

# Razvoj programske podpore za web

- predavanja -  
2021./2022.

---

## 3. CSS

1/2

# Creative Commons



- slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **prerađivati** djelo



- pod sljedećim uvjetima:

- **imenovanje:** morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno:** ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima:** ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



*U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela.*

*Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.*

*Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.*

*Tekst licence preuzet je s <http://creativecommons.org/>*

**"If you can't make it good,  
at least make it look good."**

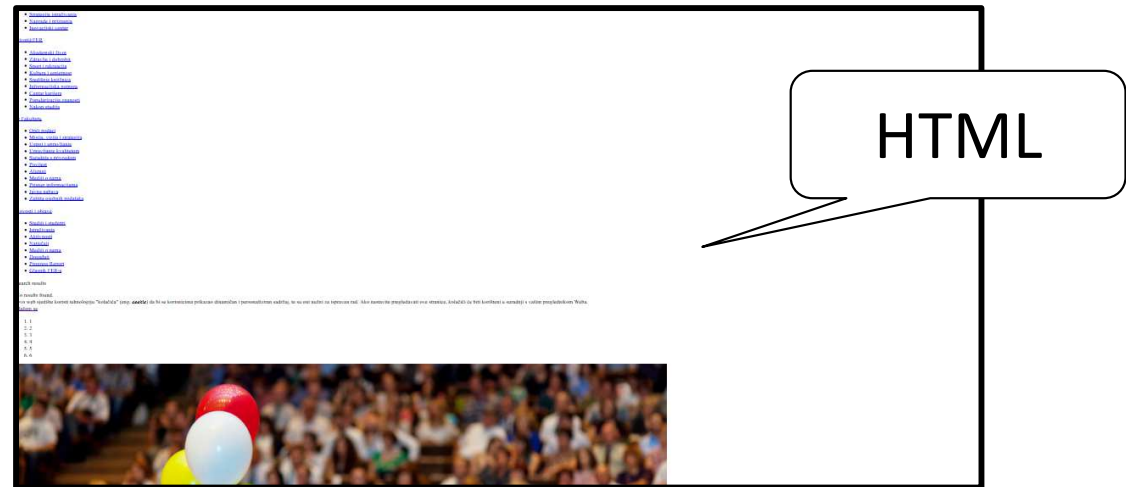
Bill Gates

# CSS - *Cascading Style Sheets*

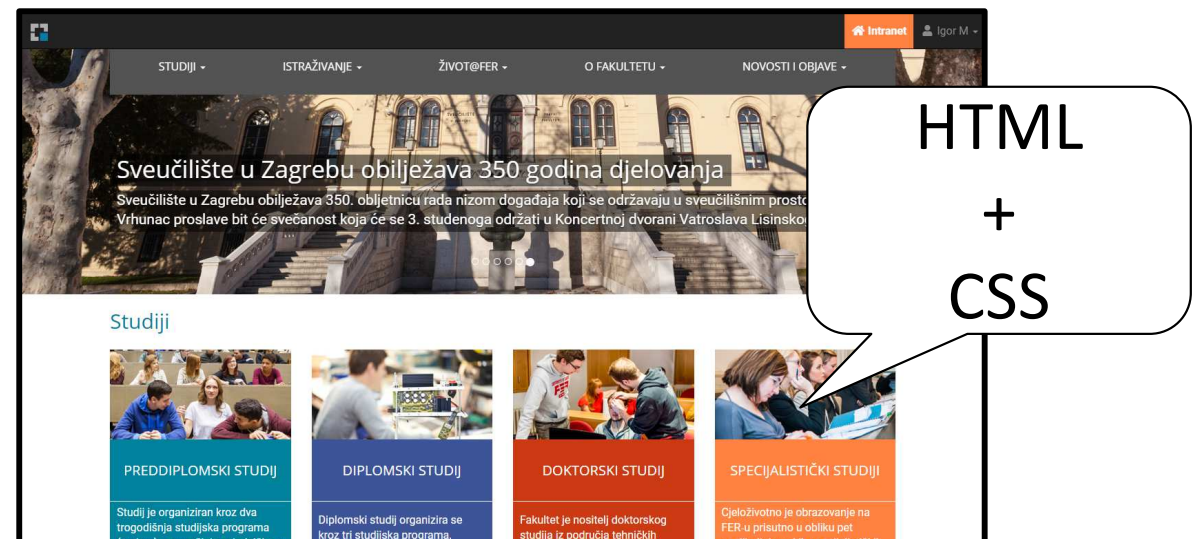
- CSS je jezik kojim se opisuje **prezentacija** dokumenta pisanog u HTML-u.

<https://www.fer.unizg.hr/>

- Razdvajanje:
  - Sadržaja (HTML)



- Prezentacije (CSS)
  - Smještaj
  - Boja
  - Fontovi
  - ...



# CSS primjer ½: HTML

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
  <h1>Moja web stranica</h1>
  <p>Danas je lijep dan.</p>
</body>
</html>
```



# CSS primjer 2/2: HTML+CSS

```
<!DOCTYPE html><html>
<head>
  <style>
    body {
      background-color: lightyellow;
    }
    h1 {
      color: blue;
      text-align: center;
    }
    p {
      font-family: helvetica;
      font-size: 20px;
    }
  </style>
</head>
<body><h1>Moja web stranica</h1>
  <p>Danas je lijep dan.</p></body>
</html>
```

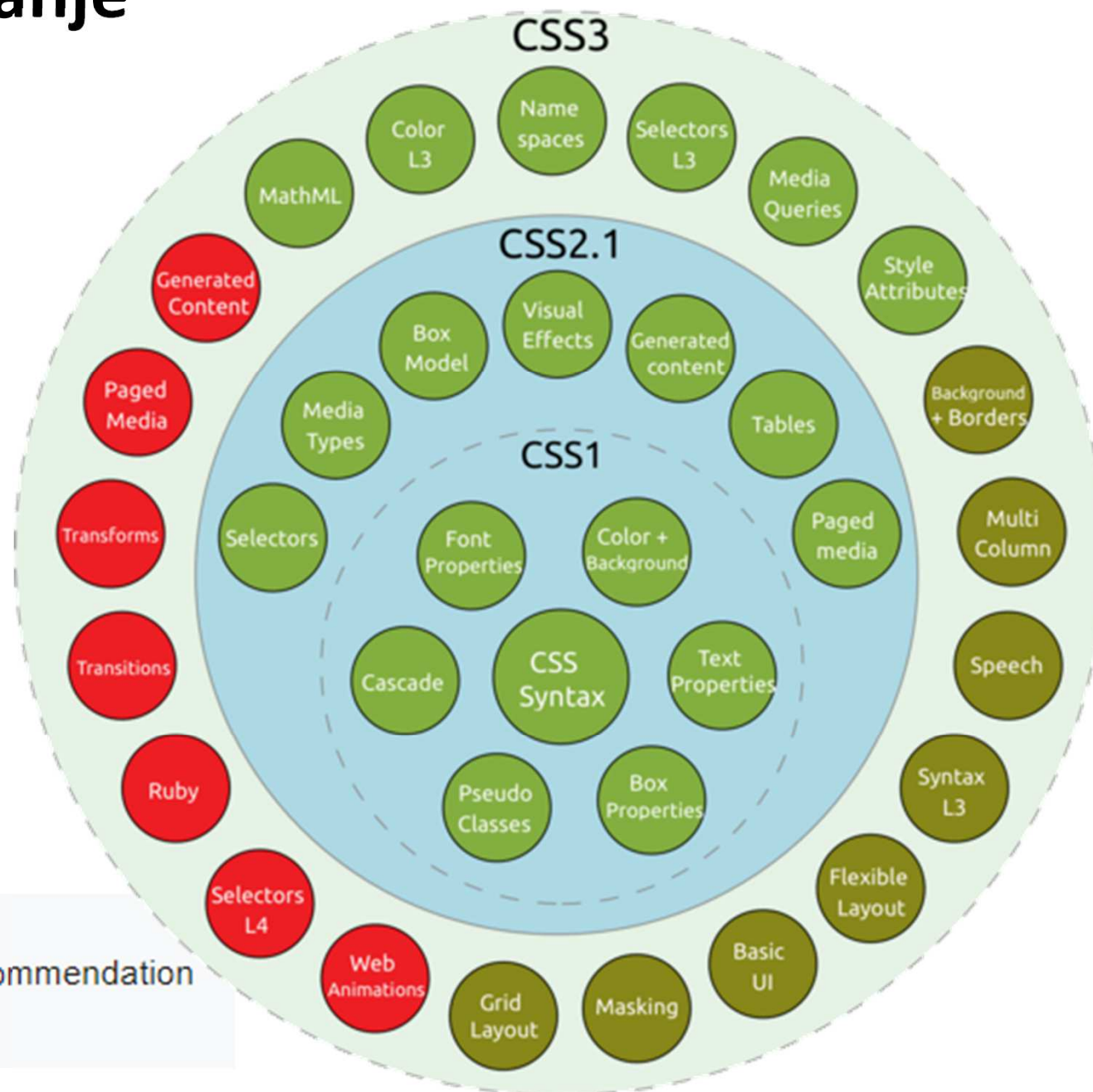


# CSS - povijest

- 1994: prvu verziju predložio Håkon Wium Lie
- 1996: **CSS1** (CSS level 1)
  - (uz značajan doprinos Berta Bosa).
- 1997: W3C osnovao **CSS Working Group** koji se brine za razvoj jezika
- 1998: **CSS2** (CSS level 2)
- 2011: **CSS 2.1** (CSS level2 revision 1).
  - Ali 2015 počeo rad na CSS 2.2
- U razvoju: **CSS3** (rad počeo još 1998.)
  - Podijeljen na više modula
- Neće biti CSS4, u razvoju: **CSS moduli 4**
  - Nastavak rada na modulima iz CSS3, npr. text3, color4

# CSS - trenutno stanje

- Podrška u preglednicima je raznolika
  - CSS2.1+CSS3
- Korisna stranica:  
<https://caniuse.com>



Taxonomy and status of CSS3 **modules**.

- Recommendation
- Candidate Recommendation
- Last Call
- Working Draft.

[#Cascading Style Sheets](#)



# Osnovna sintaksa

- pravila CSS-a se sastoje od selektora i proizvoljnog broja deklaracija (parovi *svojstvo-vrijednost*):

**selector { property: value; property: value; ...}**

declaration  
block start

declaration  
separator

declaration  
block end

- Npr.

selector

**h1 {**

**color: blue;**

**text-align: center;**

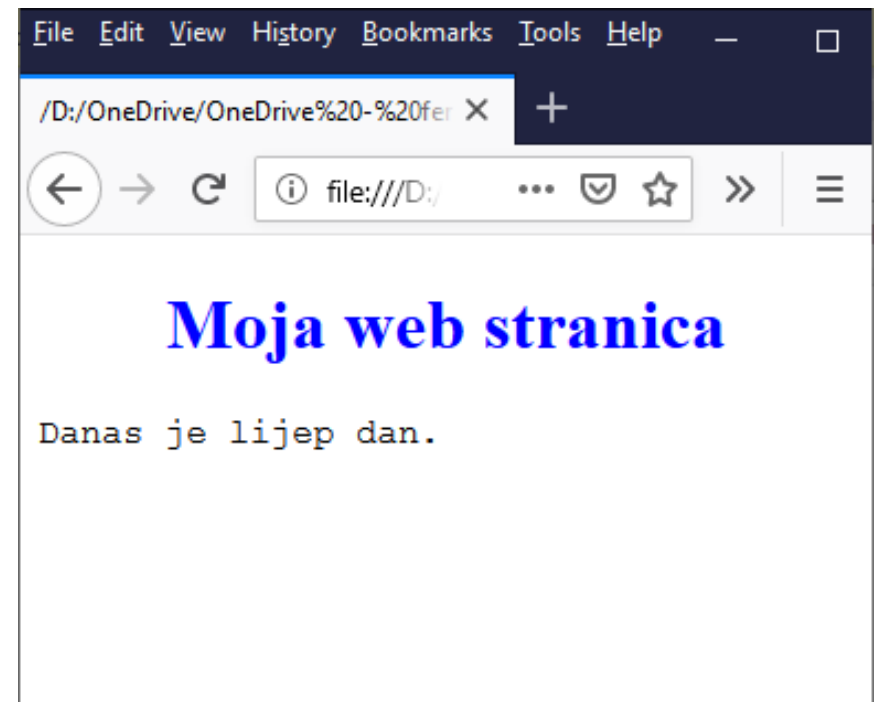
**} /\* komentar, nema // \*/**

declaration

## Gdje pisati CSS: opcija 1 - *inline*

- **Najlošija opcija** - CSS se može pisati unutar elementa koji se želi stilizirati
  - Narušava se načelo "*separation of concerns*", stil se isprepliće sa sadržajem - što nije dobro
  - Kako pišemo deklaracije unutar samog elementa, selektor nije potreban

```
<body>
  <h1 style="color:blue; text-align: center;">
    Moja web stranica
  </h1>
  <p style="font-family: Courier;">
    Danas je lijep dan.
  </p>
</body>
```

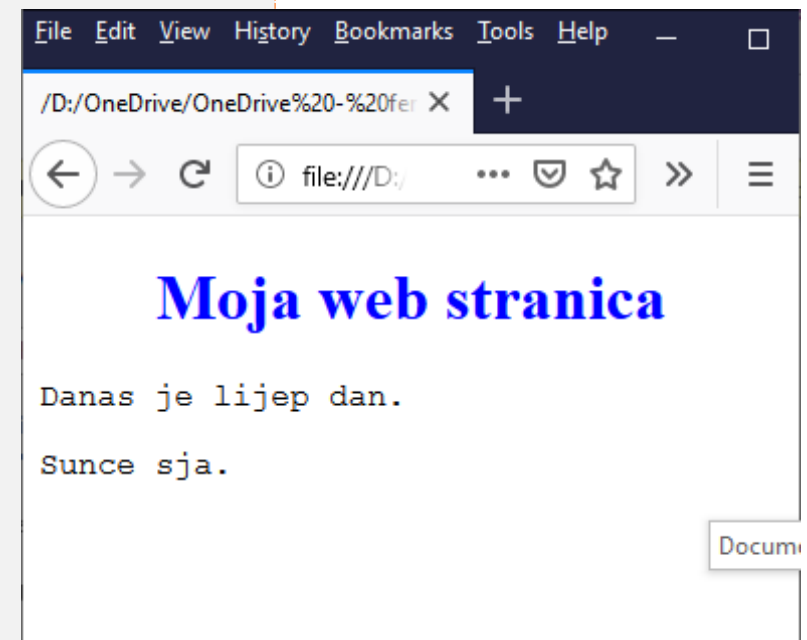


# Gdje pisati CSS: opcija 2 - unutar zaglavlja

- CSS se piše u zaglavlju unutar oznake `<style>`
  - Odabiremo elemente unutar dokumenta i postavljamo im stil

