HTML5 CSS3

Jednostavna izrada web stranica

Davor Grdić i Saša Fajković

Karlovac, 2015

SADRŽAJ:

1) UVOD – WEB STRANICA, OD KUDA ZAPOČETI 1

1. a HTML – što je? 1

1. b CSS – što je? 1

1. c HTML + CSS poveznica 1

1. d Izrada stranice 1

2) HTML DETALJNIJE 2

2. a Osnovni HTML tagovi 2

2. b Važnost i verzije DOCTYPEa 4

3) RADNI ALATI 5

3. a WSYIWYG – What You See Is What You Get 5

3. b Text-based – pisanje teksta (koda) 6

4) Inspect Element – Pregled konstrukcije stranice iz internet preglednika 7

5) HTML elementi 12

5. a paragraf 12

5. b </br>, <br> break line 13

5. c <h1></h1, <h2> Headings </h2>…<h6></h6> 14

5. d Link (poveznica) 15

5. e Charset – prepoznavanje hrvatskih znakova 16

6) Atributi i ugniježđeni tagovi/elementi 17

6. a Atributi 17

6. b Ugniježđeni tagovi 17

7) Komentari 18

8) <img> Slike 19

8. a Umetanje slike 19

8. b Umetanje slike koja je poveznica 19

9) <table> Tablice 20

9. a <table> i <td> Izrada tablica 20

9. b <th> Naslovi stupaca 21

9. c Colspan – proširi redak na širinu dva ili više stupaca 22

9. d Širina tablice i ćelija 23

10)   Liste 27

11) DIV 28

12) Tekst i mogući tagovi 29

12. a <b> Podebljani (boldani) tekst 29

12. b <i> Nakonešni (italic) tekst 29

12. c Podvučeni (underline) tekst 30
12. d Povećani tekst 30
12. e Smanjeni tekst 30
12. f <del> Uklonjen tekst 31
12. g <s> Netočan, irelevantan, nevažeći tekst 31
13) Forma, polja i dugme 32
13. a <input> elementi 32
13. b <input type="text"> 33
13. c <input type="password"> 33
13. d <input type=email"> 33
13. e <input type="radio"> 34
13. f <input type="checkbox"> 34
13. g <input type="button"> 35
13. h <input type="submit"> 35
13. i <input type="reset"> 36
13. j <input type="color"> 36
13. k <input type="file"> 37
13. l <input type="datetime"> 38
13. m <input type="date"> 39
13. n <input type="time"> 39
13. o Ostali tipovi 39
14) POST i GET metode 40
14. a GET metoda 40
14. b POST metoda 40
15) Uvod u CSS 41
16) Povezivanje HTML-a i CSS-a 42
16. a Korištenje style=" " atributa (*Inline style*) 42
16. b Pisanje CSS-a u zaglavlju HTML-a (*Internal style sheet*) 43
16. c Pisanje CSS-a u zasebnu datoteku (*External style sheet*) 43
16. d Kombinirano pisanje CSS-a 44
17) CSS selektori 45
17. a Selektor elementa 45
17. b ID selektor 46
17. c Selektor klase 46
18) CSS pozadine 47
18. a Boja pozadine 47

18. b Pozadinske slike 48
19) CSS tekst 49
19. a Fontovi 49
19. b Stilovi fonta 50
19. c CSS poveznice 51
20) CSS liste 52
21) CSS tablice 54
21. a Obrub tablice 54
21. b Dimenzije tablica 55
21. c Tekst u tablicama 56
21. d Boje tablice 57
22) CSS box model 58
22. a Sadržaj 59
22. b Obrubi 59
22. c Padding i margine 60
22. d Box-sizing 61
23) CSS pozicioniranje i prikaz elemenata 62
23. a *Inline*, *block* i *none* 62
23. b Float 64
24) Popis slika 66

1) UVOD – WEB STRANICA, OD KUDA ZAPOČETI

1. a HTML – što je?

Vjerojatno ste se već negdje susreli s ovom čudnom kraticom. HTML je skraćeni naziv za **Hypertext Markup Language**. Korištenjem HTML-a kreiramo Web stranice, a danas je u širokoj uporabi HTML5.

1. b CSS – što je?

CSS se najčešće veže usko uz HTML. Dok HTML koristimo za stvaranje kostura Web stranice, CSS koristimo za uljepšavanje prikaza sadržaja na stranici. CSS je kratica za **Cascading Style Sheets**, a danas je u masovnoj uporabi CSS3.

1. c HTML + CSS poveznica

Zamislimo da gradimo kuću. Osnovni kostur kuće (nosivi stupovi, krov, zidovi) bi bili građeni korištenjem HTML-a. Fasada, nijanse boja, načini otvaranja prozora i ostalo bi se gradilo korištenjem CSS-a. HTML dakle koristimo za izgradnju samog kostura stranice, a CSS za uljepšati prikaz svih elemenata na stranici.

1. d Izrada stranice

Web stranica je pojam koji je možda netočno preveden na hrvatski jezik. U originalu (engleski jezik) razlikujemo **Web Page** i **Website**. Konačni cilj je izgraditi Web ite, ali da bi došli do toga gradimo Web Page. Dakle, Website je izrađen na osnovu jednog ili više Web Pagea. Prebacimo li to na Hrvatski, rekli bi da Web stranicu gradimo pomoću više Web stranica… što baš nema nekog smisla. Upravo zato je ovaj dio predstavljen korištenjem engleskog jezika kako bi se lakše shvatila izgradnja Web stranice.

2) HTML DETALJNIJE

Kako smo već rekli, HTML je kratica za Hypertext Markup Language. Riječi **Markup Language** nam govore da nešto obilježavamo. Konkretno, obilježavamo korištenjem određenih oznaka koje zovemo **Markup Tags.** HTML dokumenti su obilježeni HTML oznakama koje zovemo **HTML Tags**. Svaka takva oznaka (HTML Tag) opisuje različite sadržaje HTML dokumenta.

```
<!DOCTYPE html>

<html>

  <head>
     <title> Naslov dokumenta </title>
  </head>

  <body>
     <p> Prvi odlomak u dokumentu </p>
  </body>

</html>
```

2. a Osnovni HTML tagovi

Prije nego krenemo kreirati prvu Web stranicu, potrebno je proći malo teorije (svaki početak je težak i naporan). U ovom poglavlju ćemo upoznati najosnovnije HTML tagove (oznake) koje svaki HTML dokument treba sadržavati.

DOCTYPE – opisuje vrstu HTML dokumenta. Ovisno o verziji HTML-a na kojoj želimo zasnivati izgradnju stranice, dodjeljuje se određeni DOCTYPE.

<html> </html> - opisuje cijeli HTML dokument. Za sada, ovo možemo smatrati glavnim tagom (oznakom) u koji postavljamo sve ostale tagove (oznake). Uočite kako imamo otvoreni i zatvoreni tag. Otvoreni tag (<html>) obilježava početak taga, a (</html>) obilježava kraj taga. Sve što napišemo između ta dva taga je zapravo postavljeno **unutar html taga**.

<head> </head> - daje informacije o HTML dokumentu. Ovaj tag se često naziva **glavom dokumenta.** Unutar ovog taga se postavljaju drugi tagovi koji daju osnovne informacije o dokumentu, poput *<title> </title>* taga koji opisuje naslov dokumenta. **Sadržaj koji se napiše unutar <head> </head> nije direktno vidljiv na stranici!**

<body> </body> - unutar ova dva taga stavljamo sve ono što želimo prikazati korisniku kada posjeti stranicu. Ovaj tag se ujedno naziva i **tijelom dokumenta**. Unutar njega postavljamo paragrafe, elemente poput gumba, padajućeg izbornika i slično. Primjerice * * Ovo je moj prvi paragraf * * će ispisati tekst „Ovo je moj prvi paragraf“ na stranicu, a biti će obilježen tagom *paragraf*. Vrste i značenja tagova ćemo uskoro detaljnije obraditi.

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

Slika 1 - Prikaz strukture HTML dokumenta[1]

Područja obojena u sivu boju se ne prikazuju korisniku, odnosno sve unutar *<body> </body>* će se prikazati korisniku, a ostatak dokumenta neće.

[1] Preuzeto s: http://www.w3schools.com/html/html_intro.asp

2. b Važnost i verzije DOCTYPEa

DOCTYPE opisuje verziju HTMLa standarda koju koristimo prilikom izgradnje Web stranice. Važno je navesti ispravnu verziju kako bi Internet preglednik znao točno interpretirati (prepoznati) elemente koje smo postavili u naš dokument. Danas je u uporabi najnoviji HTML5 standard koji je predstavljen 2012. godine. Prije HTML5 se koristio XHTML (2000. godina) i HTML 4.01 (1999. godina). Prva verzija HTMLa zaživjela je davne 1991. godine. Između su postojale još neke verzije, no ove se smatraju najvažnijima.

