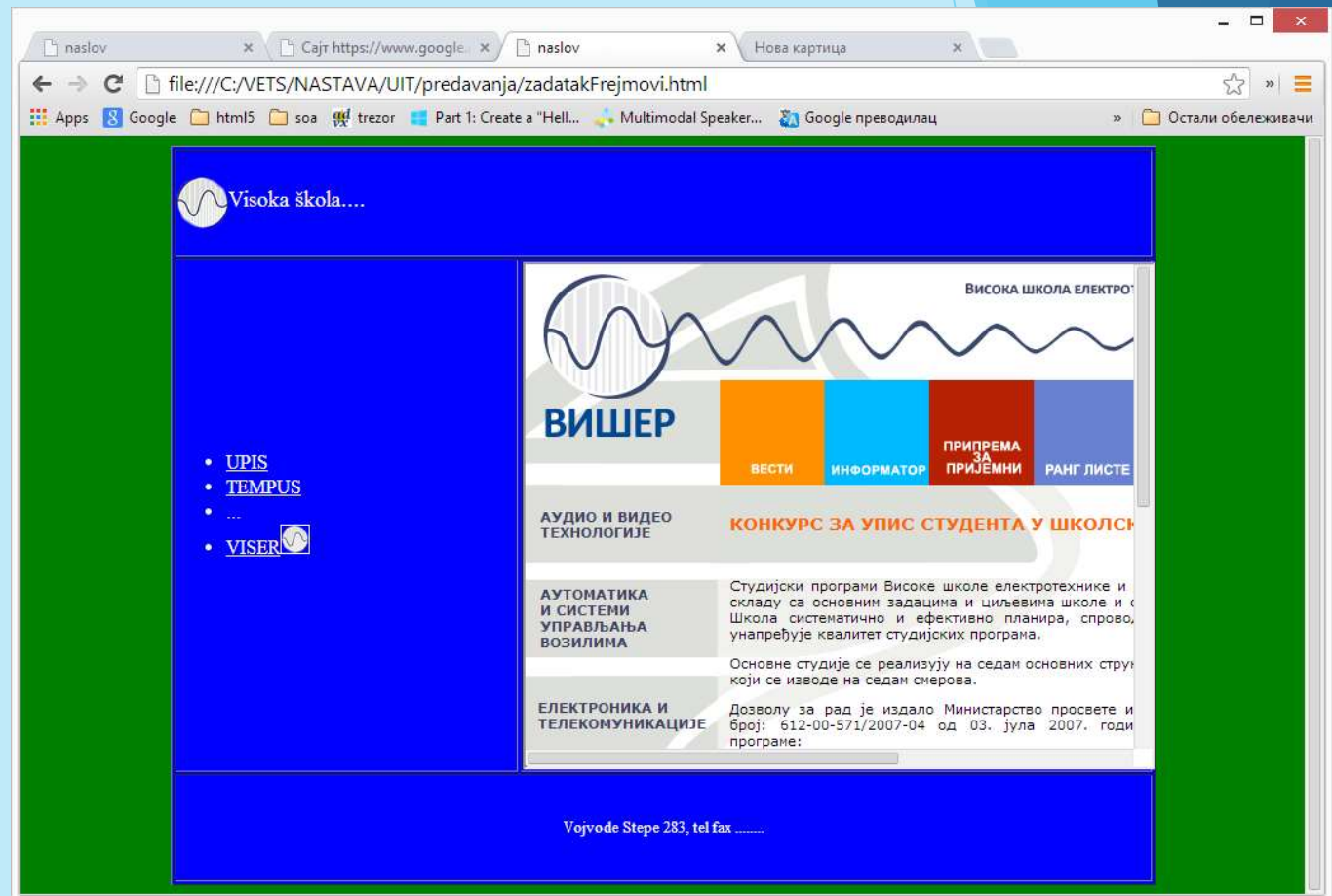


Zadatak

- Koristežći frejmove napisati html stranicu po uzoru sa slike.



The image features a minimalist design with a light gray background. On the left, a solid blue triangle points downwards. On the right, a complex arrangement of overlapping translucent blue triangles and lines creates a layered effect. A thin blue line extends from the bottom left towards the center, with the number '16.05' placed near its midpoint. Another thin blue line extends from the bottom right towards the center, with the number '2' placed near its midpoint. The text 'CSS' is centered in the upper half of the image in a large, blue, sans-serif font.

CSS

16.05

2

Uvod

- ▶ CSS je skraćenica od engleskog naziva *Cascading Style Sheets*.
- ▶ Stil (engl. Style) definiše prikaz HTML elemenata.
- ▶ CSS se može na jedinstven način primeniti da više elemenata ili grupa elemenata pa tako na dosledan način definisati izgled i funkcionalnost stranica.
- ▶ Stil se piše u elementu <head>, u eksternim stilskim stranicama ili u samom elementu.
- ▶ Češće korišćena varijanta je pisanje CSS-a u eksternim stranicama.
- ▶ *Napomena: Zbog lakšeg učenja u našim primerima biće češće korišćen interni stil pisanje stilova.*

Struktura zapisa

- ▶ Između tagova <head> i </head> dodaje se:

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

HTML5: type nije obavezan!
Označava tip internet sadržaja (MIME)
Podrazumevana vrednost je text/css

- ▶ Na primer, da bi promenili boju teksta na stranici u plavo:

- <style type="text/css">
- body { color: blue; }
- </style>

Struktura zapisa stila

- ▶ U terminologiji CSSa dve glavne sekcije pravila su:
 - **selektor** koji definiše jedan element ili više elementa na koje će se odnositi i
 - **deklaracije** koje se odnose na formiranje izgleda.

declaration

```
selector { property: value; }
```

declaration block

```
selector {  
  property1: value1;  
  property2: value2;  
  property3: value3;  
}
```

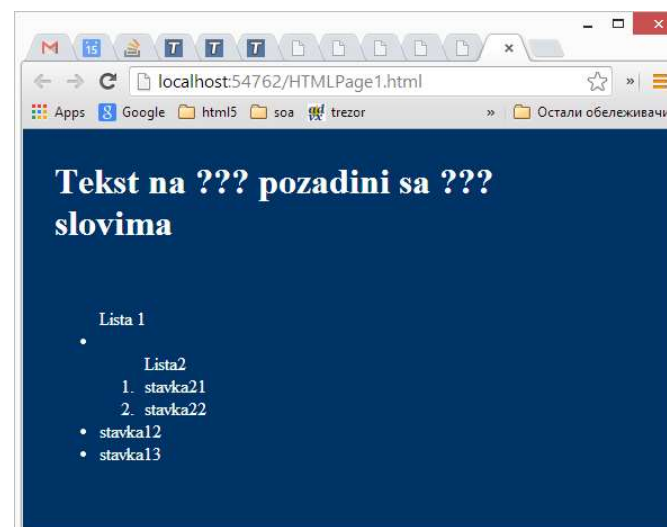
Napomena: Nazivi svojstava u bloku za deklaraciju biće kasnije detaljno rađena. Za sada uglavnom koristimo neke razumljive primere.

