

H T M L

Hypertext Markup Language

HTML

- HTML је структурирани описни језик који се користи за израду веб страна.
- Представља поједностављени подскуп SGML-а (Standard Generalized Markup Language).
- Омогућава формирање изгледа целе стране, њених саставних делова као и мрежно структурираног документа.
- HTML елементи указују WWW читачу шта треба да предузме и како треба да прикаже текст.
- HTML документ је plain-text (ASCII) фајл који креирамо помоћу било којег текст едитора. (нпр. Notepad на Windows рачунарима).
- WYSIWYG HTML Едитори (Dreamweaver, Home Page, PageMill).
- WYSIWYG "what you see is what you get";

WYSIWYG

The screenshot displays a web development environment with a WYSIWYG editor on the left and a code inspector on the right. The WYSIWYG editor shows a webpage titled "Using Cascading Style Sheets" with a navigation menu, a main content area with text and images, and a footer. The code inspector shows the corresponding HTML and CSS code for the selected element. A red circle highlights the "Code Inspector" window title, and a red underline highlights the page title in the WYSIWYG editor.

Code Inspector Content:

```
<body BGCOLOR="#FFFFFF" LEFTMARGIN="0" TOPMARGIN="0" MARGINWIDTH="0" MARGINHEIGHT="0">

<table BORDER="0" CELLSPACING="0" CELLPADDING="0" WIDTH="100%">

  <tr BGCOLOR="#000000">

    <td>

      <p><font color="#FFFFFF" size="2"><b><font face="Arial, Helvetica, sans-serif">Using Cascading Style Sheets<br>

      </font></b></font></p>

      <p>&nbsp;</p>

    </td>

  </tr>

</table>

<table BORDER="0" CELLPADDING="0" CELLSPACING="0" HEIGHT="90%">

  <tr>

    <td ROWSPAN="2" VALIGN="top" BGCOLOR="#333333">

    <td>

      <p>Compass works hard to bring you the best packages for you and your family for your rugged lifestyle. Here you'll find packages suited for the adventurous traveler.

      <p>Be ready to go on a moment's notice when the surf is pumping and fly on the mornings of hurricane inspired seas.

    </td>

  </tr>

  <tr>

    <td>

      <p>Shopping Cart  
Customer Service  
Your Account

    </td>

  </tr>

</table>

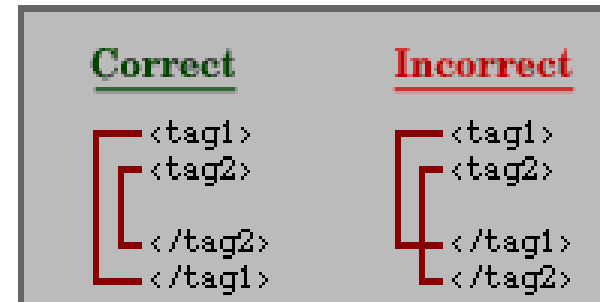
<br>
```

Правила језика HTML

- HTML документи су структурирани
- Језик HTML не прави разлику између малих и великих слова
- HTML препознаје само један размак
- HTML елементи треба да су затворени, осим ако су празни
- Атрибуте језика HTML треба смештати у знакове навода
- Читачи игноришу непознате атрибуте и елементе језика HTML
- Тестирање – CSE Validatore

HTML ELEMENTI

- Описни језици попут HTML састоје се од кодова, који се називају елементи (тагови), и који се користе за описивање структуре и изгледа документа.
- Читач веба интерпретира HTML елементе и на основу њих приказује документ на екрану.
- HTML тагови служе за означавање различитих елемената у HTML документу.
- HTML таг се означава са знаком "мање од" (<), имена тага, и знаком "веће од" (>). Тагови су обично упарени.
- <tag> ... </tag>
- Неки тагови могу имати атрибуте.
- <title> <TITLE> <Title>.



Tag	Опис
<html> ... </html>	Декларише да је Web страница писана у HTML-у
<head> ... </head>	Заглавље странице
<title> ... </title>	Дефинише наслов странице
<body> ... </body>	Ограничава тело странице
<h n> ... </h n>	Ограничава наслов нивоа <i>n</i>
 ... 	Задебљана слова (bold)
<i> ... </i>	Искошена слова <i>italik</i>
<center> ... </center>	Хоризонтално центрирање ... на страници
 ... 	Ограничава неуређену листу
 ... 	Ограничава уређену листу
<il> ... </il>	Ограничава ставку уређене листе
 	Умеће нови ред (прелом линије)
<p>	Започиње нови параграф
<hr>	Умеће хоризонталну линију по целој дужини стране
	Приказује слику
 ... 	Дефинише hiperlink

Тело документа

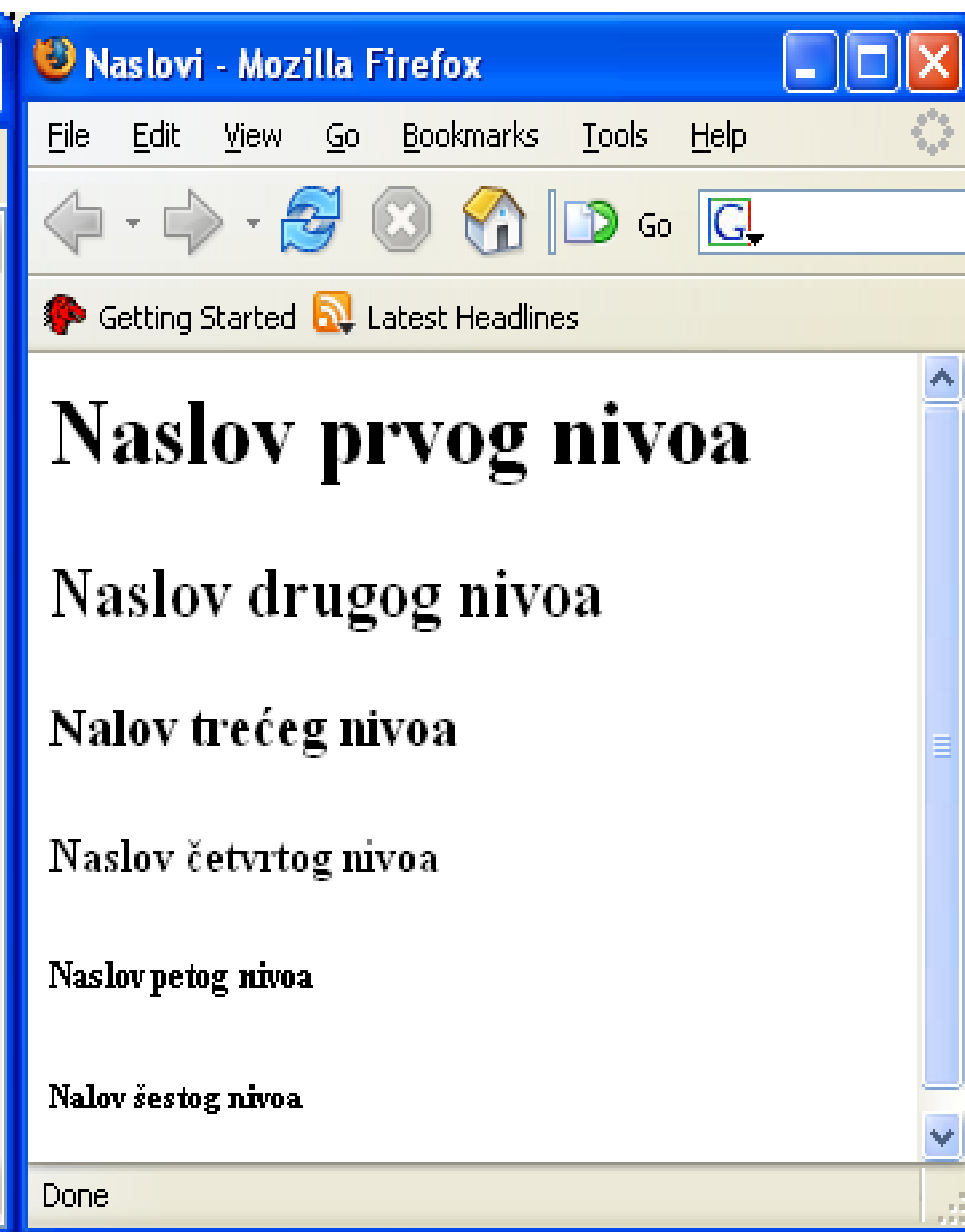
- Тело документа садржи сав текст и путање до слика које чине страну, заједно са свим елементима који контролишу изглед стране.
- ➡ Тело документа почиње маркером `<BODY>` и завршава се маркером `</BODY>`.
- ➡ Елемент `BODY` је опциони.

Форматирање блокова текста

- Елементи за форматирање блокова текста се користе за форматирање читавих пасуса текста у документу.
- Користе се у телу документа.
- За форматирање наслова користе се елементи **H1**, **H2**, **H3**, **H4**, **H5** и **H6**.
- Користе се за одвајање страна у мање целине.

- Сви елементи имају почетне и завршне маркере.
- Званично правило о насловима је следеће: што је наслов већег нивоа (што је мањи број), то ће оно бити истакнутије приказано.
- Већина читача их приказује на исти начин: фонтом Times New Roman, појачано, са 24, 18, 14, 12, 10 и 8 пиксела, респективно.
- Могу се користити стилови за форматирање наслова одређеним фонтом, величином или бојом.

```
primer2 - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Naslovi</title>
</head>
<body>
<h1>Naslov prvog nivoa</h1>
<h2>Naslov drugog nivoa</h2>
<h3>Nalov trećeg nivoa</h3>
<h4>Naslov četvrtog nivoa</h4>
<h5>Naslov petog nivoa</h5>
<h6>Nalov šestog nivoa</h6>
</body>
</html>
```



- Читачи аутоматски преламају текст према ширини блока или прозора, правећи нове редове ако је потребно.
- За креирање новог параграфа користи се елемент **P**.
- Елемент има почетни и завршни маркер, али се завршни маркер може изоставити.
- Параграф је окружен вертикалним простором од једног или пола реда.
- Могу се користити стилови за формирање пасуса одређеним фонтом, величином или бојом.

primer3 - Notepad

```
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Paragrafi</title>
</head>
<body>
<p>Tekst prvog paragrafa</p>
<p>Tekst drugog paragrafa
<p>Tekst trećeg paragrafa</p>
</body>
</html>
```

Paragrafi - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

← → ↻ × 🏠 | Go

📡 Getting Started 📡 Latest Headlines

Tekst prvog paragrafa

Tekst drugog paragrafa

Tekst trećeg paragrafa

Done

- Ручни прекид реда остварује се елементом **BR**.
- Елемент има само почетни маркер и не садржи друге елементе.
- Ручни прекид реда се користи за кратке редове текста који треба да се појаве један после другог без много простора између.
- Могу се користити вишеструке ознаке **BR** да би се направио додатни простор између редова или параграфа.

