravnat uz obe ivice. </p>
    <p> Tekst drugog pasusa poravnat uz obe ivice. </p>
    <p class="right"> Vrednost svojstva u CSS pravilima koja se od
    roditelja prosleđuju potomcima poništavaju se primenom novog CSS
    pravila na potomku. </p>
  </body>
</html>
```

Snimite dokument pod nazivom Primer12.html na D disk (D:\Grupa I\CSS\Primer12.html)



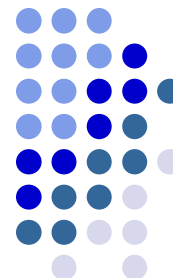
Primer 12.





Upravljanje marginama

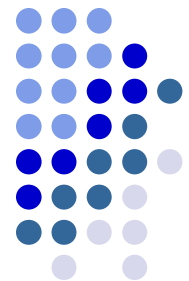
- ❑ CSS svojstva margina omogućavaju precizno upravljanje prostorom koji čitač ostavlja između elementa i ivice stranice ili između dva susedna elementa
- ❑ Postoje 4 svojstva margina: **margin-left**, **margin-right**, **margin-top** i **margin-bottom**.
- ❑ Svako svojstvo označava stranu elementa na kojoj je razmak
- ❑ Za zadavanje širine margine mogu se koristiti sve CSS jedinice za meru dužine (px, pt, em, in, mm, cm, itd.)



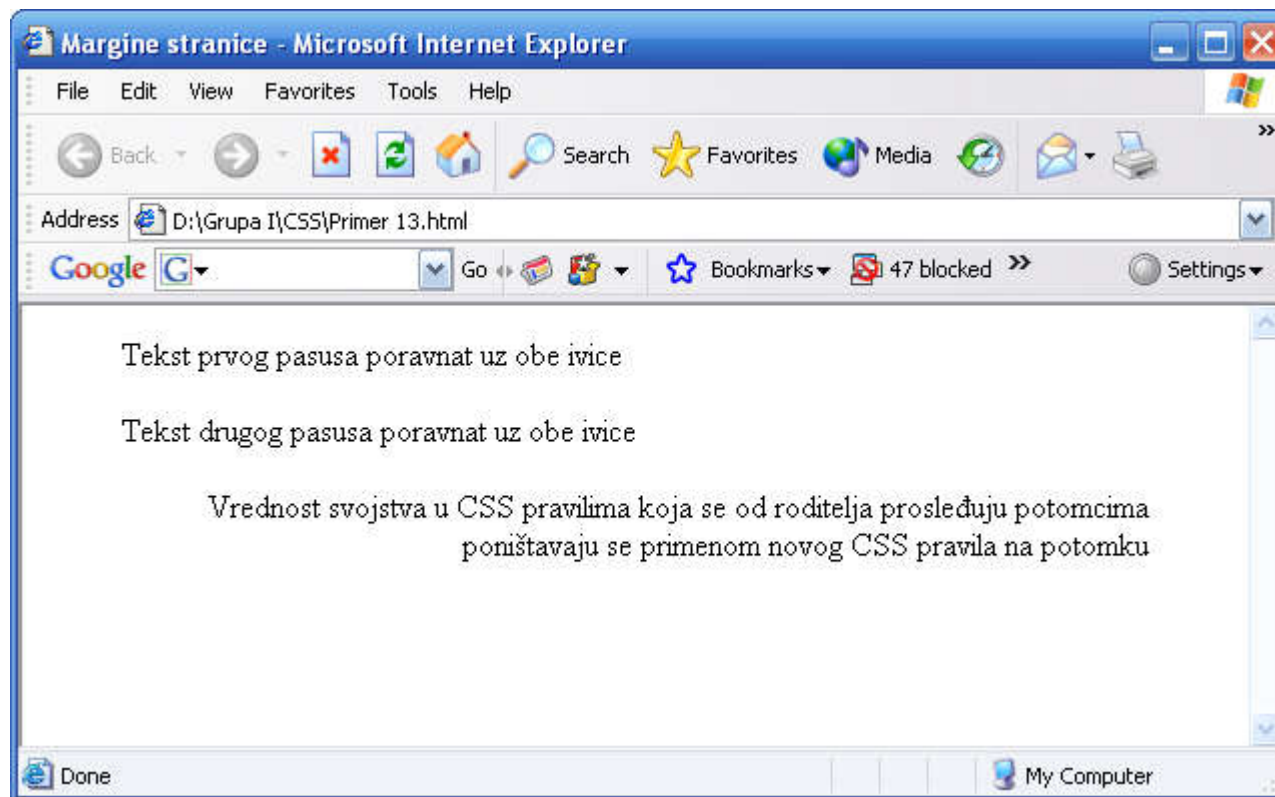
Primer 13.

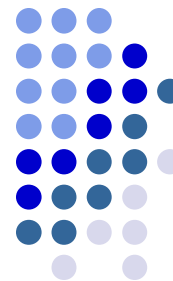
```
<html>
  <head>
    <title> Margine stranice </title>
    <style type="text/css">
      <!--
      .left {text-align:left}
      .right {text-align:right}
      .center{text-align:center}
      .justify{text-align:justify}
      body {margin-left:50px; margin-right:50px}
      -->
    </style>
  </head>
  <body class="justify">
    <p> Tekst prvog pasusa poravnat uz obe ivice </p>
    <p> Tekst drugog pasusa poravnat uz obe ivice </p>
    <p class="right"> Vrednost svojstva u CSS pravilima koja se od
    roditelja prosleduju potomcima poništavaju se primenom novog CSS
    pravila na potomku </p>
  </body>
</html>
```

Snimite dokument pod nazivom Primer13.html na D disk (D:\Grupa I\CSS\Primer13.html)



Primer 13.

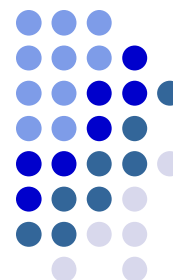




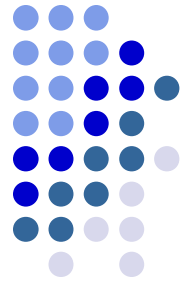
Podešavanje izgleda liste

- ❑ CSS svojstvo **list-style-type** koristi se za zadavanje izgleda oznaka elemenata liste
- ❑ CSS svojstvo **list-style-position** koristi se za određivanje mesta oznake elementa liste u odnosu na tekst elementa liste. Ovo svojstvo može imati jednu od dve vrednosti:
 - ❖ **Outside** – čitač postavlja oznaku liste izvan teksta elementa liste
 - ❖ **Inside** – čitač prikazuje oznaku elementa liste kao deo teksta elementa

Vrednosti svojstva list-style-type



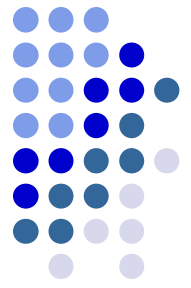
<u>Vrednost</u>	<u>Opis</u>	<u>Vrednost</u>	<u>Opis</u>
disk	Pun crn krug	upper-latin	A, B, C, D, itd.
circle	Nepopunjen crn krug	hebrew	Tradicionalno hebrejsko numerisanje
square	Pun crn kvadrat	armenian	Tradicionalno jermensko numerisanje
decimal	1,2,3,4,5, itd.	georgian	Gruzijsko numerisanje
decimal-leading-zero	01,02,..98,99 ili 001,002,..,099,100, itd.	CJK-ideographic	Obični ideografski brojevi
lower-roman	i,ii,iii,iv,v, itd.	hiragana	a, i, u, e, o, ka, ki, itd.
upper-roman	I, II, III, IV, V, itd.	katana	A, I, U, E, O, KA, KI, itd.
lower-greek	Tradicionalno grčko numerisanje (alfa, beta, itd.)	hiragana-iroha	i, ro, ha, ni, ho, he, to, itd
lower-alpha	a, b, c, d, itd.	katakana-iroha	I, RO, HA, NI, HO, HE, TO, itd
lower-latin	a, b, c, d, itd.	none	Bez oznake, samo razmak
upper-alpha	A, B, C, D, itd.	inherit	Koristi stil liste roditeljskog elementa



Primer 14.