```
<!DOCTYPE html> <html>
<head>
  <style>
    h1 {color:blue; text-align: center;}
    p {font-family: Courier;}
  </style>
</head>
<body>
  <h1>Moja web stranica</h1>
  <p> Danas je lijep dan.</p>
  <p> Sunce sja.</p>
</body>
</html>
```

Želimo primijeniti na  
sve odlomke!  
Kako prethodnim  
pristupom?



# Gdje pisati CSS: opcija 3 - posebna datoteka

- CSS se piše u posebnoj datoteci koju referenciramo (uključujemo) iz zaglavlja pomoću oznake `<link>`
  - Smanjuje se HTML datoteka, CSS se može keširati
  - **Preporučeni način**

site.html

```
<!DOCTYPE html> <html>
<head>
  <link rel="stylesheet" href="site.css">
</head>
<body>
  <h1>Moja web stranica</h1>
  <p>Danas je lijep dan.</p>
  <p>Sunce sja.</p>
</body>
</html>
```

site.css

```
h1 {
  color: blue;
  text-align: center;
}
```



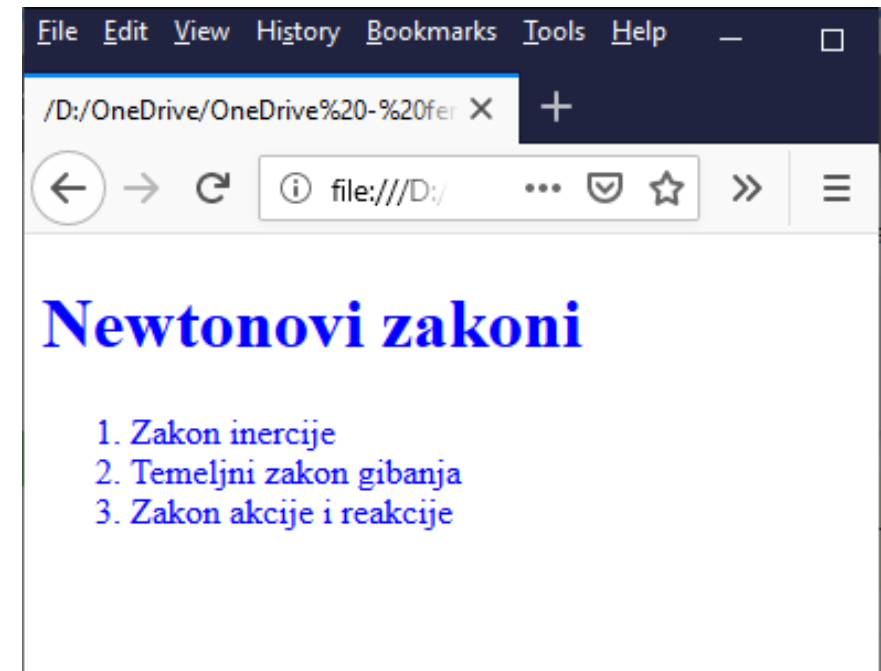
# ■ Selektori

- Selektori se koriste za odabir elemenata kojima se želi pridijeliti stil
- Moguće je odabrati nula, jedan ili više elemenata
- Dijelimo ih u pet kategorija:
  1. Jednostavni selektori (*simple selectors*)
  2. Atributni selektori (*attribute selectors*)
  3. Kombinirani selektori (*combinator selectors*)
  4. Selektori pseudo klasa (*pseudo-class selectors*)
  5. Selektori pseudo elemenata (*pseudo-elements selectors*)

# 1. Jednostavni selektori: (a) univerzalni selektor

- Odabire sve elemente na stranici!
  - Rijetko se koristi

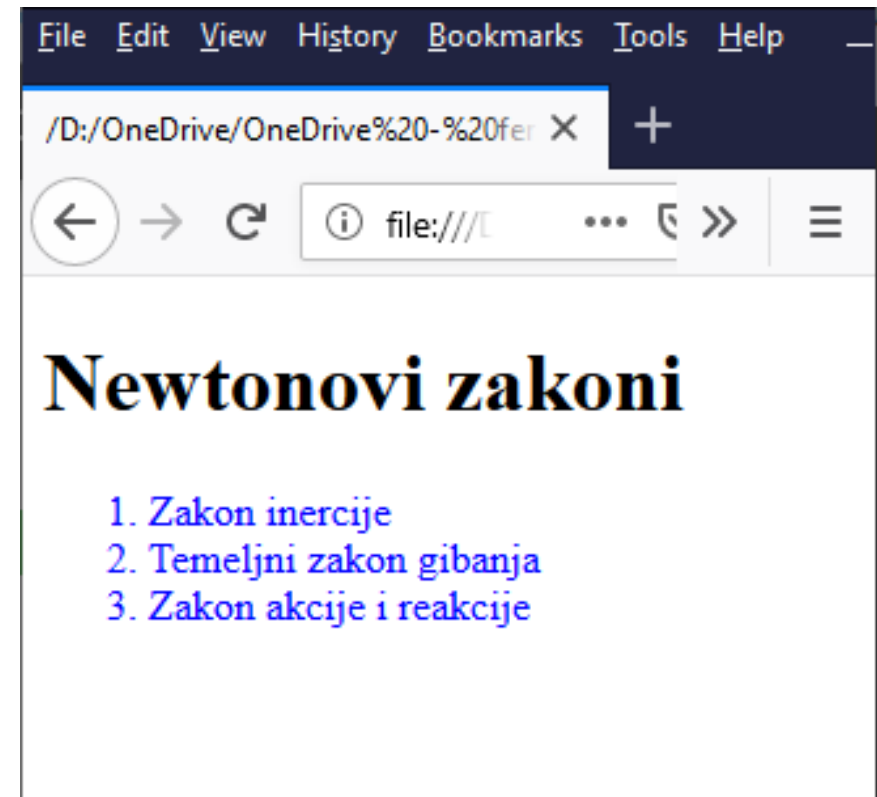
```
<!DOCTYPE html><html><head>
  <style>
    * { color: blue; }
  </style>
</head>
<body>
  <h1>Newtonovi zakoni</h1>
  <ol>
    <li>Zakon inercije</li>
    <li>Temeljni zakon gibanja</li>
    <li>Zakon akcije i reakcije</li>
  </ol>
</body></html>
```



# 1. Jednostavni selektori: (b) element selektor

- Odabire elemente na temelju imena elementa

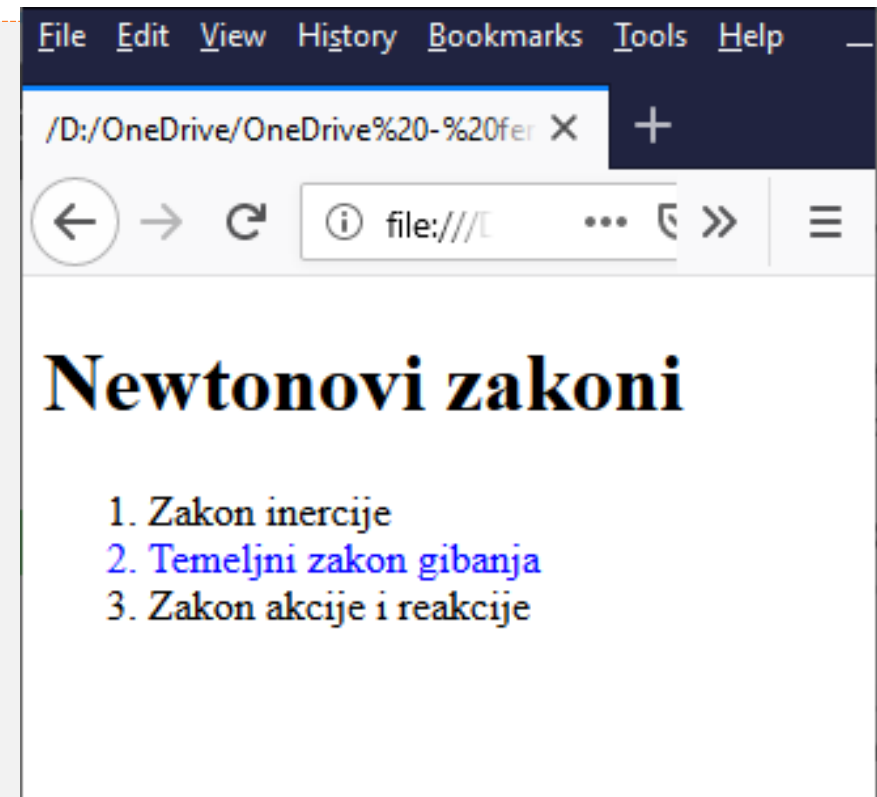
```
<!DOCTYPE html><html><head>
  <style>
    li { color: blue; }
  </style>
</head>
<body>
  <h1>Newtonovi zakoni</h1>
  <ol>
    <li>Zakon inercije</li>
    <li>Temeljni zakon gibanja</li>
    <li>Zakon akcije i reakcije</li>
  </ol>
</body></html>
```



# 1. Jednostavni selektori: (c) id selektor

- Odabire elemente na temelju **identifikatora**
  - Koristi se znak "#" i identifikator

```
<!DOCTYPE html><html><head>
  <style>
    #z2 { color: blue; }
  </style>
</head>
<body>
  <h1>Newtonovi zakoni</h1>
  <ol>
    <li id="z1">Zakon inercije</li>
    <li id="z2">Temeljni zakon gibanja</li>
    <li id="z3">Zakon akcije i reakcije</li>
  </ol>
</body></html>
```

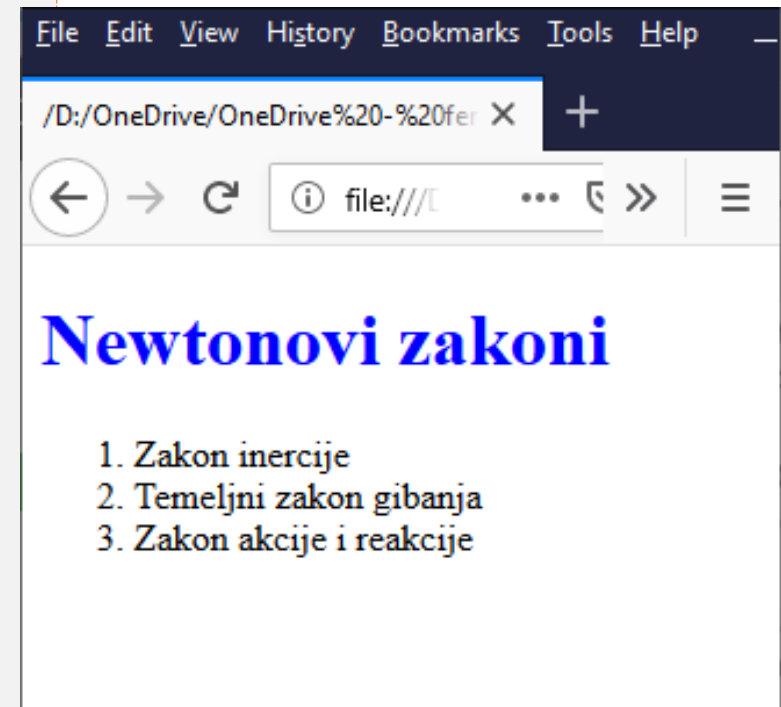




# 1. Jednostavni selektori: (d) class selector

- Odabire elemente na temelju klase (*class*)
  - Koristi se znak "." i ime klase

```
<!DOCTYPE html><html><head>
  <style>
    .naslov { color: blue; }
  </style>
</head>
<body>
  <h1 class="naslov">Newtonovi zakoni</h1>
  <ol>
    <li id="z1">Zakon inercije</li>
    <li id="z2">Temeljni zakon gibanja</li>
    <li id="z3">Zakon akcije i reakcije</li>
  </ol>
</body></html>
```



# 1. Jednostavni selektori: kombiniranje i grupiranje

- Moguće je kombinirati element i selektor class, npr.:
  - odabiru se samo elementi **h1** koji imaju klasu "naslov":

```
<style>  
  h1.naslov { color: blue; }  
</style>
```

- Moguće je grupirati više elemenata (ako npr. želimo postaviti isti stil), npr.:
  - odabrani su svi elementi **h1** i **li**:

```
<style>  
  h1, li { color: blue; }  
</style>
```

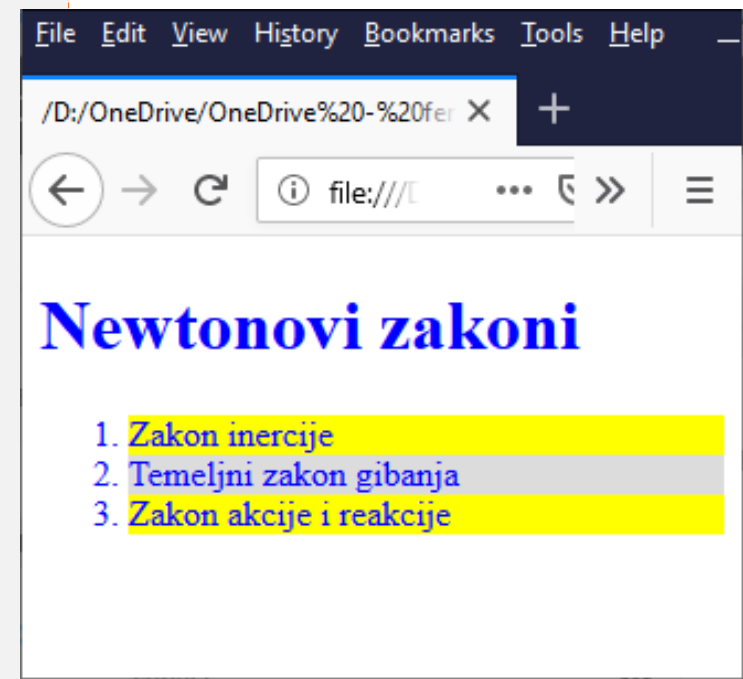
## 2. Atributni selektori

- Sintaksa: `element[atribut="vrijednost"]`
  - mogu se izostaviti i element i vrijednost

```
<!DOCTYPE html><html><head>
  <style>
    [id] { color: blue; }
    li[id] { background-color: yellow;}
    li[id="z2"] {background-color: gainsboro;}
  </style>
</head><body>
  <h1 id="n1">Newtonovi zakoni</h1>
  <ol>
    <li id="z1">Zakon inercije</li>
    <li id="z2">Temeljni zakon gibanja</li>
    <li id="z3">Zakon akcije i reakcije</li>
  </ol>
</body></html>
```

Pojednostavljeno,  
više opcija  
pogledati na:

[https://www.w3schools.com/Css/css\\_attribute\\_selectors.asp](https://www.w3schools.com/Css/css_attribute_selectors.asp)



## 3. Kombinirani selektori

- Jednostavne selektore je moguće kombinirati s četiri moguća kombinatora (*combinators*):

Operator (kombinator)	Primjer	Objašnjenje
_ (razmak)	div span	Odabire sve span element unutar div elementa, bez obzira koliko duboko
>	div>span	Odabire span elemente koji su djeca div elementa, tj. neposredno ispod
+	div+span	Odabire neposrednog susjeda iste razine ( <i>siblings</i> ), za ovaj primjer odabire span koji je neposredno nakon diva i imaju zajedničkog roditelja
~	div~span	Odabire susjede iste razine koji slijede prvog i ne moraju nužno biti neposredno iza prvog. Za ovaj primjer odabire sve span elemente koji slijede nakon diva i imaju zajedničkog roditelja.

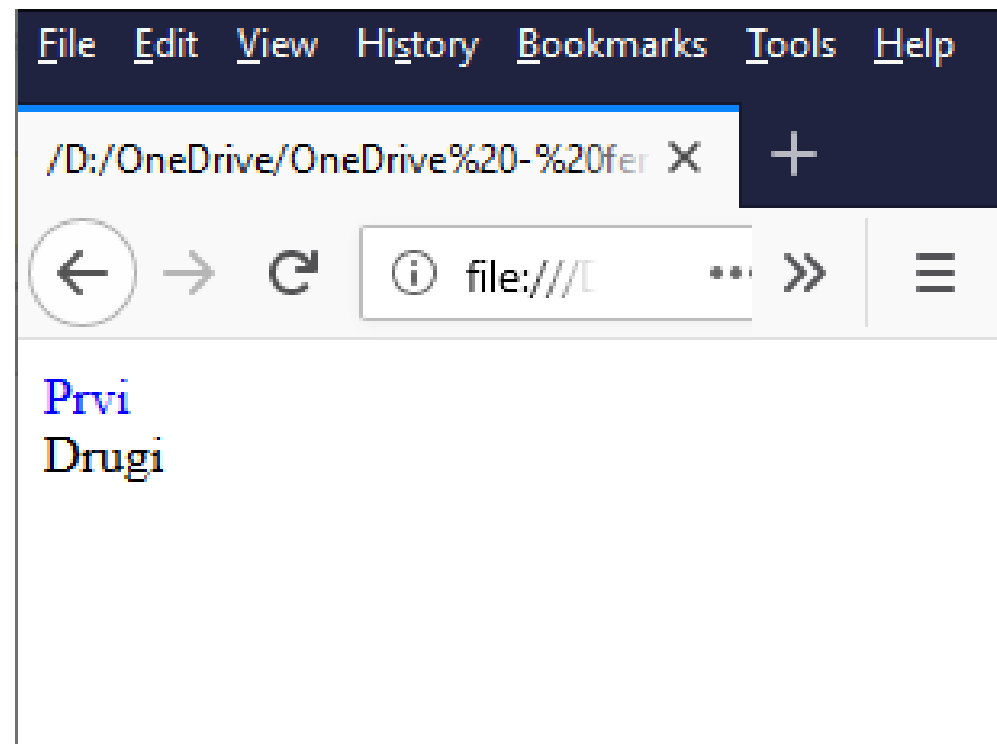
hijerarhija

susjednost

### 3. Kombinirani selektori - primjer

```
<!DOCTYPE html><html><head>
  <style>
    div > span {
      color: blue;
    }
  </style>
</head>
<body>
  <div>
    <span>Prvi</span>
    <section>
      <span>Drugi</span>
    </section>
  </div>
</body></html>
```

Što će biti ako  
obrišemo ">", tj.  
zamijenimo s  
razmakom?



## 4. Selektori pseudo-klasa

- **Pseudo-klase** se koriste kako bi se opisala posebna stanja elementa, npr. posjećena poveznica, fokusirani element, itd.

- Kombiniraju se sa selektorima:

**selektor:pseudo-klasa { ... }**

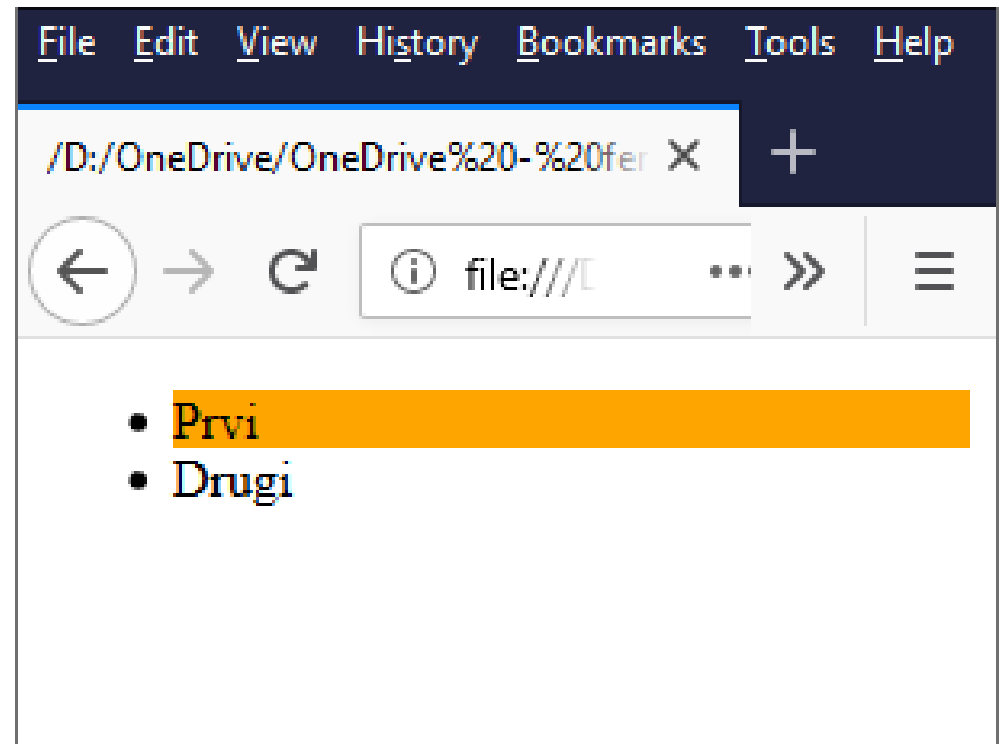
- Par primjera:

Selektor	Primjer	Objašnjenje
<b>:hover</b>	<b>div:hover</b>	Odabire div elemente nad kojima je pokazivač miša
<b>:first-child</b>	<b>li:first-child</b>	Odabire li element koji je prvo dijete svog roditelja
<b>:required</b>	<b>input:required</b>	Odabire sve input elemente koji imaju atribut required
itd. ima ih desetke, više na: <a href="https://www.w3schools.com/Css/css_pseudo_classes.asp">https://www.w3schools.com/Css/css_pseudo_classes.asp</a>		

## 4. Selektori pseudo-klasa: primjer

```
<!DOCTYPE html><html><head>
  <style>
    li:first-child {
      background-color: orange;
    }
  </style>
</head>

<body>
  <ul>
    <li>Prvi</li>
    <li>Drugi</li>
  </ul>
</body>
</html>
```



## 5. Selektori pseudo-elemenata

- **Pseudo-element** se koriste kako bi se stilizirali određeni dijelovi elementa, npr. prvo slovo, prvi redak.
- Kombiniraju se sa selektorima:  
**selektor::pseudo-element { ... }**
- Pseudo elementi:

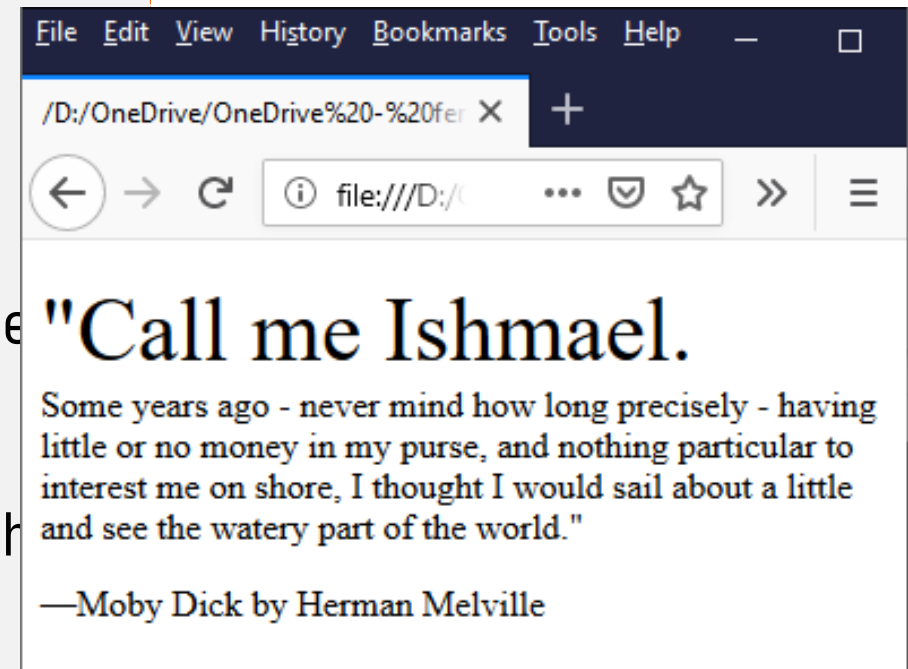
Selektor	Objašnjenje
<b>::after</b>	Ubacivanje sadržaja nakon odabranog elementa
<b>::before</b>	Ubacivanje sadržaja prije odabranog elementa
<b>::first-letter</b>	Odabire prvo slovo
<b>::first-line</b>	Odabire prvi redak
<b>::selection</b>	Odabire dio elementa koji je odabrao korisnik





## 5. Selektori pseudo-elemenata: primjer

```
<!DOCTYPE html><html><head>
  <style>
    p.quote::first-line {font-size: 40px;}
    p.quote::before,
    p.quote::after {content: "\"";}
  </style>
</head><body>
  <p class="quote">Call me Ishmael. Some
ars ago - never mind how long precisely
having little or no money in my purse,
nothing particular to interest me on sh
, I thought I would sail about a little
see the watery part of the world.</p>
<p>—Moby Dick by Herman Melville</p>
</body>
</html>
```

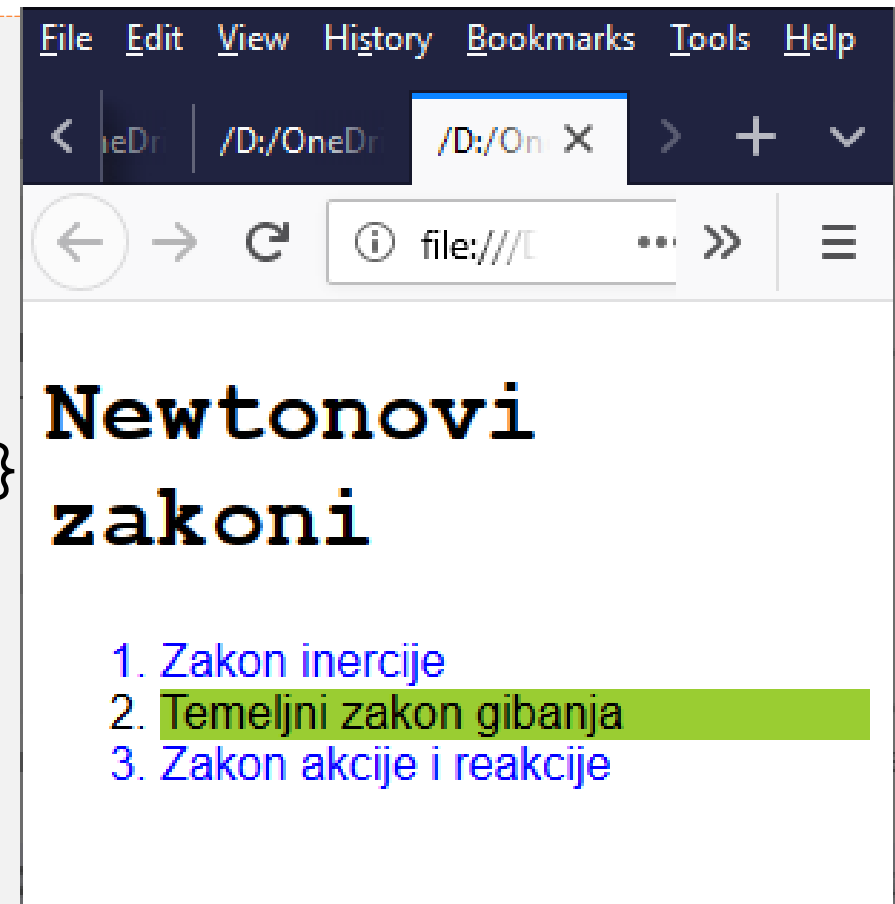


# ■ Podsjetnik: Chrome Developer Tools

- Chrome DevTools je skup alata koji su ugrađeni u preglednik Google Chrome
- Uvelike olakšavaju razvoj, moguće je mijenjati stranicu uživo, pratiti mrežni promet, debugirati javascript, itd.
- Demo: *inspect element*
- Za više informacija pogledati video:  
Build better sites faster with Chrome DevTools  
<https://www.youtube.com/watch?v=VYyQv0CSZOE>
- I drugi preglednici imaju slične alate, npr. Firefox Developer Tools

# Koji stil se primjenjuje?