primer4 - Notepad

File Edit Format View Help

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Ručni prekid reda</title>
</head>
<body>
<p>Tekst paragrafa <br> Ručni prekid
reda
</body>
</html>
```

Ručni prekid reda - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

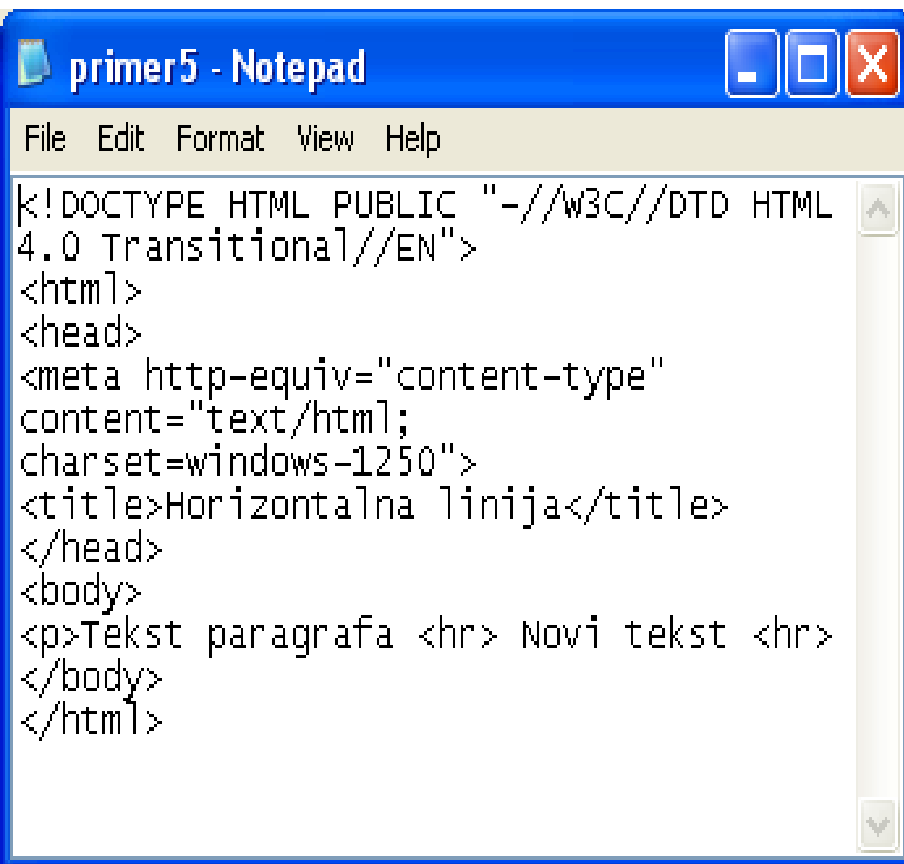
← → ↻ ⓧ 🏠 | 📄 Go

🔴 Getting Started 📡 Latest Headlines

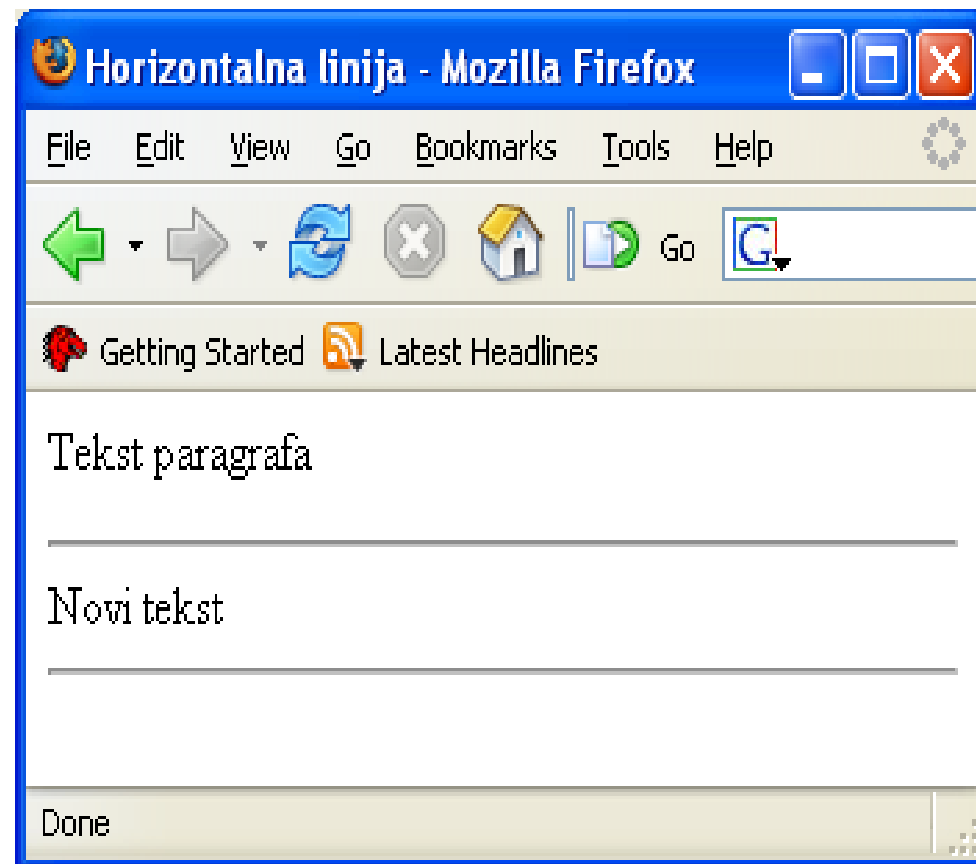
Tekst paragrafa
Ručni prekid reda

Done

- За приказивање хоризонталних линија користи се елемент **HR**.
- Елемент има само почетни маркер и не садржи друге елементе.

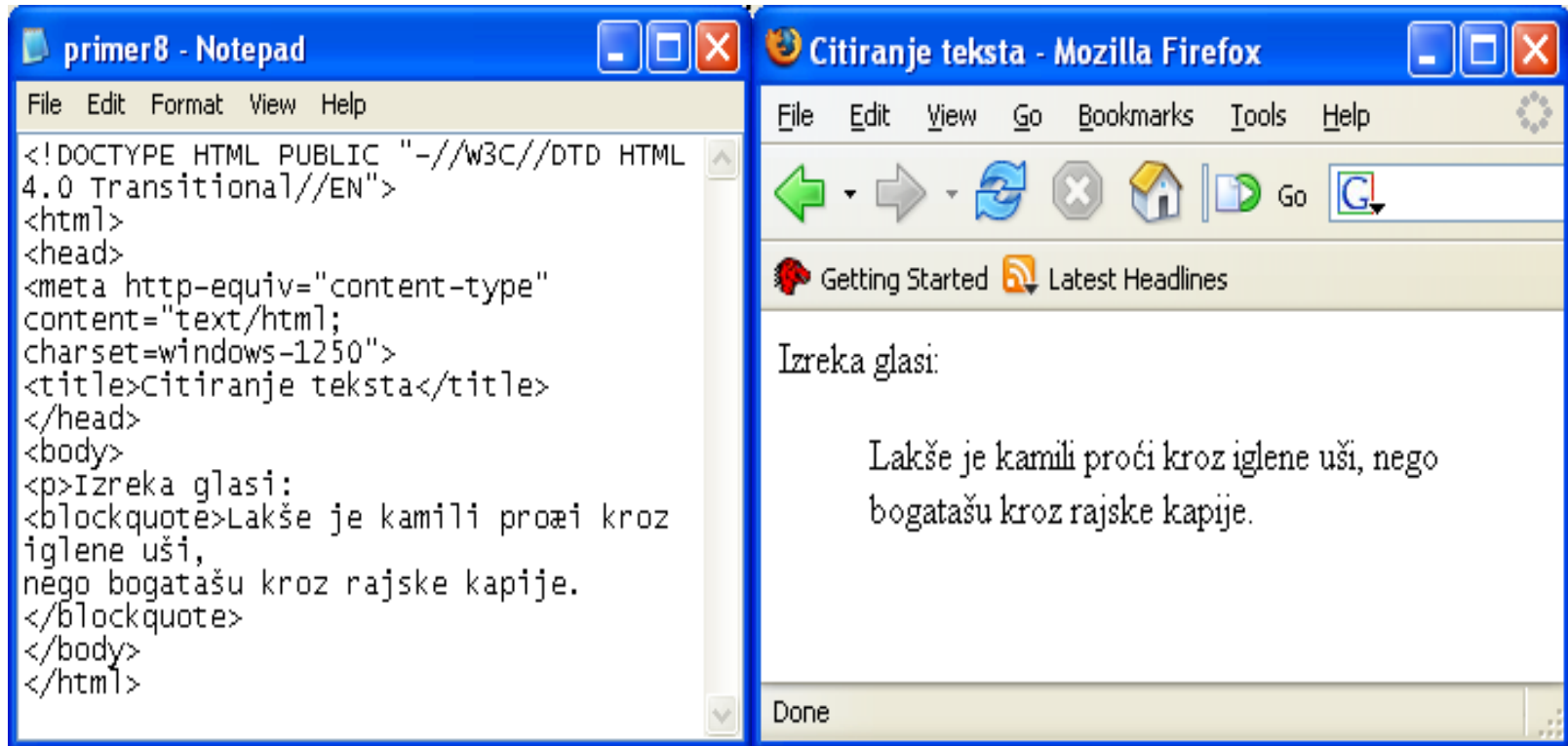