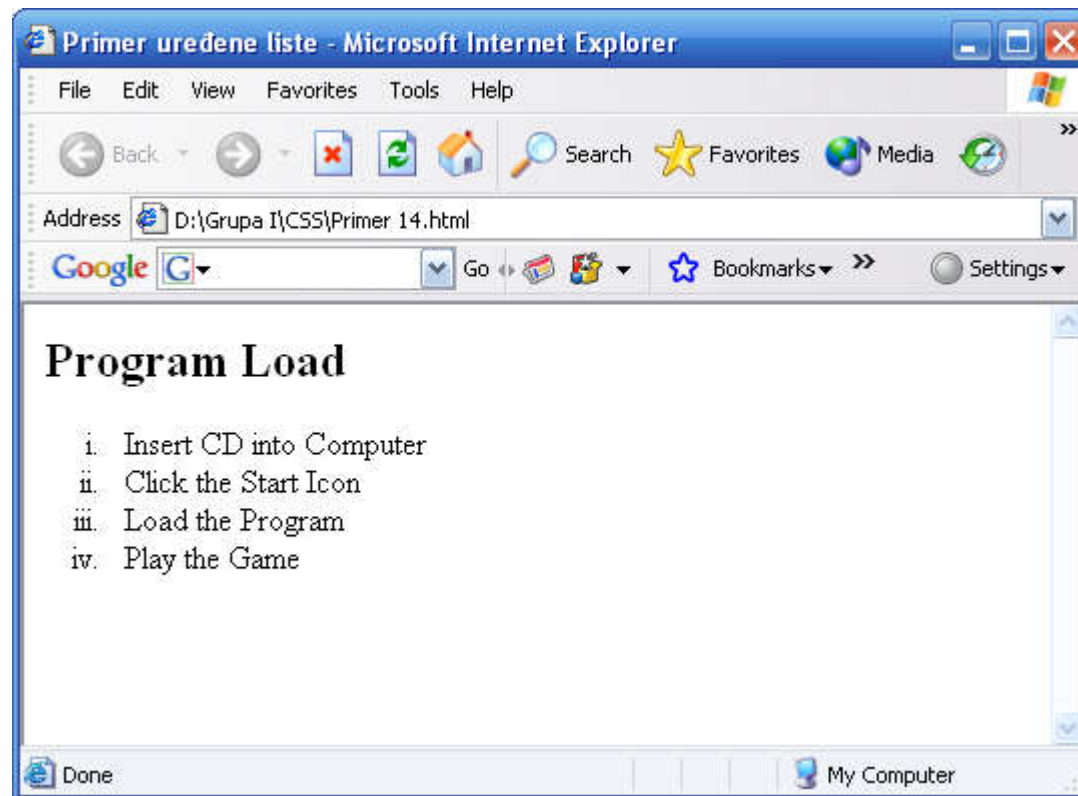
(D:\CSS\Primer 14.)

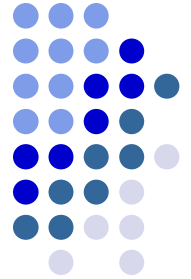
```
<html>
<head>
  <title> Primer uredene liste </title>
</head>
<body>
  <h2> Program Load </h2>
  <ol style="list-style-type:lower-roman">
    <li> Insert CD into Computer </li>
    <li> Click the Start Icon </li>
    <li> Load the Program </li>
    <li> Play the Game </li>
  </ol>
</body>
</html>
```



Primer 14.

(D:\CSS\Primer 14.)

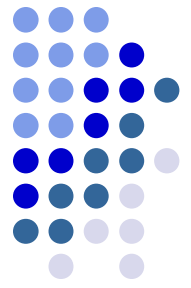




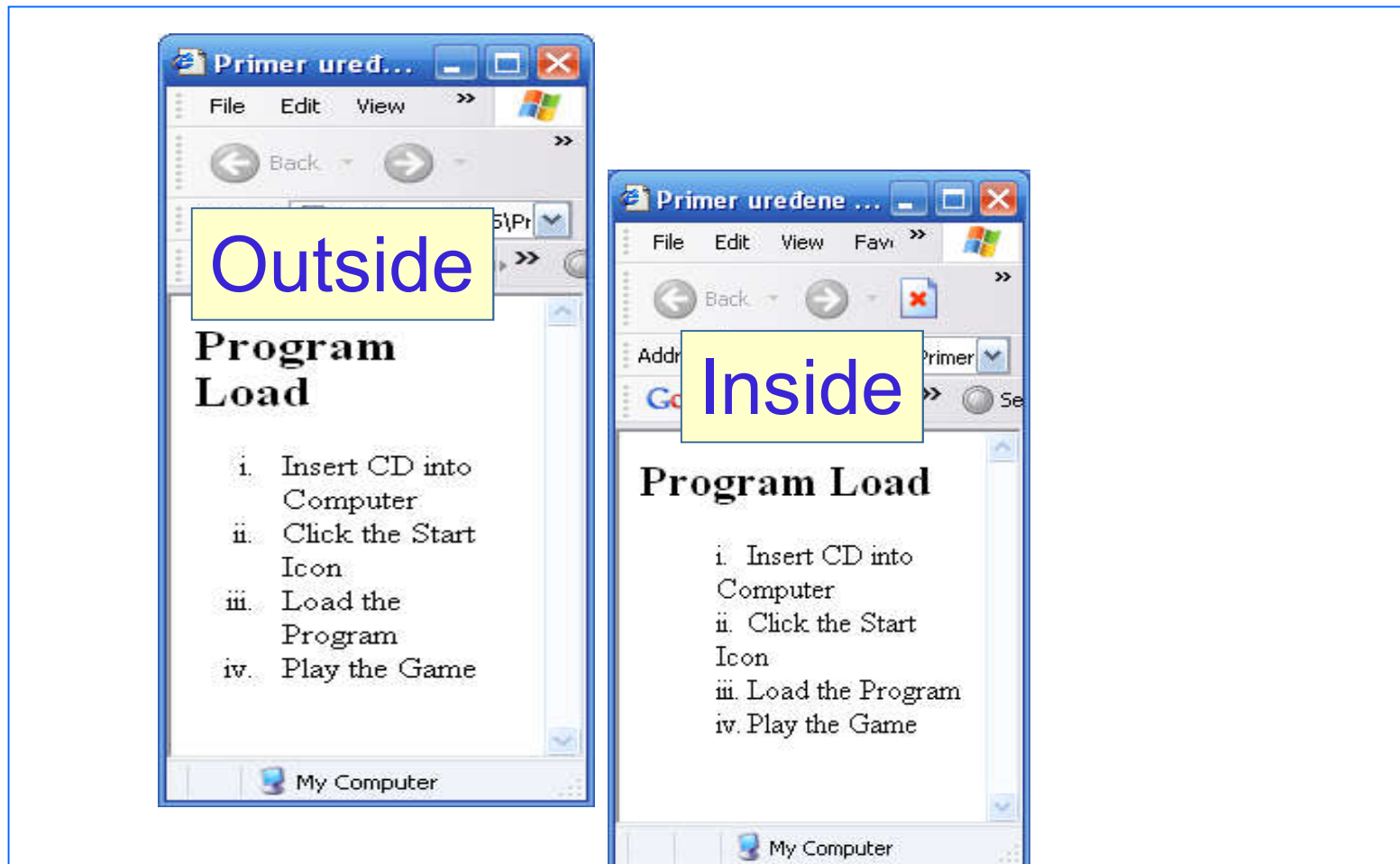
Primer 15.

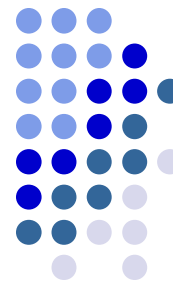
```
<html>
<head>
  <title> Primer uredene liste </title>
</head>
<body>
  <h2> Program Load </h2>
  <ol style="list-style-type:lower-roman; list-style-
position:outside">
    <li> Insert CD into Computer </li>
    <li> Click the Start Icon </li>
    <li> Load the Program </li>
    <li> Play the Game </li>
  </ol>
</body>
</html>
```

Snimite dokument pod nazivom Primer15.html na D disk (D:\Grupa I\CSS\Primer15.html)



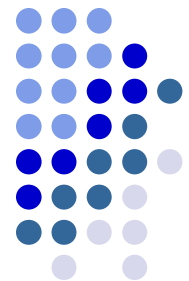
Primer 15.