- ▶ Deklaracija zapravo određuje svojstva izgleda i sastoji se iz **naziva** svojstva (npr. color) i **vrednosti** svojstva (npr. green).
- ▶ Jedna ili više deklaracija se smeštaju u **vitičaste zagrade**, odvojene međusobno **tačka zarezom**.

```
#cssPodloga  
{  
  background-color: #003366;  
  padding: 20px;  
  border: 1px;  
  color: #FFFFFF;  
  height: 100%;  
  width: 50px;  
}
```

• • •

```
<body id="cssPodloga">
```



Selektori

- ▶ Na primer:
- ▶

```
h1 { color: green; }  
p { font-size: small; font-family: sans-serif; }
```
- ▶ U primerima, `h1` odnosno `p` su HTML elementi koji su u CSS iskazu korišćeni kao selektori, pa se ovi selektori nazivaju **selektorima tipa elementa** (eng. element type selector) i predstavljaju osnovi tip selektora.
- ▶ Svojstva koja su definisana za svako od pravilo biće primenjena na svaki `h1` odnosno `p` element dokumenta.
- ▶ Postoje mnogo napredniji selektori koje ćemo upoznati kasnije.

Deklaracije

- ▶ Sastoje se od parova **svojstvo/vrednost**. Može da postoji **više od jedne deklaracije za jedno pravilo**, npr. pravilo za element `p` koje je navedeno u prethodnom primeru, uključuje veličinu fonta i naziv fonta.
- ▶ Svako pravilo se mora završiti sa **tačka-zapetom**, kako bi se odvojilo od ostalih. Ako se izostavi tačka-zapeta, deklaracija se ignoriše. **CSS ignoriše praznine i nove redove**. Zato se često deklaracije pišu u zasebnim redovima i sa belinama.
- ```
p{
 font-size: small;
 font-family: sans-serif;
}
```

# Primer 1

- ▶ U folderu cssPrimer1 otvorite kod.
- ▶ Prikažite navedenu stranicu.
- ▶ Uključite stilove koji su bili pod komentaram.
- ▶ Prikažite novu stranicu.
- ▶ Pogledajte kod i razmislite kako utiče na prikaz.

# Načini povezivanje stilova i dokumenata

- ▶ U prethodnom primeru smo povezali stil sa nazivom elementa i to u tekućem dokumentu. To je jedan od 3 načina povezivanja.
- ▶ **Spoljni** - zaseban fajl koji mora imati ekstenziju **.css**. Stilovi se naknadno povezuju ili importuju u jedan ili više HTML dokumenata. Na ovaj način ceo web sajt deli iste stilove, što je dobar način za održavanje.
- ▶ **Unutrašnji** - Ovaj stil često koristimo u našim primerima i na vežbama. Ubačen je u HTML dokument i deluje samo na taj dokument. Element **style** je smešten u zaglavlju dokumenta

```
<head>
 <title>Required document title here</title>
 <style>
 /* ovako se pisu komentari, npr <style type="text/css">
 se pise za XHTML i HTML 4.01 verzije*/
 </style>
</head>
```

- ▶ **Linijski** (engl. inline) - Moguće je da se stil primeni na samo 1 element koristeći sopstvene attribute tog elementa. Na primer:
- ▶ `<h1 style="color: red"></h1>`
- ▶ Da bi koristili višestruka svojstva, samo je potrebno da ih razdvojimo tačka-zapetom, na primer:
- ▶ `<h1 style="color:red; margin-top:2em">Introduction</h1>`
- ▶ **Linijske stilove treba izbegavati**, osim ako je apsolutno neophodno kako bi se promenilo delovanje spoljašnjih stilova. Eventualna izmena sajta zahteva pretraživanje pojedinačnih elemenata, što je izuzetno nepraktično.
- ▶ **Povezivanje HTML stranice sa CSS fajlom**
- ▶ Pomoću dva pravila:
- ▶ `link`
- ▶ `@import`



- ▶ **Link** element se koristi za povezivanje spoljnih fajlova sa HTML dokumentom. Navodi se u **head** elementu. Link je prost element. Osnovni atributi su zajednički za HTML5 i HTML4.01. To su:
- ▶ **rel="stylesheet"** - definiše relaciju sa tekucim dokumento. Kada je css fajl onda je uvek vrednost **stylesheet**.
- ▶ **href="url"** - lokacija **.css** fajla.
- ▶ Na primer:

```
<head>
 <title>Titles are required.</title>
 <link rel="stylesheet" href="/path/stylesheet.css">
</head>
```

- ▶ Drugi način da se neki **.css** fajl uključi u oblikovanje nekog HTML dokumenta je koristeći pravilo **@import**. Može se primeniti takođe i u spoljnom **.css** fajlu kao i u html dokumentu.

```
<head>
 <style>
 @import url("/path/stylesheet.css");
 /*moze i bez url-a, navodeci samo vrednost*/
 p { font-face: Tahoma;}
 </style>
 <title>Naslov dokumenta</title>
</head>
```
- **@import** čita pravila pre nego što budu primenjivana. Na taj način, pravila koja su naknadno definisana preklapaju ova pravila.

- ▶ Mala je razlika između @import i link načina uključivanja spoljnih css fajlova. Import dozvoljava da se spoljni css fajl uključi u definiciju stila, na primer:
- ▶ `<style type="text/css">@import url("styles.css");</style>`
- ▶ Stariji čitači ne prepoznaju import što takođe može biti iskorišćeno. Na primer: `@import url(../style.css);` neće biti primenjeno u slučaju IE3,4 ili Netscape 4....

## Kaskadni koncept stilova

### ▶ Nasleđivanje

- ▶ HTML elementi se koriste za prikaz sadržaja elemenata. Sadržaj nije nezavisan po elementima, već povezuje te elemente. Na primer, ako se koristi `em` element u nekom paragrafu u kom je definisana veličina i oblik karaktera, očekivano je da veličina ostane ista i za naglašeni tekst.

Ovaj tekst je u *paragrafu* sa naglasenom reci.