HTML 4.01 DOCTYPE:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN""http://www.w3.org/TR/html4/loose.dtd">
```

HTML 5 DOCTYPE:

```
<!DOCTYPE html>
```

HTML 5 još nije do kraja definiran. Neki nagađaju kako će krajnja definicija HTML5 standarda biti postignuta tek 2020. godine. Iako još nije do kraja definiran, HTML5 je danas preferirana verzija HTML standarda i svi imalo ozbiljni preglednici podržavaju sve ili većinu njegovih značajki.

3) RADNI ALATI

Za izradu Web stranica možemo koristiti izuzetno puno alata. Osnovne dvije kategorije u koje možemo podijeliti ove alate su WYSIWYG i tekstualno bazirani alati.

3. a WSYIWYG – What You See Is What You Get

Ova vrsta alata je jako privlačna samim početnicima koji nemaju prethodnog iskustva u izradi stranica, nemaju daljnje tendencije učiti kako raditi web stranice, a žele na brzinu napraviti stranicu. Kao što samo ime kaže, radi se po principu „Što vidiš, to dobiješ“. Korisnik na jednostavan način može razmještati elemente po stranici i mijenjati im svojstva poput boje, debljine obruba, ispune i slično. Ovakav pristup je jednostavan ali ne nudi puno opcija čime je i sam *developer* (osoba koja radi stranicu) ograničena dostupnim mogućnostima.

Primjeri WYSIWYG alata:

- Kompozer (http://kompozer.net/)
- Adobe Dreamweaver (http://www.adobe.com/products/dreamweaver)

3. b Text-based – pisanje teksta (koda)

Pisanje teksta odnosno HTML koda nudi potpunu slobodu u kreiranju stranice i developeru stoje sve opcije na raspolaganju jer u ovom slučaju vidi svaki detalj stranice i može utjecati i na najmanje detalje. Mi ćemo se bazirati na Text-based editorima (uređivačima) jer želimo naučiti kvalitetno i s razumijevanjem raditi web stranice. Korištenjem ovih alata radimo direktno po ***source code-u***. Source code (izvorni kod) je skupina raznih riječi i slova odnosno koda koji ćemo polako upoznavati.

Primjeri Text-based alata:

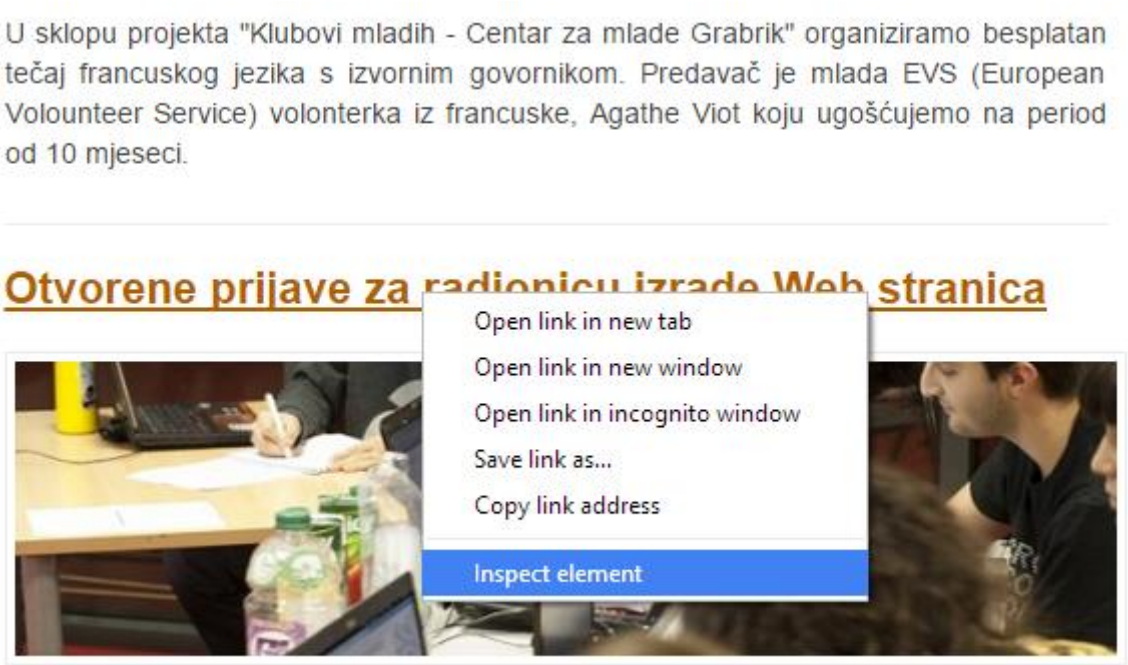
- ActiveState Komodo (http://komodoide.com)
- Eclipse (https://eclipse.org)
- Notepad++ (http://notepad-plus-plus.org)
- NetBeans IDE (https://netbeans.org)
- Sublime Text (http://www.sublimetext.com)
- TextMate (http://macromates.com)

U ovim primjerima ćemo koristiti **Sublime Text 2.**

4) Inspect Element – Pregled konstrukcije stranice iz internet preglednika

Kako je svaka web stranica zapravo jednostavan HTML dokument koji sadrži kod koji Internet preglednik interpretira na određeni način, svaku stranicu koju pregledavamo možemo i dubinski pregledati odnosno vidjeti na koji način je ona kreirana.

Google Chrome



Slika 2 - Google Chrome - Inspect element

U Google Chrome pregledniku napravimo desni klik na nekoj lokaciji na web stranici i odaberemo Inspect Element. Nakon toga se otvara dodatni prozor s prikazom svih komponenti na stranici. Te komponente je moguće i direktno mijenjati kako bi vidjeli promjene odmah, no one naravno neće biti nigdje spremljene i ponovnim učitavanjem stranice će sve takve promjene nestati.

Mozilla Firefox

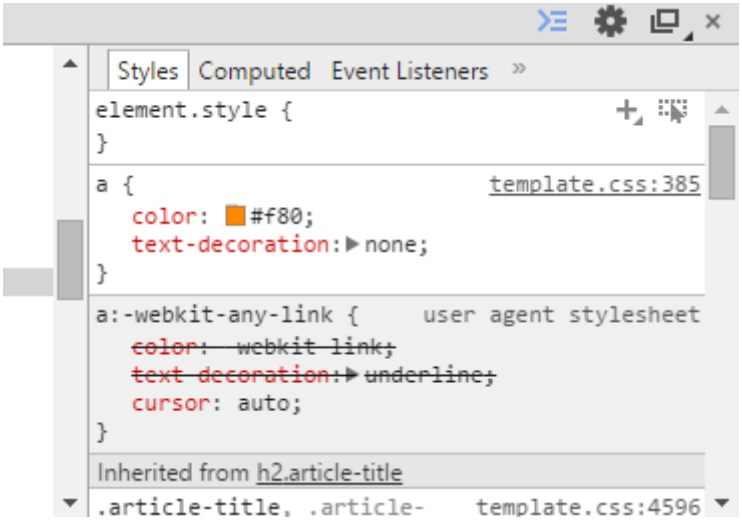
Slika 3 - Mozilla Firefox - Inspect Element

Kao i kod Google Chrome preglednika, napravimo desni klik i odaberemo „Inspect Element“. Otvorit će se novi prozor s prikazom informacija kako je stranica građena.

Pregled stranice kroz konzolu

Slika 4 - Konzola za pregled elemenata na stranici

Za svaki odabrani element možemo vidjeti i njegova svojstva s desne strane konzole.



Slika 5 - Pregled svojstava elemenata

Prikazane vrijednosti možemo mijenjati kako bi vidjeli promjenu. Za primjer je korištena stranica http://www.carpediemka.info i trenutno je odabran naslov članka „Otvorene prijave za izradu Web stranica“.

```
<a href="/index.php/105-otvorene-prijave-za-radionicu-izrade-web-stranica"> Otvorene prijave za izradu web stranica </a>
```

Naslov članka je trenutno narančaste boje.

Otvorene prijave za radionicu izrade Web stranica

Slika 6 - Naslov članka za uređivanje

Korištenjem konzole ćemo lako promijeniti boju u recimo crvenu. Treba samo paziti da je u konzoli odabran ovaj element.

```
▼<h2 class="article-title">
  <a href="/index.php/105-otvorene-prijave-za-radionicu-izrade-web-stranica"> Otvorene prijave za radionicu izrade Web stranica</a>
 </h2>
```

Slika 7 - Odabir elementa u konzoli

S desne strane konzole vidimo svojstva tog elementa i jedno od svojstava je boja teksta.