Upravljanje kursorom

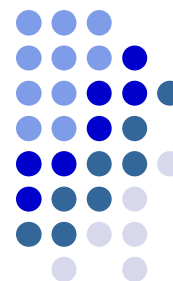
- ❑ CSS svojstvo **cursor** ukazuje čitaču koji oblik pokazivača miša da upotrebi
- ❑ Promenom podrazumevanog kursora, posetiocima se daje do znanja da su tekst ili slika ispod njega po nečemu posebni
- ❑ Primer:
 - ❖ `abbr, acronym, .help {border-bottom:1px dotted black; cursor:help}`
- ❑ Kursor dobija oblik znaka pitanja kad god posetilac pređe mišem preko elementa `<acronym>`, `<abbr>` ili elementa klase `help`
- ❑ Atribut **title** može se postaviti u HTML oznake da bi ukazao na pomoćni tekst koji će čitač pokazati kada se pokazivač miša nalazi iznad elementa čija oznaka sadrži atribut



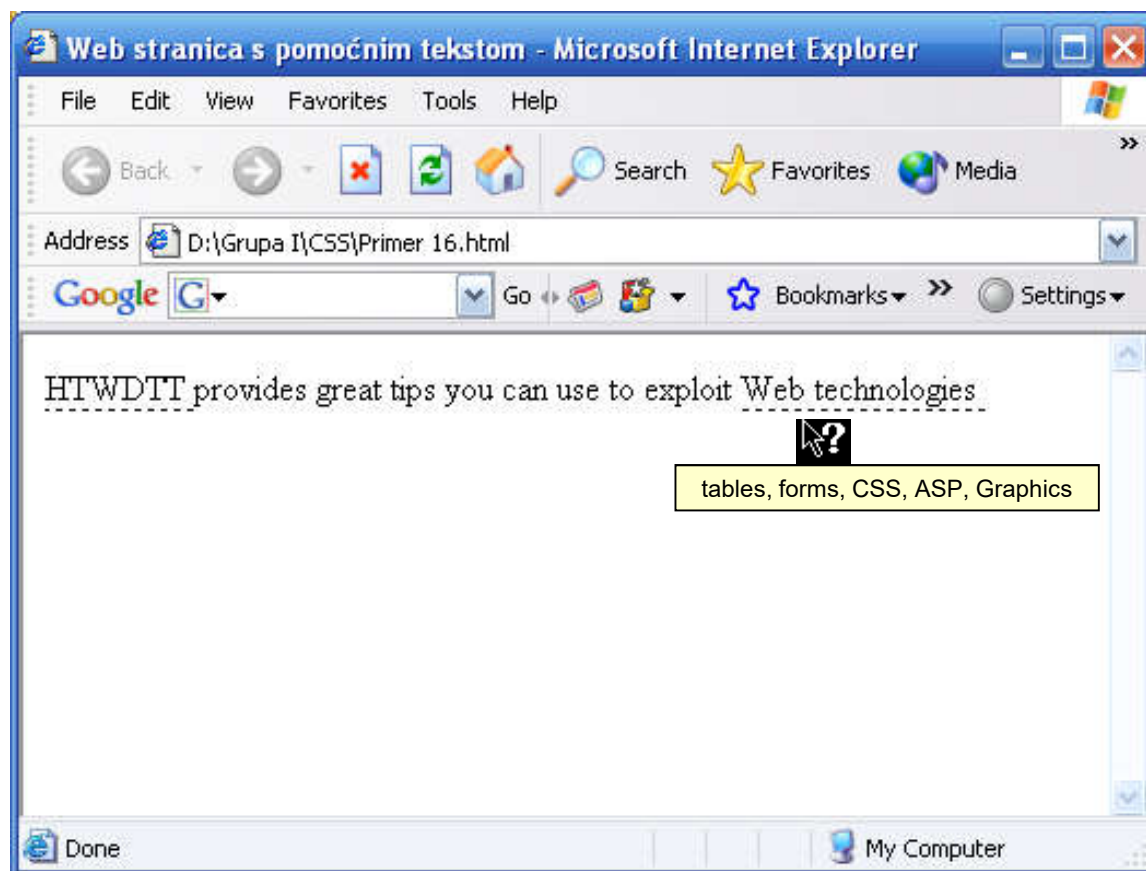
Primer 16.

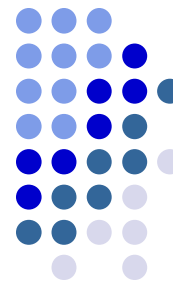
```
<html>
<head>
  <title> Web stranica s pomoćnim tekstom </title>
  <style type="text/css">
    abbr, acronym, .help {border-bottom:1px dotted black;
    cursor:help}
  </style>
</head>
<body>
<acronym title="HTML & Web Design Tips and Techniques"> HTWDTT
  </acronym> provides great tips you can use to exploit
<span class="help" title="tables, forms, CSS, ASP, Graphics">
  Web technologies </span>
</body>
</html>
```

Snimite dokument pod nazivom Primer1.html na D disk (D:\Grupa I\CSS\Primer1.html)



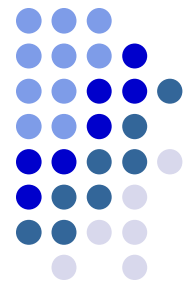
Primer 16.





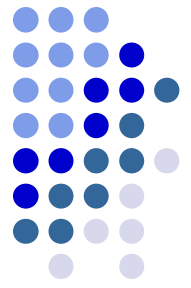
Tipovi kursora u CSS-u

Vrednost kursora	Opis
auto	Prikazani kursor zavisi od konteksta
default	Podrazumevani kursor čitača, obično strelica
hand	Šaka, obično iznad mesta koje može da se pritisne u okviru slike ili hipertekstualne veze
move	Strelica sa četiri vrha, koja naznačava da element može da se pomera
e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize	Kursor za promenu veličine elementa. Obično je u pitanju strelica koja pokazuje u naznačenom smeru, e=istok, strelica na desno, s=jug, strelica nadole, itd
text	Kursor u obliku slova I za obeležavanje teksta ili pronalaženje mesta za umetanje
wait	Kursor za čekanja, obično peščani sat
help	Pomoćni kursor, obično znak pitanja



Oznaka `` - Web homonim

- ❑ `` je osnova savremenog pristupa organizaciji Web stranica, koji se zasniva na kombinaciji jezika HTML verzije 4.01 i kaskadnih opisa stilova (CSS).
- ❑ Oznaka `` može stajati uz bilo koji objekat, jer ga ona samo obeležava za promenu.
- ❑ Ako se bilo koji objekat nađe između oznaka `` i `` ništa se neće promeniti, a neće se pojaviti ni greška.



Oznaka `` - Web homonim

□ Oznaka `` prepoznaje i klase stilova.

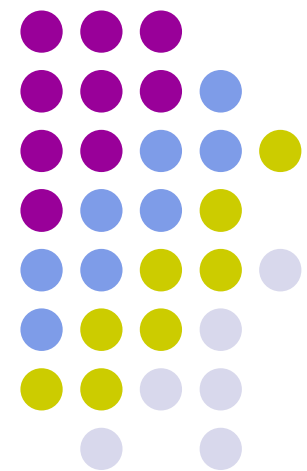
❖ Na primer, može se napraviti klasa:

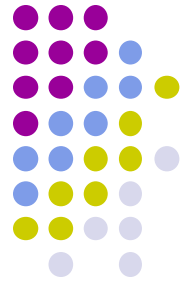
```
<style type="text/css">  
.novitekst {font-size:18pt; color:#ff0000;  
font-family:arial}  
</style>
```

❖ a kasnije se ona može iskoristiti unutar oznaka `` na sledeći način:

```
<span class="novitekst"> Prikaz pomoću klase </span>
```

Dinamičko generisanje HTML-a



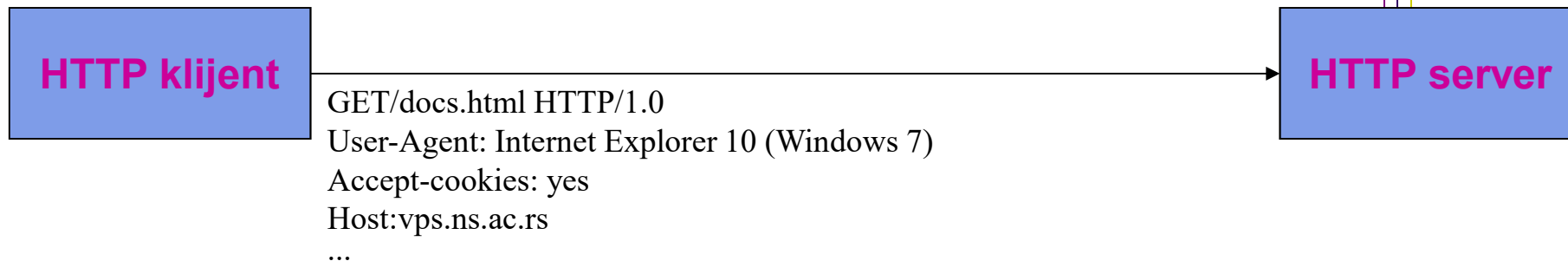


Dinamičko generisanje HTML-a

HTTP protokol

- ❑ Web čitači su namenjeni za prikazivanje Web stranica koje im isporučuju odgovarajući Web serveri.
- ❑ Struktura i izgled samih stranica se opisuje jezikom HTML.
- ❑ Komunikacija između Web klijenta (tj. čitača) i Web servera odvija se po standardnom HTTP (*HyperText Transfer Protocol*) protokolu.

Slanje zahteva HTTP klijenta



Prvi red poruke je najvažniji - on sadrži:

- ❖ komandu koju klijent upućuje serveru (u ovom slučaju to je GET komanda kojom se zahteva određena datoteka sa Web servera),
- ❖ putanju datoteke u okviru Web sajta servera (*/docs.html*) i
- ❖ oznaku verzije protokola (*HTTP/1.0*).