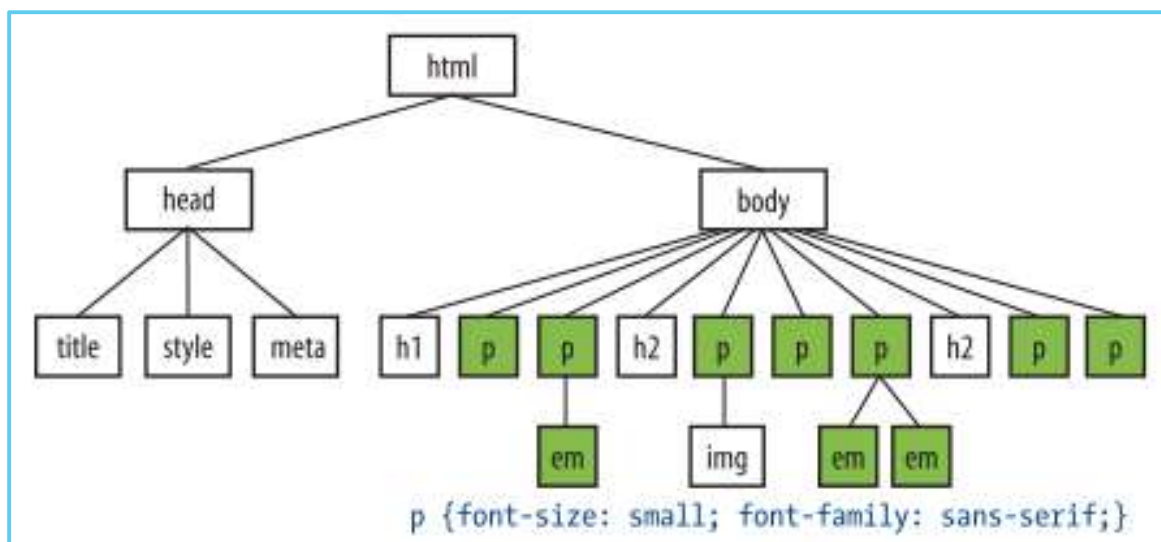
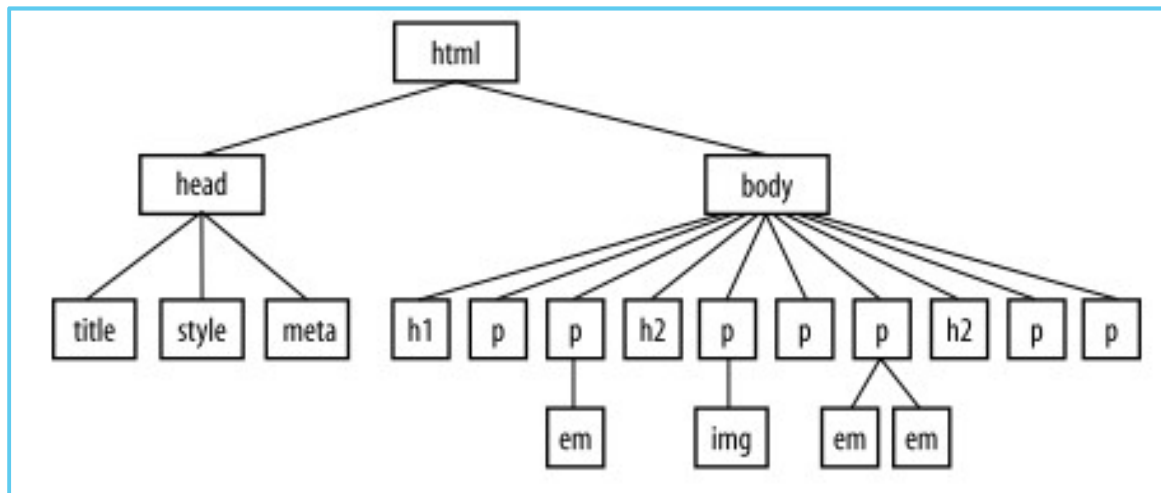
Bez primene stila

Ovaj tekst je u *paragrafu* sa naglasenom reci.

Sa primenom stila:

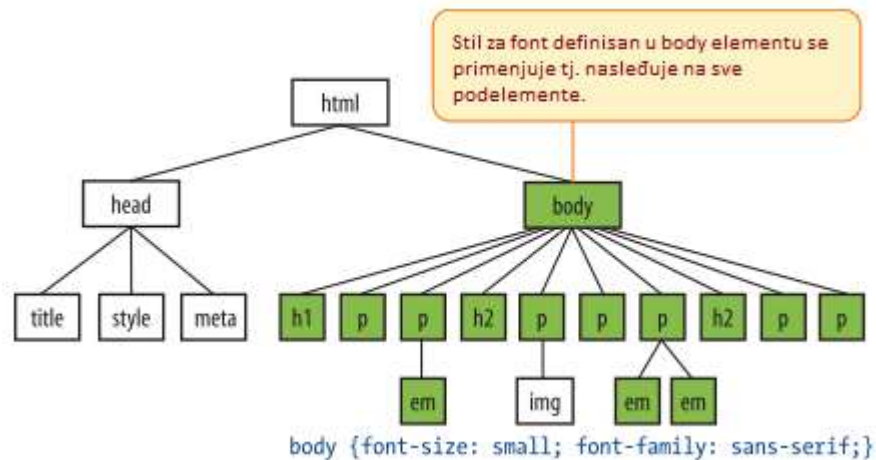
```
p {
 font-size: small;
 font-family: sans-serif;
}
```

- Objašnjenje: Zbog ovog koncepta, struktura HTML dokumenta je bitna za CSS. Kada se napiše stil za definisanje fonta za **p** element (pasus) kao selector, pravilo se primenjuje na sve paragrafe u dokumentu kao i na sve elemente koje ti paragrafi sadrže. Prethodni primer to pokazuje. Na slici se vidi na koje elemente važi primena stila. Zapazite da je **img** element isključen iz stabla primene jer se font ne može primeniti na slike.



# Konflikti

- ▶ Ukoliko se stil **definiše** u **body** elementu primenjuje se na sve elemente HTML stranice.
- ▶ Međutim, nastaje **konflikt** sa stilovima koji su posebno definisani ili se primenjuju na unutrašnje elemente, kao što je bio slučaj sa stilom koji se primenjuje na paragraf **p**.



ili??

```
<p style="color: red">
<h1 style="color: green">
<body style="color: blue">
```

- ▶ CSS dozvoljava da se primenjuje nekoliko stilova na istom dokumentu. Međutim, višestruka primena znači i konflikte. Na primer, ako se na **h1** element primeni jedan spoljni stil koji nalaže da naslov bude crven, a takođe postoji i ugrađeni stil koji za naslove nalaže da budu zeleni, koje boje će naslov biti prikazan?