```
a {                                template.css:385
   color: #f80;
   text-decoration: none;
}
```

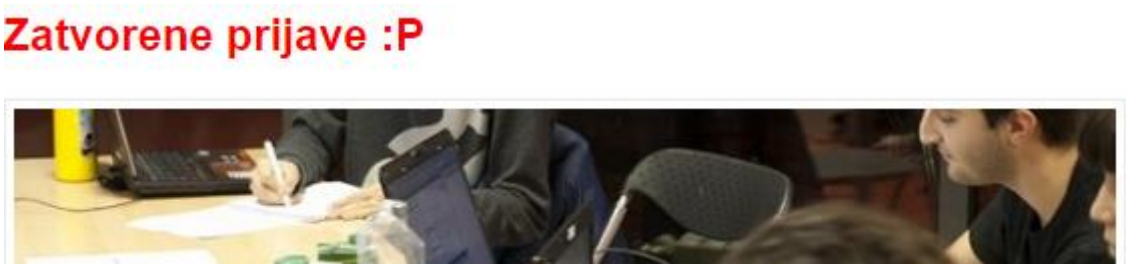
Slika 8 - Pregled boje elementa

Korištenjem konzole ćemo sada promijeniti boju teksta u naslovu u crvenu tako da kliknemo na narančastu boju (***#f80***) i upišemo *„**red**“.*

```
a {                                    template.css:385
   color: red;
   text-decoration: ▶ none;
}
```

Slika 9 - Promjena boje u crvenu

Pogledamo li naslov, vidjet ćemo da je on sada crvene boje.



Slika 10 - Promijenjena boja naslova u crvenu

Na sličan način možemo promijeniti i **tekst naslova**. U konzoli pronađemo gdje je upisan tekst naslova

```
▼<h2 class="article-title">
   <a href="/index.php/105-otvorene-prijave-za-radionicu-izrade-web-stranica"> Otvorene prijave za radionicu izrade Web stranica</a>
 </h2>
```

Slika 11 - Naslov članka

odaberemo (dvoklik) tekst koji se ispisuje kao naslov.

```
stranica"> Otvorene prijave za radionicu izrade Web stranica</a>
```

Slika 12 - Promjena teksta naslova 1

i promijenimo tekst koji piše u nešto drugo.

```
rade-web-stranica">Zatvorene prijave :P</a>
```

Slika 13 - Promjena teksta naslova 2

Na stranici će sada biti prikazan naslov s izmijenjenim tekstom i izmijenjenom bojom.

Slika 14 - Izmijenjeni naslov

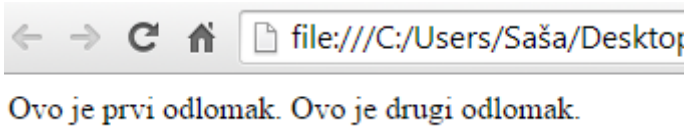
5) HTML elementi

HTML elementom smatramo sve što je napisano u kodu od početka nekog HTML taga do kraja HTML taga. Vidjet ćemo da je princip pisanja uvijek isti <tag> sadržaj </tag>. Treba ima na umu kako skoro svi tagovi imaju svoju **otvorenu** i **zatvorenu** varijantu.

5. a paragraf

Paragraf možemo smatrati odlomkom. Dakle, to je skupina teksta. Iako tekst možemo napisati direktno unutar <body> </body> vidjet ćemo zašto to nije ispravan način i što se događa sa strukturom stranice.

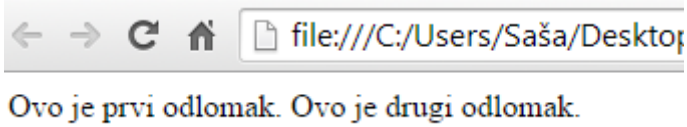
```
<body>
        Ovo je prvi odlomak.
        Ovo je drugi odlomak.
</body>
```



Slika 15 - Odlomci 1

Vidimo da je tekst napisan u jednom redu, a nije odvojen u dva reda kako smo napisali unutar HTML editora. Razlog tome je što prazni prosto u editoru (*whitespace*) zapravo ničem ne služi osim za ugodniji prikaz koda samom *developeru*.

```
<body>
        Ovo je prvi odlomak.
        Ovo          je drugi odlomak.
</body>
```

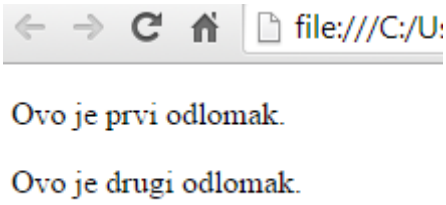


Slika 16 - Odlomci 2

Vidimo kako bez obzira na whitespace (upisano je desetak razmaka između „Ovo“ i „je“), prikaz na stranici će izgledati drugačije odnosno suvišni whitespace će biti ignorira.

Ispravan kod bi trebao izgledati ovako:

```
<body>
        <p>Ovo je prvi odlomak.</p>
        <p>Ovo je drugi odlomak.</p>
</body>
```



Slika 17 - Odlomci 3

U ovom slučaju će Internet preglednik znati ispravno interpretirati ovaj tekst koji je sada podijeljen u dva paragrafa.

5. b </br>, <br> break line

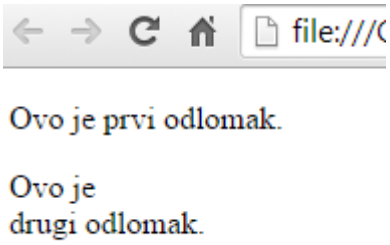
Nastavno na prošli primjer, želimo napraviti prekid u odlomku. Za ovu potrebu koristimo **line break** tag. Za razliku od većine tagova, breakline nema otvorenu i zatvorenu varijantu nego samo ili otvorenu ili zatvorenu. Ovisno koji standard se primjenjuje, koristi se ili otvoreni ili zatvoreni.

 - otvoreni – koristi se u HTML

</br> - zatvoreni – koristi se u XHTML

MI ćemo dakle koristiti <br> tag koji je naveden po HTML specifikaciji. Iako je moguće koristiti i </br>, preporuka je držati se specifikacije.

```
<body>
        <p>Ovo je prvi odlomak.</p>
        <p>Ovo je <br> drugi odlomak.</p>
</body>
```

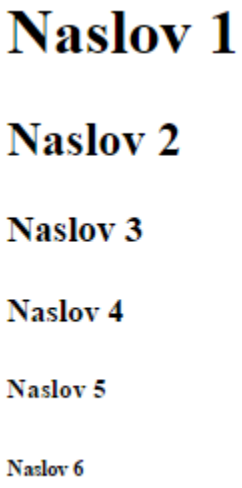


Slika 18 - Prijelom linije

5. c <h1></h1, <h2> Headings </h2>…<h6></h6>

Heading tagovi se koriste za definiranje naslova i razlikujemo šest vrsta ovih tagova. Što je veći broj uz slovo „h“ odnosno što je taj tag većeg broja, njegova vrijednost je manja. Iako ćemo ubrzo vidjeti kako se oni grafički razlikuju, treba imati na umu kako imaju i različite logičke vrijednosti. Tako primjeri <h1> treba koristiti za obilježavanje najvažnijeg teksta. Nešto manje važnu informaciju obilježavamo s tagom <h2> i tako dalje.

```
<body>
        <h1>Naslov 1</h1>
        <h2>Naslov 2</h2>
        <h3>Naslov 3</h3>
        <h4>Naslov 4</h4>
        <h5>Naslov 5</h5>
        <h6>Naslov 6</h6>
</body>
```



Slika 19 – Headings

5. d Link (poveznica)

Ako želimo napraviti link (poveznicu), koristimo element <a> koji unutar sebe sadrži **atribut href. Atribute** ćemo uskoro upoznati detaljnije. Važno je upamtiti kako se piše ovaj element. Prvo otvaramo tag <a> unutar kojeg imamo atribut href. Za sada bi kod izgledao ovako:

```
<a href>
```

Zatim je potrebno dodijeliti poveznicu tako da atributu href dodijelimo adresu na koju nas treba odvesti, obavezno unutar dvostrukih navodnika pa će to sada izgledati ovako:

```
<a href = „http://www.google.com“>
```

Nakon što smo završili s otvorenim tagom, upisujemo tekst koji će se pojaviti na stranici. Imajte na umu da link koji smo naveli za href atribut nije vidljiv na stranici kao tekst. Kod će sada izgledati ovako:

```
<a href = „http://www.google.com“> Google tražilica
```

Na stranici će se sada prikazati tekst Google tražilica na koji je moguće kliknuti te će nas odvesti na dodijeljenu web adresu (www.google.com). Još samo trebamo kreirati zatvoreni <a> tag kako bi sve bilo ispravno napisano. Završna verzija koda će izgledati ovako:

```
<a href=“http://www.google.com“>Google tražilica</a>
```

Izrada linka za slanje Emaila

Korištenjem <a> taga lako možemo napraviti i link koji će omogućiti pokretanje Email klijenta kada korisnik klikne na taj link.

```
<a href=“mailto:a@example.com “>Pošalji Email Saši</a>
```

Važno je uočiti ključnu riječ „**mailto**“ koja označava da se radi o slanju elektroničke pošte te će rezultirati pokretanjem Email klijenta na korisnikovom računalu.

5. e Charset – prepoznavanje hrvatskih znakova

Ako ste napisali identično kod kako je to gore navedeno, možete primijetiti kako se slovo „ž“ ne prikazuje ispravno. To je zato što Internet preglednik ima određeni način na koji interpretira simbole te u ovom slučaju slovo „ž“ nije opisano kako da ga interpretira.

Google traÅ¾ilica

Slika 20 - Character encoding problem

Rješenje problema je jednostavno. Trebamo samo specificirati da se koristiti UTF-8 standard. Ovakva podešavanja zovemo **Character Encoding**. Ako vas više zanima, bacite oko na Google ☺.

Sjetimo se da ovakve tehničke informacije stavljamo unutar glave dokumenta odnosno unutar <head></head> tagova. Ovaj put nam treba element **<meta>** unutar kojeg specificiramo atribut **charset**.