```
primer5 - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Horizontalna linija</title>
</head>
<body>
<p>Tekst paragrafa <hr> Novi tekst <hr>
</body>
</html>
```

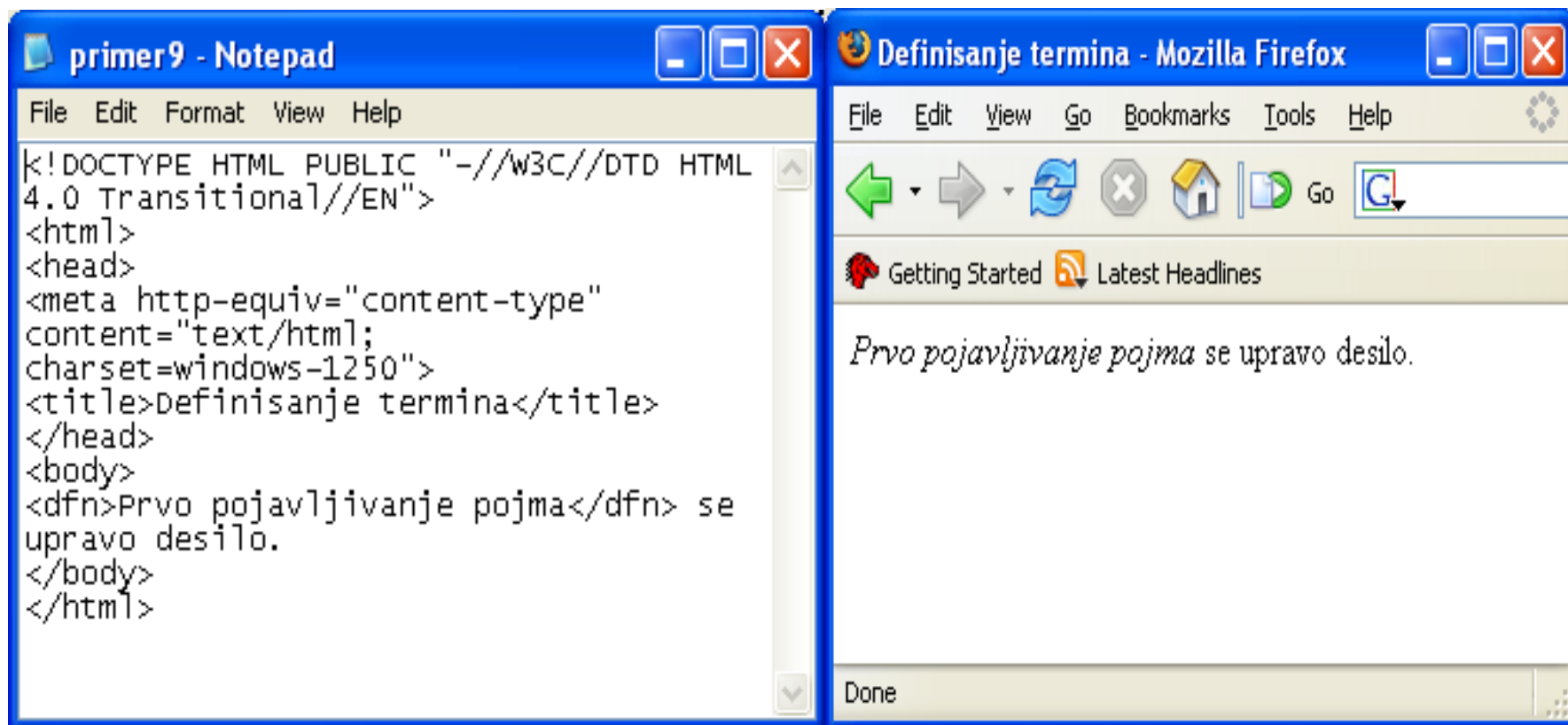


- Преформатирани текст омогућава да се одрже оригинални прекиди и размаци.
- То је идеално за кôд, песме, табеле итд.
- За приказивање преформатираног текста ко-ристи се елемент **PRE**.
- Елемент има почетни и завршни маркер.
- Преформатирани текст се обично приказује са фонтом са једнаким размаком као што је Courier.

- За цитирање текста на нивоу блока користи се елемент **BLOCKQUOTE**.
- Елемент има почетни и завршни маркер.
- Уобичајено је да текст буде померен с лева и десна, формира нови параграф и додаје простор изнад и испод цитата.



- За дефинисање термина користи се елемент **DFN**.
- Елемент има почетни и завршни маркер.
- Текст се приказује у курзиви.



Форматирање речи

- Поједине речи или делови текста могу се приказати различитим стиловима.
- Силови могу бити:
 - логички
 - физички.
- Код логичких стилова начин приказивања је одређен читачем.
- Код физичких стилова начин приказивања одређује сам корисник.

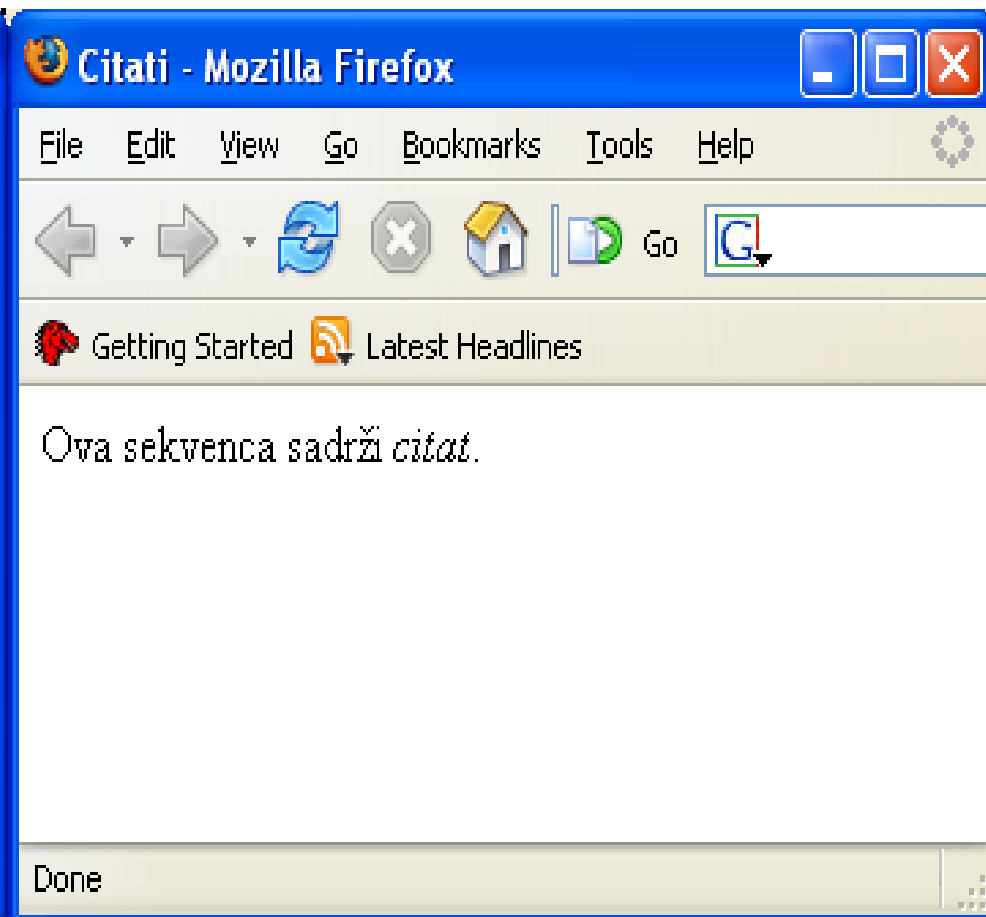
Логички стилови су:

- CITE
- CODE
- EM
- KBD
- SAMP
- STRONG
- VAR

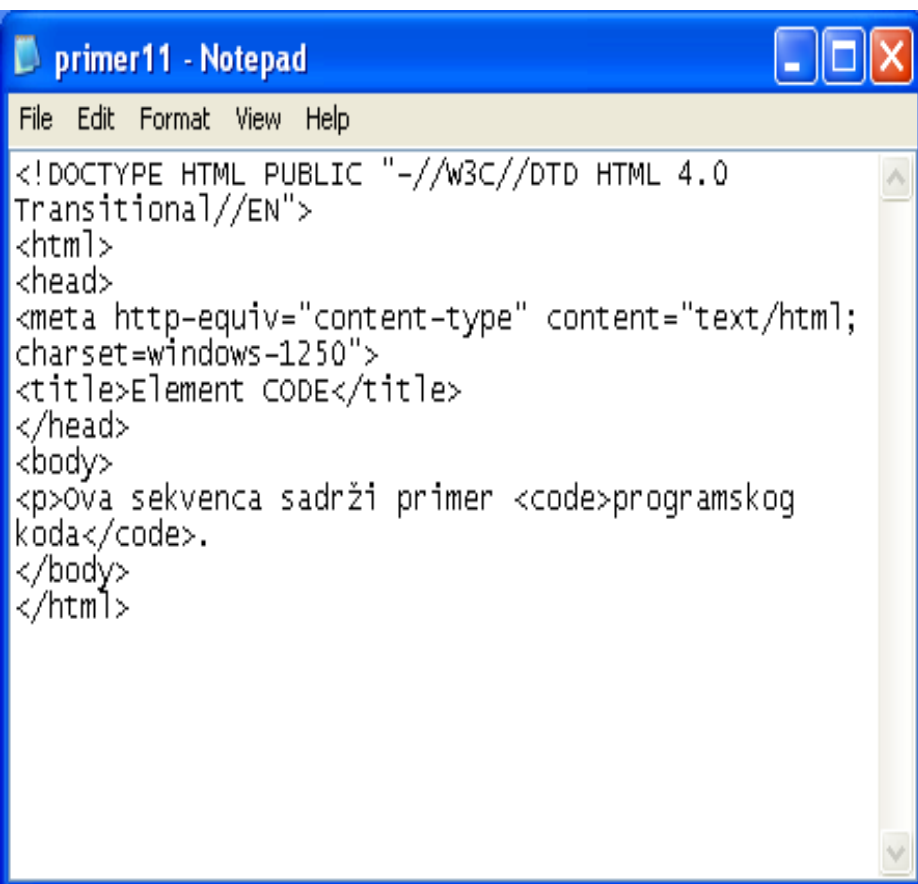
•Сви елементи логичких стилова имају почетне и завршне маркере.

- Елемент **CITE** се користи за приказивање цитата или референци. Текст је у курзиву.