# Kaskadno rešavanje konflikata

- ▶ Razrešavanje ovog slučaja je toliko važno da je način važenja ugrađen u sam naziv stilova - "kaskadni". **Dakle, pravilo se se primenjuje "odozgo na dole" sve dok ne bude preklopljeno stilom koji ima veću težinu.** Na primer, ako se ne primeni ni jedan stil, važeći stil je interni stil čitača, koji nazivamo podrazumevani stil za renderovanje tj formiranje prikaza.
- ▶ Stilovi koji se definišu u web stranici, dakle, stilovi koje definiše autor web stranice i važe za istu, preklapaju i korisničke i podrazumevane stilove.
- ▶ Jedini izuzetak su stilovi koje korisnik identifikuje kao „important“.
- ▶ Dakle, css stilovi su hijerarhijski i na taj način se određuje koji će biti važeći.
- ▶ Uopšteno, stil koji je „bliže“ definisan za neki element ima veću težinu. Na primer, unutrašnji stil ima veću težinu od spoljnog stila.
- ▶ Na primer, `h1` element iz našeg primera primeniće zelenu boju, kako je navedeno u unutrašnjem stilu.
- ▶ Stilovi u elementu imaju najveću težinu pošto se ne može naći bliži stil za taj element.

# !important

- ▶ Da bi se promenio podrazumevani način primene kaskadnih stilova, moguće je sprečiti preklapanje primenom važnosti na neki stil.
- ▶ Ovo se postiže ubacivanje indikatora `!important`.
- ▶ Na primer, da bi tekst u paragrafu bio plav uvek, koristi se pravilo:
- ▶ `p {color: blue !important;}`
- ▶ Čak i u slučaju da se naknadno uključi drugi „`inline`“ stil, na primer:
- ▶ `<p style="color: red">`
- ▶ Paragraf će i dalje da ima plav tekst jer se stil sa indikatorom `!important` ne može preklapati sa drugim stilovima.
  
- ▶ Dakle, da bi promenili boju paragrafa, ne utičući na ostatak web sajta, postavili smo indikator `!important`.
- ▶ `p {color: blue !important;}`
- ▶ Ipak, da li je moguće promeniti ovaj stil?
- ▶ Jedini način da se neki `!important` stil preklopi jeste da drugi stil koristi takođe indikator `!important`, ali da bude sa većim prioritetom.

# Redosled primene stilova

- ▶ Informacija o stilu može da potiče od različitih izvora. Izvori su navedeni u listi, od opštijih ka specifičnijim. Tako će stavke u listi na nižoj poziciji preklopiti stavke iznad.
- 1. Podrazumevani stilovi browser-a
- 2. Korisnički definisani stilovi u browser-u
- 3. Spoljni stilovi
- 4. Importovani stilovi (dodati pomoću @import funkcije)
- 5. Ugrađeni stilovi (added with the style element)
- 6. Linijski (Inline) stilovi (dodaju se u početnom tagu)
- 7. Stilovi označeni sa !important

- ▶ Šta se događa ako imamo konflikt između stilova jednake težine?
- ▶ Na primer:

```
<style>
 p { color: red; }
 p { color: blue; }
 p { color: green; }
</style>
```

▶ Ili

```
<style>
 p { color: red;
 color: blue;
 color: green; }
</style>
```

- ▶ U ovom scenariju, važeći stil će biti onaj koji je poslednji definisan!



# Box model

Svi HTML elementi su u sopstvenim "kutijama". Ovaj termin „kutije“, engl. *Box*, koristi se kada se govori o prikazu odnosno dizajnu. Box model je model koji predstavlja oivičeni HTML element jednim pravougaonikom koji se sastoji od: margina, ivica, padding, i od sadržaja.

Ovaj model nam dozvoljava da pomoću CSS bolje sagledavamo formatiranje, definišemo raspored, umećemo ivice, praznine i sl. u odnosu na druge elemente.



## The Back of the New \$20

Have you seen the "Series 2004 \$20 Notes"? The U.S. Treasury has rolled out yet another revamp of the U.S. twenty dollar bill in an effort to stop those pesky counterfeiters once and for all. It features high-tech fake-busting elements like a watermark, a security thread, and color-shifting ink. It also features *crappy design*.

I'm not going to concern myself here with a critique of the front of the bill (my friend Jeff says "it looks like spilled on it"). It's the *back* of the note that's driving me crazy.

### Too Many 20s



In particular, it's all those little 20s haphazardly sprinkled in the white space. They are nails-on-a-chalkboard design series.

Are they supposed to be another security feature? ("They'll NEVER be able to duplicate this \$20... look at those 20s... they're all *GIVER* the place!") Did they let a summer intern at the Bureau of Engraving and Printing design it? ("Hey, let Jimmy try it!") Were they concerned the \$20 bill might be confused with a \$10? ("What this 20 needs is a LOT more 20s.")

### Connect-the-Dots

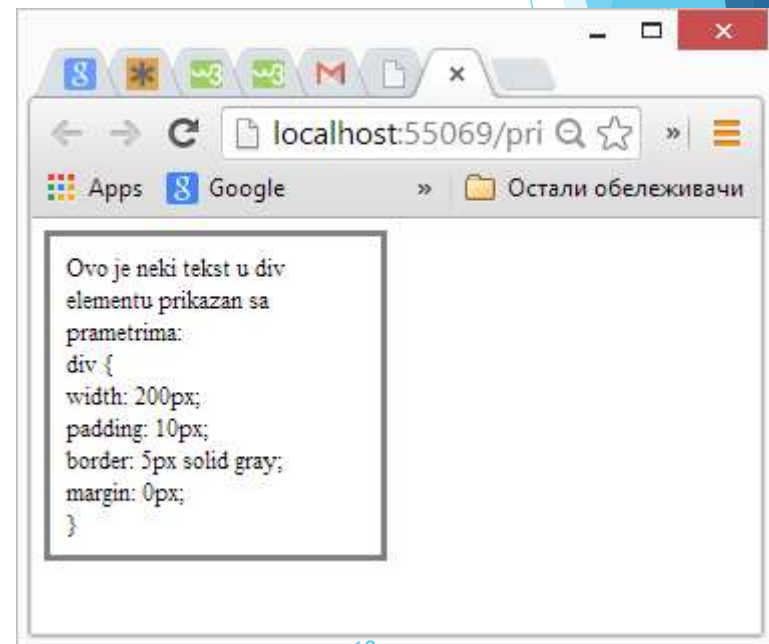
There must be more to it. My theory: the new 20s contain subliminal connect-the-dots messages, like tiny constellations. So, perhaps the 20s connect to form a secret message designed to stimulate the economy ("SPEND MORE") or boost patriotism ("WE'RE NO.1").

I'm not sure I've successfully cracked the code, so I'm asking for your help. I encourage you all to get a new \$20 bill, connect the dots to find the message on the back (pencil is best), and mail it to me for review. Together, we can get to the bottom of this.