```
<meta charset="UTF-8">
```

Cijeli kod unutar <head> će sada izgledati ovako:

```
<head>
        <meta charset="UTF-8">
</head>
```

Google tražilica

Slika 21 - Character encoding ispravno

Vidimo da se sada tekst ispravno prikazuje i ako kliknemo na njega, odvest će nas na stranicu Google tražilice.

6) Atributi i ugniježđeni tagovi/elementi

6. a Atributi

Kao što smo vidjeli kod <a> taga, moguće je, a ponekad i potrebno dodavati atribute tagovima. Atributi dodatno opisuju/specificiraju određeni tag. Detaljno ćemo se s atributima upoznavati kako prolazimo kroz tagove.

Primjeri:

```
<a href="http://www.google.com">Poveznica na Google</a>
```

Atribut je *href* kojem je pridodana vrijednost „http://www.google.com“

```
<img src="http://full-monty.net/wp-content/uploads/2014/01/cropped-Full-Monty1.png"
width="104" height="142">
```

Tag <img> (koristimo za umetanje slika) u ovom primjeru ima tri atributa. Atribut *src* opisuje lokaciju slike koja se u ovom slučaju nalazi na adresi „http://full-monty.net/wp-content/uploads/2014/01/cropped-Full-Monty1.png“. Dodatno je tu atribut *width* koji specificira širinu slike te atribut *height* koji specificira visinu slike.

6. b Ugniježđeni tagovi

Osim jednostavnog pisanja tagova jednog ispod drugog, većinu vremena ćete provoditi kreirajući elemente jednog unutar drugog. Pritom je važno paziti na pravovremeno zatvaranje otvorenih tagova.

7) Komentari

Komentari su jako važan dio pisanja bilo kojeg koda. Komentari se neće prikazati na stranici niti će se ono što obilježeno kao komentar izvršiti/procesirati. Njihova svrha je obilježavanje i pojašnjavanje napisanog koda samom programeru. Posebice su korisni ukoliko na nekom projektu sudjeluje više osoba i kada je potrebno raditi naknadne prepravke jer pomoću njih možemo lijepo opisati što koji dio koda predstavlja.

Komentare pišemo unutar **<!-- -->** simbola

```
<p> Prvi naslov </p>
<!-- </p> Drugi naslov </p> -->
```

Prvi naslov

Slika 22 – Komentari 1

Kako vidimo na slici, ispisan je samo tekst „Prvi naslov“ a drugi tekst („Drugi naslov“) nije prikazan jer je obilježen kao komentar.

Ipak, ako bi pogledali konstrukciju stranice iz Internet preglednika, vidjet ćemo da je komentar ipak vidljiv.

```
<!DOCTYPE html>
▼<html>
  ▶<head>…</head>
  ▼<body>
      <p> Prvi naslov </p>
      <!-- </p> Drugi naslov </p> -->
    </body>
  </html>
```

Slika 23 - Komentari 2

8) <img> Slike

8. a Umetanje slike

<img> tag koristimo prilikom umetanja slike u HTML dokument. Atribut koji ćemo uvijek navoditi je *src* koji određuje lokaciju slike. Slika se može nalaziti na našem računalu ili negdje na internetu odnosno na nečijem serveru.

Primjer kada želimo u dokument ubaciti sliku s našeg računala koja se nalazi na desktopu:

```
<img src="C:\Users\Saša\Desktop\CarpeDiem.jpg">
```

Primijetite kako je uz naziv slike potrebno upisati i ekstenziju datoteke koja je u ovom slučaju jpg. Ako bi se slika nalazila negdje na internetu odnosno na nekom serveru, primjer koda bi izgledao ovako:

```
<img src="http://www.codeproject.com/KB/GDI-plus/ImageProcessing2/img.jpg">
```

U ovom drugom primjeru vidimo da se slika povlači negdje s interneta i ponovno je potrebno unijeti potpunu lokaciju s ekstenzijom datoteke. Ako bih sada želio promijeniti dimenzije slike to mogu napraviti korištenjem atributa za širinu i visinu pa bi kod izgledao ovako:

```
<img src="http://www.codeproject.com/KB/GDI-plus/ImageProcessing2/img.jpg" width=100
height=200>
```

8. b Umetanje slike koja je poveznica

Ako bismo željeli napraviti da kada korisnik klikne na sliku da ga se odvede na neku drugu stranicu, potrebno je <img> tag ubaciti unutar <a> taga. Ovdje vidimo primjer ugniježđenih tagova.

```
<a href="http://www.carpediem.hr"><img src="http://www.codeproject.com/KB/GDI-
plus/ImageProcessing2/img.jpg" width=100 height=200></a>
```

Vidimo da smo započeli s tagom <a> te smo njegovom atributu *href* dodali lokaciju na koju želimo da nas odvede i zatim navodimo što će biti to na što korisnik treba kliknuti kako bi otišao na http://www.carpediem.hr stranicu. U ovom slučaju radi se o <img> elementu (slici) koju povlačimo s interneta.

9) <table> Tablice

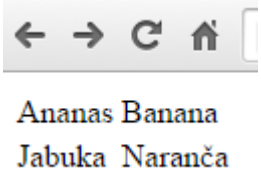
Tablice su još jedan važan HTML element. Još uvijek možete naići na stranice koje su više-manje u cijelosti kreirane korištenjem tablica. Ovakav pristup je zastario i smatra se lošim, no do pred 10 godina su mnogi kreirali stranice koristeći tablice.

9. a <table> i <td> Izrada tablica

<table> </table> su tagovi koje koristimo za kreiranje tablica. Unutar ta dva taga navodimo koliko će tablica sadržavati redova i stupaca. Tag **<tr>** (Table row) označava jedan red u tablici. Unutar svakog reda stavljamo **<td>** koji označavaju Table Data na osnovu kojih se kreiraju stupci.

```
<table>
        <tr>
                <td>Ananas</td>

                <td>Banana</td>
        </tr>
        <tr>
                <td>Jabuka</td>
                <td>Naranča</td>
        </tr>
</table>
```

Ovaj kod će stvoriti tablicu s dva reda i dva stupca.

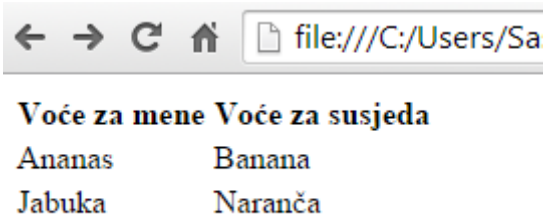


Slika 24 - Stvaranje tablice

9. b <th> Naslovi stupaca

Uz <td> tag, koristimo i <th> Table Header koji bi trebao opisivati podatke u pojedinom stupcu. Pošto se smatra „važnijim“ od podataka, biti će i drugačije prikazan.

```
<table>
    <tr>
        <th>Voće za mene</th>
        <th>Voće za susjeda</th>
    </tr>
    <tr>
        <td>Ananas</td>
        <td>Banana</td>
    </tr>
    <tr>
        <td>Jabuka</td>
        <td>Naranča</td>
    </tr>
</table>
```



Slika 25 - Stvaranje naslova stupaca

9. c Colspan – proširi redak na širinu dva ili više stupaca

Ako želimo kreirati tablicu koja ima primjerice jedan glavni stupac pod kojim se nalaze ostali stupci možemo koristiti tag <th> ili <td> ali moramo specificirati atribut **colspan**. Ovom atributu dodajemo vrijednost koja opisuje širinu jednog retka naspram broju stupaca. U našem slučaja je to broj 2 pa će se taj jedan redak proširiti na dimenziju dva stupca koja se nalaze ispod njega.

```
<table>
        <tr>
                <th colspan="2">Shopping lista </th>
        </tr>
        <tr>
                <th>Voće za mene</th>
                <th>Voće za susjeda</th>
        </tr>
        <tr>
                <td>Ananas</td>
                <td>Banana</td>
        </tr>
        <tr>
                <td>Jabuka</td>
                <td>Naranča</td>
        </tr>
</table>
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 26 – Colspan

Dodatno možemo dodati crni obrub kako bi lakše uočili kostur tablice. Imajte na umu da ovaj dio već polako ulazi u CSS koji se obrađuje nakon HTML dijela.

```
<table border=1 >
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 27 - Border tablice

9. d Širina tablice i ćelija

Do sada smo vidjeli da se tablica automatski prilagođava širini teksta koji se nalazi unutar neke ćelije.

Širina tablice

Ako želimo točno precizirati širinu tablice, možemo korištenjem atributa width odrediti širinu u pikselima (px).

```
<table border=1 width=500px>
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 28 - Uređivaje širine tablice 1

Širinu tablica odnosno ćelija je moguće postaviti i korištenjem postotka. Pogledajmo ovaj primjer:

```
<table border=1 width=50%>
```

Shopping lista	
Voće za mene	**Voće za susjeda**
Ananas	Banana
Jabuka	Naranča

Slika 29 - Uređivanje širine tablice 2

Vidimo da je sada tablica zauzela 50% prostora onoga u čemu se nalazi. U ovom slučaju se ona nalazi unutar HTML dokumenta (često će se nalaziti unutar nekog drugog HTML elementa) i sada zauzima 50% širine cijelog dokumenta.