```
primer10 - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Citati</title>
</head>
<body>
<p>Ova sekvenca sadrži
<cite>citat</cite>.
</body>
</html>
```

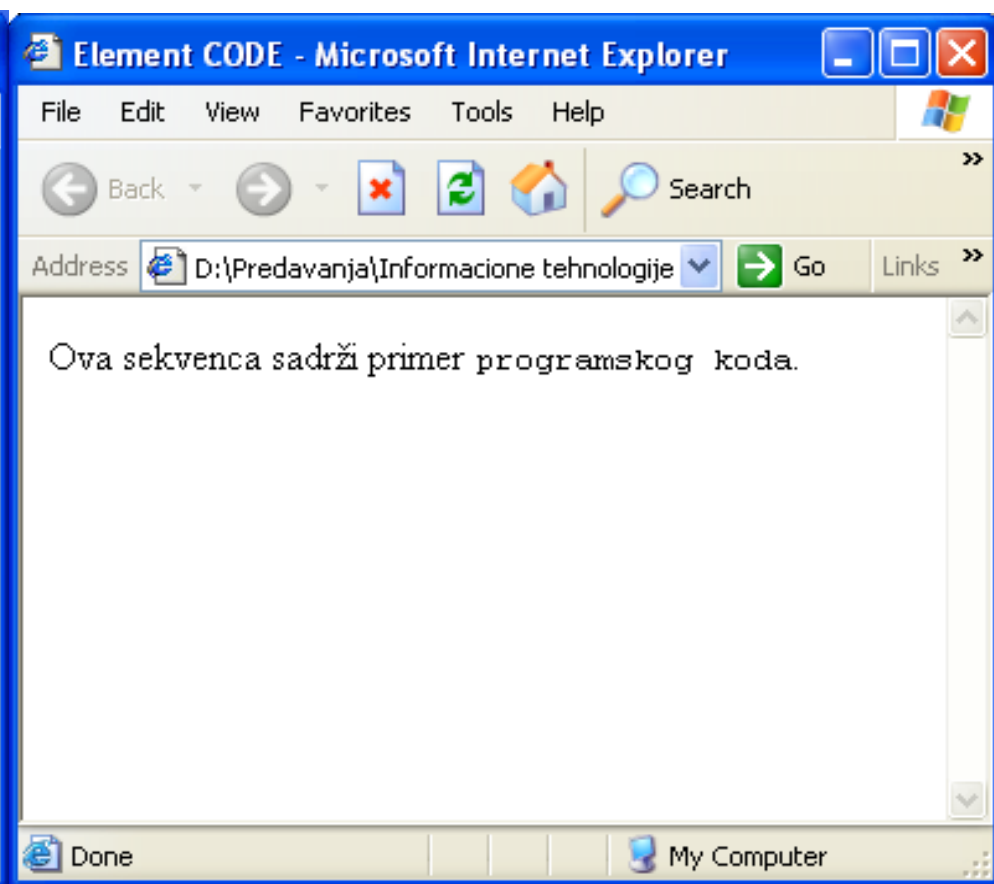


- Елемент **CODE** се користи за приказивање програмског кода. Текст је приказан непропорционалним фонтом.

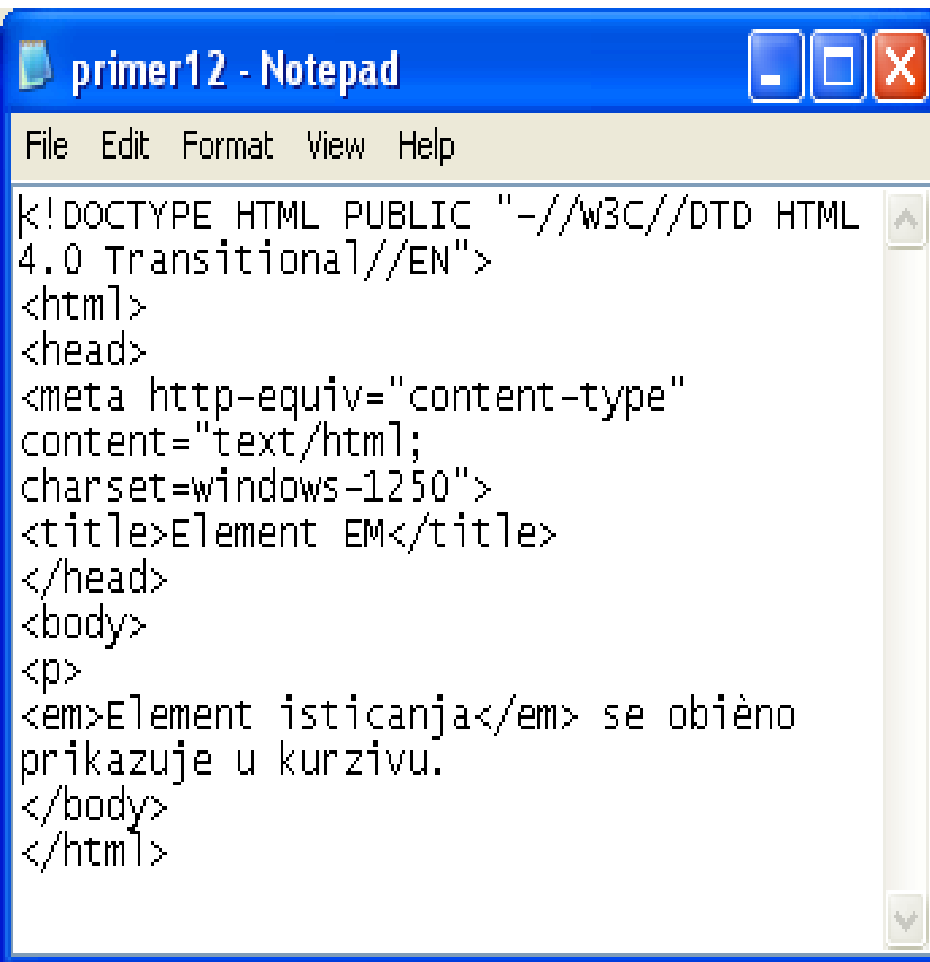


primer11 - Notepad

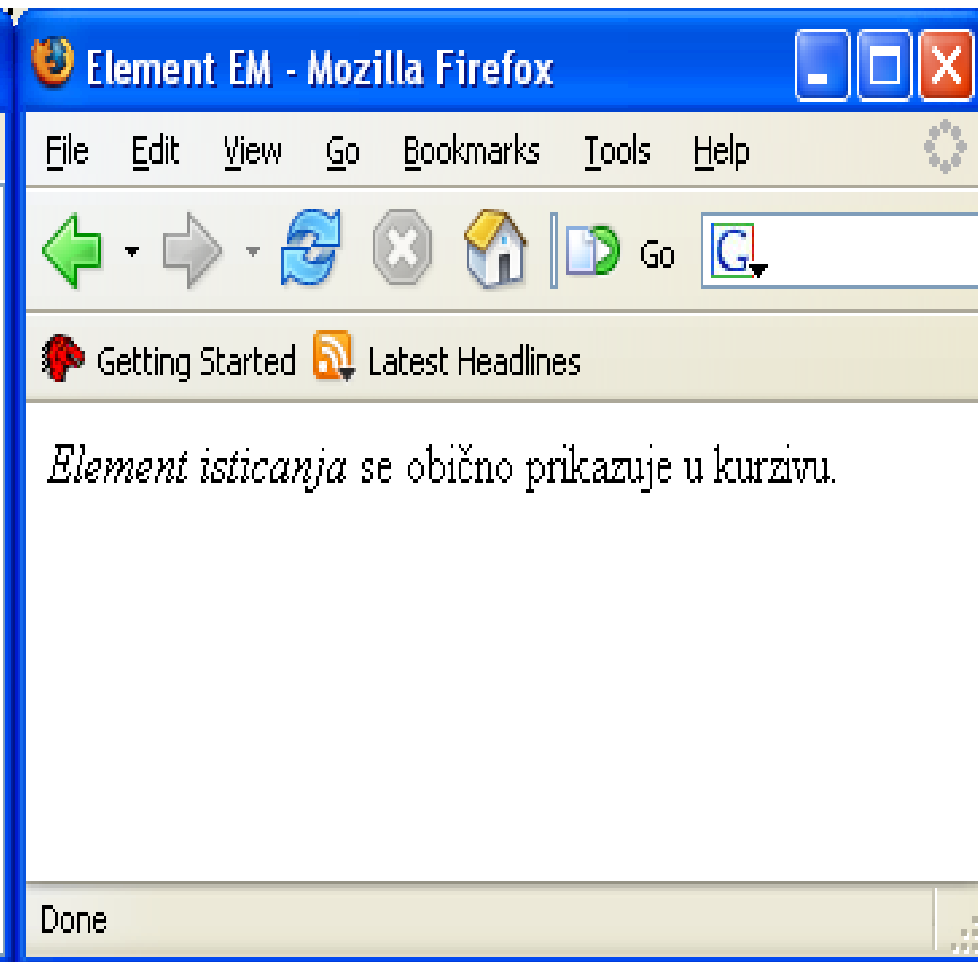
```
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=windows-1250">
<title>Element CODE</title>
</head>
<body>
<p>Ova sekvenca sadrži primer <code>programskog
koda</code>.
</body>
</html>
```



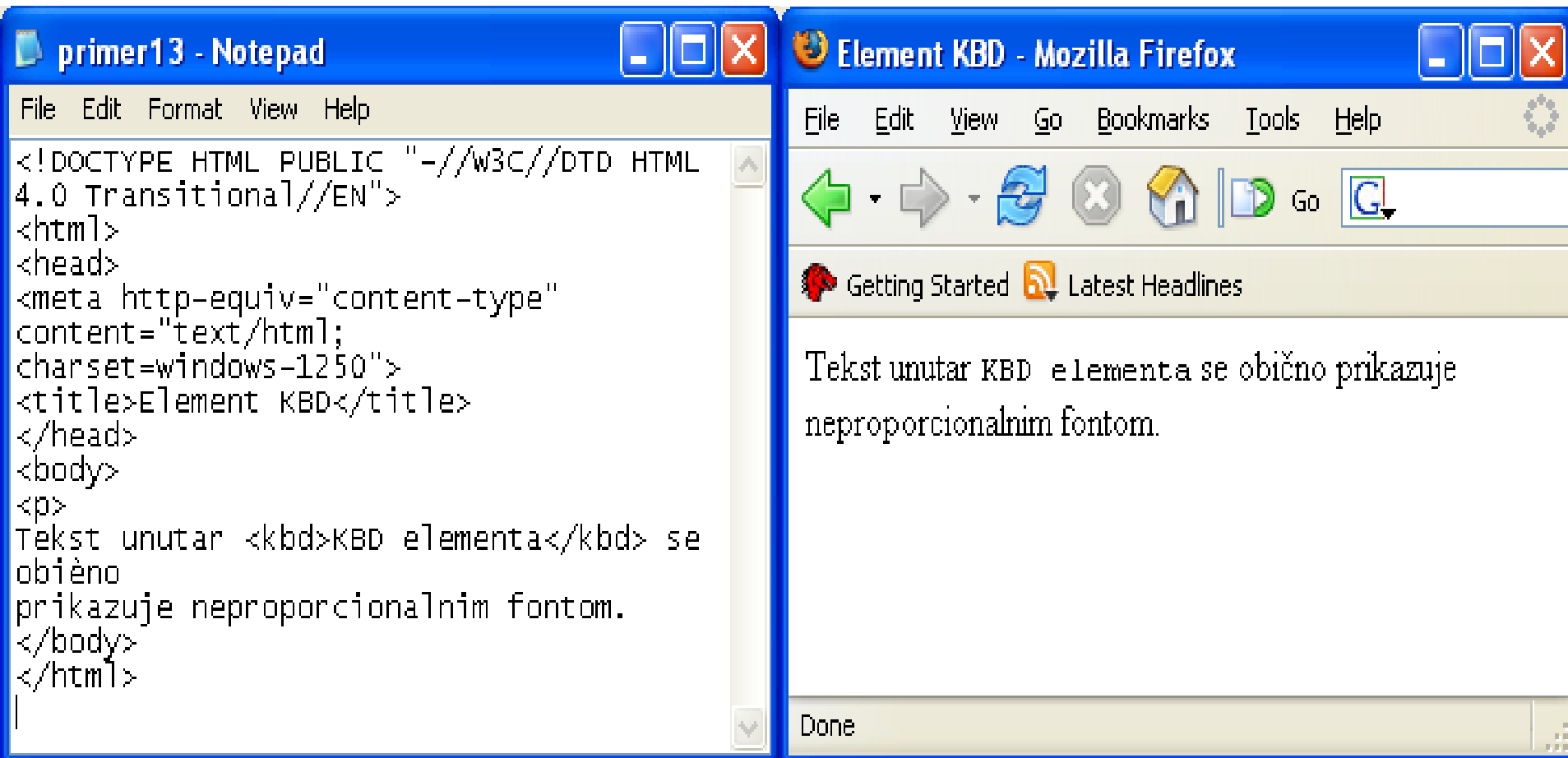
- Елемент **EM** се користи за истицање текста. Текст је у курзиву.



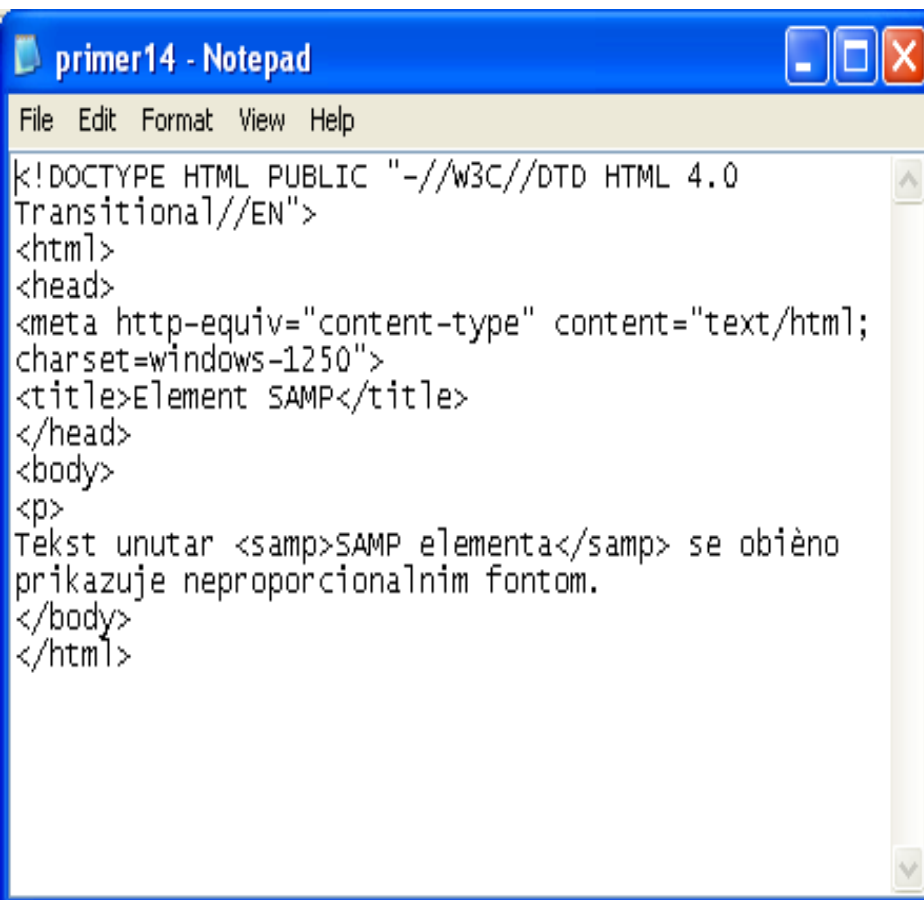
```
primer12 - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Element EM</title>
</head>
<body>
<p>
<em>Element isticanja</em> se obièno