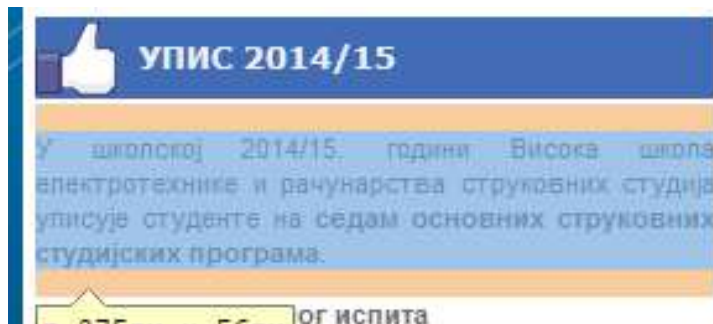
ncern myself  
the *back* of the

# Primer 2

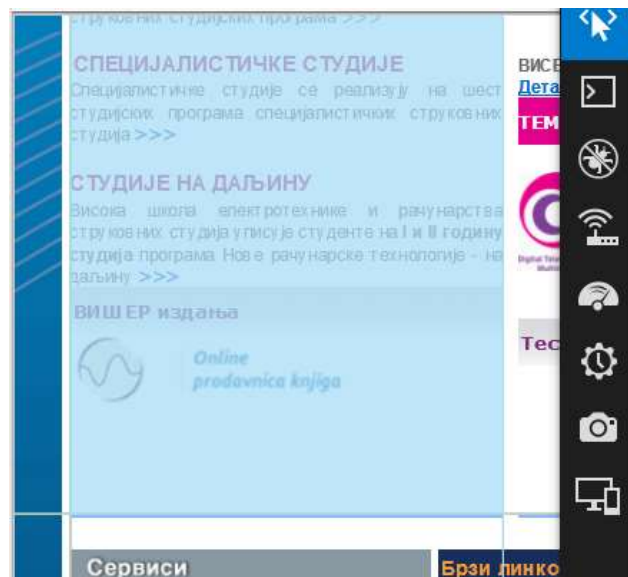
- ▶ `<html>`
- ▶ `<head>`
- ▶ `<style>`
- ▶ `div {`
- ▶ `width: 200px;`
- ▶ `padding: 10px;`
- ▶ `border: 5px solid gray;`
- ▶ `margin: 0px;`
- ▶ `}`
- ▶ `</style>`
- ▶ `</head>`
- ▶ `<body>`
- ▶ `<div class="ex">Ovo je ne. . . .`
- ▶ `. . . .`
- ▶ `</div>`
- ▶ `</body>`
- ▶ `</html>`
- ▶ Kolika je **ukupna** širina / visina „div“ oblasti?



# Box model kroz pomoćne razvojne alate



```
<td valign="top">
 <p align="justify">...</p>
 <p align="justify">...</p>
 <p align="justify">...</p>
 <n align="justify"> </n>
```



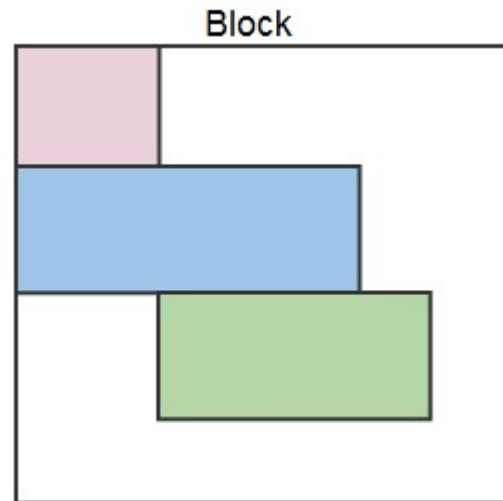
```
<td width="934" class="td_belaPozadina" rowspan="1" colspan="1" sizcache="2" sizset="0">
 <!-- Tabela sa promenljivim sadrzajem -->
 <table width="934" bgColor="#ffffff" border="0" cellSpacing="0" cellPadding="0" sizcache="2" sizset="0">
 <tbody sizcache="2" sizset="0">
 <tr>...</tr>
 <tr sizcache="2" sizset="0">
 <td width="282" valign="top" rowspan="2" colspan="1">...</td>
 <td width="10" class="putokazTekst" valign="top" rowspan="2" colspan="1">...</td>
 <td width="250" valign="top" rowspan="2" colspan="1">...</td>
 <td width="10" rowspan="6" colspan="1">...</td>
 <td width="320" align="center" valign="top" rowspan="6" colspan="1" sizcache="2" sizset="0">...</td>
 </tr>
 <tr>...</tr>
 <tr>...</tr>
 <tr>...</tr>
```

# Blok elementi

- ▶ Neki elementi se uvek prikazuju u novom redu počev od leve ivice prozora čitača tj. u novom redu a u zavisnosti od veličine sadržaja imaju širinu.
- ▶ Ovi elementi se nazivaju **blok** elementima.

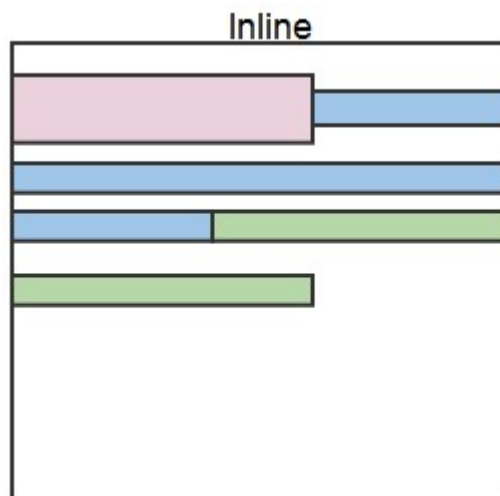
- ▶ Primeri:

- `<div>`
- `<form>`
- `<h1>`
- `<p>`
- `<ul>`
- `<li>`



# Linijski elementi

- ▶ Neki elementi se uvek prikazuju kao reči jednog teksta tj. tako što se nadovezuju kroz linijski prikaz susednih elemenata.
- ▶ Ovi elementi su poznati kao **linijski** (engl. inline) elementi.
- ▶ Na primer:
  - ▶ `<a>`
  - ▶ `<b>`
  - ▶ `<em>`
  - ▶ `<img>`
  - ▶ `<span>`



# Primer

```
<!DOCTYPE html>
<html>
<head>
 <style>
 p {
 display: inline;
 }
 </style>
</head>
<body>

 <p>This is a paragraph.</p>
 <p>This is a paragraph.</p>
 <p>This is a paragraph.</p>
 <p>This is a paragraph.</p>
 <p>This is a paragraph.</p>

</body>
</html>
```

## Napomena:

- Svojstvo display definiše načine prikaza:
- Osnovne vrednosti su:  
`inline`, `block`, `inherit`, `flex` (HTML5).

This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph. This is a paragraph.

# Formiranje selektora

16.05

23

# Vrste selektora

- ▶ Selektor definiše elemente na koje se stil primenjuje. Definisanje tih elemenata se vrši koristeći:
- ▶ **Univerzalni selektor**
  - ▶ `* {color:blue; }`
  - ▶ Odnosi se na sve elemente
- ▶ **Tip elementa**
  - ▶ `H2 { color:blue; }`
  - ▶ Odnosi se na H2 elemente. U opštem slučaju stoji tip elementa.