```
<head>
<style>
  * { font-family: Courier; }
  #z2 { background-color: yellowgreen;
        color: black; }
  li { font-family: Arial; color: red; }
  li { color: blue; }
</style></head><body>
  <h1>Newtonovi zakoni</h1>
  <ol>
    <li id="z1">Zakon inercije</li>
    <li id="z2">Temeljni zakon gibanja</li>
    <li id="z3">Zakon akcije i reakcije</li>
  </ol>
</body>
```



# ■ Kaskadnost (1/3)

- Iz prethodnog primjera možemo neformalno zaključiti:
  1. Pravila se **kombiniraju**
    - Moguće je definirati više pravila koja odabiru neki element i onda se različita svojstva kombiniraju kaskadno
  2. Ista svojstva se **premošćuju** (*override*) **po nekom načelu**
- Preciznije, kaskadno (**CSS**) određivanje koje pravilo/svojstvo će se primijeniti uzima u obzir tri koncepta:
  1. **Izvor odnosno važnost** (*origin/importance*)
  2. **Specifičnost** (*specificity*)
  3. **Poredak koda** (*source order*)
- Na sljedeća dva slajda pogledajmo 1 i 2+3, a onda ćemo iskazati konačni algoritam

# 1. Izvor i važnost (*origin & importance*)

- pravila CSS-a mogu doći iz različitih **izvora**:
  - **User-agent**: svaki preglednik ima neke postavke
  - **Author**: stilovi definirani unutar web-aplikacije
  - **User**: korisnici mogu lokalno premostiti određene postavke, rijetko korišteno
- **Važnost** pravila je moguće modificirati naredbom tj. **iznimkom** "**!important**" (što se ne preporuča)
  - Više na: <https://developer.mozilla.org/en-US/docs/Web/CSS/Specificity>
- Primjenjuje se (najvažniji prvi):
  - **User-agent !important**
  - **User !important**
  - **Author !important**
  - **Author (normal)**
  - **User (normal)**
  - **User agent (normal)**

Prvo iznimke (koje treba izbjegavati)

Normalan i tipičan slijed koji najčešće vidimo u praksi

## ■ 2. Specifičnost (*specificity*) i 3. poredak

- Neformalno: "precizniji, specifičniji selektori imaju veći prioritet".
- Primjenjuje se (najvažniji prvi):
  - *Inline* stilovi (1000)
  - #ID selektori (100)
  - .class, :pseudo-class i [attribute] selektori (10)
  - <TAG> i ::pseudo-element selektori (1)
  - <body> i \* (0)
- Specifičnost se izračunava tako da se zbrajaju težine
- Ako je specifičnost ista, gleda se **poredak i pobjeđuje zadnji**



# Primjer izračuna specifičnosti

Hello World!

```
Elements Console Recorder >>
<!DOCTYPE html>
<html>
  <head>
    <style>
      #d123 {
        color: black;
      }

      p {
        color: pink;
      }

      #b123 p {
        color: blue;
      }
    </style>
  </head>
  <body id="b123">
    <p id="d123" style="color:red"> H
```

$S = 100$

$S = 1$

$S = 100 + 1 = 101$

$S = 1000$

```
Styles Computed Layout Event Listeners DOM Breakpoints >>
Filter :hov .cls + <
element.style {
  color: red;
}
#b123 p {
  color: blue;
}
#d123 {
  color: black;
}
p {
  color: pink;
}
p {
  display: block;
  margin-block-start: 1em;
  margin-block-end: 1em;
  margin-inline-start: 0px;
  margin-inline-end: 0px;
}
```

# ■ Kaskadnost - ukupan algoritam

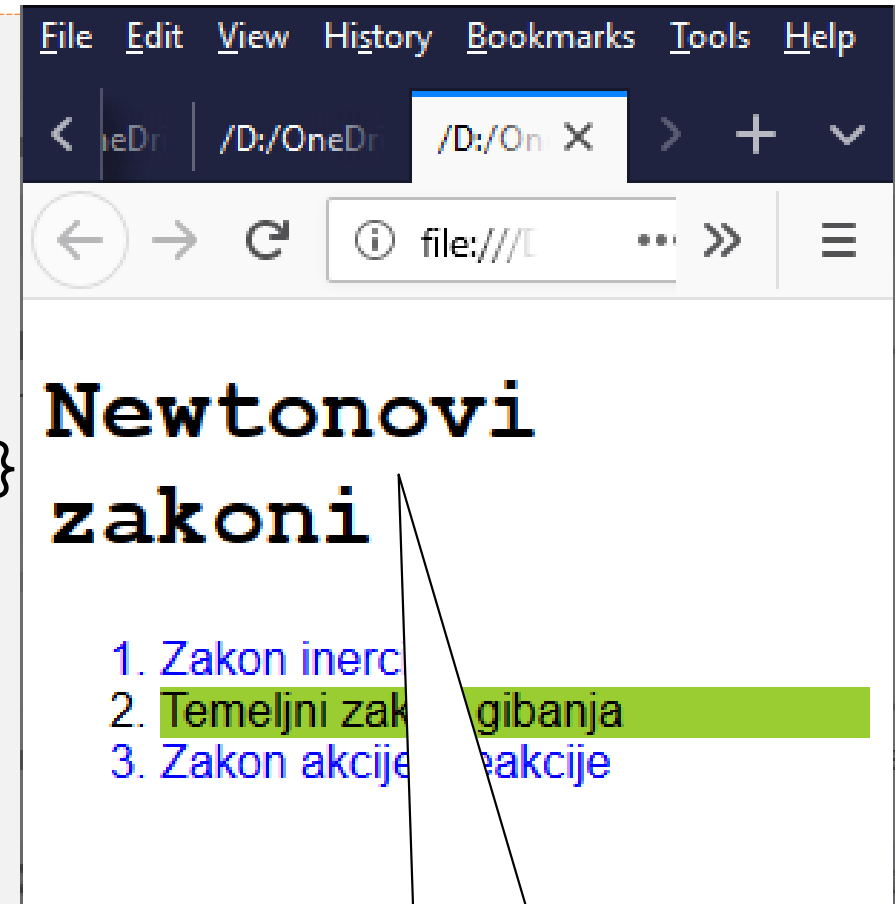
- Prema specifikaciji\*, kaskadna pravila se evaluiraju u četiri koraka:
  1. Pronaći sve deklaracije koje se odnose na neki element odnosno svojstvo
  2. Sortiraj po **(1) važnosti i izvoru**:
    1. Ako su različite važnosti/izvora primijeni pravilo (i stani)
    2. Inače, nastavi sa sljedećim korakom
  3. Sortiraj po **(2) specifičnosti**
    1. Ako su različite specifičnosti, primijeni pravilo najveće specifičnosti (i stani)
    2. Inače, nastavi:
  4. Primijeni **(3) poredak** tj. zadnje navedeno pravilo (*last wins*)

\* <https://www.w3.org/TR/2011/REC-CSS2-20110607/cascade.html#cascade>



# Zamijenimo \* s body, što se promijenilo?

```
<head>
<style>
  body { font-family: Courier; }
  #z2 { background-color: yellowgreen;
        color: black; }
  li { font-family: Arial; color: red; }
  li { color: blue; }
</style></head><body>
  <h1>Newtonovi zakoni</h1>
  <ol>
    <li id="z1">Zakon inercije</li>
    <li id="z2">Temeljni zakon gibanja</li>
    <li id="z3">Zakon akcije i reakcije</li>
  </ol>
</body>
```



Isto Courier?  
Zašto?

# Nasljeđivanje u CSS-u

- Elementi nasljeđuju neka svojstva od roditelja(predaka), a neka ne, npr.:
  - Color i font-family se nasljeđuju
  - Width, height, border, itd. se ne nasljeđuju
- *"Which properties are inherited by default and which aren't is largely down to common sense."*

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Cascade\\_and\\_inheritance](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance)

- Naslijeđena svojstva imaju **najmanji prioritet**, manji čak od user-agent stilova
- Ipak, vrlo koristan mehanizam - možemo npr. postaviti isti font svugdje tako da ga postavimo u body selektoru
- Moguće je kontrolirati nasljeđivanje s inherit, initial i unset opcijama

# Kontroliranje nasljeđivanja - primjer

```
<head> <style>
  body {
    color: green;
    border: 1px solid black;
  }
  ol { border: inherit; }
  #z1 { border: inherit; }
  #z2 { color: initial; }
  #z3 { color: unset; }
</style></head>
<body><h1>Newtonovi zakoni</h1>
<ol>
  <li id="z1">Zakon inercije</li>
  <li id="z2">Temeljni zakon gibanja</li>
  <li id="z3">Zakon akcije i reakcije</li>
</ol></body>
```

## Newtonovi zakoni

1. Zakon inercije
2. Temeljni zakon gibanja
3. Zakon akcije i reakcije

"uključuje"  
nasljeđivanje od  
**roditelja**

Vraća na  
default

"isključuje"  
nasljeđivanje i vraća na  
postavke preglednika

### *Text Hand*

A B C D E F G H I J  
 K L M N O P Q R  
 S T U V W X Y Z  
 a b c d e f g h i j k l m n o p q r s t u v w x y z  
 1 2 3 4 5 6 7 8 9 0

## Fonts/typography

*Typography is the art and technique of arranging type to make written language legible, readable, and appealing when displayed. The arrangement of type involves selecting typefaces, point sizes, line lengths, line-spacing (leading), and letter-spacing (tracking), and adjusting the space between pairs of letters (kerning[1]). The term typography is also applied to the style, arrangement, and appearance of the letters, numbers, and symbols created by the process*



# Font families

Generic family	Font family
serif	"Times New Roman", Georgia, Book Antiqua, ...
sans-serif	Verdana, Arial, Tahoma, ...
cursive	<i>"Brush Script MT", ...</i>
monospace	"Courier New", "Lucida Console"

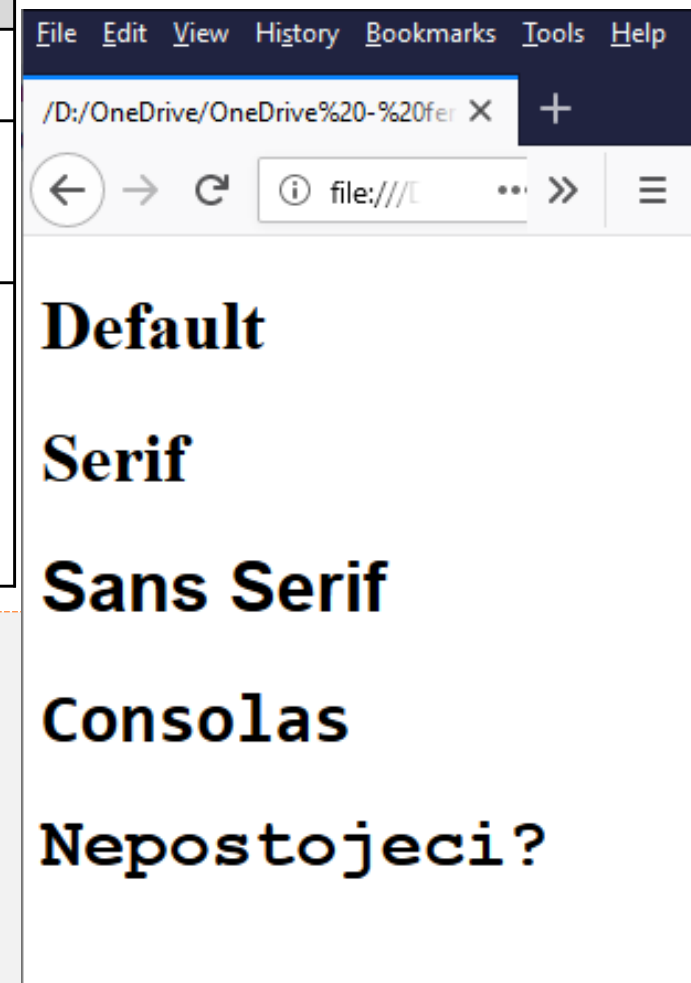
Fontovi koji imaju više riječi odvojenih razmakom se pišu u navodnicima

AaBbCc Sans-serif font  
AaBbCc Serif font  
AaBbCc Serif font (red serifs)  
#Serif

# Font families - koji font se prikazuje?

Zadano	Što se prikazuje i odakle?
Ništa	Defaultni font definiran u pregledniku
Generic family	Predstavnik te familije definiran u pregledniku
Font family	Zadani font ako preglednik raspolaže s njim (spada u uski krug "sigurnih fontova" ili postoji na lokalnom računalu).