Naravno, i kod mnogih drugih elemenata je moguće na više načina korigirati širinu, visinu, udaljenost od drugog elementa, itd.

Širina i udaljenost ćelija

Korištenjem **cellpadding** atributa možemo odrediti razmak između sadržaja u ćeliji i njenih obruba.

```
<table border=1 width=200px cellpadding=10>
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 30 - Cellpadding 10px

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 31 - Cellpadding 2px

Vidimo kako je u prvoj tablici svaki tekst u svakoj ćeliji udaljen 10px od rubova te ćelije. U drugoj tablici je cellpadding atributu dodijeljena vrijednost 2.

Uz udaljenost sadržaja ćelije od njenih rubova, možemo utjecati i na širinu ćelije neovisno o količini sadržaja u ćeliji korištenjem atributa **width**.

```
<table border=1 cellpadding=2>
                    <tr>
                        <th colspan="2">Shopping lista </th>
                    </tr>
                    <tr>
                        <th width=250px >Voće za mene</th>
                        <th>Voće za susjeda</th>
                    </tr>
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 32 - Cell width

Treći atribut koji ćemo upoznati je **cellspacing**. On nam omogućava postavljanje razmaka između ćelija unutar tablice.

```
<table border=1 cellspacing=20>
                    <tr>
                        <th colspan="2">Shopping lista </th>
                    </tr>
                    <tr>
                        <th width=250px >Voće za mene</th>
                        <th>Voće za susjeda</th>
                    </tr>
```

Shopping lista	
Voće za mene	Voće za susjeda
Ananas	Banana
Jabuka	Naranča

Slika 33 – Cellspacing

Vidimo da je sada svaka ćelija od svake udaljena 20px. Lijevi stupac tablice ima širinu 250px dok za desni stupac nije definirana širina odnosno automatski se prilagođava ovisno o tekstu upisanom u ćelije.

10)   Liste

U HTML-u liste dijelimo na dvije vrste; sortirane i nesortirane liste. Sortirana lista će ispred svake stavke dodati broj, dok će nesortirana lista dodati kružić.

  su tagovi za sortiranu list – Ordered List.

 su tagovi za nesortiranu listu – Unordered list.

Nakon što specificiramo kakvu listu radimo, dodajemo u nju stavke. Svaka stavka ima svoj HTML tag a to je <li> - List item.

```
<ul>
        <li>Auto</li>
        <li>Brod</li>
        <li>Motor</li>

</ul>
```

- Auto
- Brod
- motor

Slika 34 - Nesortirana lista

```
<ol>
        <li>Auto</li>

        <li>Brod</li>

        <li>Motor</li>

</ol>
```

1. Auto
2. Brod
3. Motor

Slika 35 - Sortirana lista

11) DIV

DIV objekti se učestalo koriste prilikom izrade Web stranica jer za razliku od tablica, omogućavaju puno fleksibilniji prikaz te nude određena svojstva koje tablice ne nude. DIV objekt zamislite kao kutiju unutar koje stavljate ostale elemente.

```
<div>
                        <table>
                                <tr>
                                        <td>Jabuke</td>
                                        <td>Kruške</td>
                                        <td>Mandarine</td>
                                </tr>
                                <tr>
                                        <td>Špinat</td>
                                        <td>Paprika</td>
                                </tr>
                        </table>
</div>
```

Jabuke Kruške Mandarine
Špinat Paprika

Slika 36 - DIV 1

Kao što vidimo, u prikazu nema razlike u odnosu da kreiramo tablicu bez DIV objekta no uskoro ćemo vidjeti zašto se DIV objekti često koriste te kako ih koristiti.

12) Tekst i mogući tagovi

Do sada smo vidjeli kako tekst možemo pisati unutar paragrafa/odlomaka korištenjem HTML taga ** ** i kao naslove korištenjem nekog od **<h></h>** tagova gdje uz slovo „h“ navodimo broj između 1 i 6 pri kojem manji broj označava veću prioritetnost. Za razdvajanje teksta u redove smo koristili tag **
**.

12. a   Podebljani (boldani) tekst

Želimo li podebljati neki tekst koristimo HTML tag ****. Unutar taga navodimo tekst koji želimo podebljati. Unutar jednog paragrafa je moguće kombinirati više podebljanih dijelova teksta. Iako izgledaju identično, razlika je u logičkom razumijevanju ova dva taga. **Strong** bi trebao označavati vrstu teksta (važan tekst) tekst dok je **Bold** zapravo stil fonta.

```
<p>Tekst koji je <b>boldan</b></p>
```

Drugi tag koji možemo koristiti je **.** Iako daju

Tekst koji je **boldan**

Slika 37 - Podebljani tekst

12. b   Nakonešni (italic) tekst

Želimo li nakositi neki tekst, stavljamo ga unutar **** tagova. Kao i kod prethodnog taga, moguće je kombinirati više tagova unutar jednog paragrafa.

```
<p>Tekst koji je <i>nakošen</i></p>
```

Postoji i drugi HTML tag koji će napraviti isti rezultat (barem vizualno). To je HTML tag ****. Iako izgledaju identično, razlika je u logičkom razumijevanju ova dva taga. **Emphasized** bi trebao označavati vrstu teksta dok je **italic** zapravo stil fonta.

```
<em>Tekst koji je <i>nakošen</em></p>
```

Tekst koji je *nakošen*

Slika 38 - Nakonešni tekst

12. c Podvučeni (underline) tekst

Ako želimo podvući neki tekst, koristimo **<u></u>** tagove. Također, kao i kod prethodna dva, moguće je kombinirati više tagova unutar jednog paragrafa odnosno HTML elementa.

```
<p>Tekst koji je <u>podvučen</u></p>
```

Tekst koji je podvučen

Slika 39 - Podvučeni tekst

12. d Povećani tekst

Želimo li neki dio teksta povećati možemo koristiti HTML tag **<big></big>**.

```
<p>Tekst koji je <big>povećan</big></p>
```

Tekst koji je povećan

Slika 40 - Povećani tekst

12. e Smanjeni tekst

Želimo li pak neki dio teksta smanjiti u odnosu na ostatak teksta, možemo koristiti **<small></small>**.

```
<p>Tekst koji je <small>smanjen</small></p>
```

Tekst koji je smanjen

Slika 41 - Smanjeni tekst

12. f <del> Uklonjen tekst

Iako se možda na prvu čini čudno uopće koristiti tag za neki tekst koji je obrisan, vidjet ćemo da ovaj tag imaju svoju primjenu posebice u novinarstvu. Tekst obilježen **<del></del>** tagom neće biti obrisan (korisnik će i dalje vidjeti taj tekst) nego će biti prekrižen. Na taj način se sugerira kako taj tekst više važeći već da postoji novi tekst.

```
<p>Proizvođač mog laptopa je <del>HP</del> Asus.</p>
```

Proizvođač mog laptopa je ~~HP~~ Asus.

Slika 42 - Obrisani tekst

12. g <s> Netočan, irelevantan, nevažeći tekst

Korištenjem **<s>** taga obilježavamo tekst koji više nije ispravan tj. točan. Primjerice radimo analizu najboljeg mobitela na svijetu i napišemo da je Mobitel 1 najbolji, no netom nakon objave članka se pojavio još bolji mobitel koji je ipak bolji od navedenog.

```
<p>Najbolji mobitel je <s>iPhone</s> Nexus.</p>
```

Najbolji mobitel je ~~iPhone~~ Nexus.

Slika 43 - Nevažeći tekst

Iako izgledom **<del> i <s> izgledaju isto**, razlika je opet u semantici tj. logici korištenja. <del> tag bi trebali koristiti samo za obilježavanja teksta koji bi trebao biti uklonjen dok bi za netočan ili nevažeći tekst trebali koristiti <s> tag.

13) Forma, polja i dugme

Kada želimo kreirati HTML formu, koristimo tagove **<form></form>**. Unutar forme postavljamo elemente poput polja za unos teksta, odabir datuma, razne dugme (button) i slično.

Kada napišemo samo ovaj kod:

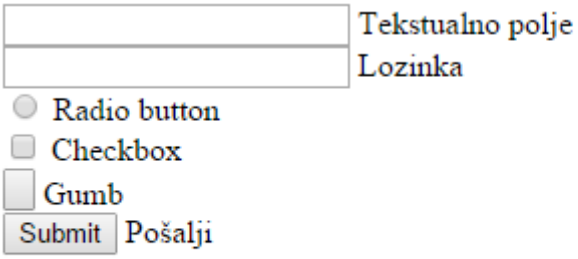
```
<form>

</form>
```

Primijetiti ćemo da se ništa ne prikazuje kada otvorim HTML dokument. Tim kodom smo samo kreirali formu u koju postavljamo druge HTML elemente.