Naredni redovi u poruci - dodatne informacije koje server može da iskoristi za svoje potrebe. U ovom primeru:

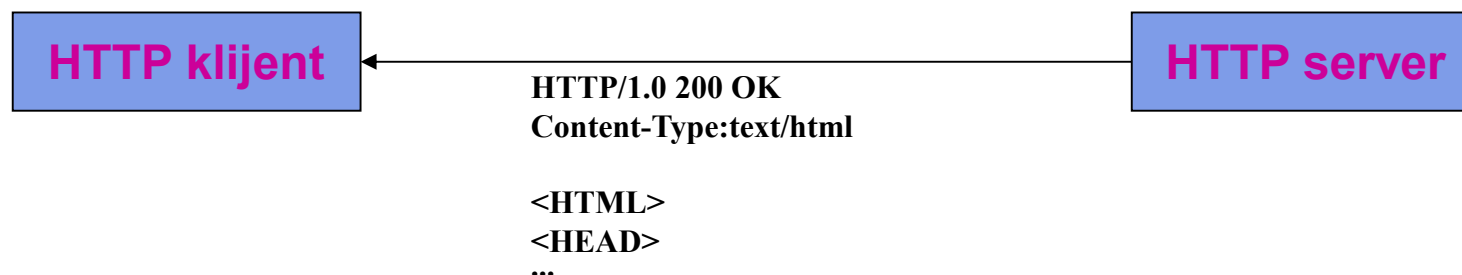
- ❖ polje *User-Agent* predstavlja opis klijentskog softvera (tip i verzija Web čitača i operativnog sistema),
- ❖ polje *Host* predstavlja simboličku adresu klijenta, itd.



Slanje odgovora HTTP klijentu

- ❑ Zadatak servera je da po prijemu zahteva HTTP klijenta odgovori na njega.
- ❑ U ovom slučaju odgovor servera treba da sadrži traženu datoteku (**docs.html**), pri čemu je format odgovora takođe definisan HTTP protokolom.

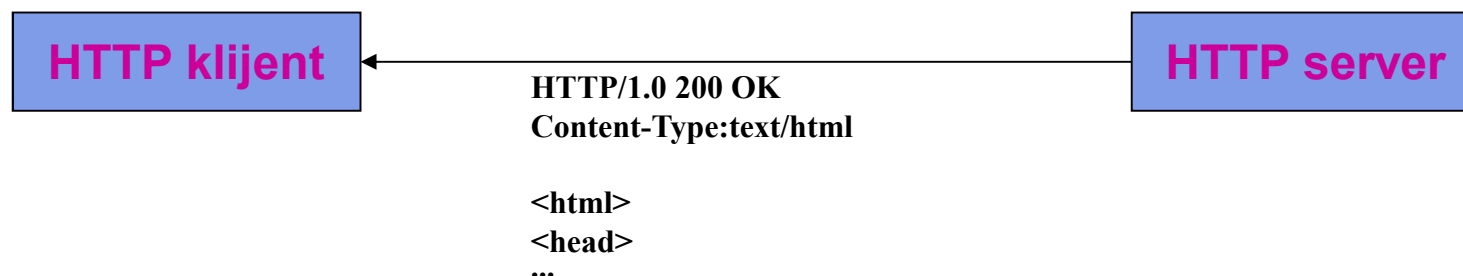
Slanje odgovora HTTP klijentu



Prvi red odgovora sadrži:

- ❖ oznaku protokola, trocifreni broj koji predstavlja status izvršene operacije (200) i tekstualni opis tog statusa (OK). - Konstanta 200 označava da je zahtev uspešno izvršen i da se tražena datoteka nalazi u nastavku poruke.
- ❖ Druge konstante koje se češće sreću su:
 - 💧 404 (tražena datoteka nije pronađena),
 - 💧 407 (pristup datoteci nije dozvoljen), i
 - 💧 302 (datoteka premeštena na drugo mesto).

Slanje odgovora HTTP klijentu



Naredni redovi u poruci sadrže:

- ❖ **Content-Type** - oznaka tipa sadržaja koji se vraća (standardizovane oznake propisane u odgovarajućim **RFC** dokumentima (*Request For Comments*) - zvanična serija dokumenata koja opisuje tehničke i organizacione detalje Interneta počevši od 1969
- ❖ Na primer:
 - 💧 HTML datoteke imaju oznaku **text/html**,
 - 💧 datoteke sa ASCII tekstem bez formatiranja imaju oznaku **text/plain**,
 - 💧 GIF slike **image/gif**, JPEG slike **image/jpeg**, itd.
- ❖ Jedan prazan red koji razdvaja zaglavlje odgovora od samog sadržaja datoteke.



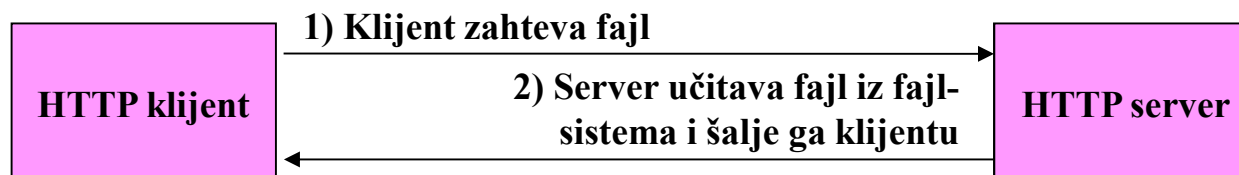
Statički i dinamički Web sadržaji

- ❑ Jedino što klijent može da zatraži od servera je **datoteka**.
- ❑ Na serveru je da tu datoteku pronađe (eventualno i modifikuje!) i pošalje klijentu.
- ❑ Web sadržaji koji se smeštaju na server vidljivi su klijentima kao pojedine datoteke:
 - ❖ datoteke mogu biti unapred pripremljene (npr. u editoru kakav je *Adobe Dreamweaver*) i smeštene u fajl-sistem Web servera.
 - ❖ Mogu biti i generisane “u letu” po prijemu zahteva klijenta na neki poseban način;
- ❑ U tom smislu, Web sadržaje (zapravo, datoteke) možemo podeliti na **statičke** i **dinamičke**.
- ❑ Klijent ne zna da li je datoteka koju je tražio generisana statički ili dinamički.



Statički Web sadržaji

- Statički Web sadržaji su datoteke koje su unapred smeštene u odgovarajući direktorijum fajl-sistema Web servera i spremne su za isporuku klijentima po njihovom zahtevu.

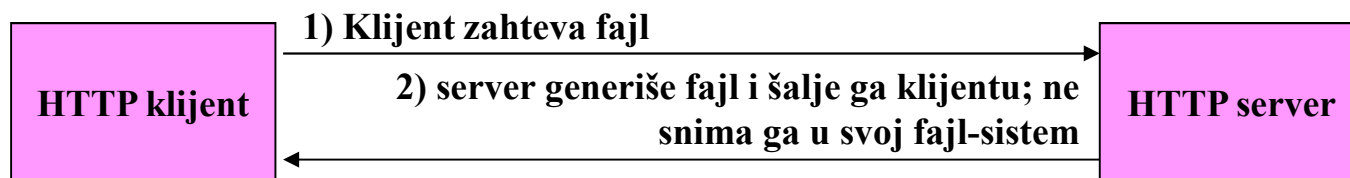


- Isporuka statičkih sadržaja -

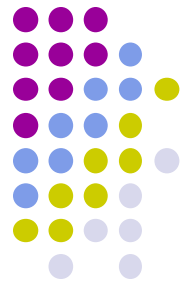


Dinamički Web sadržaji

- ❑ Dinamički sadržaji nisu uskladišteni unapred već se generišu za svaki zahtev klijenta posebno.
- ❑ U ovom slučaju server neće tražiti datoteku u okviru fajl-sistema; server “zna” da je u pitanju dinamički generisana datoteka i poziva odgovarajući potprogram koji će je generisati.
- ❑ Najčešće nema potrebe ovako generisanu datoteku čuvati na serveru



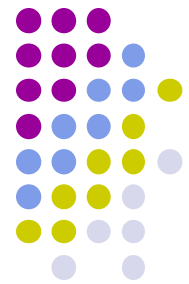
- Isporuka dinamičkih sadržaja -



HTTP komunikacija

Ukupna sekvenca aktivnosti klijenta i servera u HTTP komunikaciji

- 1. klijent otvara konekciju sa serverom**
- 2. klijent šalje zahtev serveru**
- 3. server vraća odgovor**
- 4. zatvara se konekcija**



HTTP komunikacija

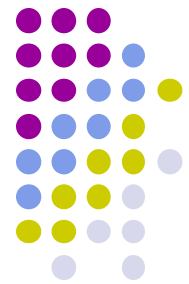
- ❑ Komunikacija između klijenta i servera zasnovana na **zahtev/odgovor** principu.
- ❑ Svaki par zahtev/odgovor smatra se nezavisnim od ostalih.
- ❑ Recimo, u slučaju da prvi klijent pošalje zahtev serveru i dobije odgovor, zatim drugi klijent pošalje zahtev i dobije odgovor, pa potom ponovo prvi klijent pošalje novi zahtev, nema načina da se ustanovi da je prvi klijent poslao dva zahteva (prvi i treći).