```
<body>
  <h1>Default</h1>
  <h1 style="font-family: serif;">Serif</h1>
  <h1 style="font-family: sans-serif;">Sans Serif</h1>
  <h1 style="font-family: Consolas, monospace;">
    Consolas</h1>
  <h1 style="font-family: Nepostojeci-font, monospace;">Nepostojeci?</h1>
</body>
```



Možemo navesti i listu!



# ■ Koje fontove možemo koristiti?

- Postoje tri opcije:
  1. Fontovi s lokalnog (korisničkog) računala
    - Postoje mali skup fontova koji su zastupljeni na gotovo svim operacijskim sustavima, npr. TNR i Arial
    - "*Web safe fonts*", npr. <https://www.cssfontstack.com/>
  2. Web fonts
    - Fontovi koji preglednik dohvaća preko mreže i koji su pohranjeni na poslužiteljima neke treće strane
    - Npr. <https://fonts.google.com/>
  3. Poslužiteljski fontovi
    - Poput prethodnog, samo što fontove poslužujemo s vlastitog poslužitelja

# Koje fontove možemo koristiti: tri opcije

```
<head>
  <link href="https://fonts.googleapis.com/css?family=PT+Mono&display=swap"
rel="stylesheet">
  <style>
    @font-face {
      font-family: mojPTMono;
      src: url(PTMono-Regular.ttf);
    }
  </style>
</head>
<body>
  <h1 style="font-family: Consolas, monospace;">1. Consolas</h1>
  <h1 style="font-family: 'PT Mono', monospace;">2. PT Mono@Google</h1>
  <h1 style="font-family: mojPTMono, serif;">3. Moj PT Mono@localhost</h1>
</body>
```

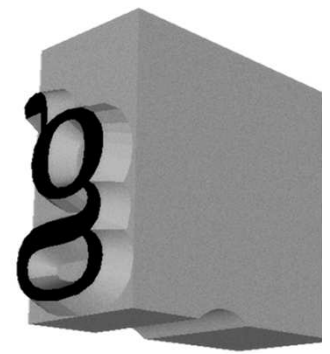
Diagram illustrating the three font options available in the code:

1. Consolas
2. PT Mono@Google
3. Moj PT Mono@localhost

The diagram shows the code structure with callouts indicating the font options used in the HTML output:

- Callout 1 points to the `font-family: Consolas, monospace;` in the first `<h1>` tag.
- Callout 2 points to the `font-family: 'PT Mono', monospace;` in the second `<h1>` tag.
- Callout 3 points to the `font-family: mojPTMono, serif;` in the third `<h1>` tag.





## @font-face direktiva

- S pristupima 2 i 3 više nismo ograničeni na mali skup uobičajenih fontova
- @font-face direktivom je moguće definirati novi font
- Moguće je definirati još detaljnija svojstva fontova osim same familije, npr:
  - font-weight: {normal, bold, 100, 200, ..., 900}
  - Itd. pogledati npr. <https://developer.mozilla.org/en-US/docs/Web/CSS/@font-face>
- ... ali za svaku kombinaciju mora postojati datoteke (definicija) tog font-facea, npr. ovime ćemo s GF-a dohvatiti Cousine familiju, ali weight 700, italic:

```
<link href="https://fonts.googleapis.com/css?family=Cousine:700i&display=swap" rel="stylesheet">
```

# Formati fontova

- **TTF** - TrueType Font, 1991. Apple
  - Precizna kontrola nad prikazom fontova
  - Najrašireniji i najprihvaćeniji format na OS-ovima
- **OTF** - OpenType, 1996, Microsoft + Adobe
  - Nadgradnja na TTF
- **EOT** - Embedded OpenType (EOT), 2008. Microsoft
  - Autorska prava, kompresija, ALI: podržan samo u IE
- **WOFF** - Web Open Font Format, 2009., Mozilla, Opera, Microsoft
  - = OTF/TTF + metapodatci (autorska prava) + kompresija
- **WOFF2** - 2018. Google, **još 30% bolja kompresija**
- **SVG font**
  - Nema font-hinting, aktualan za stare Safari korisnike

# Koji format koristiti?

- **WOFF(2) su budućnost**
- WOFF već jako dobro zastupljen: <https://caniuse.com/#search=woff>
- Može se probati redom: WOFF2, WOFF, TTF, npr.

```
<style>
  @font-face {
    font-family: mojFont;
    src: url(mojFont.woff2) format("woff2");
        url(mojFont.woff) format("woff");
        url(mojFont.ttf) format("truetype");
  }
</style>
```

# Još neka CSS svojstva fontova

```
<p style="font-weight: 200; font-style: italic;">
```

“Courage is going from failure to failure without losing enthusiasm.”

– Winston Churchill</p>

```
<p style="font-weight: bold;">
```

“Courage is going from failure to failure without losing enthusiasm.”

– Winston Churchill</p>

```
<p style="font-size:30px;">
```

“Courage is going from failure to failure without losing enthusiasm.”

– Winston Churchill</p>

*“Courage is going from failure to failure without losing enthusiasm.” – Winston Churchill*

**“Courage is going from failure to failure without losing enthusiasm.” – Winston Churchill**

“Courage is going from failure to failure without losing enthusiasm.” – Winston Churchill

- font-size
- font-weight
- font-style

# font-size, font-weight, font-style

Svojstvo	Moguće vrijednosti
<b>font-weight</b>	normal bold bolder lighter number initial inherit
<b>font-style</b>	normal italic oblique initial inherit
<b>font-size</b>	medium xx-small x-small small large xx-large smaller larger length initial inherit

- Dodatna objašnjenja mogućih vrijednosti pogledati npr. na <https://www.w3schools.com>
- Osim px za *length* se mogu koristiti i druge jedinice
  - Objašnjeno uskoro u posebnom poglavlju

# Još neka svojstva teksta: text-align, text-indent, letter-spacing, line-height

```
<head><style>
  p.author {  text-align: right;
               letter-spacing: 10px; }
</style></head>
<body style="text-align: center; text-indent: 50px;">
  <p>"In three words I can sum up everything I've learned about life: it goes on."<br> — Robert Frost</p>
  <p style="line-height: 16px;">
    "Here's to the crazy ones (...)"
    they can change the world, are the
    ones who do." </p>
  <p class="author">— Rob Siltanen</p>
  <p style="line-height: 32px;">
    "Here's to the crazy ones. The misfits
    ones who do."
  </p>
  <p class="author">— Rob Siltanen</p>
</body>
```

"In three words I can sum up everything I've learned about life: it goes on."  
— Robert Frost

"Here's to the crazy ones. The misfits. The rebels. The troublemakers. The round pegs in the square holes. The ones who see things differently. They're not fond of rules. And they have no respect for the status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things. They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do."

— R o b S i l t a n e n

"Here's to the crazy ones. The misfits. The rebels. The troublemakers. The round pegs in the square holes. The ones who see things differently. They're not fond of rules. And they have no respect for the status quo. You can quote them, disagree with them, glorify or vilify them. About the only thing you can't do is ignore them. Because they change things. They push the human race forward. And while some may see them as the crazy ones, we see genius. Because the people who are crazy enough to think they can change the world, are the ones who do."

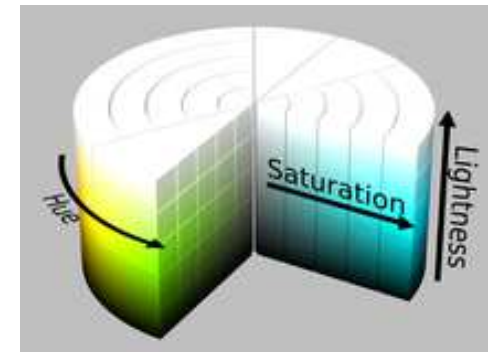
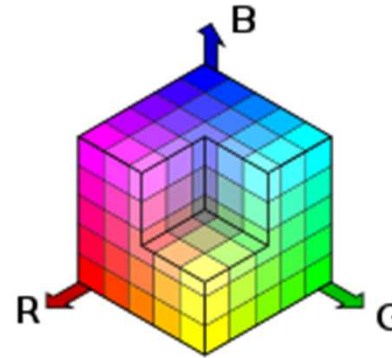
— R o b S i l t a n e n

A vibrant, multi-colored background with vertical stripes of red, orange, yellow, green, and blue, resembling a rainbow.

# Boje

# Model boja

- Model boja je apstraktni matematički model koji opisuje kako boje predstaviti uređenim n-torkama brojeva, tipično trima ili četirima vrijednostima odnosno komponentama boje.
- Neki modeli:
  - RGB** - (Red, Green, Blue)
  - HSL - (Hue, Saturation, Lightness)
  - HSV - (Hue, Saturation, Value)
  - CMYK
  - Munsell, Natural, ...
- Komplicirana tema, CSS podržava RGB i HSL
- U okviru predmeta ćemo koristiti samo RGB






# RGB: kako zadati boju?

- RGB je aditivni model u kojem se boja postiže "zbrajanjem" crvene, zelene i plave komponente
- Zadajemo intenzitete u rasponu od 0 do 255 (jedan bajt)
  - 24-bitna boja, tzv. "truecolor",  $256^3 = 16.777.216$  boja
- Opcije:
  1. `rgb(r, g, b)`
  2. `rgba(r, g, b, alpha)`, *alpha*  
definira (ne)prozirnost, (*opacity*) u intervalu [0, 1]
  3. Heksadecimalna notacija: `#rrggbb`
  4. Skraćena hex notacija  
(ako su obje hex znamenke iste): `#rgb`
  5. Putem **imena**, 140 imena: [https://www.w3schools.com/colors/colors\\_names.asp](https://www.w3schools.com/colors/colors_names.asp)
    1. Npr. blue, red, aqua, beige, ...
  6. `hsl(h, s, l)`, `hsla(h, s, l, alpha)`

 `rgb(0, 0, 255);`

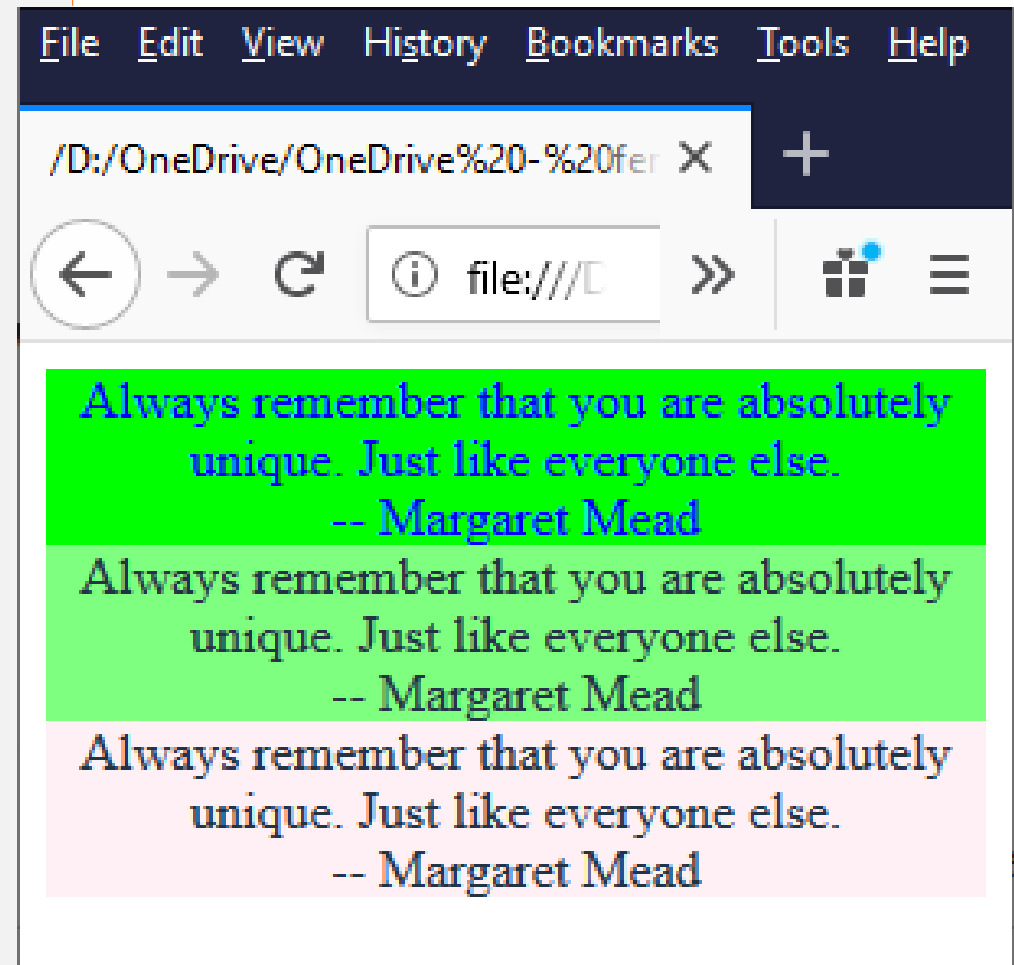
 `rgba(0, 0, 255, 0.5);`

 `#0000ff;`

 `#00f;`

# Primjer: color, background(-color)

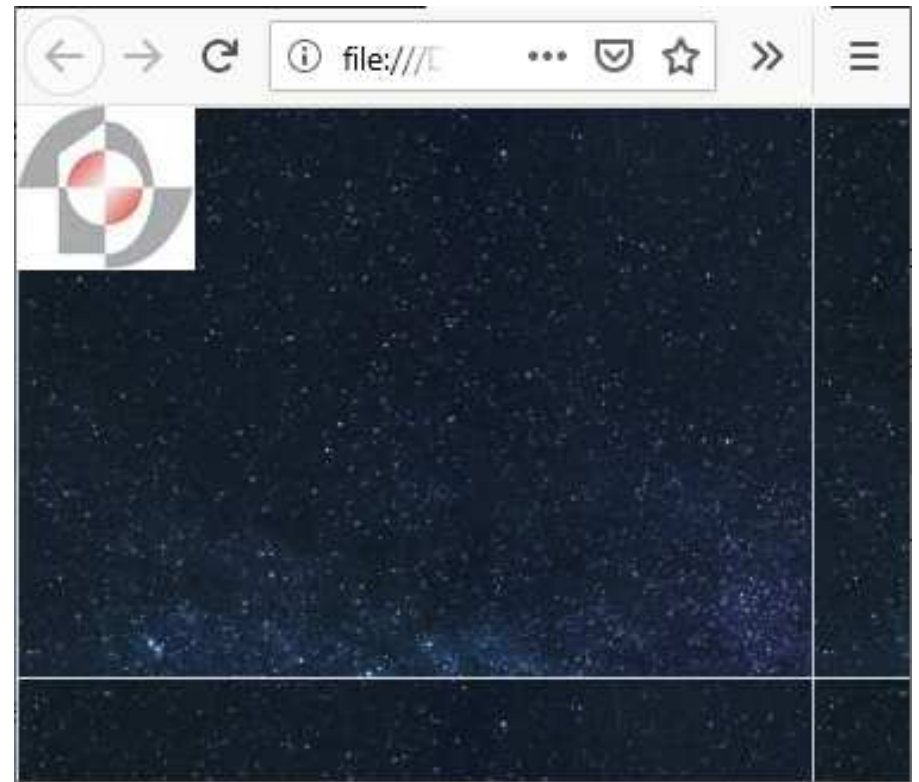
```
<head> <style>
  #div1 {      color: rgb(0, 0, 255);
    background-color: rgb(0, 255, 0)
  }
  #div2 {      color: #223344;
    background: rgba(0, 255, 0, 0.5)
  }
  #div3 {      color: #234;
    background: LavenderBlush;
  }
</style>
</head>
<body style="text-align: center;">
  <div id="div1">Always...</div>
  <div id="div2">Always...</div>
  <div id="div3">Always...</div>
</body>
```



**background-image**

# Slika kao pozadina

- Osim boje, možemo koristiti i sliku kao pozadinu
  - Moguće je postaviti N slika
  - Podrazumijevano ponašanje:
    - Gornji lijevi kut
    - Ponavlja se horizontalno i vertikalno