13. a <input> elementi

Input element je jedan od najkorištenijih elemenata unutar formi i postoji više vrsta ovog elementa. Atributu **type** dodajemo jednu od moguće tri vrijednosti kako bi odredili vrstu elementa.

```
<form>
        <input type="text"> Tekstualno polje <br>
        <input type="password"> Lozinka <br>
        <input type="radio"> Radio button <br>
        <input type="checkbox"> Checkbox <br>
        <input type="button"> Gumb <br>
        <input type="submit"> Pošalji
</form>
```



Slika 44 - Input elementi

13. b <input type="text">

Ovaj atribut koristimo kada želimo kreirati polje za unos tekstualne vrijednosti.

```
<form>
        Ime: <input type="text"> <br>
        Prezime <input type="text"> <br>
</form>
```

Ime:
Prezime

Slika 45 - Input type = Text

13. c <input type="password">

Password koristimo kada želimo kreirati polje za unos lozinke. Iako ovo polje izgleda poput polja za unos teksta, prilikom unosa vrijednosti prikazuju se točkice kako bi sakrile ono što je napisano.

```
<form>
        Korisničko ime: <input type="text"> <br>
        Lozinka: <input type="password"> <br>
</form>
```

Korisničko ime: Sasa
Lozinka: ••••••••

Slika 46 - Input type = password

13. d <input type=email">

Ovaj atribut će kreirati element za unos adrese elektroničke pošte. Iako izgleda poput klasičnog polja za unos teksta, posjetimo li stranicu s mobilnog uređaja, mnoge tipkovnice će prepoznati da se radi o polju za unos elektroničke pošte pa će izgled tipkovnice prilagoditi tome (dodat će se @ znak u prvi plan).

Unesi Email: info@carpediem.hr

Slika 47 - input type = email

13. e <input type="radio">

Radio type koristimo kada želimo kreirati izbor više opcija, no želimo dopustiti odabir samo jedne od mogućih opcija. Primjer toga je odabir spola, grada u kojem osoba rođena i slično.

```
<form>
        <p> Spol: </p> <br>
        <input type="radio">Muško <br>
        <input type="radio">Žensko <br>
</form>
```

Spol:

Muško
Žensko

Slika 48 - input type = radio

13. f <input type="checkbox">

Za razliku od *radio* kontrole, *checkbox* omogućava odabir više opcija iz nekog popisa. Ove elemente možemo koristiti kod kreiranja popisa izbornih predmeta gdje student mora odabrati dva ili više predmeta.

```
<form>
        <p> Predmet: </p> <br>
        <input type="checkbox">Programiranje <br>
        <input type="checkbox">Web <br>
        <input type="checkbox">Mreže <br>
</form>
```

Predmet:

Programiranje
Web
Mreže

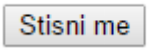
Slika 49 - Input type = checkbox

13. g <input type="button">

Button koristimo kada želimo kreirati običan gumb. Prilikom kreiranja samog elementa, u gumbu se neće ispisati nikakav tekst što najčešće nije opcija koju želimo napraviti. Ako već radimo gumb koji služi nekoj vrsti interakcije s korisnikom, na gumb želimo napisati neki tekst.

Za potrebu upisivanja teksta u gumb koristimo atribut **value** kojem pridodajemo vrijednost teksta koja će biti upisana u taj gumb.

```
<form>
        <input type="button" value="Stisni me">
</form>
```

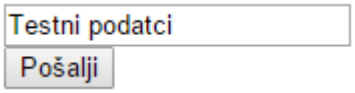


Slika 50 - input type = button

13. h <input type="submit">

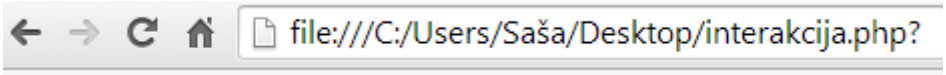
Submit se izgledom ne razlikuje od klasičnog button elementa no njihova funkcionalnost je različita. Submit button se koristi za slanje forme. Kako bi gumb postao aktivan, potrebno je samoj formi dodati atribut **action** kojem dodajemo vrijednost što se događa kada se pritisne gumb Submit. Za slanje se koriste dvije metode; POST i GET metoda koje ćemo obraditi kasnije.

```
<form action="interakcija.php">
        <input type="text"> <br>
        <input type="submit" value="Pošalji">
</form>
```



Slika 51 - input type = submit

Pritiskom na gumb, otvara se stranica interakcija.php.



Slika 52 - Otvaranje stranice nakon Submit akcije

13. i <input type="reset">

Kao što imamo poseban gumb za slanje forme, imamo i poseban gumb za resetiranje podataka upisanih u formu. Ovaj gumb će sve upisane podatke unutar forme u kojoj se nalazi vratiti na početno stanje

```
<form>
        <input type="reset">
</form>
```

Reset

Slika 53 - input type = reset

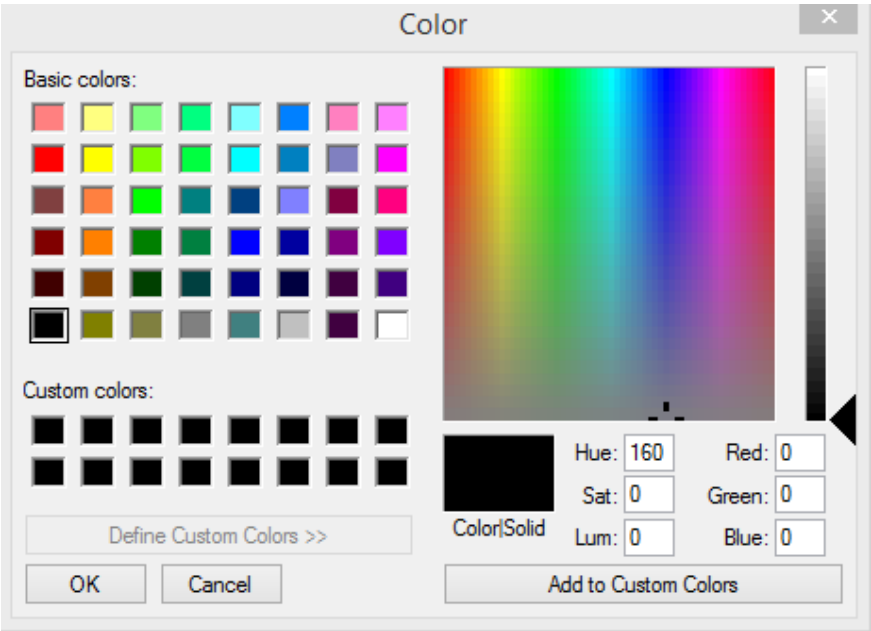
13. j <input type="color">

Odabir boja (Color picker) je predstavljen u HTML5 standardu. Kao što ime kaže, omogućava postavljanje elementa za odabir boje u HTML dokument.

```
<input type="color"> Odaberi boju
```

Odaberi boju

Slika 54 - input type = color



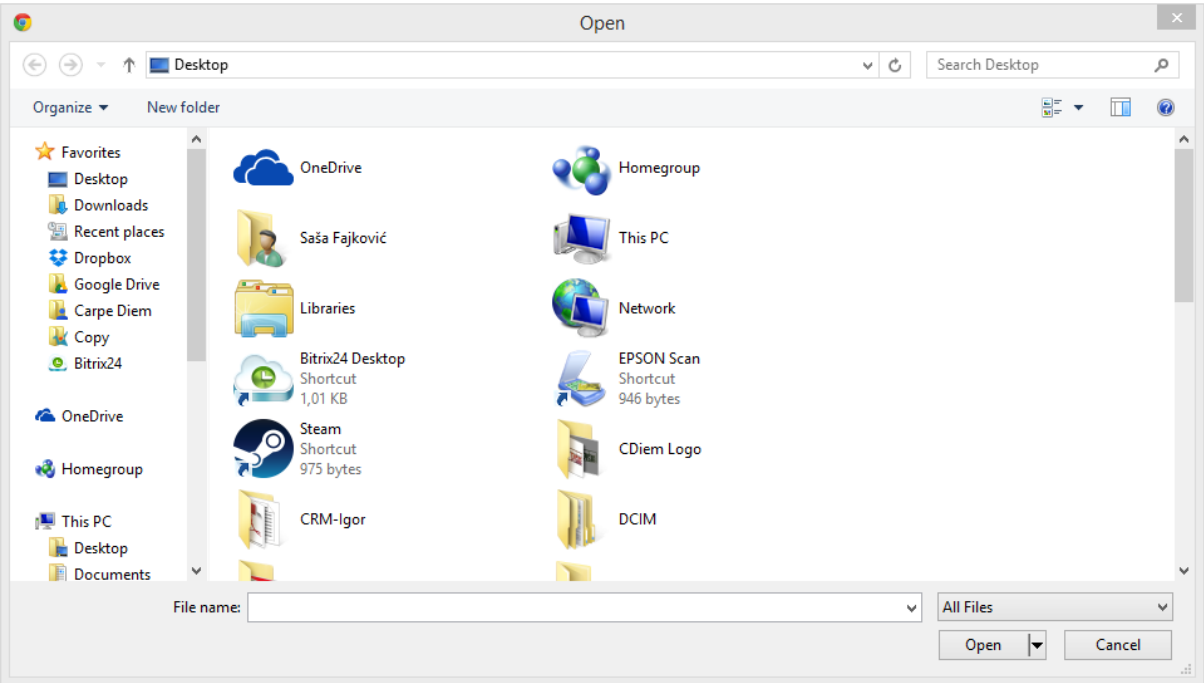
Slika 55 - Color picker

13. k <input type="file">

Ponekad očekujemo od korisnika da doda datoteku prilikom primjerice slanja životopisa. U tom slučaju, koristimo HTML element koji omogućava pretraživanje datoteka na korisnikovom računalu.