- ❑ Server svaki zahtev opslužuje nezavisno od ostalih zahteva.

HTTP je stateless protokol: ne omogućava praćenje stanja korisničke sesije između slanja više različitih zahteva.



Praćenje sesije korisnika

- ❑ HTTP protokol ne omogućava praćenje sesije korisnika.
- ❑ Kako bi se ovaj cilj ipak postigao definisano je pomoćno rešenje - mehanizam slanja “**kolačića**” (*cookies*) između klijenta i servera
- ❑ Uveo ga je *Netscape Navigator*, a kasnije je postao standardan mehanizam za ovu namenu podržan od svih Web čitača i servera.
- ❑ **Šta je jedan *cookie* zapravo?**
- ❑ Možemo ga shvatiti kao string od tipično 20-30 nerazumljivih znakova koji je namenjen za jednoznačno identifikovanje korisnika na serveru.

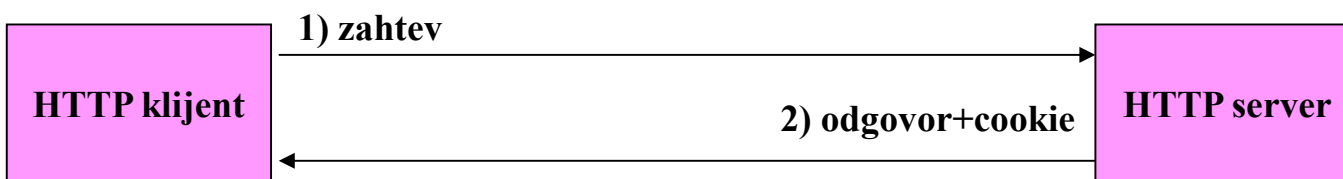


Praćenje sesije korisnika

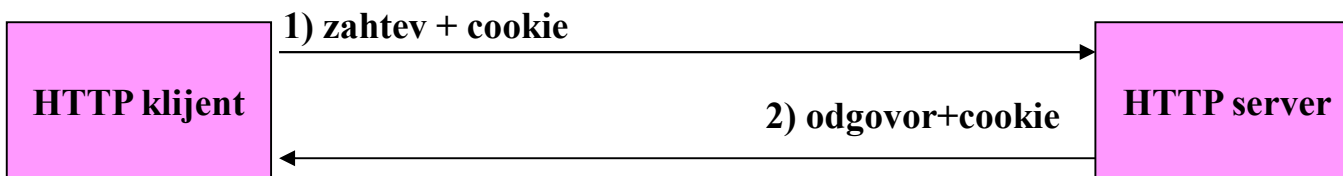
- ❑ Prilikom slanja **prvog zahteva** server će ustanoviti da mu klijent nije poslao *cookie* kao jednu stavku u zaglavlju zahteva.
- ❑ U odgovor na taj zahtev server će dodati *cookie*.
- ❑ Ukoliko je Web čitač podešen tako da radi sa *cookie-ima*, on će u svim sledećim zahtevima koje bude slao tom serveru uključiti i *cookie*, tako da će server moći da prepozna klijenta koga je već ranije opsluživao.



Praćenje sesije korisnika



a) slanje prvog zahteva i prijem odgovora koji uključuje cookie



b) svi sledeći zahtevi sadrže cookie



Svrha kolačića

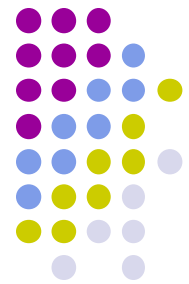
- ❑ **Glavna svrha** kolačića jeste da sačuva informacije:
 - ❖ Oni se uglavnom koriste za čuvanje korisničkog imena i lozinke – korisnik ne mora da ih unosi svaki put kada pristupi lokaciji
 - ❖ Mogu se koristiti za čuvanje svih podešavanja koja je korisnik zadao kada je poslednji put pristupao lokaciji

- ❑ **Sa programerske tačke gledišta:**
 - ❖ Kolačići se koriste da reše probleme vezane za upravljanje stanjem – problem stanja je povezan sa činjenicom da se između dve posete lokaciji server ne seća korisnika



Upotreba kolačića

- ❑ Nekim korisnicima se ne dopada kada neko prati njihovo ponašanje na Webu, pa će oni isključiti kolačiće
- ❑ Da bi se to izbeglo i steklo poverenje korisnika, autori Web lokacije bi trebalo da sačine:
 - ❖ poseban **dokument o privatnosti** ili
 - ❖ **izjavu o upotrebi** i tako obaveste korisnika za šta će kolačići biti upotrebljeni
 - ❖ **samo jedan kolačić** koji će istovremeno i da prati korisnika i da čuva informacije o podešavanjima koja je on zadao – više kolačića zahteva od korisnika niz potvrda, jer svaki mora biti prihvaćen
 - ❖ **alternativno rešenje** za korisnike koji ne žele da prihvate kolačiće



Razvoj dinamičkih veb-stranica

- ❑ Često nazivan “**serverski HTML**” ili “**serverski skriptovi**” – pridruživanje programiranja Web stranama na relativno jednostavan i brz način
- ❑ Cilj serverskih skriptova: izrada posebnih HTML datoteka ili šablona koji sadrže smešu skripta i HTML elemenata, koje će server po potrebi učitati
- ❑ Postoje mnoga okruženja za serverske skriptide:
 - ❖ **PHP –Personal Home Page**
 - ❖ **ASP - Active Server Pages**
 - ❖ **ASP.NET**
 - ❖ **JSP - JavaServer Pages**
- ❑ Razlike su veoma male – glavna razlika između okruženja jeste sintaksa

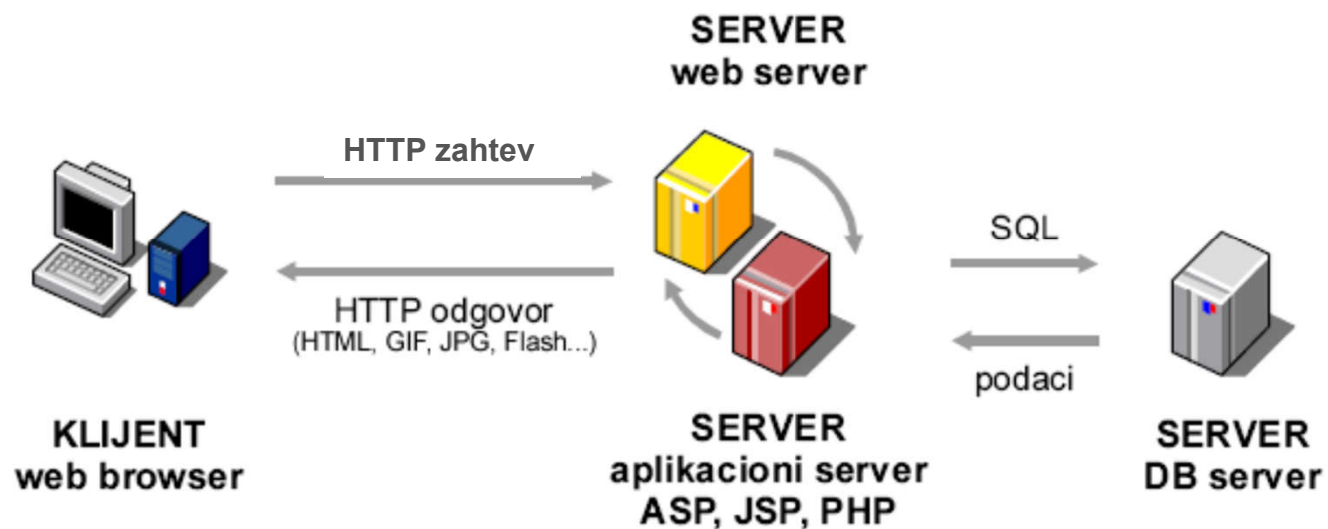
Razvoj dinamičkih veb-stranica

76

- **PHP (Hypertext Preprocessor)** je jedan od veb orijentisanih jezika serverske strane koji omogućava implementaciju poslovne logike u veb-aplikacijama.
 - Veliki broj veb-sajtova i aplikacija je razvijen u PHP programskom jeziku. Istu namenu imaju
- **ActiveServerPages**, Majkrosoftova tehnologija za razvoj dinamičkih veb-stranica, i
- **JavaServerPages** tehnologija za razvoj dinamičkih veb-stranica zasnovana na Java programskom jeziku.