```
<head><style>
  body {
    background-color: #ccc;
    background-image: url('./fer-logo-xs.jpg'), url('./stars-xs.png');
    background-repeat: no-repeat, repeat;
  }
</style></head>
<body></body>
```

Relativni put od mjesta gdje je taj CSS napisan

Što bi se prikazalo da nema ove linije?

# Skraćena sintaksa: background

- Moguće je sva svojstva pozadine nanizati u jednoj deklaraciji (*shorthand property*), npr.:

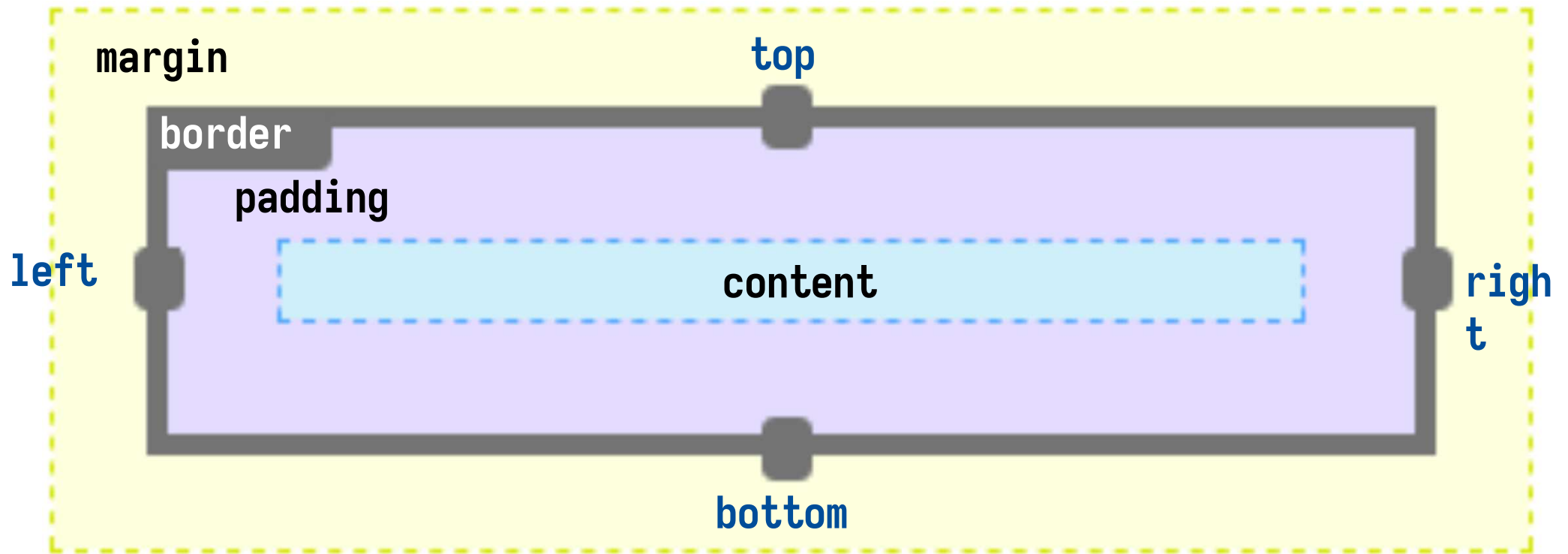
```
background: #ccc url('./fer-logo-xs.jpg') no-repeat;
```

- Dakle, background je kratica za ova svojstva:

- [background-color](#)
- [background-image](#)
- [background-position](#)
- [background-size](#)
- [background-repeat](#)
- [background-origin](#)
- [background-clip](#)
- [background-attachment](#)

Samostalno pogledati još  
barem podebljana svojstva na:  
[https://www.w3schools.com/cssref/  
/css3\\_pr\\_background.asp](https://www.w3schools.com/cssref/css3_pr_background.asp)

# Box model



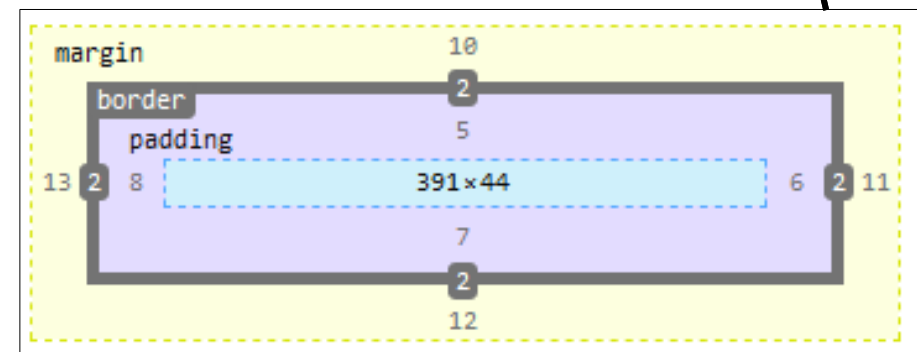
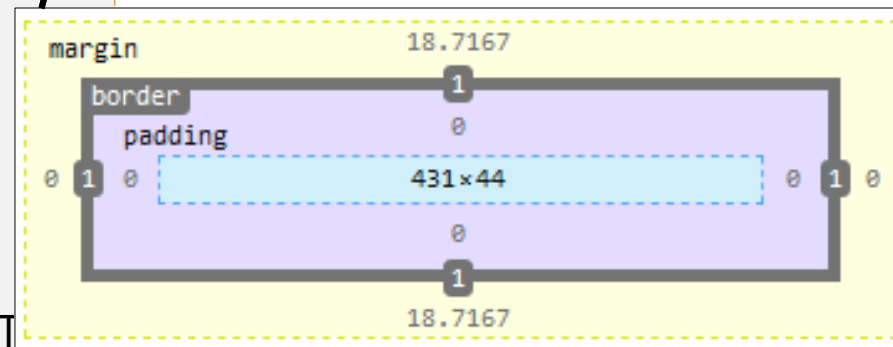
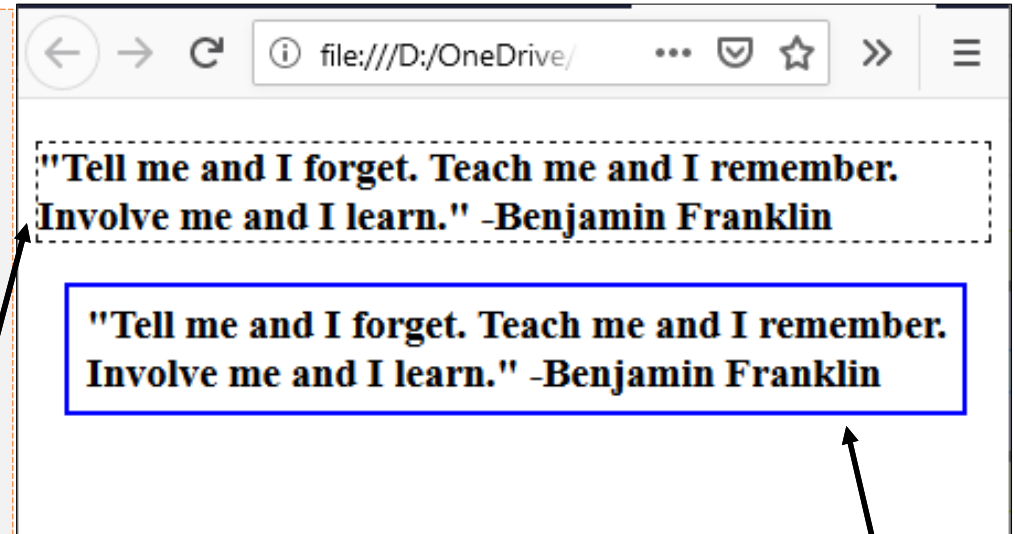
# Box model: margin, border, padding

```
<head><style>
  #q1 {
    border: 1px dashed black;
  }
  #q2 {
    border: 2px solid blue;
    margin: 10px 11px 12px 13px;
    padding: 5px 6px 7px 8px;
  }
</style></head>
<body>
```

```
  <h3 id="q1">"Tell me and I forget. Teach me and I remember. Involve me and I learn." -Benjamin Franklin</h3>
```

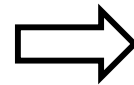
```
  <h3 id="q2">"Tell me and I forget. Teach me and I remember. Involve me and I learn." -Benjamin Franklin</h3>
```

```
</body>
```



## ■ Digresija: *shorthand properties*

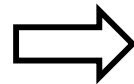
```
border-width: 2px;  
border-style: solid;  
border-color: blue;
```



Poredak nije bitan

```
border: 2px solid blue;
```

```
margin-top: -10px;  
margin-right: 11px;  
margin-bottom: 12px;  
margin-left: 13px;
```



Poredak bitan

```
margin: -10px 11px 12px 13px;  
top right bottom left
```

```
margin: 20px 25px;  
top left  
bottom right
```

```
margin: 30px;  
top left  
bottom right
```



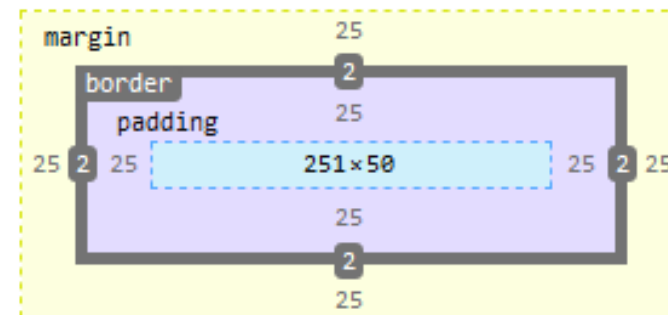
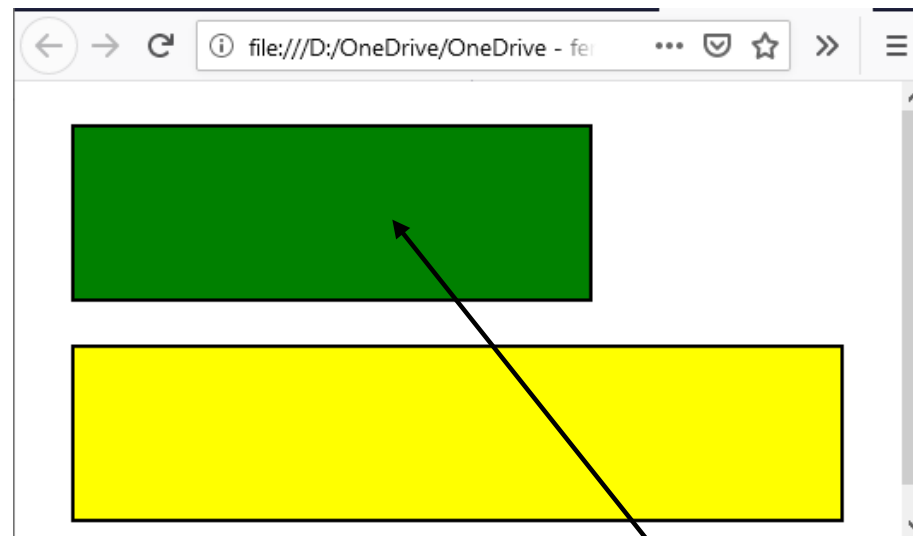
# ■ Box model i inline/block elementi?

- HTML podsjetnik:
  - **Block** elementi zauzimaju cijeli horizontalni prostor
    - Npr. `div`, `h1`, `p`
    - Na njih se u potpunosti primjenjuje box model
  - **Inline** elementi zauzimaju samo prostor koji im je potreban da se prikažu, pa ih može biti više u jednom redu
    - Npr. `span`, `img`
    - Koristi se box model, ALI:
      - `margin-top` i `margin-bottom` nemaju učinka
      - `padding-top` i `padding-bottom` ne pomiču susjedni sadržaj već samo vlastiti border
    - Ne može im se postaviti `width` i `height` (sljedeći slajd)
- CSS omogućuje da se elementu promijeni narav prikaza pomoću `display: block | inline | inline-block | ...`

# Block elementi: width i height