Odaberi datoteku: Choose file No file chosen

Slika 56 - input type = file

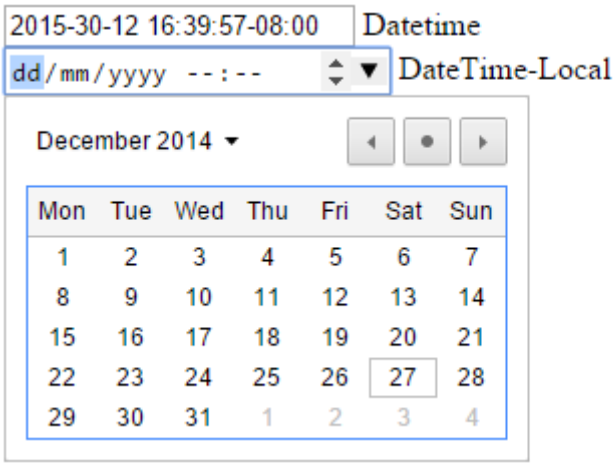


Slika 57 - Odabir datoteke

13. l <input type="datetime">

Ako želimo korisniku omogućiti odabir datuma i vremena, koristimo **datetime**. Ovom kontrolom korisnik može definirati dan, mjesec, godinu, sate, minute i sekunde. Pritom razlikujemo dvije varijacije ovog elementa – **datetime** i **datetime-local**. Prva verzija će bazirati vrijednosti na osnovi UTC vremenske zone dok se druga verzija neće bazirati na niti jednoj vremenskoj zoni.

```
<form>
        <input type="datetime" value="2015-30-12 16:39:57-08:00"> Datetime <br>
        <input type="datetime-local" value="2015"> DateTime-Local
</form>
```

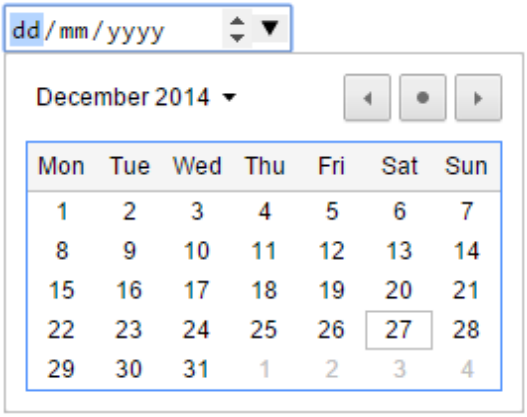


Slika 58 - input type = datetime

13. m <input type="date">

Ako želimo omogućiti unos samo datuma (bez vremenske komponente) koristimo **date**.

```
<form>
        <input type="date">
</form>
```

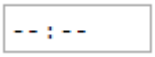


Slika 59 - input type = date

13. n <input type="time">

Ako želimo omogućiti unos samo vremena (bez datuma), koristimo **time**.

```
<form>
        <input type="time">
</form>
```



Slika 60 - Input type = time

13. o Ostali tipovi

Do sada spomenuti tipovi elemenata unutar forme nisu svi. Ako želite, pogledajte kako rade i ovi elementi:

hidden	month	number	range	search	tel	url	week

14) POST i GET metode

POST i GET su dvije metode koje dio **HTTP** protokola. HTTP je kratica za **Hypertext Transfer Protocol**. Ovja protokol nam omogućava komunikaciju između klijentskog računala i serverskog računala, a komunikacija se odvija dvostrano po principu upita i odgovora. To znači da klijentsko računalo (dakle osoba koja posjećuje Web stranicu) šalje upit na server. Taj upit može biti da se učita neki HTML dokument. Server obrađuje upit i šalje podatke korisniku kako i taj HTML dokument se prikazuje.

Često korištene metode u sklopu HTTP-a su **GET** i **POST.**

14. a GET metoda

GET metodu koristimo za dohvaćanje podataka s nekog striktno određenog odredišta. Podatci dobiveni GET metodom se spremaju u *cache* memoriju, ostaju pohranjeni u povijesti pretraživanja Internet preglednika i možemo kreirati *bookmarke* na osnovu tih podataka. S obzirom da su podatci koji se šalju GET metodom vidljivi u adresnom polju (*Adress Bar*), nikako ne bismo smjeli obrađivati osjetljive informacije ovom metodom (primjerice lozinke, bankovne kartice i sl.). Preporuka je da se ova metoda koristi samo za dohvaćanje podataka.

14. b POST metoda

Za razliku od GET metode, koristimo li POST metodu podaci koje šaljemo prema serverskom računalu neće biti vidljivi u *Adress Baru*. Podatci dobiveni POST metodom se ne spremaju u *cache* memoriju, ne spremaju se u povijest pretraživanja i ne mogu se *Bookmarkati*. S obzirom da podatci koji se šalju prema serveru nisu vidljivi, ovu metodu možemo koristiti za slanje osjetljivih informacija prema serveru.

15) Uvod u CSS

CSS (Cascading Style Sheets) jezik je kojim se opisuje izgled i ponašanje HTML elemenata na Internet stranicama. CSS ima vrlo jednostavnu sintaksu koju lako možemo čitati te se sastoji od nekoliko osnovnih elemenata:

```
h1 {
        color: black;
        text-align: left;
}
```

Element *h1* predstavlja **selektor** te nam njegov naziv govori na koji HTML element se CSS kod u vitičastim zagradama odnosi. Unutar vitičastih zagrada nalaze se **deklaracije** koje opisuju izgled odabranog HTML elementa. Deklaracija se sastoji od **svojstva** (*color*) i **vrijednosti** (*#ffffff*) te uvijek završava znakom točka-zarez (*semicolon*). U ovom početnom primjeru, CSS kod odnosi se na sve naslove *h1* kategorije u HTML dokumentu te im boju teksta postavlja u crnu a poravnanje po lijevoj strani.

Gornji CSS kod može se pisati i u sljedećem obliku:

```
h1 { color: black; text-align: left; }
```

no, preporučeno ga je pisati na način prikazan prvim primjerom zbog bolje čitljivosti.

CSS kod može sadržavati i komentare koji pomažu autoru koda (ili drugoj osobi koja će ga u nekom trenutku čitati ili uređivati) pri snalaženju. Komentari se mogu protezati kroz jedan ili više redova, počinju oznakom /* a završavaju oznakom */. Primjer:

```
p {
   color: red;
   /* komentar kroz jedan red */
   text-align: center;
}

/* Komentar kroz
više redova */
```

16) Povezivanje HTML-a i CSS-a

Kako bi HTML elementi poprimili određena obilježja, potrebno je povezati napisani HTML kod sa CSS-om. CSS jezik može se povezati sa HTML dokumentom (stranicom) na više načina:

1. Korištenjem style=“ “ atributa unutar HTML oznaka
2. Pisanjem CSS koda u zaglavlju HTML dokumenta
3. Zapisom u samostalnu .css datoteku i njenim povezivanjem s HTML dokumentom

16. a Korištenje style=“ “ atributa (*Inline style*)

Ovaj način pisanja CSS koda danas se rijetko koristi jer jako smanjuje preglednost zapisanog HTML koda zbog toga što se CSS kod upisuje u atribut unutar HTML oznake (taga). Primjer:

```
<a href="http://www.google.com" style=“text-decoration: none; color: red; font-size:
14px“>Poveznica na Google</a>
```

Unatoč manjku preglednosti, ovakav način zapisa CSS koda nužan je i prihvatljiv pri izradi HTML elektroničke pošte (*e-mail*) koja ne podržava vanjske .css datoteke te cjelokupan kod (HTML i CSS) mora biti u jednoj datoteci. CSS deklaracije unutar *style* atributa razdvajaju se znakom točka-zarez, a isti znak opcionalan je na posljednjoj deklaraciji (ne mora se pisati).

16. b Pisanje CSS-a u zaglavlju HTML-a (*Internal style sheet*)

Pisanje CSS koda u zaglavlju HTML dokumenta jednostavan je i pregledan način zapisa CSS koda. Ovaj način pisanja koristi se kada neka pod-stranica *websitea* treba imati unikatan stil, različit od svih ostalih dijelova *websitea*. Piše se unutar <head>*</head> oznaka HTML-a, a primjer je sljedeći:

```
<head>
<style>
body {
   background-color: black;
}
h1 {
   color: white;
   margin-left: 40px;
}
</style>
</head>
```

Kod ovakvog načina pisanja, napisani CSS kod odnosi se samo na HTML dokument u kojem je napisan.

16. c Pisanje CSS-a u zasebnu datoteku (*External style sheet*)

Ovaj način zapisa CSS-a najprikladniji je kada se napisani stilovi trebaju odnositi na više HTML stranica (*webpage*). Kada uređujemo eksternu CSS datoteku, mijenjanjem koda u njoj utječemo na cijelu stranicu (*website*). Kako bismo povezali HTML dokument sa CSS datotekom, potrebno je u zaglavlju HTML-a napisati sljedeće:

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

Umjesto „style.css“ možemo pisati naziv naše CSS datoteke. Gore napisani kod mora se nalaziti u zaglavlju svakog HTML dokumenta koji čini *website*.