Primer serverske analize HTML dokumenta



Primer serverske analize dokumenta



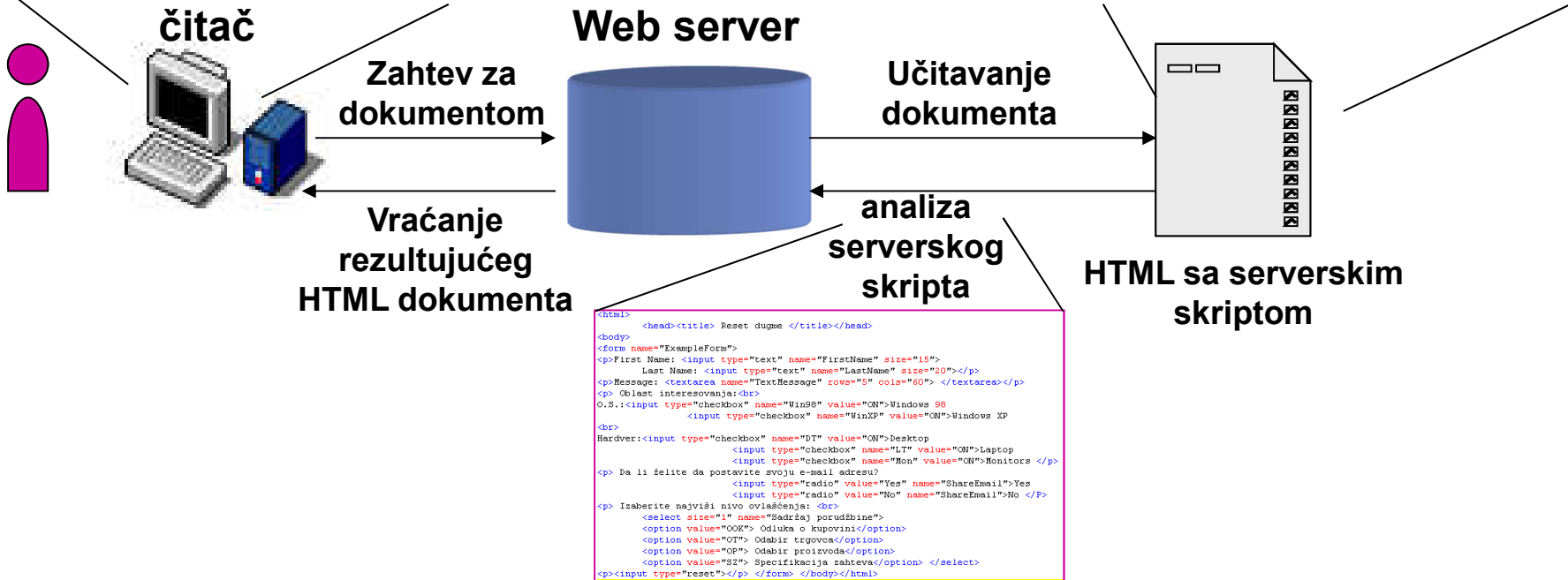
DEMO COMPANY sponsored by PINT

Help

About	Find out company history and information about this sites sponsor, Powell Internet Consulting.
Products	Get more details on all of Demo Company's products, including Games , ESVs , Ballair Packs and Robots .
News	All of the latest Press Releases are in this area for you to view.
Jobs	Looking for work? All of our latest job listings for both Corporate and Production are online.
Intranet	Here you will gain access to Demo Company's sample intranet. (no password required)
Contact	If you want to contact us, simply fill out the form to let us know what you need.
Site Map	The site map will give you an immediate overview of the entire site and allow you to jump to any page that you want to.

[About](#) [Confused?](#) [Products](#) [News](#) [Jobs](#) [Intranet](#)

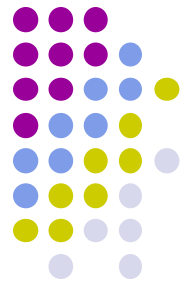
[Contact](#) [Site Map](#) [Help](#)



ASP



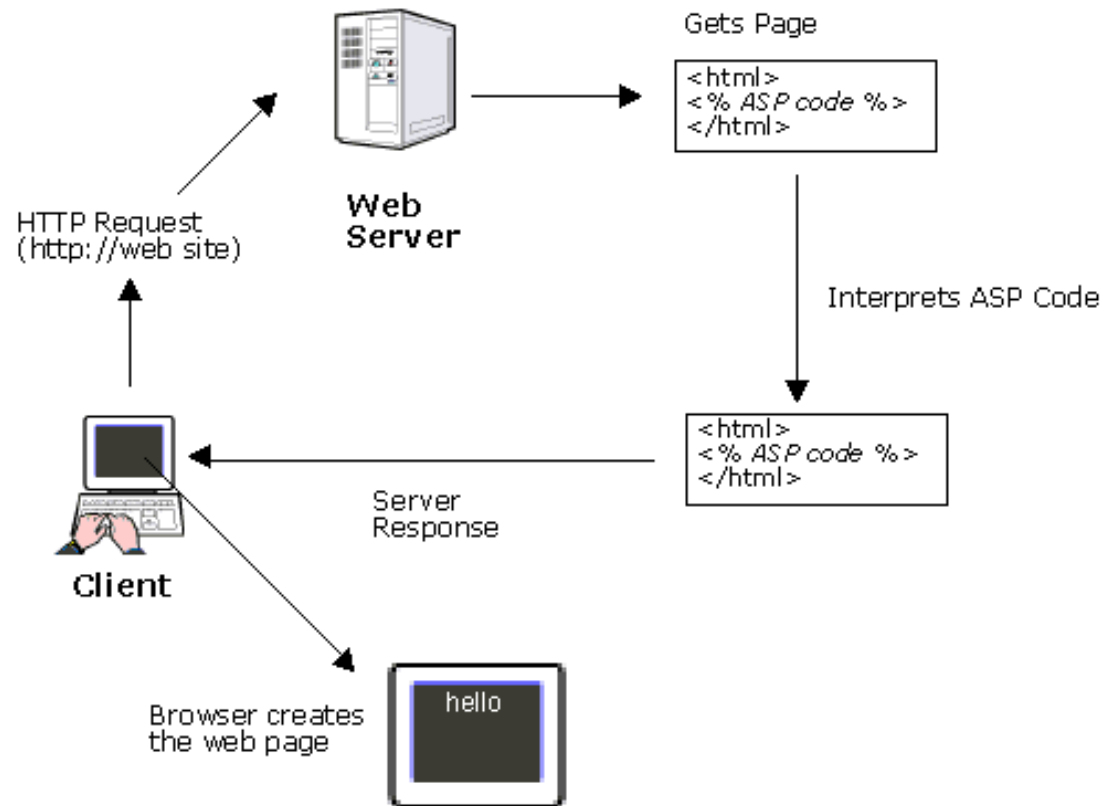
- ❑ ASP ("Stranice aktivne na serveru", eng. Active Server Pages) predstavlja jedan od skriptnih jezika za razvijanje Internet sajtova, razvijen od strane kompanije Microsoft.
- ❑ Njegov glavni takmac na Internet tržištu je PHP, koji ima tu osobinu da je besplatan i slobodan, dok su ASP i svi proizvodi potrebni za njegovo korišćenje komercijalni i u vlasništvu kompanije Microsoft.
- ❑ Sa druge strane, Microsoft konstantno održava i dodaje nove mogućnosti svom paketu, na taj način ga čineći modernim, čvrstim i pružajući svojim klijentima svu moguću tehničku podršku.



ASP

- ❑ Nakon uvođenja .NET tehnologije, Microsoft je i ASP prilagodio istom, tako da je u upotrebi verzija ASP.NET
- ❑ ASP je **skriptni jezik**, kao i PHP, i potreban je interpreter na serverskoj strani da bi isti mogao da radi.
- ❑ ASP nije programski jezik, već tehnologija koja omogućava programiranje Web servera, dok programski jezik zavisi od izbora autora:
 - ❖ **VBScript** – najčešće korišćen
 - ❖ **JScript** – Microsoftova verzija JavaScript jezika
 - ❖ **C#** (samo za ASP.NET)
 - ❖ **Više od 19 jezika** (samo ASP.NET) razvijenih od nezavisnih firmi

ASP



Okviri (frameworks) za razvoj aplikacija

- U razvoju savremenih poslovnih sistema na webu, podrazumeva se sveobuhvatnost, visok nivo kompleksnosti i kvaliteta rešenja, uz minimalne troškove i kratak vremenski rok.
- U praksi se sve više primenjuju okviri (frameworks) za razvoj aplikacija.
- Na ovaj način se olakšava i ubrzava razvoj aplikacija, omogućava modularnost sistema, poboljšava sigurnosti itd.
- Primeri **CSS** okvira su: **Bootstrap i Foundation**.
- Zastupljeni **PHP** okviri su: **Symfony 2, Laravel, CodeIgniter** i drugo.



Sistemi za upravljanje sadržajem

- Aplikacije koje omogućavaju objavljivanje, ažuriranje, brisanje i održavanje veb-sadržaja na webu nazivaju se **sistemi za upravljanje sadržajem (Content management systems, CMS)**.
- Pod sadržajem se u ovom kontekstu podrazumeva bilo koja jedinica digitalne informacije: tekst, slika, animacija, i-mejl poruka, video/audio zapis i slično.