```
<head>
  <style>
    div {
      border: 2px solid black;
      margin: 10px;
      padding: 15px;
      height: 50px;
    }
  </style>
</head>
<body>
  <div style="background: green;
              width: 50%;">

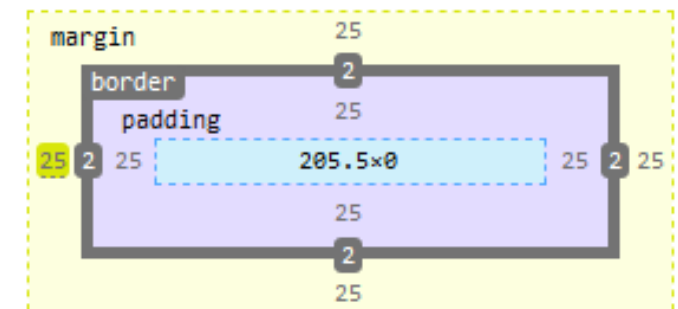
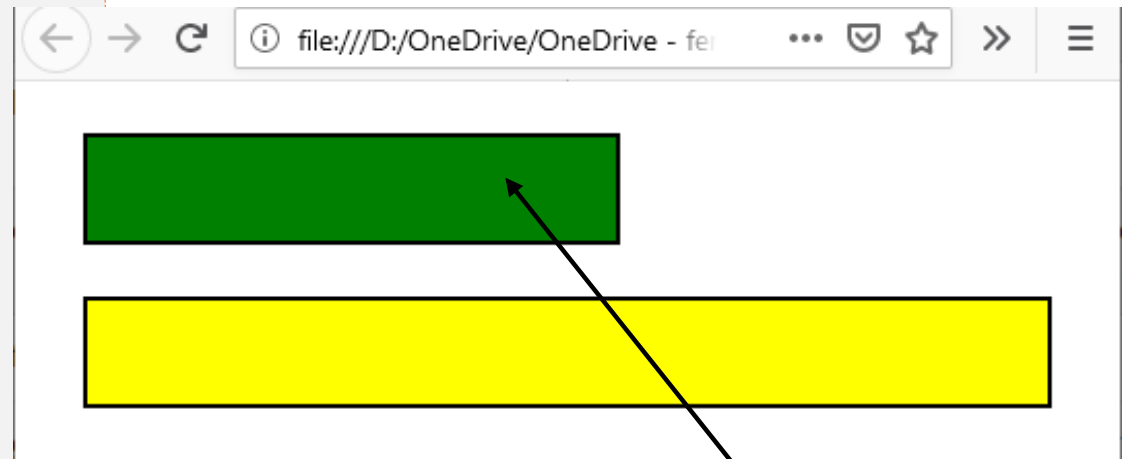
  </div>
  <div style="background: yellow;">
  </div>
</body>
```



Očigledno se širina i visina pretpostavljeno primjenjuju samo na sadržaj. To se može promijeniti s **box-sizing** svojstvom (sljedeći slajd).

# Block elementi: box-sizing

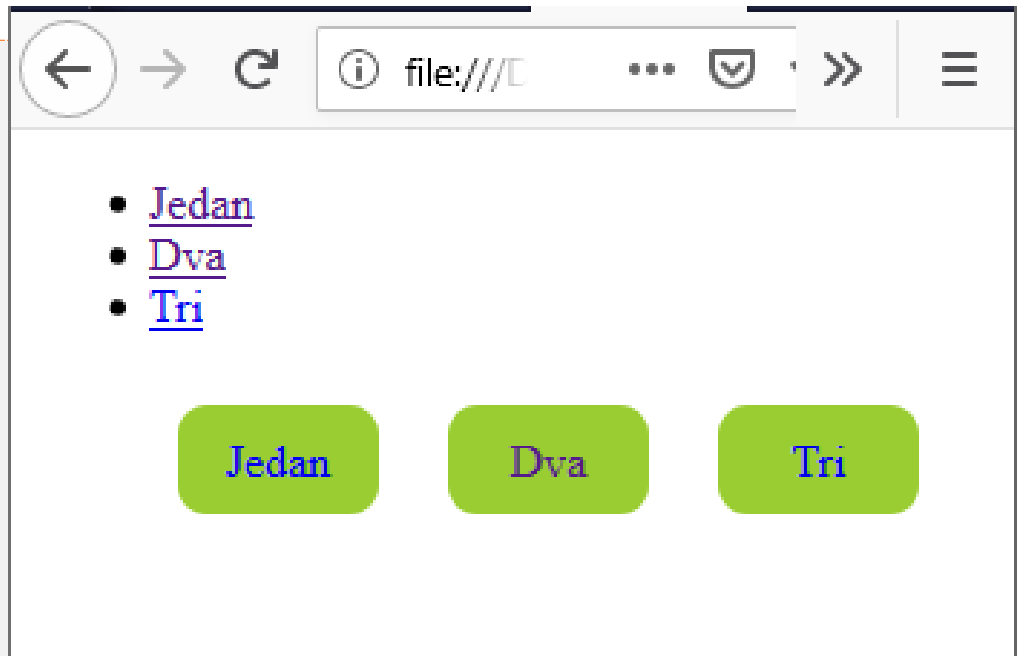
```
<head><style>
  * {
    box-sizing: border-box;
  }
  div {
    border: 2px solid black;
    margin: 10px;
    padding: 15px;
    height: 50px;
  }
</style></head>
<body>
  <div style="background: green;
              width: 50%;">
  </div>
  <div style="background: yellow;">
  </div>
</body>
```



I dalje je zeleni > 50% zato jer border-box ide do bordera, a margine su i dalje posebno - nije ih moguće uklopiti, ali to je logično - onda ih ne treba postaviti.

# Svojstvo display, stiliziranje linkova

```
<head><style>
  ul.hrzntl li { display: inline; }
  ul.hrzntl a {
    background: yellowgreen;
    border-radius: 10px;
    padding: 10px;
    margin: 10px;
    text-align: center;
    text-decoration: none; /*-underline*/
    width: 50px; /*ne radi bez sljedećeg*/
    display: inline-block;
  }
</style></head><body>
  <ul>
    <li><a href="#">Jedan</a></li>
    <li><a href="#">Dva</a></li>
    <li><a href="#">Tri</a></li>
  </ul> ...
```



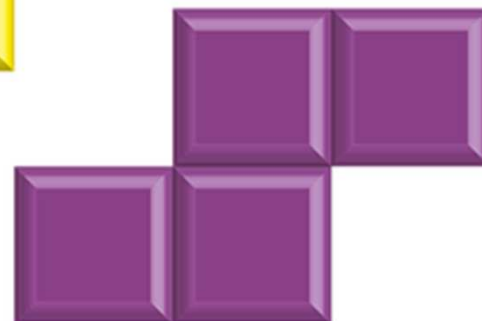
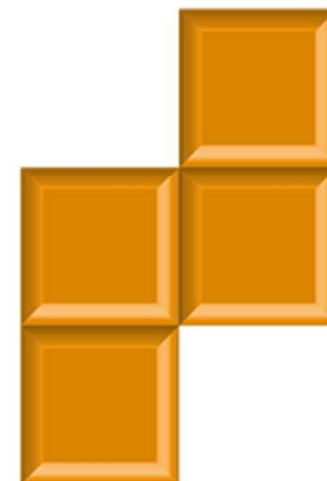
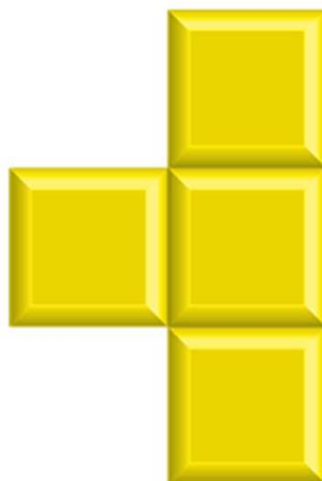
```
...
<ul class="hrzntl">
  <li><a href="#1">Jedan</a></li>
  <li><a href="#2">Dva</a></li>
  <li><a href="#3">Tri</a></li>
</ul>
</body>
```

# ■ Domaća zadaća: samostalno pogledati

- Links:
  - [https://www.w3schools.com/Css/css\\_link.asp](https://www.w3schools.com/Css/css_link.asp)
- Tables:
  - [https://www.w3schools.com/Css/css\\_table.asp](https://www.w3schools.com/Css/css_table.asp)
- Lists:
  - [https://www.w3schools.com/Css/css\\_list.asp](https://www.w3schools.com/Css/css_list.asp)
- Buttons:
  - [https://www.w3schools.com/csS/css3\\_buttons.asp](https://www.w3schools.com/csS/css3_buttons.asp)
- Icons:
  - [https://www.w3schools.com/Css/css\\_icons.asp](https://www.w3schools.com/Css/css_icons.asp)
- Forms (osim Responsive forms):
  - [https://www.w3schools.com/Css/css\\_form.asp](https://www.w3schools.com/Css/css_form.asp)

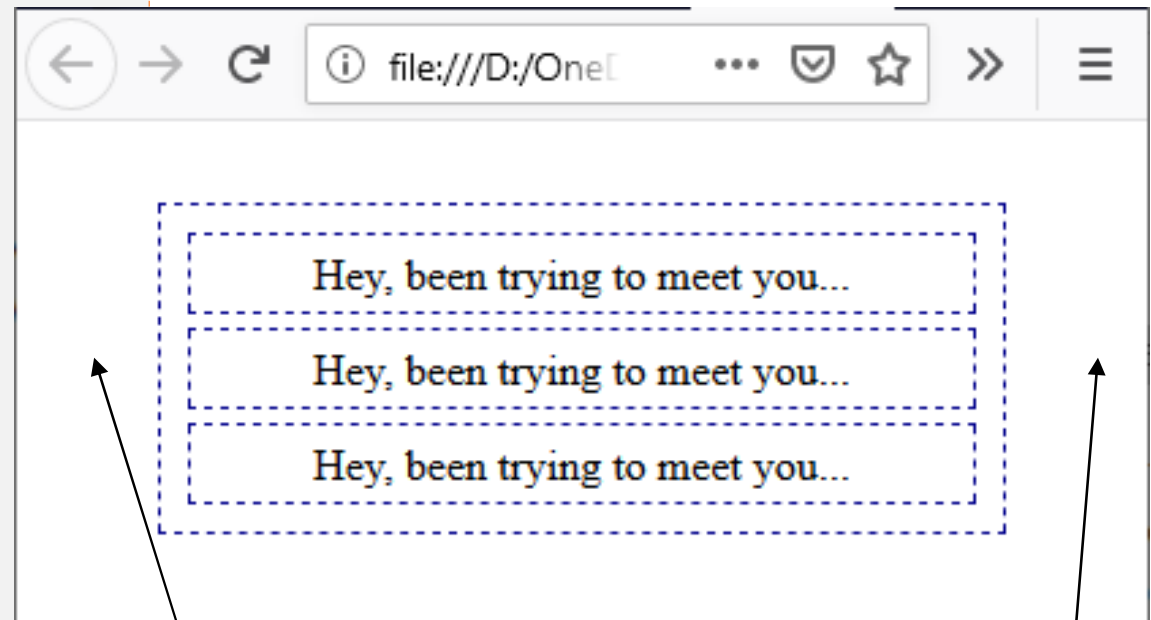
# Razmještaj

uvod: float, position



# Kako horizontalno centrirati sadržaj?

```
<head><style>
  div {
    border: dashed darkblue 1px;
    padding: 5px;
    margin: 5px;
    text-align: center;
  }
  div.container {
    margin: 30px auto;
    max-width: 300px;
  }
</style></head>
<body>
  <div class="container">
    <div>Hey, been trying to meet you...</div>
    <div>Hey, been trying to meet you...</div>
    <div>Hey, been trying to meet you...</div>
  </div>
</body>
```

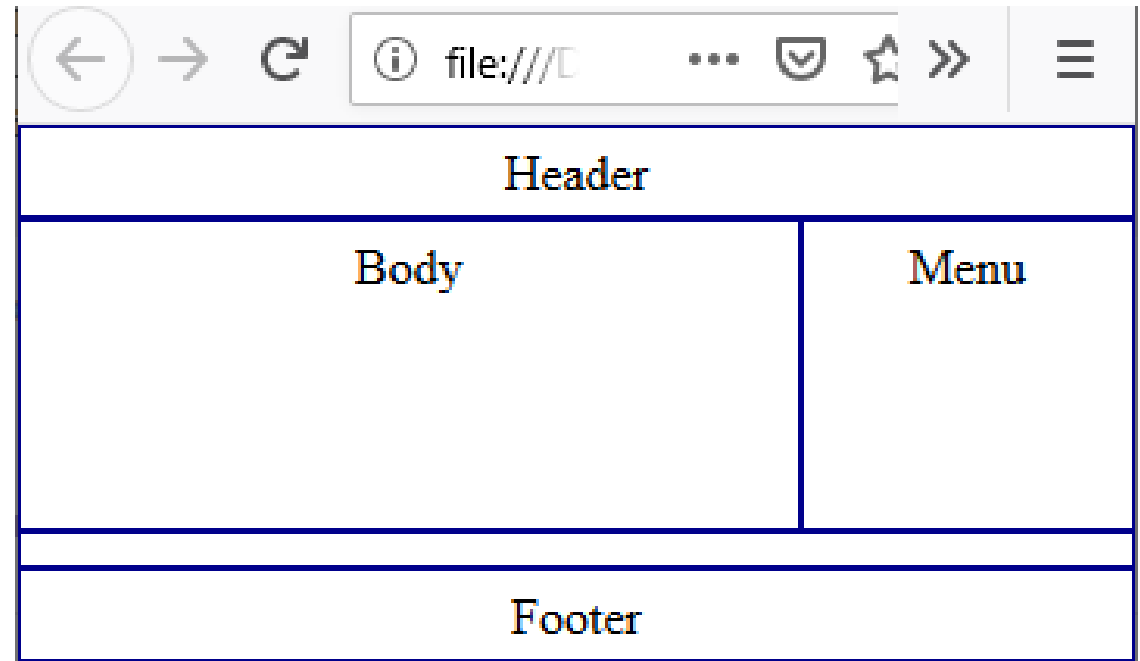


**auto:** jednake margine

samo **width** se ne ponaša dobro kod promjene veličine - nije responzivno (probajte!)

# CSS Layout: float and clear (danas zastarjelo)