## ➤ Selektor po ID

- ▶ Da bi se stil primenio na samo jedan određen element, taj element mora biti jedinstveno definisan, dakle neophodno je da bude označen **ID**.
- ▶ Upotreba atributa ID uz element, na primer `id="glavniNaslov"` znači da element ima jedinstvenu vrednost ID. Pomoću ID vrednosti se vrši referisanje na određeni elemenat.
- ▶ Slično načinu kako element `<a>` referiše sidro, tako selektor koristi ID elementa tako da se prvo stavi `#` pa vrednost ID atributa elementa na koji se stil odnosi. Na primer:
- ▶ `#glavniNaslov { color: red; }`
- ▶ `#zakljucak { color:blue; }`
- ▶ Prvi selektor se odnosi na element čiji je `id="glavniNaslov"`, dok drugi označava primenu na element čiji je `id="zakljucak"`.
- ▶ `<h1 id="glavniNaslov">...</h1>`
- ▶ `<p id="zakljucak">...</p>`

## ➤ Selektor po klasi

- ▶ Selektor koji treba da označi primenu stila na grupi elemenata koristi istu vrednost za atribut klase na tim elementima. Na primer, ako nekoliko elemenata ima zajedničko svojsvo **class**:
- ▶ `<h3 class="podNaslovi">. . .</h3>`
- ▶ tada se pomoću tog svojstva i primenom selektora po klasi može izvršiti foramtiranje svih elemenata iste klase, na primer:
- ▶ `.podNaslovi{ color:red; }`
- ▶ Obratiti pažnju da se ispred naziva klase stavlja tačka. **Tačka je obavezna.**

# Kombnacija selektora

- ▶ Selektori mogu koristiti kombinaciju nekoliko vrsta označavanja. Na primer, može se kombinovati tip elementa sa nekim od ova dva tipa.
- ▶ Na primer: Ime klase se može koristiti zajedno sa tipom elementa. U tom slučaju se pravilo primenjuje na elemente tog tipa iz označene klase.
- ▶ `H2.podNaslovi{ color:red; }`
- ▶ Ili se tip može koristiti sa id tipom selektora
- ▶ `H1#naslov{ color:red; }`

# Više klasa za isti element

- ▶ Za istovremenu primenu više definisanih klasa na jedan element navode se klase razdvojene razmakom. Na primer:

```
<style>
 .bojateksta {
 color: blue;
 }
 .bojapozadine{
 background-color:lightblue;
 }
</style>
=====
<body>
 <p class="bojateksta bojapozadine" >Ovo je paragraf
 namenjen testiranju primene više klasa...</p>
```

# Selektor za dete element

- ▶ Koristeći CSS selektore moguće je odabrati elemente koji su u određenoj relaciji sa drugim elementima.
- ▶ Neke relacije su: roditelj, dete, preci, potomci ili su na istom nivou u hijerarhijskoj strukturi (blizanci).
- ▶ **Selektor za sve dete elemente:**
- ▶ Sintaksa je: *selektorRoditelj* > *selektorDete*. Na primer:
- ▶ `div#zakljucak > p { color:blue; }`
- ▶ Selektor se odnosi na paragrafe koji su dete elementi u odnosu na div sa id= "zakljucak"

▶ Selektori za potomke:

- ▶ Elementi u HTML stranici su hijerarhijski uređeni poštujući na taj način strukturu sadržaja u dokumentu. Primena selektora takođe je hijerarhijska.
- ▶ Zadatak: Crvenom bojom naglasiti tekst uokviren sa elementom `em` u naslovu `h1`.
- ▶ `<h1 class="podNaslovi">Podnaslov1 u grupi <em>podNaslovi<mark>br.<em>1</em></mark></em></h1>`

▶ Pokušaj 1:

- ▶ `h1 > em { color: red }`
- ▶ Promena će se odraziti i na `em` elemente čiji je roditelj `h1`. Problem je što se neće važiti za dalje podnivoje.

▶ Pokušaj 2:

- ▶ `h1 { color: blue }`  
`em { color: red }`
- ▶ Promena će se odraziti na sve `em` elemente, a ne samo na `em` element unutar `h1` elementa.

▶ Rešenje:

- ▶ `h1 em { color: red }`, Važi i za slučaj kada je `em` potomak elementa `h1` ne samo dete:

▶ **Selektor za element istog nivoa odmah iza elementa:**

- ▶ Blizanci nazivamo element koji imaju istog roditelja. Za označavanje blizanca koristi se oznaka +
- ▶ Zadatak: Crvenom bojom naglasiti tekst u paragrafu **p** koji su na istom nivou iza elementa **div**.

▶ Rešenje:

▶ `div + p { color: red }`

▶ **Selektor za sve elemente blizance:**

- ▶ Svi blizanci istog nivoa se označavaju karakterom ~
- ▶ Zadatak: Crvenom bojom naglasiti tekst u paragrafima **p** koji su na istom nivou u elementu **div**.

▶ Rešenje:

▶ `div ~ p { color: red }`

# Selektori na osnovu atributa

## ▶ **[atribut]**

- ▶ Izbor elemenata se može izvršiti na osnovu atributa u tim elementima odnosno na osnovu vrednosti tih atributa. Na primer:

- `a[href] {`
- `text-decoration: none;`
- `}`

- ▶ Rezultat je da, gde god se pojavljuje *href* atribut u elementu *a*, link neće biti podvučen.

## ▶ **[atribut=val]**

- ▶ Drugi način je da se selektuje ne samo atribut, nego i vrednost tog atributa. Na primer:

- `a[href="http://www.viser.edu.rs"] {`
- `text-decoration: none;`
- `}`

- Rezultat je da, gde god se pojavio *href* atribut u nekom elementu, a ukazuje na školu, link neće biti podvučen.



## ▶ **[atribut~=val]**

- ▶ Izbor elemenata se vrši na osnovu vrednosti atributa. Samo oni elementi čija vrednost atributa sadrži navedenu vrednost **kao jednu od reči (odvojenu razmaknicom)** odgovaraju selektoru. Na primer:

- `a[naziv~="upis"] {  
 text-decoration: none;  
}`
- Rezultat primene su elementi sa atributom *naziv* koji imaju vrednost `"upis"`, `"junski upis"`, `"upis septembarski"`

## ▶ **[atribut |= val]**

- ▶ Izbor elemenata se vrši na osnovu vrednosti atributa i to tako da atribut **ima navedenu vrednost ili da vrednost počinje tom reči sa crticom iza nje**. Praznina znači da se neće selektovati. Ne uključuje mogućnost da ima više reči. Na primer:

- `a[naziv|="main"] {  
 color: red;  
}`
- Rezultat primene su elementi sa atributom *naziv* a deluje samo ukoliko a element ima atribut naziv sa vrednosti `"main"`, `"main-1"`, dakle ili je vrednost baš `"main"` ili počinje sa `"main-"`.
- Neće odgovarati vrednost `"main main-1"` mada obe reči zadovoljavaju posebno kriterijum.