- CMS se obično koriste za upravljanje blogovima, portalima s vestima, intranet portalima, elektronskim prodavnicama i slično.
- Osnovna prednost primene CMS je upravljanje sadržajem veb-sajta, od strane korisnika Interneta, koji ne moraju poznavati veb tehnologije kao što su: HTML, CSS, JavaScript, PHP i drugo.



Sistemi za upravljanje sadržajem

- Tipične **funkcionalnosti** sistema za upravljanje sadržajem (CMS) su:
 - **Kreiranje sadržaja.** CMS obezbeđuje alate za jednostavno kreiranje sadržaja (tzv.WYSIWYG editori – What You See Is What You Get).
 - **Upravljanje sadržajem.** Svaka kreirana stranica se čuva u bazi podataka. Na ovaj način se skladišti sav sadržaj, zajedno s detaljima kao što su verzije, dozvole nad sadržajem, status i slično.
 - **Objavljivanje.** Mehanizmi za objavljivanje omogućavaju da se izgled i raspored strana na sajtu formiraju automatski.
 - **Prezentacija.** Primenom sistema šablona (templejta) sadržaj se prikazuje krajnjim korisnicima. Šabloni omogućavaju da se na lak i brz način promeni izgled sadržaja.



Sistemi za upravljanje sadržajem

- Arhitektura CMS obuhvata dva dela:
 - deo za korisnike (front-end) i
 - deo za administraciju (back-end).



Sistemi za upravljanje sadržajem

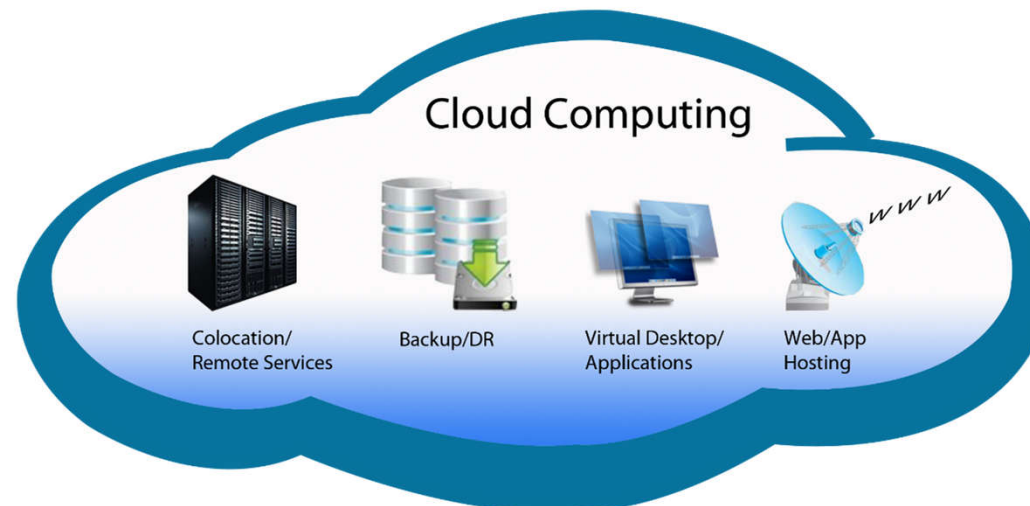
- **WordPress** je besplatna platforma za upravljanje sadržajem i alat za upravljanje blogovima zasnovan na PHP i MySQL tehnologijama.
- WordPress obuhvata veći broj funkcionalnosti koje se odnose na upravljanje veb-sadržajem:
 - upravljanje linkovima,
 - SEO funkcionalnosti,
 - mogućnost dodeljivanja više kategorija jednom članku,
 - podrška za tagovanje elemenata sadržaja u člancima,
 - automatski filteri u pretrazi,
 - standardizovano formatiranje i upravljanje stilovima teksta u okviru članaka.
- Osim WordPress-a, često korišćeni sistemi za upravljanje sadržajem su: Drupal, Joomla!, Liferay, Alfresco i drugi.



- **Cloud computing** je koncept u računarstvu koji omogućava da se računarski resursi isporučuju korisnicima na osnovu njihovog zahteva u vidu usluge, i putem Interneta.
- Cloud computing se zasniva na tehnologiji virtuelizacije i podrazumeva da se servisi i podaci hostuju u deljenom, dinamički skalabilnom skupu resursa provajdera.
- Teorijski, ova tehnologija predstavlja pogodno rešenje za realizaciju infrastrukture poslovnih sistema:
 - uz smanjenje troškova ulaganja u infrastrukturu i
 - postizanja visokog stepena skalabilnosti i dostupnosti.



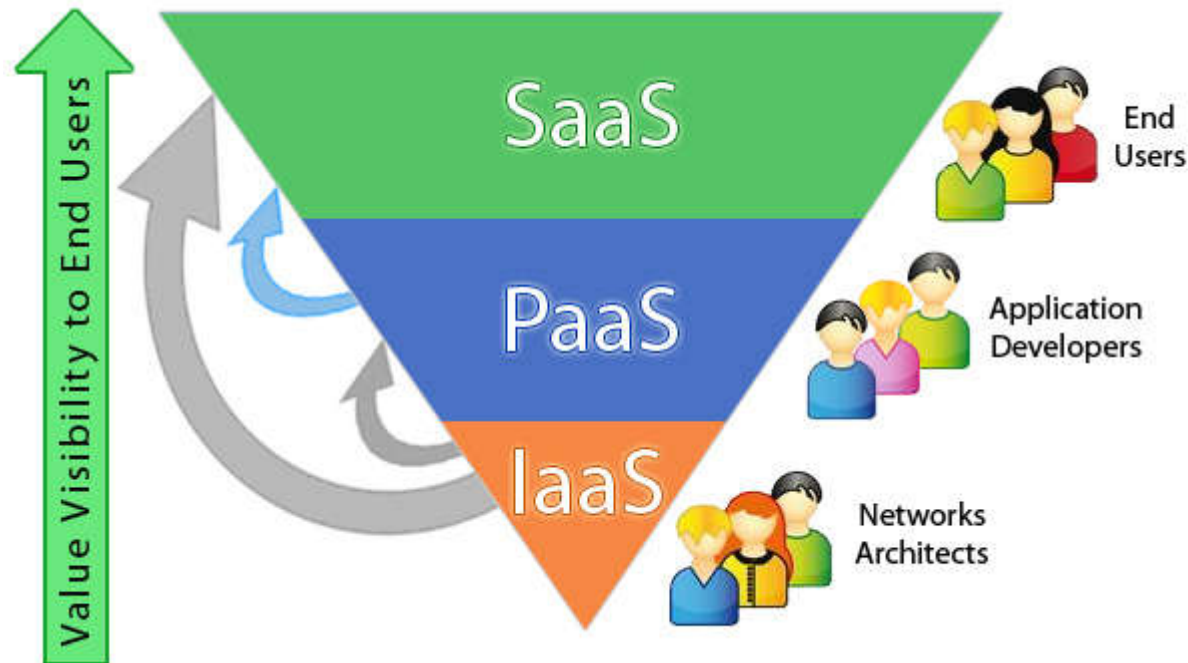
- Uloge koje se mogu uočiti u cloud computing-u su:
 - **Prodavac** je preduzeće koje kao posrednik prodaje cloud usluge krajnjim korisnicima.
 - **Provajder** je preduzeće koje je vlasnik cloud servisa, koje nudi korisnicima.
 - **Korisnik** je pojedinac ili preduzeće koje koristi cloud servise.



Cloud computing - razvojni modeli

89

- Cloud computing omogućava tri osnovna tipa razvojnih modela, u zavisnosti od tipa usluge koja se isporučuje korisniku:
 - infrastruktura kao servis (IaaS),
 - platforma kao servis (PaaS) i
 - softver kao servis (SaaS).



- **Infrastruktura kao servis (IaaS).** Računarski i mrežni resursi, kao što su procesorska snaga, prostor na disku, memorija, mrežni kapacitet, ruteri, fajervol, i sl.) korisniku se isporučuju kao usluga; instalacija i konfiguracija sistema su u nadležnosti korisnika.
- Korisnik može kod provajdera oblaka rezervirati resurse koji su mu potrebni i dinamički menjati te resurse u skladu s potrebama.
- Korisnik provajderu plaća za infrastrukturu koju koristi na sličan način kao što plaća račun za električnu energiju ili za mobilni telefon.
- Iznos računa zavisi od potrošnje i/ili od predefinisiranog tarifnog paketa.



- Međutim, da bi IaaS bio dostupan korisnicima, neophodan je softver koji provajderima cloud-a omogućava administraciju infrastrukture, jednostavno dodeljivanje resursa korisnicima, upravljanje infrastrukturom i merenje performansi.
- Neki od alata za upravljanje cloud-om su:
 - OpenStack,
 - OpenNebula,
 - Eucalyptus,
 - AppLogic i drugi.