prikazuje u kurzivu.
</body>
</html>
```



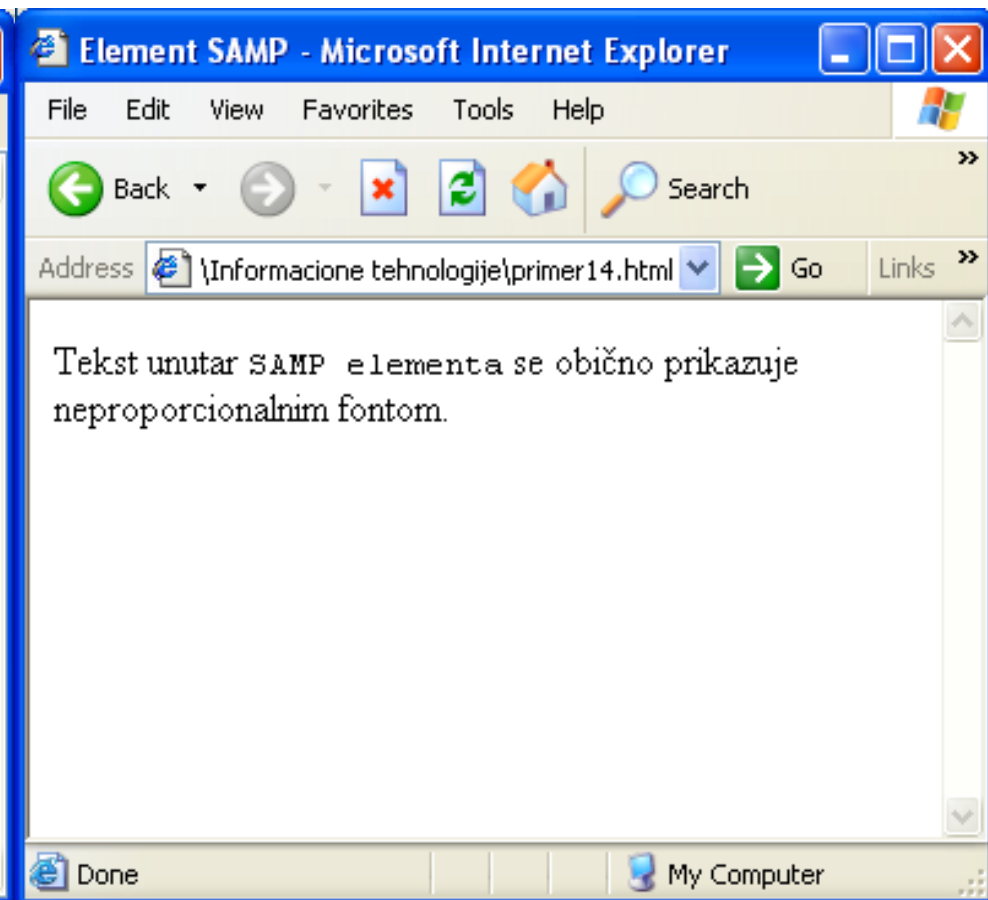
- Елемент **KBD** се користи за истицање текста. Текст се приказује непропорционалним фонтом.



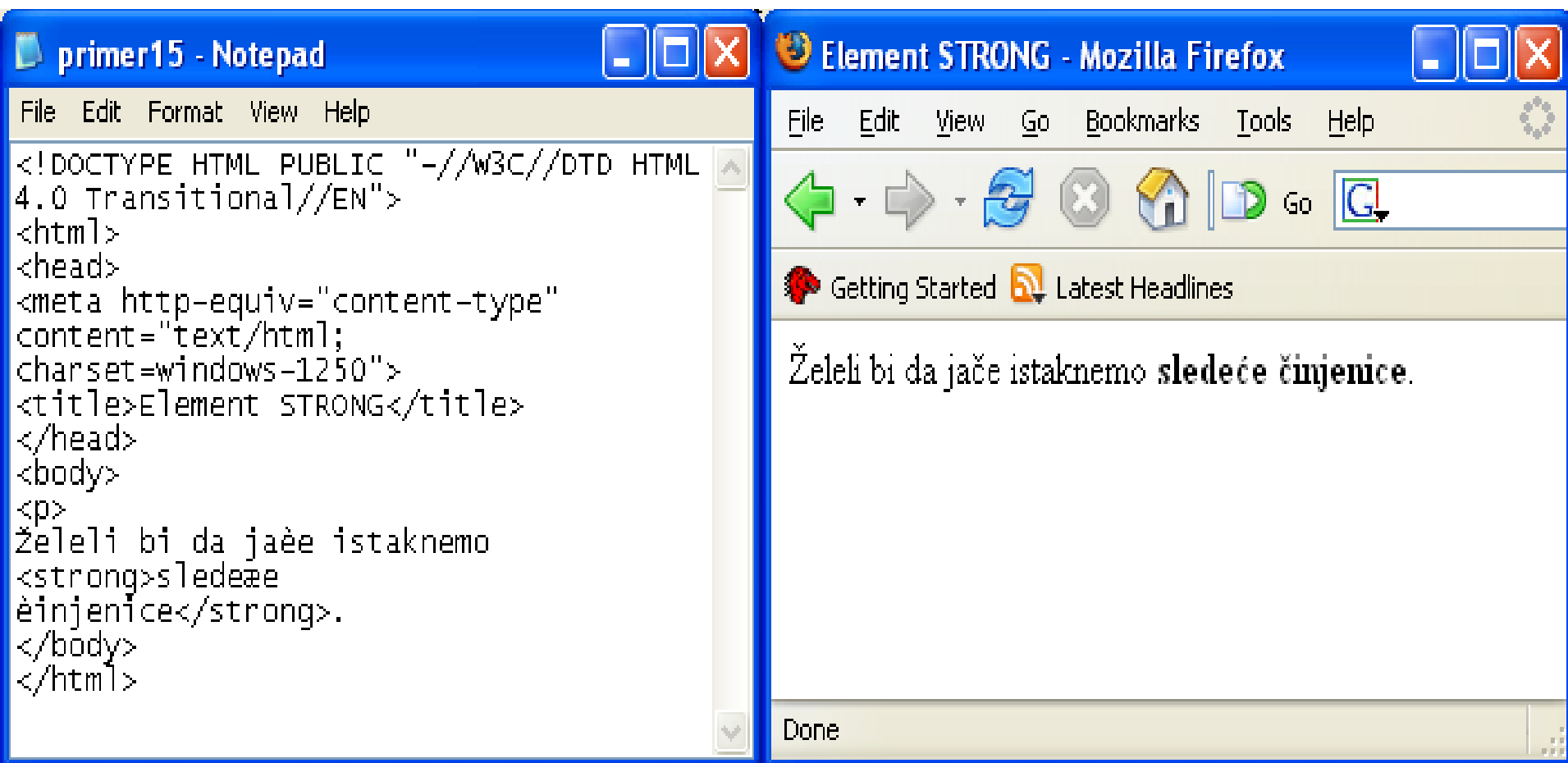
- Елемент **SAMP** се користи за приказивање примера. Текст је приказан непропорционалним фонтом.



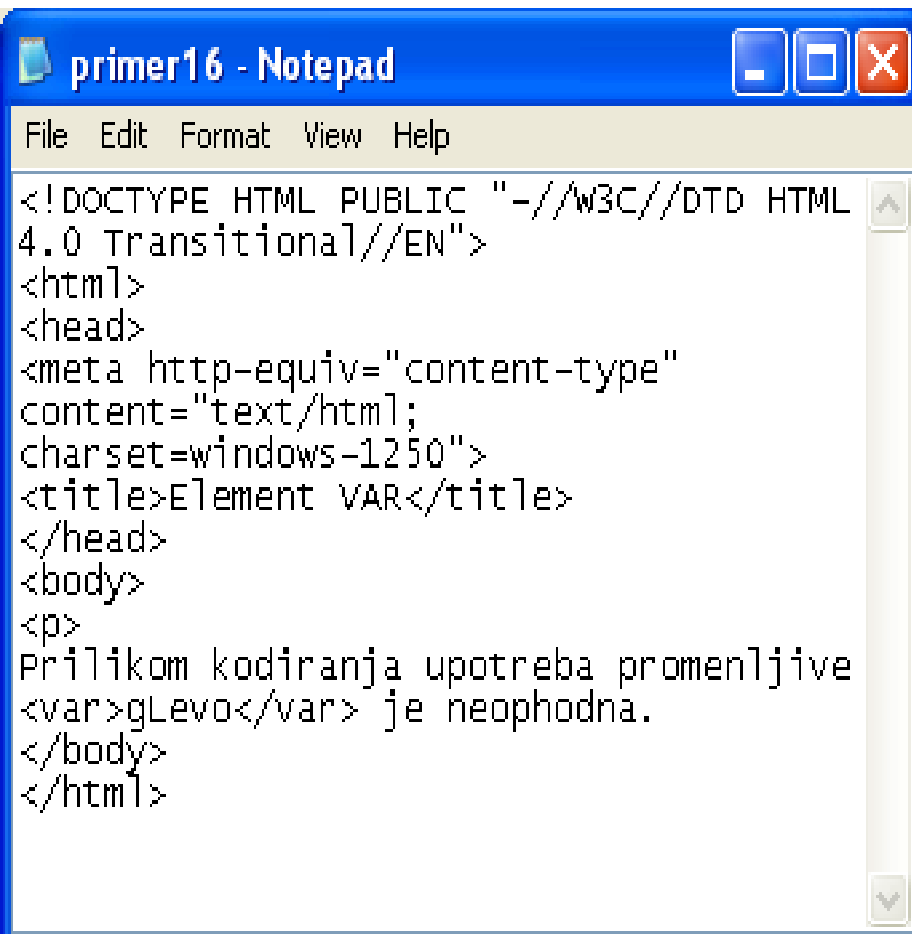
```
primer14 - Notepad
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN">
<html>
<head>
<meta http-equiv="content-type" content="text/html;
charset=windows-1250">
<title>Element SAMP</title>
</head>
<body>
<p>
Tekst unutar <samp>SAMP elementa</samp> se obièno
prikazuje neproporcionalnim fontom.
</body>
</html>
```



- Елемент **STRONG** се користи за јаче истицање текста. Текст је приказан појачаним фонтом.

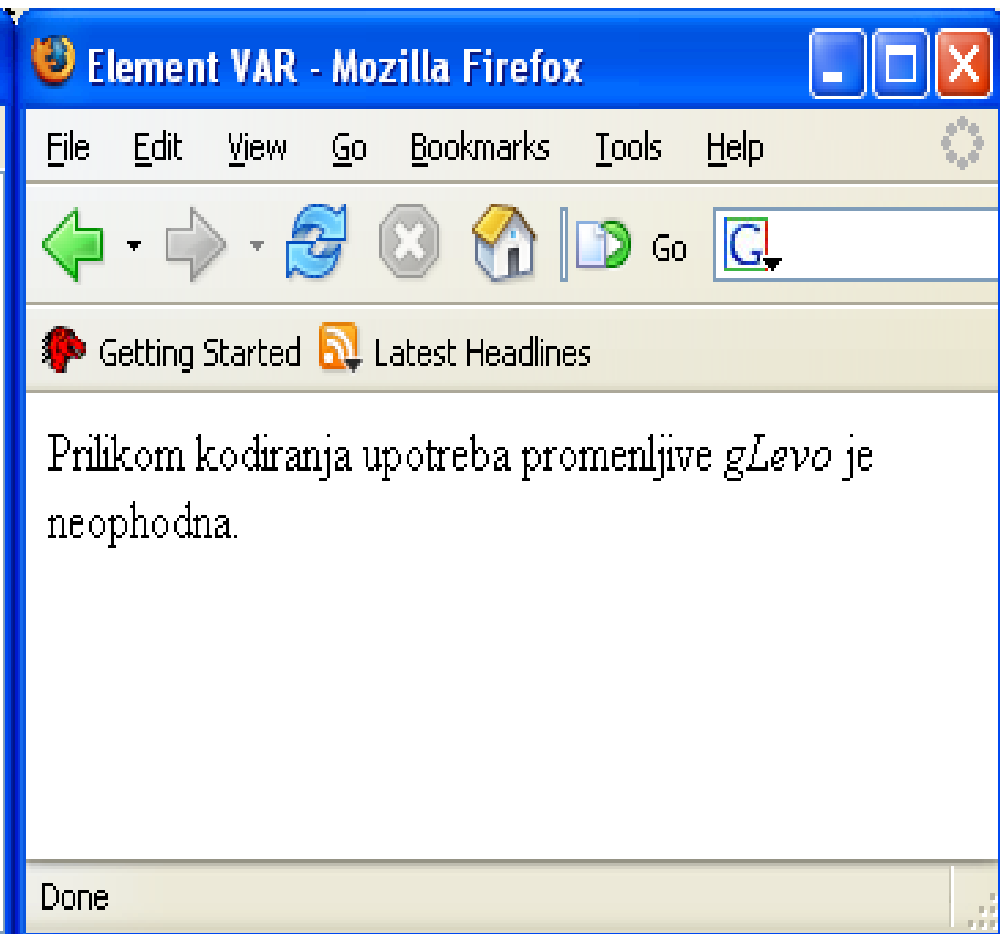


- Елемент **VAR** указује на име променљиве. Текст је у курзиву.



primer16 - Notepad