► `[atribut ^= val]`

- Izbor elemenata se vrši na osnovu vrednosti atributa i to tako da **atribut ima navedenu vrednost ili počinje sa tom vrednošću**. Ako ima više reči, vrednost atributa se tumači kao jedna vrednost tj. samo ako počinje na pravi način onda je selektovano. Na primer:

- `a[naziv ^= "ns13"] {`
- `color: red;`
- `}`
- Rezultat primene su elementi sa atributom *naziv* a deluje samo ukoliko a element ima atribut naziv sa vrednostima **"ns13"**, **"ns133"**, **"ns13 ns133"**.

► **[atribut \$= val]**

- Izbor elemenata se vrši na osnovu vrednosti atributa i to tako da atribut **ima navedenu vrednost ili se završava sa tom vrednošću**. Na primer:

- `a[naziv$=".png"] {`
- `color: red;`
- `}`

- Rezultat primene su elementi sa atributom *naziv* a deluje samo ukoliko a element ima atribut naziv sa vrednostima `"slika1.png"`, `"slika2.png"`.

► **[atribut \*= val]**

- Izbor elemenata se vrši na osnovu vrednosti atributa i to tako da atribut **ima ili sadrži navedeni podstring**. Na primer:

- `a[naziv*="viser"] {`
- `color: red;`
- `}`

- Rezultat primene su elementi sa atributom *naziv* a deluje samo ukoliko a element ima atribut naziv sa vrednostima `"www.viser.edu.rs"`.

# Primena više selektora atributa

- ▶ Primenjuju se kada je potrebno da se stil primeni za nekoliko atributa jednog elementa ili za nekoliko različitih testova istog atributu. Na primer:
- ▶ `span[hello="Cleveland"][goodbye="Columbus"] { color: blue; }`

## Primena grupe selektora

- ▶ Grupe selektora se pišu tako što se pre narednog stavi **zapeta** i tako dalje. Na primer:
- ▶ `h1,h2 { color:red; }`
- ▶ Boja će biti primenjena na naslove h1 i h2.
- ▶ Osim primene grupe selektora, možete koristiti i bilo koje kombinacije navedenih načina označavanja selektora.

# Pseudo -elementi -selektori

- ▶ Uvod
- ▶ Mada smo prikazali veliki broj različitih selektora, ipak pomoću do sada objašnjenih tehnika stilizacije dokumenta nije moguće uraditi selekcije u nekim slučajevima. Na primer, ni jedan element ne referiše prvu liniju paragrafa. Napomena autora: zapazite da prva linija paragrafa zavisi od širine prozora čitača. Za takve slučajeve je uveden koncept **pseudoelemenata** i **pseudoklasa**.
- ▶ Ovim tehnikama kreiraju se apstrakcije koje nisu obuhvaćene stablom dokumenta. Dakle, omogućava se dizajnerima primena stila na sadržaj koji ne postoji u izvornom dokumentu (na primer :before daje pristup u kontekstu).

# Pseudoelementi

- ▶ Pseudoelementima se označava deo elementa.

Selektor	Opis
<b>::first-line</b>	Prva linija u prikazu
<b>::first-letter</b>	Prvo slovo
<b>::before</b>	Ubacivanje sadržaja i stila ispred selektora
<b>::after</b>	Ubacivanje sadržaja posle selektora
<b>::selection</b>	Selektuje deo elementa koji je selektovan od strane korisnika

- ▶ Sintaksa pseudoelemenata
- ▶ `selector::pseudo-element {  
    property:value;  
}`
- ▶ Zapazite notaciju sa dve dvotačke u odnosu na jednu dvotačku - `::first-line` u odnosu na `:first-line`
- ▶ Dvostruka dvotačka je zamenila jednostruku u pseudoelementima u CSS3 standarda. Cilj dvostruke dvotačke je razdvajanje pseudo-klasa i pseudo-elemenata.
- ▶ Jedna dvotačka je korišćena i za pseudo-kalse i za pseudo-elemente u standardima CSS2 odnosno CSS1.

## ▶ `::first-line`

▶ Definiše stil koji se odnosi na prvu liniju teksta.

```
▶ first-line {
 color: #ff0000;
 font-variant: small-caps;
}
```

## ▶ `::first-letter`

▶ Definiše stil prvog slova u tekstu.

▶ Primer:

```
▶ p::first-letter {
 color: #ff0000;
 font-size: xx-large;
}
```



## ▶ `::before` `::after`

- ▶ Omogućava ubacivanje nekog sadržaja pre ili posle određenog elementa.

- ▶ `a[naziv*=novo]::after` {  
    `content: url(novo.gif);`  
}

## ▶ `::selection`

- ▶ Sledeća svojstva mogu biti primenjena: `color`, `background`, `cursor`, and `outline`.

- ▶ Primer:

- ▶ `::selection` {  
    `color: blue;`  
    `background: yellow;`  
}

# Pseudo-klase

- ▶ Slično pseudo-elementima mogu se koristiti unapred definisane klase. Radi se o vrstama selektora za znakove koji nisu opipljivi i koje ne možete da „ručno“ označite.

Sintaksa:  
selector:**pseudo-class** {  
    property:value;  
}

- ▶ Na primer:
- ▶ `div > p:first-child { color: green; }`

- ▶ Važi za:

```
<div class="note">
 <p> Prvi paragraf.
</div>
```

- ▶ Ne važi za:

```
<div class="note">
 <h2>Note</h2>
 <p> Prvi paragraf.
</div>
```