- Kao globalni provajderi cloud-a ističu se: Amazon AWS i Rackspace.
- Brojni lokalni provajderi Internet ili telekomunikacionih usluga proširuju svoju ponudu nudeći infrastrukturu kao servis.



- **Platforma kao servis (PaaS).** Platforma omogućava korisniku skup alata i aplikativnih programskih interfejsa (API) koji se mogu koristiti za razvoj aplikacija.
- Korisnik ne može da upravlja hardverskim resursima na kojima se aplikacija izvršava, već samo ima kontrolu nad aplikacijom koju razvija. Primeri platformi su:

- Google AppEngine,
- Microsoft Azure.



- **Softver kao servis (SaaS).** Korisniku se omogućava da koristi gotov softver koji je razvio provajder. Softver se pokreće na cloud computing infrastrukturi, a korisnik mu pristupa najčešće kroz veb-brauzer.
- Brojni su primeri softvera koji se korisnicima nudi kao servis, besplatno ili kao freemium model:
 - elektronska pošta (Gmail, Hotmail, Yahoo, i drugi),
 - čuvanje podataka (Dropbox, Google Drive, Microsoft OneDrive, iCloud i drugi),
- ali i onih koji nude softver za različite poslovne funkcije, kao što su:
 - Aplikacije za kancelarijsko poslovanje (Microsoft Office 365, Google Apps for Business, Zoho Office).
 - ERP rešenja (Netsuite, Epicor).
 - Upravljanje odnosima s klijentima i korisnička podrška (SalesForce).



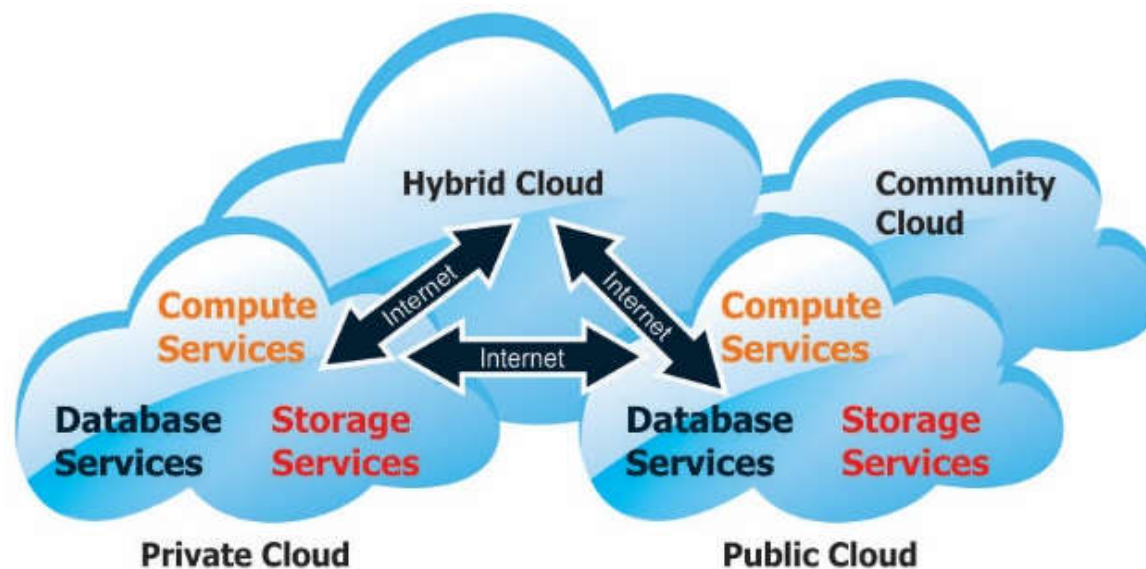
- U zavisnosti od vlasništva, postoje četiri modela razvoja cloud computing infrastrukture: privatni, javni, hibridni i zajednički oblak.
- **Privatni oblak.** Preduzeće koristi koncepte računarstva u oblaku za organizaciju sopstvene računarske infrastrukture.
- Posedovanje sopstvenih računarskih resursa može biti skupo, ali organizaciji daje potpunu kontrolu nad infrastrukturom i podacima koji se na toj infrastrukturi čuvaju.
- Ovakav pristup pogodan je za organizacije kojima je potrebna kompleksna računarska infrastruktura, a koje moraju obezbediti privatnost i zaštitu podataka, kao što su na primer banke, državna uprava itd.
- Ovaj pristup najčešće nije pogodan za mala i srednja preduzeća.



Cloud computing - razvojni modeli

95

- **Javni oblak.** Resursi se dinamički dodeljuju korisnicima, pri čemu korisnik ne zna gde se podaci fizički čuvaju, niti ima potpunu kontrolu nad njihovom zaštitom.
- **Hibridni oblak.** Kombinacija javnog i privatnog oblaka, u smislu da se deo podataka drži u privatnom a deo u javnom oblaku.
- **Zajednički oblak.** Više organizacija deli istu cloud infrastrukturu. Oblakom upravlja jedna od tih organizacija, ili neko treće lice.

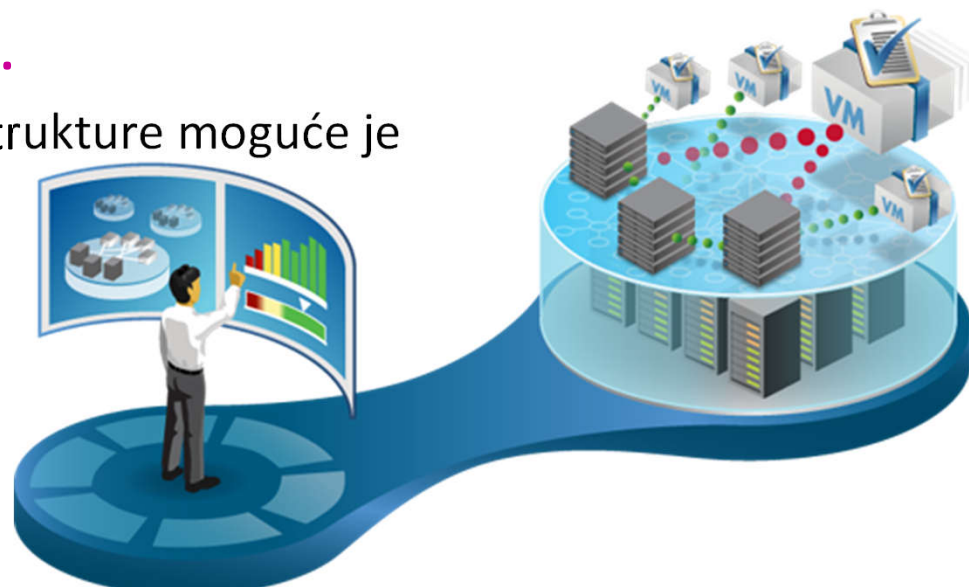


- Osnovna tehnologija računarstva u oblaku je **virtuelizacija**.
- U najširem smislu, **pojam virtuelizacije** se odnosi na logičku podjelu računarskih resursa između više klijenata ili udruživanje više fizičkih resursa u jednu logičku celinu.
- Osnovne prednosti primene tehnologija virtuelizacije su konsolidacija resursa, ušteda energije, jednostavnost održavanja, jednostavnost kreiranja rezervnih kopija sistema i oporavka sistema, nezavisnost od operativnog sistema itd.
- Virtuelizacija zapravo omogućava skalabilnost i ekonomičnost infrastrukture oblaka.



- Postoji više tipova virtuelizacije od kojih svaka ima primenu u okviru cloud computing koncepta:
- **Virtuelizacija servera.** Jedan ili više fizičkih servera se organizuju kao niz virtuelnih servera nezavisnih od fizičke infrastrukture.
- **Virtuelizacija aplikacija.** Aplikacije se izvršavaju bez potrebe za instalacijom na računaru s kog se pokreću. Omogućeno je pokretanje aplikacije nezavisno od operativnog sistema.
- **Virtuelizacija desktopa.** Desktop se razdvaja od fizičkog korisničkog računara i prikazuje korisniku na primer kroz veb-brauzer.
- **Virtuelizacija mrežne infrastrukture.**

U okviru jedne fizičke mrežne infrastrukture moguće je kreirati veći broj logički razdvojenih mreža.



Prednosti i nedostaci cloud computinga

- Osnovni **nedostaci** cloud computing pristupa odnose se na:
 - još uvek neadekvatnu pravnu regulativu i nejasnoće u nadležnostima. Na primer, preduzeća iz jedne države mogu koristiti usluge provajdera oblaka iz druge države, a podaci se mogu fizički čuvati u računarskom centru lociranom u trećoj državi.

- Bez usklađenosti u pravnoj regulativi svih država učesnica, vlasništvo nad podacima može biti nejasno.

- Takođe, postoji nedostatak jasnoće u pogledu licenci za softver, problemi u upravljanju digitalnim identitetima i pristupom, kao i problemi u garanciji dostupnosti servisa od strane provajdera.