```
<head><style>
  * { box-sizing: border-box; }
  body { margin: 0px; }
  div {
    border: solid darkblue 1px;
    padding: 5px;
    text-align: center;
  }
  div.body, div.menu {
    height: 100px;
    float: left;
  }
  div.body { width: 70%; }
  div.menu { width: 30%; }
</style></head>
<body>...
```



```
...
<div class="header">Header</div>
<div class="body">Body</div>
<div class="menu">Menu</div>
<div style="clear: left"></div>
<div class="footer">Footer</div>
</body>
```



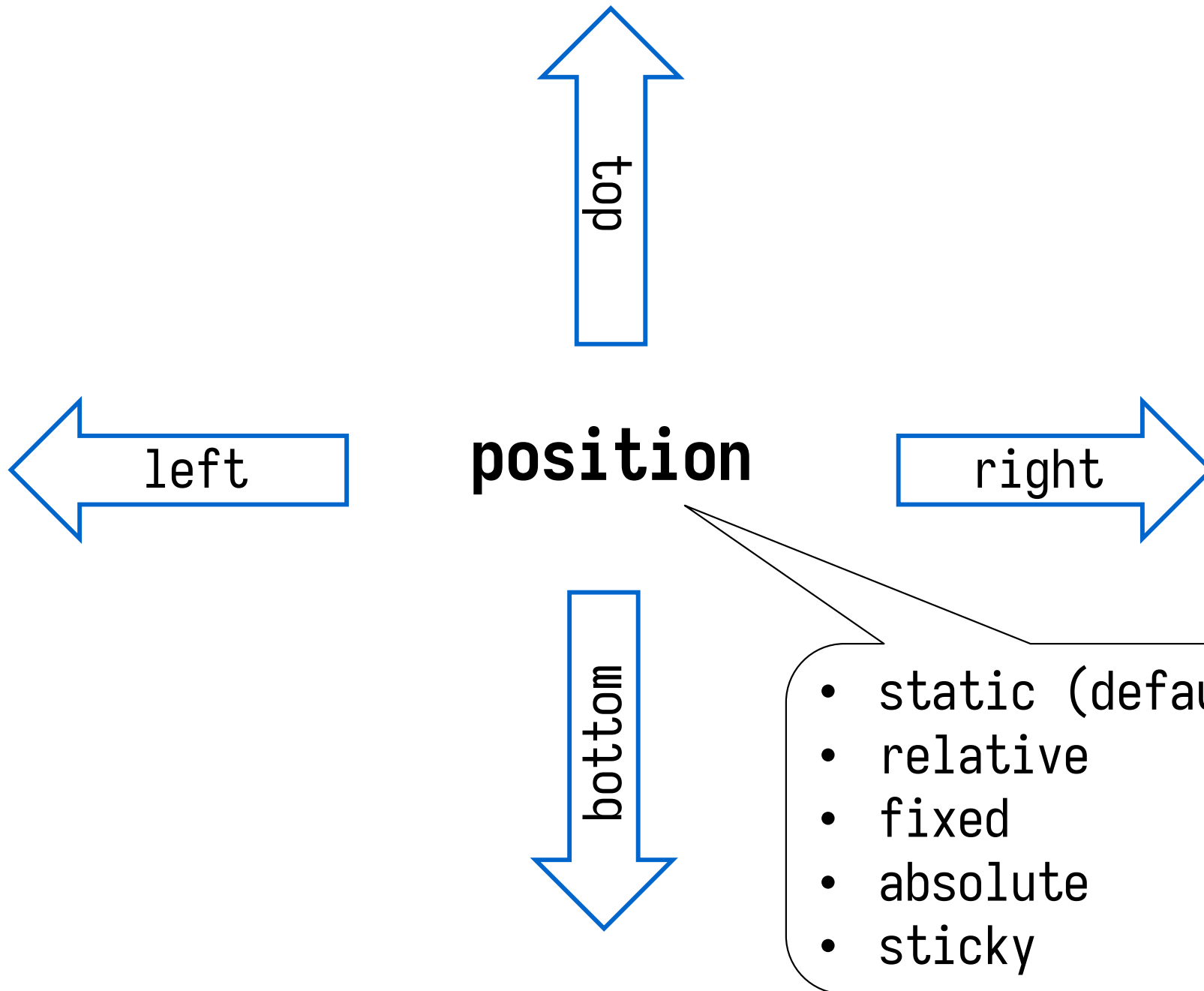
# Još jedan float primjer - ispravna uporaba

```
<head>
  <style>
    img.left { float: left; }
    img.right { float: right; }
  </style>
</head>
```

```
<body>
  <div>
    
    
    <h2>Povijesni pregled</h2>
    <p>Iako povijest samostalnog djelovanja Fakulteta započinje
1. srpnja 1956. upisom prve generacije studenata na samostalni Ele
ktrotehnički fakultet, tradicija studija elektrotehnike seže u 191
9. godinu, kada je na Kraljevskoj visokoj tehničkoj školi uspostav
ljen studij elektrotehnike, a nastavlja se uspostavom Tehničk
kulteta 1926. godine na kojem je djelovao Strojarsko-elektro
inženjerski odjel.</p>
  </div>
</body>
```



Domaća zadaća:  
isprobajte **clear left**,  
**right** i **both** na **p**  
elementu.



# Primjer: position (1/2): static i relative

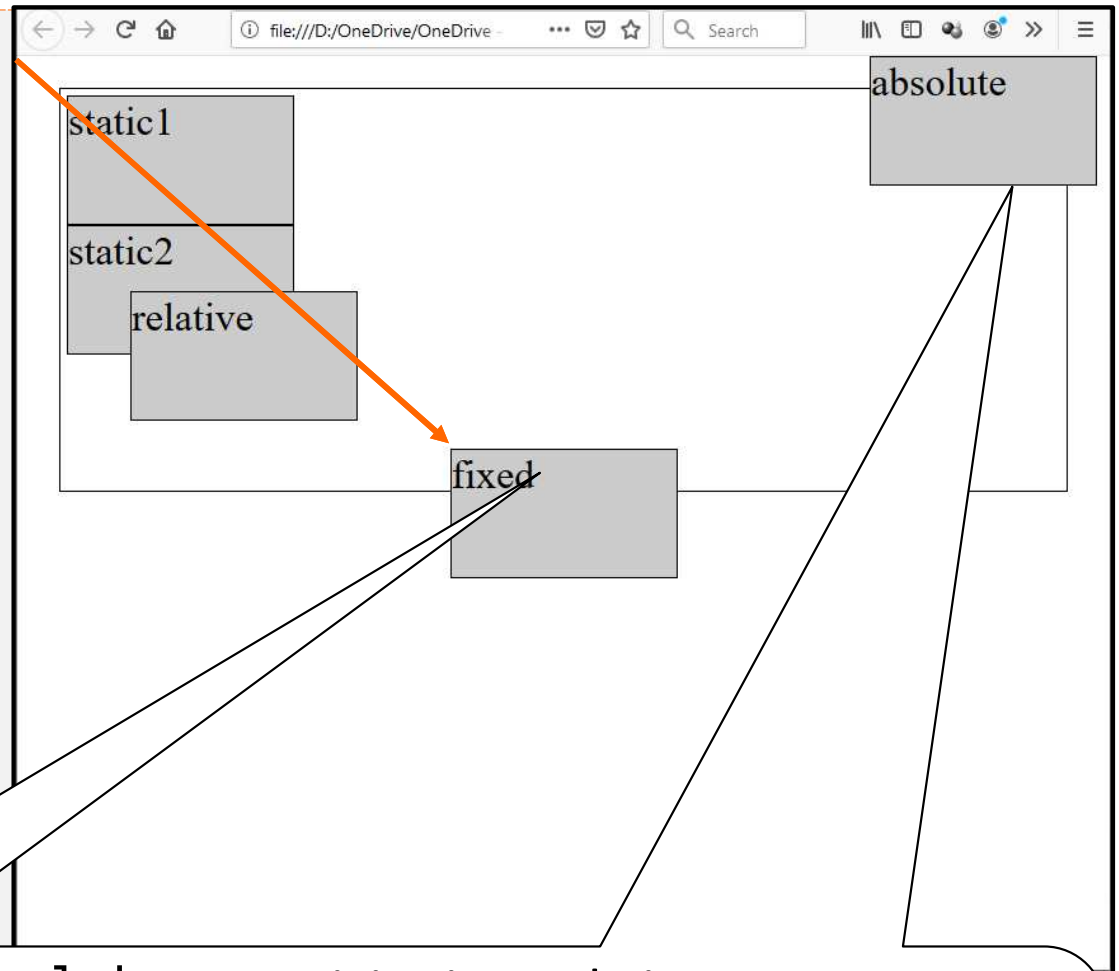
```
<head><style>
  div {
    border: 1px solid black;
    background: #ccc;
    font-size: 32px; }
  div.container {
    margin: 25px 25px;
    padding: 5px;
    background: unset;
    /* position: relative; */
  }
  div.lego {
    width: 177px;
    height: 100px;
  }
  div.relative {
    position: relative;
    top: -50px;
    left: 50px; } ...
```



```
... </style></head><body>
  <div class="container">
    <div class="lego">static1</div>
    <div class="lego">static2</div>
    <div class="lego relative">relative</div>
    <div class="lego absolute">absolute</div>
    <div class="lego fixed">fixed</div>
  </div></body>
```

# Primjer: position (2/2): absolute i fixed

```
... div.absolute {  
  position: absolute;  
  top: 0px;  
  right: 10px; }  
div.fixed {  
  position: fixed;  
  top: 50%;  
  left: 50%;  
  transform: translate(-50%, -50%); }  
</style></head><body>  
  <div class="container">  
    <div class="lego">static1</div>  
    <div class="lego">static2</div>  
    <div class="lego relative">relative</div>  
    <div class="lego absolute">absolute</div>  
    <div class="lego fixed">fixed</div>  
  </div></body>
```



**fixed** - u odnosu na *viewport*.  
Ne miče se skrolanjem.

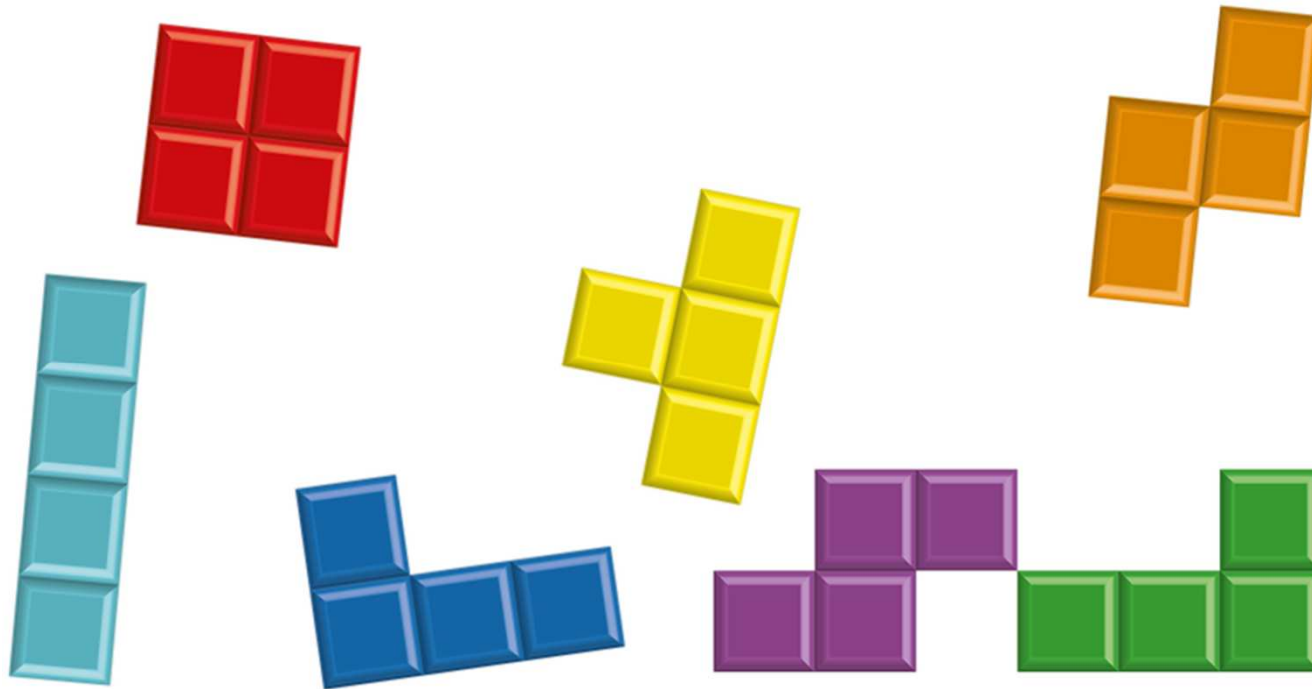
**absolute** se pozicionira s obzirom na najbližeg pozicioniranog pretka, inače u odnosu na *body*.  
Probajte odkomentirati `position relative` njegova roditelja na prethodnom slajdu.

# ■ Position - domaća zadaća

- Još samostalno pročitati:
  - [https://www.w3schools.com/Css/css\\_positioning.asp](https://www.w3schools.com/Css/css_positioning.asp)
  - Svakako proučite:
    - sticky
    - z-index

## ■ Razmještaj - "kraj početka"

- S dosad pokazanim svojstvima nije lako napraviti složenije razmještaje elemenata na stranici



- Stanje od prije 10-15 godina
- **Sljedeće predavanje** objašnjava moderna rješenja