# Pregled pseudo-klasa

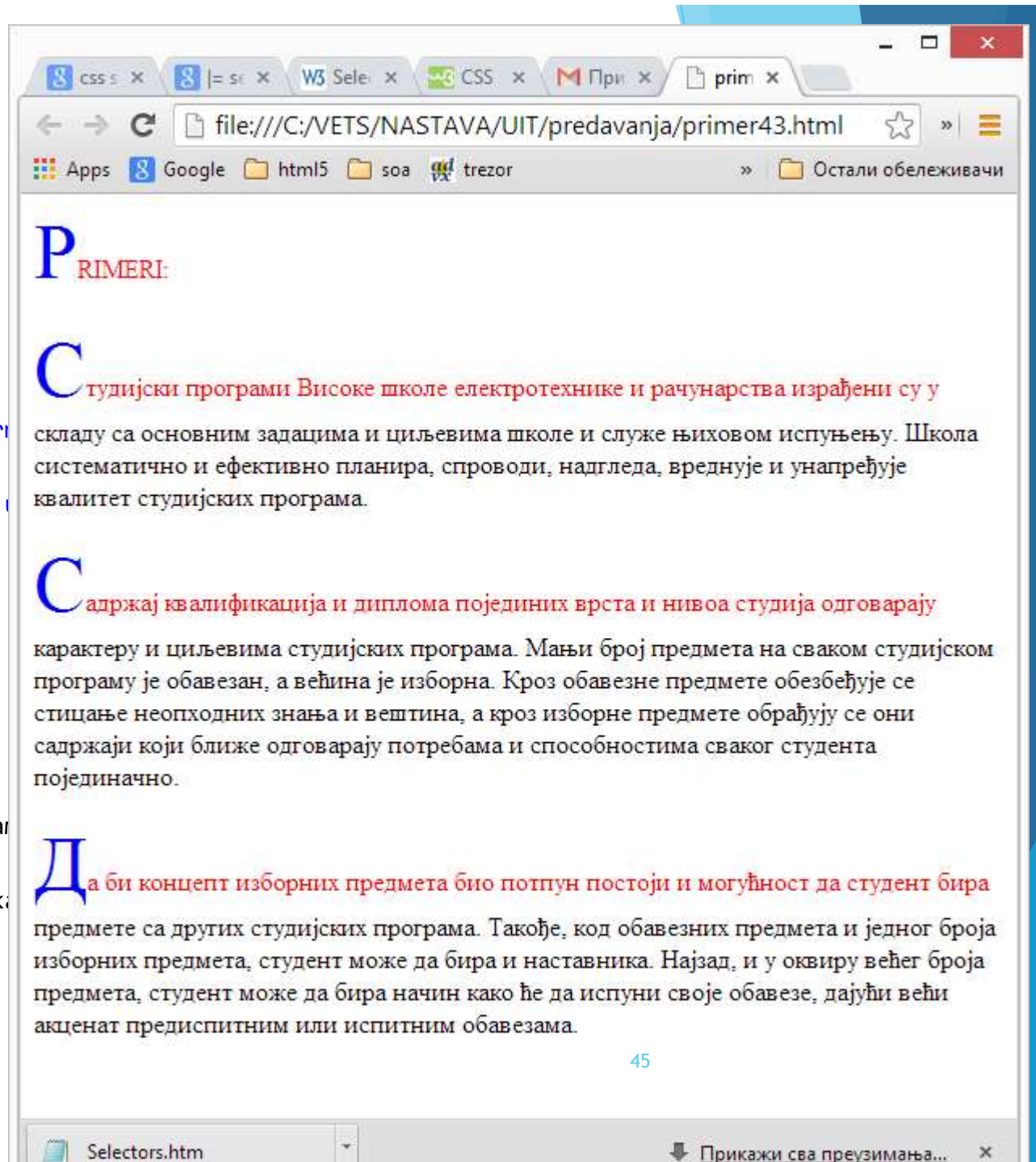
Selektor	Opis
<b>:first-child</b>	prvo dete element
<b>:last-child</b>	poslednje dete element
<b>:nth-child(n)</b>	n-to dete element
<b>:nth-last-child(n)</b>	n-to dete od poslednjeg dete elementa
<b>:hover</b>	Selektuje stanje miša iznad
<b>:empty</b>	Selektuje sve elemente koji su prazni
<b>:link</b>	
<b>:visited</b>	Selektuje stanje veza, dakle ne element već stanje!
<b>:active</b>	
<b>:hover</b>	
<b>:checked</b>	Selektuje svaki čekirani ulazni element forme
<b>:disabled</b>	Selektuje svaki koji ima stanje disabled
<b>:enabled</b>	Selektuje svaki koji je omogućen za unos
<b>:focus</b>	Selektuje onaj koji ima fokus

# Pseudoselektori za stanja linkova

- ▶ Poseban slučaj je da se selektor odnosi ne na element već na stanje elementa. Tako se za definisanje stanja linkova koristi se sledeci stil:
- ▶ A :naziv { color:red; }
  - naziv ima moguće vrednosti:
    - *link* - odnosi se na veze na koje nije kliknuto niti je miš iznad njih
    - *visited* - odnosi se na veze koje su korišćene
    - *focus* - ako je veza u fokusu
    - *hover* - kada se na vezu pokaže
    - *active* - kada se na njih pritisne

# Primer 3

```
<head>
 <style>
 div p:first-line {
 color:red;
 }
 p:first-letter {
 color: blue;
 font-size: 3em;
 font-weight: normal;
 }
 p#naslov {
 text-transform: uppercase;
 }
 </style>
</head>
<body>
 <div>
 <p id="naslov">
 Primeri:
 </p>
 </div>
 <div>
 <p>
 Студијски програми
 </p>
 . . .
 <p>
 Садржај квалификација
 </p>
 <p>
 Да би концепт
 </p>
 </div>
</body>
</html>
```



# Primer 4

```
<html>
<head>
 <style>
 input:focus {
 background-color: yellow;
 }
 </style>
</head>
<body>
```

<p>Pozadina postaje žuta kada se element nadje u fokusu:</p>

```
 <form>
 First name: <input type="text"
name="firstname" />

 Last name: <input type="text"
name="lastname" />
 </form>
```

<p>Fokus radi na većini novih čitača, ne i na starijim.</p>

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

Pozadina postaje žuta kada se element nadje u fokusu:

First name:

Last name:

Fokus radi na većini novih citaca, ne i na starijim.

# Primer 5

```
> <html>
> <head>
> <style>
> form:before {
> content: "<hr>";
> }
> p:before {
> content: ">>";
> }
>
> blockquote:before {
> content: "##";
> font-size: 24px;
> color: purple;
> }
>
> blockquote:after {
> content: "#";
> font-size: 24px;
> color: purple;
> }
> </style>
> </head>
> <body>
>
> <p>Pozadina postaje žuta kada se element nadje u fokusu:</p>
> <form>
> First name: <input type="text" name="firstname" />

> Last name: <input type="text" name="lastname" />
> </form>
>
> <p>Fokus radi na većini <blockquote>xxxx xxx xxxxxxx xxxx xxxx xxx xxxxxx
> xxxx xxxx xxx xxxxxxx xxxx xxxx xxx xxxxxx xxxx</blockquote> novih čitača,
> ne i na starijim.</p>
>
> </body>
> </html>
```

>>Pozadina postaje žuta kada se element nadje u fokusu:

<hr> First name:   
Last name:

>>Fokus radi na većini

## xxxx xxx xxxxxxx xxxx xxxx xxx xxxxxx xxxx  
xxxx xxx xxxxxxx xxxx xxxx xxx xxxxxx xxxx#

novih čitača, ne i na starijim.

>>