```
File Edit Format View Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML
4.0 Transitional//EN">
<html>
<head>
<meta http-equiv="content-type"
content="text/html;
charset=windows-1250">
<title>Element VAR</title>
</head>
<body>
<p>
Prilikom kodiranja upotreba promenljive
<var>gLevo</var> je neophodna.
</body>
</html>
```



ФИЗИЧКИ СТИЛОВИ СУ:

B	Слова су појачана
I	Слова су искошена
TT	Слова су приказана непропорционалним фонтом
BIG	Текст је приказан великим фонтом
SMALL	Текст је приказан малим фонтом
SUB	Текст је приказан испод нормалног текста
SUP	Текст је приказан изнад нормалног текста

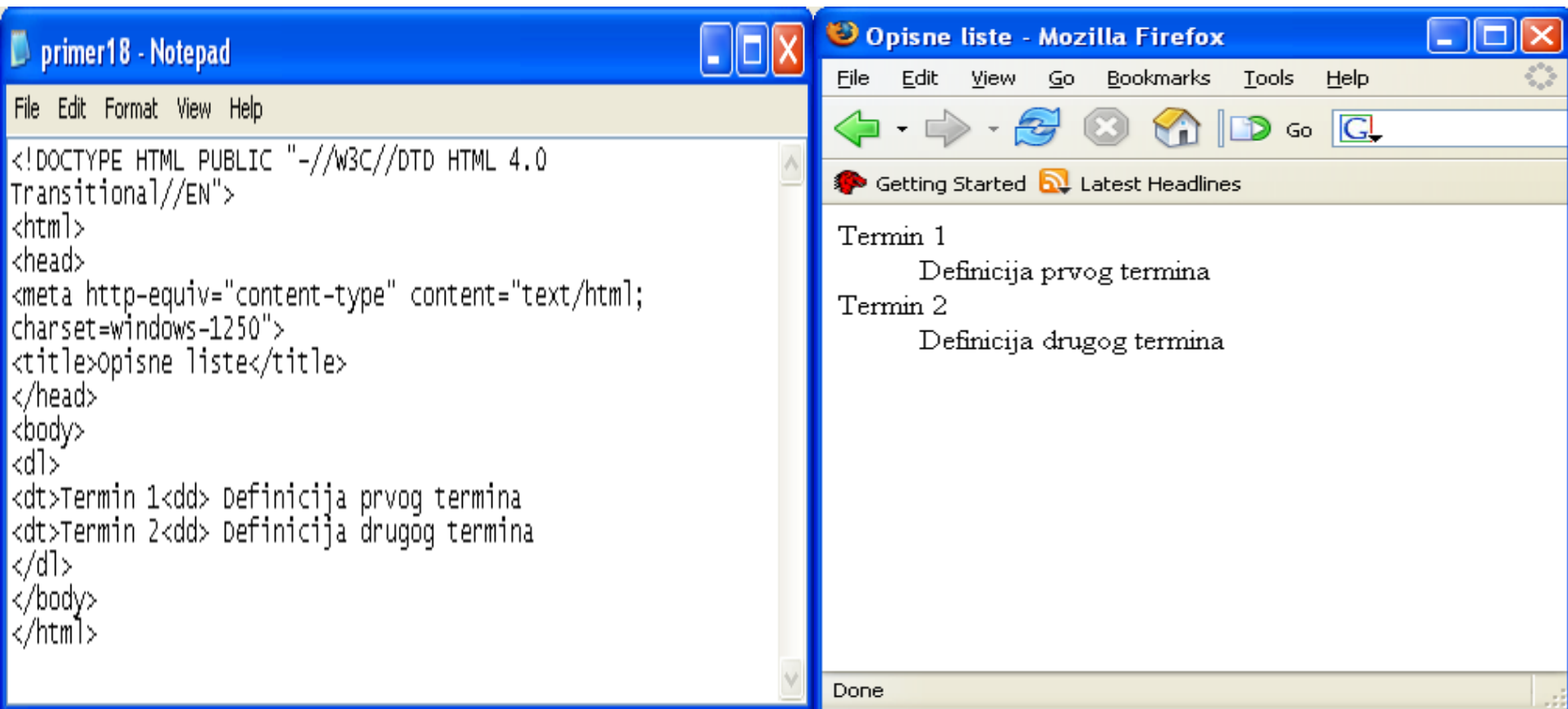
- Сви елементи имају почетне и завршне маркере.

Елементи набрајања

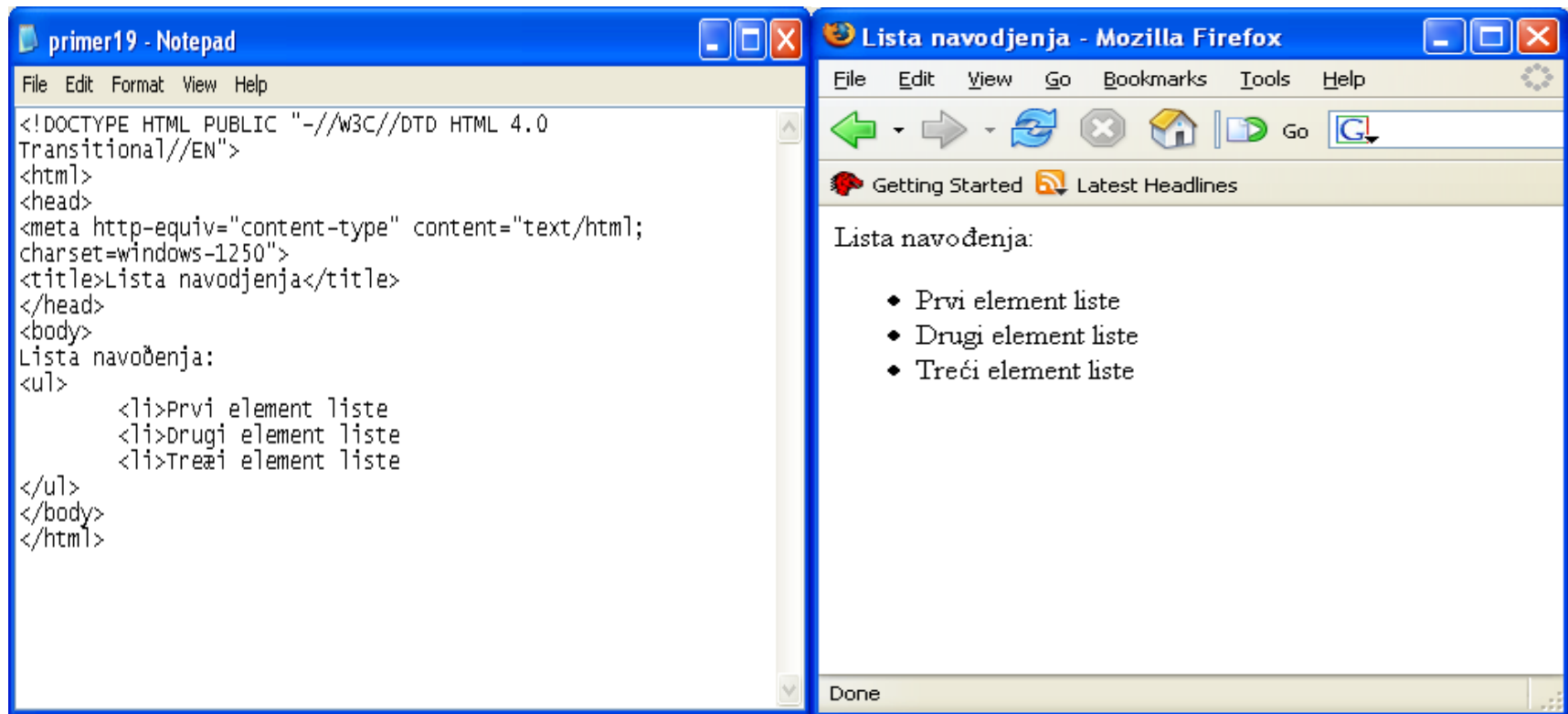
- HTML подржава следеће типове набрајања:
 - описне листе или листе за дефиниције
 - листе набрајања
 - листе навођења.
- Описне листе су листа термина и одговарајућих дефиниција.
- Овај тип листа је посебно погодан за индексе.

- Описна листа почиње маркером `<DL>`, а завршава се маркером `</DL>`.
- Елемент `DL` садржи елементе `DT` и `DD`.
- Оба елемента имају почетне и завршне маркере, али се завршни маркери могу изоставити.
- Елемент `DT` се користи за означавање термина који ће се дефинисати у листи.
- Елемент `DD` се користи за означавање дефиниције у листи.

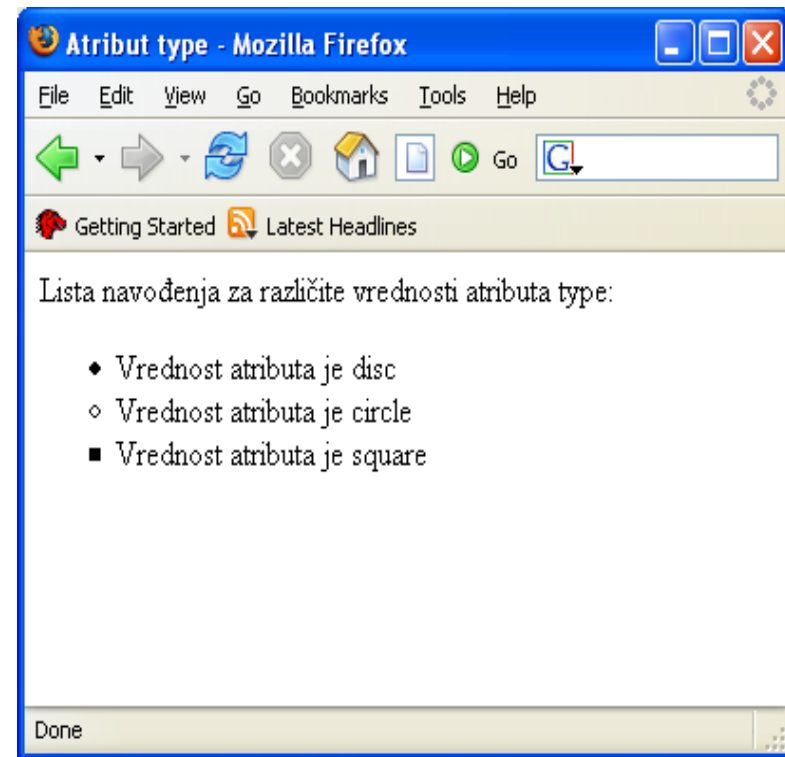
- Читачи обично увлаче дефиниције у новом реду испод термина који се дефинише.



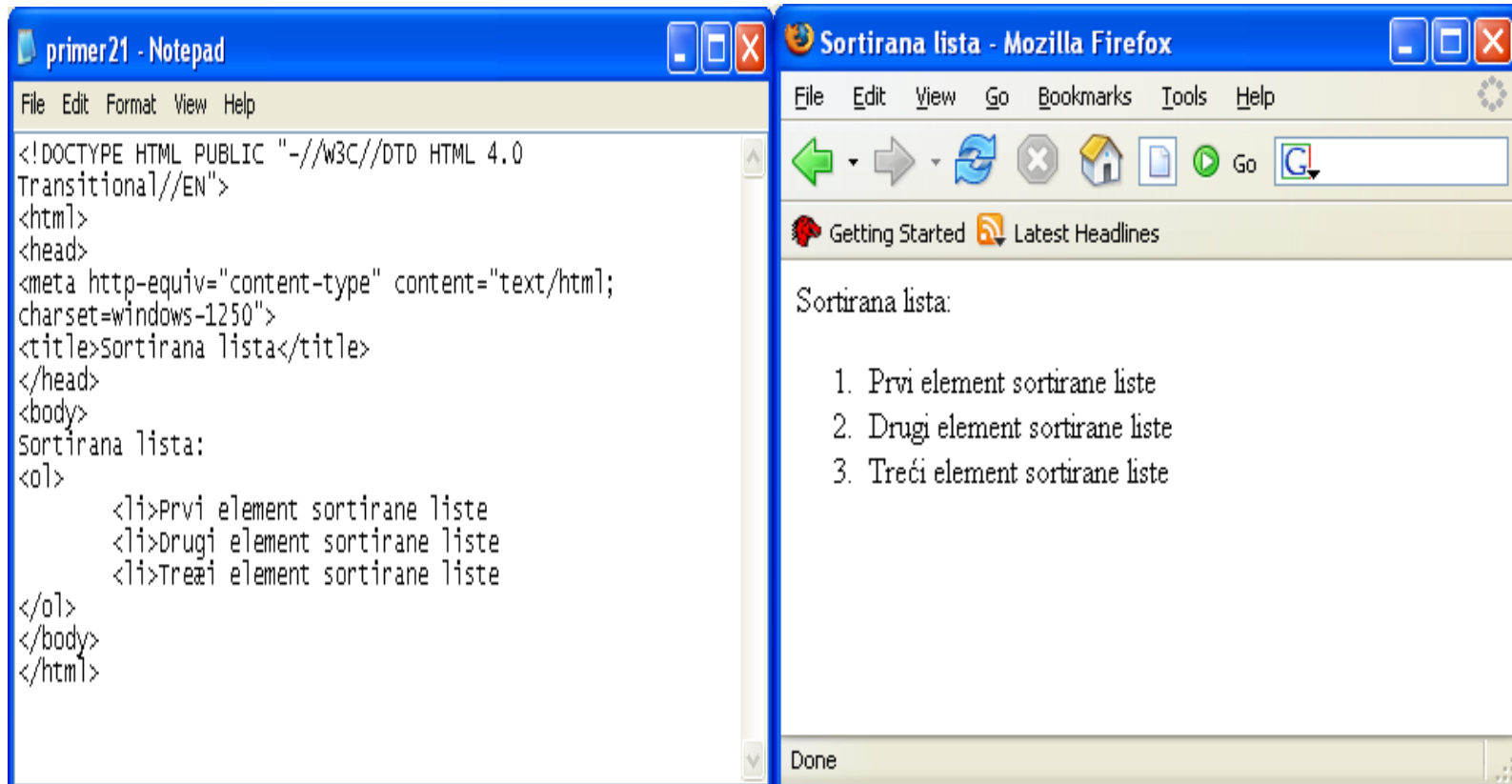
- Листа набрајања се користи за представљање листе чланова одвојене празним редом.
- Листа почиње маркером ``, а завршава се маркером ``.
- Чланови листе се представљају елементом `LI`.
- Елемент `LI` има почетни и завршни маркер, али се завршни маркер може изоставити.



- За дефинисање стила ознаке за целу листу или појединачне ставке листе може се користити превазиђени атрибут **type**.
- Вредности атрибута **type** могу бити:
 - **disc**
 - **circle**
 - **square**.
- Ако је **type=disc**, тада је ознака
- Ако је **type=circle**, тада је ознака
- Ако је **type=square**, тада је ознака



- Сортиране листе се користе за представљање листи које су сортиране према неком критеријуму.
- Листа почиње маркером ``, а завршава се маркером ``.
- Чланови листе се представљају елементом `LI`.
- Ако другачије није одређено, чланови сортиране листе се нумеришу бројевима 1, 2, 3 итд.



- Начин обележавања набрајања може се променити превазиђеним атрибутом **type**.
- Могуће вредности атрибута **type** су:

A За обележавање користимо велика слова A, B, C ...

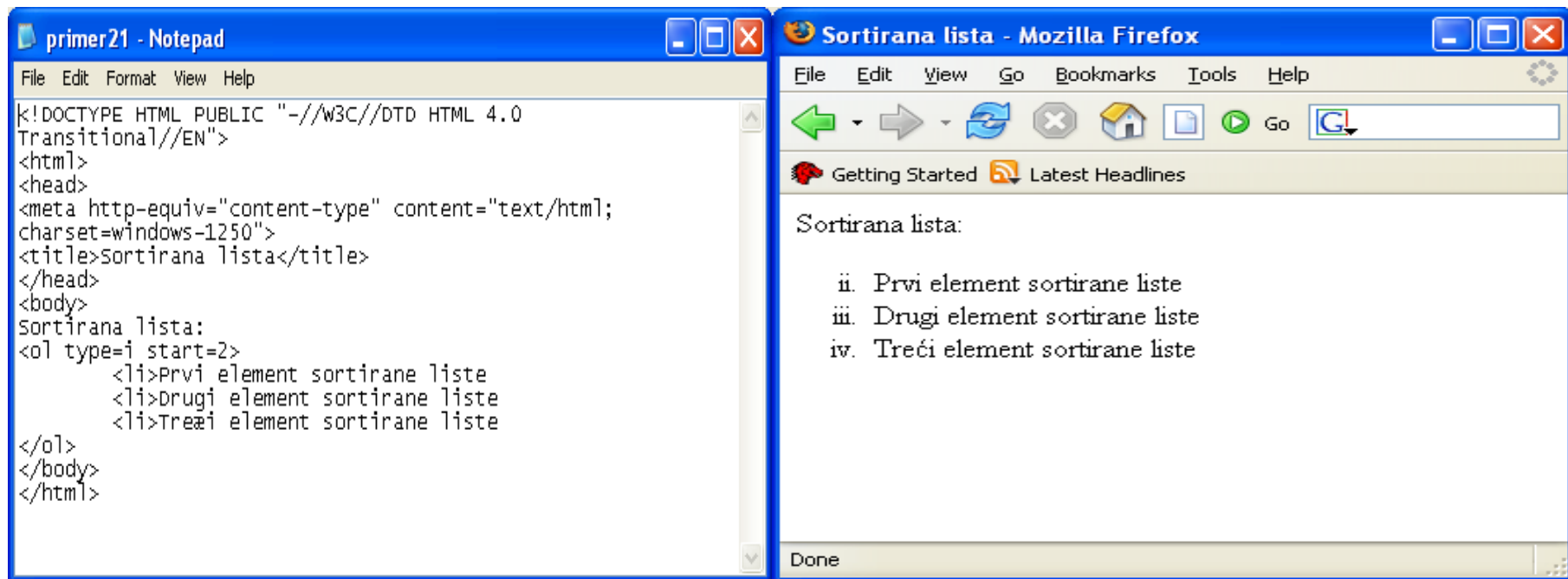
a За обележавање користимо мала слова a, b, c ...

I За обележавање користимо велике римске бројеве I, II ...

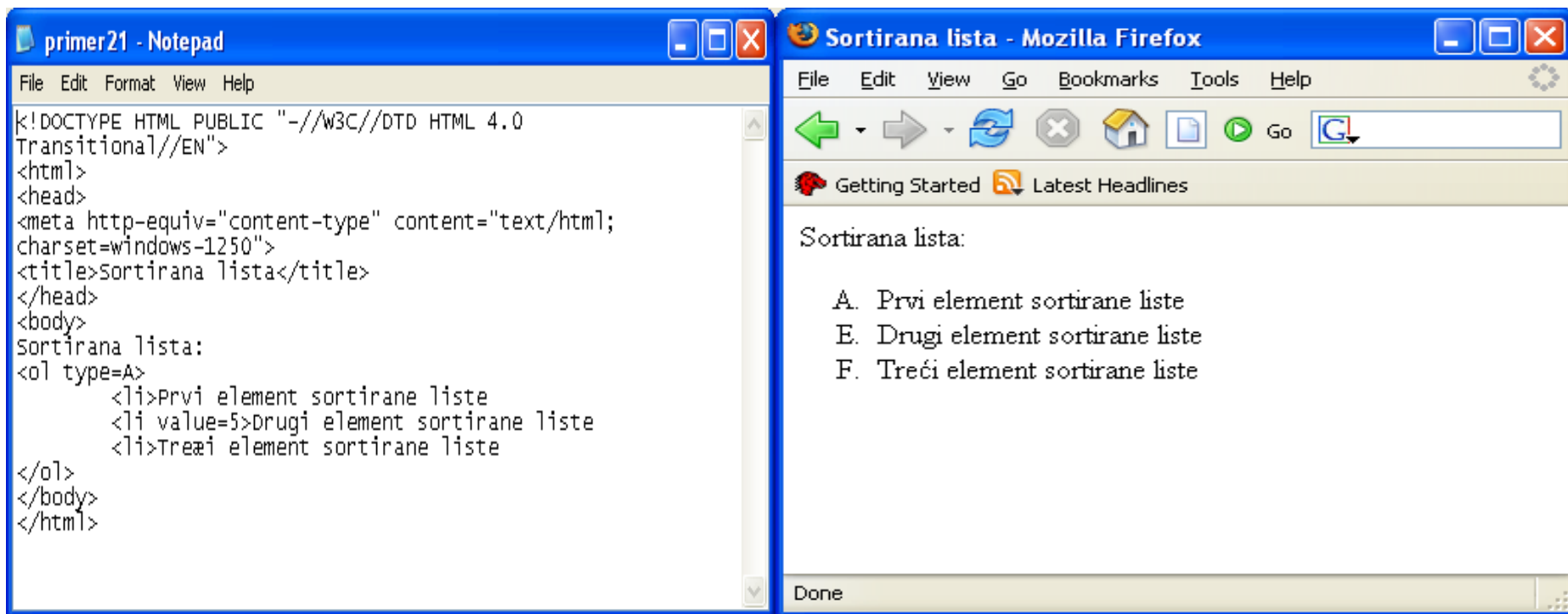
i За обележавање користимо мале римске бројеве i, ii ...

1 За обележавање користимо арапске цифре 1, 2, 3 ...

- Нумерација сортиране листе може почети бројем различитим од 1.
- То се постиже превазиђеним атрибутом **start**.
- Вредност атрибута **start** се увек поставља у бројевима, који се затим конвертују према одговарајућем типу набрајања.



- Може се променити набрајање дате ставке у листи.
- То се постиже превазиђеним атрибутом **value**.
- Промена броја ставке у листи доводи до промене следећих елемената у листи.



Mala škola Java Script-a sa primerima - uvod

JavaScript je skript jezik koji je razvila kompanija *Netscape Communications* radi povećanja interaktivnosti Web strana. Web browser-i koji podržavaju JavaScript interpretiraju kôd. JavaScript dodaje interaktivnost Web prezentaciji na strani klijenta. Pri kreiranju JavaScript kôda može se koristiti ma koji ASCII editor.

Polazni naziv JavaScript jezika bilo je LiveScript, ali u saradnji Netscape-a i Sun-a, ime je izmenjeno u JavaScript, zbog namene da Java i JavaScript budu jezici Internet-a i internet programiranja.

UKRATKO O *JAVA SCRIPT*-u i *JAVA*-i

Java je **objektno orijentisani programski jezik**, namenjen razvoju aplikacija za sisteme na računarskim mrežama.

Aplikacija napisana u Javi prenosiva je između različitih platformi, što Javu i čini specifičnom.

U Javu su ugrađeni brojni sigurnosni mehanizmi: memorija nije dostupna niti jednom Java programu; svi bajt kôdovi preneseni mrežom se proveravaju u cilju pronalaženja neželjenih izmena na putu od servera do klijenta; onemogućeno je da program-uljez kloniranjem klasa unese destruktivni kôd u program; itd.

Pojavom Jave došlo je do znatnog unapređenja komunikacije u Internet svetu.

Osim što su se otvorile mogućnosti interaktivne komunikacije, i web stranice poprimile su dinamičniji oblik...

Java Script je kompaktan i **objektno zasnovan skriptni jezik** namenjen za razvoj klijent Internet aplikacija.

Programski kôd upisuje se direktno na HTML stranicu. WEB BROWSER-I KOJI PODRŽAVAJU JavaScript interpretiraju kôd i obezbeđuju širok raspon interaktivnosti i efekata na Web stranama.

Java Script poseduje relativno mali broj tipova podataka: numeričke, boolean i string vrednosti. Takođe, podržava funkcije, ali bez posebnih zahteva za deklarisanjem.

Poređene Java Script-a i Java-e

Java Script	Java
Ne kompajlira se, već samo interpretira kod klijenta.	Kompajlira se na serveru pre izvršenja kod klijenta.
Objektno zasnovan. Koristi postojeće objekte, bez klasa ili	Objektno orjentisan. Applet i sadrže objekte sastavljene od klasa sa

nasleđa.	nasleđem.
Kôd je integrisan u HTML.	Applet je izdvojen iz HTML-a.
Tip promenljive se ne deklarise.	Tip promenljive mora biti deklarisan.
Dinamička povezanost. Objektne reference proveravaju se u run-time.	Statička povezanost. Objektne reference moraju postojati u vreme kompajliranja.
Zaštićen. Ne može zapisivati na hard disk.	Zaštićen. Ne može zapisivati na hard disk.

Načini uključivanja JavaScript-a unutar HTML dokumenta

1. u sekciji <BODY>... </BODY> unutar oznaka <SCRIPT language="JavaScript" type="text/javascript"> ... </SCRIPT>
2. u sekciji <HEAD>... </HEAD> unutar oznaka <SCRIPT>...</SCRIPT>
3. u sekciji <BODY>... </BODY> ili <FORM>... </FORM> unutar HTML elementa. Tada se ne moraju koristiti oznake <SCRIPT>...</SCRIPT>
4. u posebnim datotekama sa ekstenzijom .js, a da bi se koristila zasebna datoteka, mora se u tekućem HTML dokumentu uključiti unutar oznaka, na primer primer.js:

```
<SCRIPT language="JavaScript" type="text/javascript" src="primer.js"> </SCRIPT>
```

Načini izvršenja skripta unutar taga <SCRIPT>...</SCRIPT>

- ako je skript uključen u zaglavlju, ignorisan je do poziva
- ako je skript uključen u telu, rezultat se prikazuje na Web strani
- ako je skript rukovalac događaja, skript će se izvršiti kad se desi događaj

UKRATKO O PROMENLJIVAMA, TIPOVIMA PODATAKA, KONVERZIJI I OPERATORIMA

NAZIVI PROMENLJIVIH

- prvi znak mora biti slovo engleske abecede ili podvlaka ("_")
- mogu sadržati brojeve i slova engleske abecede
- velika i mala slova se razlikuju, ali, uobičajeno je da se pišu malim slovima!
- ključne reči (*for, if, else, class, byte, int...*) ne mogu se koristiti u nazivu identifikatora

TIPOVI PODATAKA

- numerički (npr: 3.14159 ...)
- logički (Boolean) - (true / false)
- stringovi (npr: "Zdravo, svete!")
- null (specijalne ključne reči s null vrednošću)

SPECIJALNI KARAKTERI

U stringovima je dozvoljeno koristiti sledeće specijalne karaktere:

- `\b` = pomeraj za jedno mesto ulevo (backspace)
- `\f` = pomeraj jedan red dole (form feed)
- `\n` = na početak novog reda (new line character)
- `\r` = return (carriage return)
- `\t` = tabulator (tab)

KONVERZIJA PODATAKA

JavaScript nije strogo tipiziran i ne zahteva posebno deklarisanje tipa podataka, te će se konverzija obaviti automatski.

Osnovni način za deklarisanje promenljivih:

```
var broj1 = 42  
var tekst = "Zdravo, svete!"
```

ali i:

```
broj1 = 42  
tekst = "Zdravo, svete!"
```

Generalno, u kombinaciji broja i stringa, Java Script konvertuje broj u string. Na primer:

```
x = "Upisali ste broj " + 54  
y = 54 + " je odgovor na Vas unos."
```

Za konverziju stringa u broj, koriste se funkcije:

- EVAL - ocenjuje string i ako je moguće pretvara ga u broj;
- parseInt - konvertuje string u integer, ako je moguće;
- parseFloat - konvertuje string u floating-point broj, ako je moguće.

ARITMETIČKI OPERATORI

Operacija	Objašnjenje
<code>X + Y</code>	- sabiranje
<code>X - Y</code>	- oduzimanje
<code>X * Y</code>	- množenje
<code>X / Y</code>	- deljenje
<code>X % Y</code>	- ostatak deljenja X sa Y (modul)
<code>X++</code>	- postfiksno uvećanje za 1
<code>++X</code>	- prefiksno uvećanje za 1
<code>X--</code>	- postfiksno umanjenje za 1
<code>--X</code>	- prefiksno umanjenje za 1

RELACIJSKI OPERATORI

Operacija	Objašnjenje
<code>X > Y</code>	- X veće od Y
<code>X < Y</code>	- X manje od Y
<code>X >= Y</code>	- X veće ili jednako Y
<code>X <= Y</code>	- X manje ili jednako Y
<code>X == Y</code>	- X jednako Y
<code>X != Y</code>	- X nije jednako Y

OPERATORI USLOVA

Operacija	Objašnjenje
<code>X && Y</code>	- konjunkcija (logičko "I")
<code>X Y</code>	- disjunkcija (logičko "ILI")
<code>!X</code>	- negacija (logičko "NE")

OPERATORI PRIDRUŽIVANJA

Operacija	Objašnjenje
<code>X = Y</code>	- promenljivoj X pridružuje se vrednost Y
<code>X += Y</code>	<code>X = X + Y</code>
<code>X -= Y</code>	<code>X = X - Y</code>
<code>X *= Y</code>	<code>X = X * Y</code>
<code>X /= Y</code>	<code>X = X / Y</code>
<code>X %= Y</code>	<code>X = X % Y</code> (ostatak pri celobrojnom deljenju)

USLOVNI IZRAZI

Uslovnim izrazom možemo prihvatiti jednu od dve vrednosti navedene u uslovu. Evo sintakse i primera:

```
(uslov) ? izraz1 : izraz2  
status = (godina >= 7) ? "placaju" : "besplatno"
```

Ako je uslov istinit (true), izraz ima vrednost *izraz1*, u suprotnom ima vrednost *izraz2*.

UKRATKO O KONTROLI TOKA I ISKAZIMA

IF...ELSE struktura:

```
if(izraz1){  
    ...iskazi...  
}  
else if(izraz2) {  
    ...iskazi...  
}  
else {  
    ...iskazi...  
}
```

Primer s IF... strukturom:

```

<SCRIPT LANGUAGE="JavaScript"> /* oznaka početka Java Scripta */
<!-- /* sve unutar ovog taga je nedostupno browserima koji ne podržavaju Java Script */

    var datum = new Date(); /* promenljivoj datum pridružuje tekući datum */
    var sat = datum.getHours(); /* u promenljivu sat prihvata tekući čas */

    if (sat > 5 && sat < 13){ /* logička provera */
        document.write('<BODY BGCOLOR="blue">Dobro jutro!') /* BODY i ispis poruke */
    }
    if (sat > 12 && sat < 19){ /* ako je čas veći od 12 i manji od 19 */
        document.write('<BODY BGCOLOR="green">Dobar dan!')
    }
    if (sat > 18 || sat < 6){ /* ako je čas veći od 18 ili manji od 6 */
        document.write('<BODY BGCOLOR="silver">Dobro veče!')
    }

->
</SCRIPT> /* oznaka kraja Java Scripta */

```

Navedeni primer postavlja odgovarajuću pozadinu i ispisuje poruku u HTML dokumentu u zavisnosti od tekućeg sata. Unutar zagrada naredbe `document.write(...)` može se postaviti bilo koji HTML tag, tekst ili sve to kombinovati s vrednostima svojih promenljivih ili već postojećih funkcija u JavaScriptu...

FOR... ciklus:

```

<SCRIPT LANGUAGE="JavaScript">
<!--
    for (var i = 1; i <= 10; i++){ /* i=1; ciklus traje sve dok je i<=10; i=i+1 */
        document.write('Vrednost i = ' + i); /* u HTML ispisuje: Vrednost i =... */
    }
->
</SCRIPT>

```

Navedeni primer će u HTML dokumentu, na poziciji gde je postavljen, ispisati:
Vrednost i = 1 Vrednost i = 2 ... Vrednost i = 10

FOR...IN ciklus:

```

for (promenljiva in objekt) {
    ... iskazi...
}

```

WHILE... petlja:

```
while (izraz){
  ...iskazi...;
}
```

DO...WHILE... petlja:

```
do {
  ...iskazi...;
} while (izraz);
```

WITH... iskaz:

```
with (objekt) {
  ...iskazi...
}
```

Na primer, matematičkim funkcijama mora prethoditi objekat Math. Sledeći primer podrazumeva Math ispred PI, COS() i SIN():

```
with (Math) {
  a = PI / 2;
  x = cos(a);
  y = sin(a);
}
```

BREAK iskaz:

prekida tekuću while ili for strukturu i nastavlja iza nje.

```
i = 0;
n = 0;
while (i < 5) {
  i++;
  if (i == 3) {
    continue
  }
  else {
    n += i
  }
}
```

FUNCTION i RETURN iskaz:

primer funkcije koja izračunava kvadrat broja:

```
function kvadratBroja( x ) {
  return x * x
}
---
x = kvadratBroja(5) /* poziv funkcije */
```

```
document.write("Kvadrat od 5 je " + x)
---
Kvadrat od 5 je 25
```

NEW iskaz:

definiše novi objekt na bazi postojećeg.

Donji primer promenljivih datum pridružuje tekući datum:

```
var datum = new Date();
```

THIS iskaz:

referenca na tekući objekt.

Donji primer definiše da se klikom na objekt poziva funkcija **slucajniBroj()** a kao parametar prenosi se naziv tekuće forme (**this.form**):

```
onClick="slucajniBroj(this.form)"
```

VAR iskaz:

definisanje promenljivih (opcionalno).

```
var broj1 = 0, tekst1="proba"
```

OBJEKTI, METODE, SVOJSTVA I DOGAĐAJI

Svaka web stranica poseduje objekte:

- **window**: top-level objekt; sadrži svojstva primenjiva na celokupan prozor,
- **location**: sadrži svojstva tekuće URL lokacije
- **history**: sadrži svojstva prethodno posećenih URL
- **document**: sadrži svojstva sadržaja tekućeg dokumenta, kao što su naziv (title), boja pozadine (bgcolor), forme

Primer svojstava:

- **location.href** = "http://www.matf.bg.ac.yu/uwit/proba.html" //lokacija dokumenta
- **document.title** = "Probni dokument" //naziv dokumenta (title)
- **document.fgColor** = #000000 //boja slova
- **document.bgColor** = #FFFFFF //boja podloge
- **history.length** = 7 //koliko poslednjih dokumenta da "pamti" u history

Browser može kreirati objekte bazirane na sadržaju stranice, npr.:

- `document.mojaforma //forma`
- `document.mojaforma.Check1 //check polje na formi`
- `document.bojaforma.Button1 //taster na formi`

Oni mogu imati svojstva kao što su:

- `document.mojaforma.action = "http://www.matf.bg.ac.yu/primeri/obrada.html"`
- `document.mojaforma.method = get`
- `document.mojaforma.length = 5`
- `document.mojaforma.Button1.value = "Klikni ovde"`
- `document.mojaforma.Button1.name = taster1`
- `document.mojaforma.text1.value = "sadržaj tekst polja"`
- `document.mojaforma.text1.name = TekstPolje1`
- `document.mojaforma.Check1.defaultChecked = true`
- `document.mojaforma.Check1.value = on`
- `document.mojaforma.Check1.name = CheckPolje1`

Mnogi objekti imaju metode koje emuliraju događaje.

Npr, button objekt ima click metodu koja emulira klik miša na tasteru...

Event (događaj)	Nastaje kada korisnik...	Event Handler
blur	izađe iz fokusa elementa forme	onBlur
click	klikne na elementu forme ili link	onClick
change	podesi/promeni vrednost "text", "textarea" ili izabranog elementa	onChange
focus	uđe u fokus nekog elementa forme	onFocus
load	učita stranicu u browser	onLoad
mouseover	pomera pokazivač miša preko linka ili "anchora"	onMouseOver
select	izabere "input" polje elementa forme	onSelect
submit	izvrši "submit" (slanje) forme	onSubmit
unload	"napusti" stranicu	onUnload

KOMENTARI U JAVA SCRIPTU

Za komentar do kraja reda ili onaj na kraju nekog iskaza (npr, iza `;`) koristi se oznaka `//...`

Za komentar kroz više redova ili umetnut unutar neke strukture (npr, unutar `{ }` zagrada) koristi se `/**/`