Eksterna CSS datoteka smije sadržavati samo CSS kod i komentare. Nisu joj potrebna nikakva zaglavlja ni podnožja, već se stilovi HTML elemenata nižu na prethodno pokazan način:

```
body {
   background-color: white;
}
h1 {
   color: black;
   margin-left: 20px;
}
```

16. d Kombinirano pisanje CSS-a

Svi spomenuti načini pisanja CSS-a mogu se kombinirati, no u tom slučaju primjenjuju se po standardiziranoj listi prioriteta:

1. Vlastiti stilovi Internet preglednika
2. Eksterna CSS datoteka (External style sheet)
3. CSS u zaglavlju HTML-a (Internal style sheet)
4. Inline style

Stavka s brojem 1 ima manji prioritet od stavke s brojem 2 i tako dalje, drugim riječima, stilovi se mogu prepisati jedan preko drugog ako se na dva različita nađe stil koji se odnosi na isti HTML element. Tako će, na primjer, stil zapisan metodom pod brojem 4 vrijediti iako je prethodno za isti element definiran stil eksternom CSS datotekom.

17) CSS selektori

Kako bi Internet preglednik (*browser*) znao na koji HTML element se CSS kod odnosi, moramo koristiti selektore. Selektori dolaze u više varijanti:

- Selektor elementa
- ID selektor
- Selektor klase
- Selektori pseudo-klasa i pseudo-elemenata
- Selektori atributa

Selektori nam omogućavaju da odaberemo ciljani element i na njega (ili na više njih) primijenimo neki CSS stil.

17. a Selektor elementa

Ovaj tip selektora odnosi se na nazive HTML elemenata i odjednom definira stil za sve elemente tog tipa.

```
p {
   text-align: center;
   color: red;
}
```

Prikazani kod definira da će svi elementi <p> tipa (paragraf) imati crvenu boju teksta koji će biti poravnan po sredini.

Selektori se mogu i grupirati kako bi CSS lista bila preglednija, a to se čini na sljedeći način, odvajanjem naziva zarezom:

```
h1, h2, p {
   text-align: center;
   color: red;
}
```

17. b ID selektor

Ovaj tip elektora koristi ID (identifikator) HTML elemenata kako bi primijenio neki stil na točno određeni element. Identifikator HTML elementa mora biti jedinstven i ne smije se ponavljati više puta unutar istog HTML dokumenta, npr. zaglavlje. HTML elementu se identifikator dodjeljuje na sljedeći način:

```
<div id="naziv-identifikatora">*</div>
```

Takvom elementu bismo CSS stil odredili na sljedeći način:

```
#naziv-identifikatora {
   text-align: center;
   color: red;
}
```

Naziv identifikatora ne smije počinjati brojem!

17. c Selektor klase

Za razliku od identifikatora, više HTML elemenata može dijeliti istu klasu. Korištenje klasa praktičan je način da većem broju elemenata dodijelimo iste postavke prikaza odn. stila. Kako bismo HTML elementu dodijelili klasu, napišemo sljedeće:

```
<div class="naziv-klase">*</div>
```

Kako bismo elementima s definiranom klasom dodijelili neki CSS stil, činimo to na sljedeći način:

```
.naziv-klase {
   text-align: center;
   color: red;
}
```

Ukoliko više različitih elemenata dijeli istu klasu, primjerice <h1>, <h2> i <p>, ciljani element (npr. paragraf s klasom „center“) možemo odabrati na sljedeći način:

```
p.center { /*definirani stilovi*/ }
```

Ostale vrste selektora obradit će se kasnije u skripti.

18) CSS pozadine

Uz pomoć CSS koda, gotovo svakom HTML elementu možemo definirati izgled pozadine. Mogućnosti podešavanja pozadine su mnoge – moguće je birati boju, prozirnost, sliku, položaj slike i ponavljanje uzorka. Elementi koji mogu imati pozadinu su tekst, tablice, elementi, poveznice itd.

18. a Boja pozadine

Boja pozadine može se pridodati nekom HTML elementu na sljedeći način:

```
body {
   background-color: #b0c4de;
}
```

Kod boje možemo napisati na više različitih načina, od kojih su najčešće korištene:

- HEX vrijednosti - npr. #ff0000
- RGB vrijednosti - npr. rgb(255, 0, 0)
- RGBA vrijednosti, odn. prozirne pozadine – npr. rgba(255, 0, 0, 0.7), gdje vrijednost 0.7 predstavlja 30% prozirnosti
- Nazivi boja – npr. red, blue, white itd.

Unutar iste CSS datoteke, moguće je miješati tipove kodova boja tako da za jedan element koristimo HEX vrijednosti, a za drugi npr. RGBA vrijednosti.

Budući da kodovi boja nisu odveć intuitivnog karaktera te kako ne bismo morali pogađati koju boju želimo, kod se može dobiti iz postojeće boje na više načina. Najčešće korišten način jest pomoću alata za grafičku obradu (Adobe Photoshop, GIMP, Paint) koji nam kod odabira boje automatski daju i njen HEX ili RGB kod. Drugi način jest korištenje *color picker* programa (npr. ColorPic) ili *plugina* za Internet preglednike (ColorZilla za Firefox, Color Pick za Chrome).

HEX kodove boja moguće je dobiti odabirom boje na poveznici:

http://www.w3schools.com/tags/ref_colorpicker.asp

18. b Pozadinske slike

HTML elementi kao pozadinu mogu imati sliku koja im se dodaje na sljedeći način:

```
body {
   background-image: url("naziv_slike.jpg");
}
```

Adresa „naziv-slike.jpg“ predstavlja potpunu putanju do slike koju želimo koristiti za pozadinu. Dimenzije odabrane slike ne moraju nužno odgovarati dimenzijama HTML elemenata te se one, ukoliko se to ručno ne definira, ponavljaju horizontalno i vertikalno. Na taj način može se jako malom sličicom prekriti jako velika površina nekog elementa.

Ukoliko želimo da se slika ponavlja samo u jednom smjeru koristimo:

```
background-repeat: repeat-x;
```

za horizontalno ponavljanje ili

```
background-repeat: repeat-y;
```

za vertikalno ponavljanje slike. Ukoliko ne želimo ponavljanje slike, koristimo:

```
background-repeat: no-repeat;
```

Pozadinskoj slici može se odrediti pozicija unutar HTML elementa (ukoliko je slika manja od njega), na primjer:

```
background-position: right top;
```

što će rezultirati time da je pozadinska slika pomaknuta u gornji desni kut HTML elementa.

Pozadinskim slikama mogu se definirati i dodatne postavke:

- Kretanje pozadinske slike – *background-attachment*
- Veličina pozadinske slike – *background-size*

19) CSS tekst

Tekstu u HTML dokumentima možemo mijenjati izgled na puno različitih načina – boju, veličinu, font, ukrase, poravnanje, transformacije, uvlake itd.

```
p.glavni {
        color: #ffffff;
        background-color: #ff0000;
        font-size: 18px;
        text-decoration: underline;
        text-align: center;
        text-transform: uppercase
}
```

Gore napisani kod će tekstu unutar <p> elementa s klasom „glavni“ dati bijelu boju slova, crvenu boju pozadine. Tekst će biti podvučen crtom, poravnan po sredini HTML elementa i pisan velikim slovima (bez obzira na to kako ga napišemo), na primjer:

NEKI TEKST

Osim prikazanih svojstava teksta, postoje i brojna druga te je pokrivanje cjelokupnog materijala preopširno za gradivo ove skripte.

19. a Fontovi

Tekstu unutar HTML dokumenta se uz pomoć CSS-a može promijeniti font što se čini na sljedeći način:

```
p {
   font-family: "Calibri", Arial, sans-serif;
}
```

Oznaka „Calibri“ predstavlja puni naziv fonta koji želimo koristiti za tekst nekog elementa. Ukoliko Internet preglednik ne podržava taj font, koristiti će se drugi navedeni font, u ovom slučaju Arial. Ukoliko ni taj font nije podržan, koristit će se najsličniji generički font iz familije fontova *sans-serif*.

Slika 61- Usporedba serif i sans-serif familije fontova

Za prikaz teksta na ekranima, preporučuje se koristiti *sans-serif* familiju fontova zbog bolje čitljivosti i manjeg zamora očiju.

19. b Stilovi fonta

Osim samog fonta, tekstu se može dodijeliti posebni stil fonta, na primjer:

```
p.glavni {
        font-family: "Calibri", Arial, sans-serif;
        font-style: italic;
        font-weight: bold;
}
```

Tekstu u primjeru font će se promijeniti u Calibri te će biti nakošen i podebljan.

19. c CSS poveznice

Poveznice se pomoću CSS-a uređuju slično kao i obični tekst te vrijede sve iste postavke boja, fontova, veličine, pozadine, ukrasa i sl. Web dizajneri često, iz estetskih razloga, žele ukloniti povlaku ispod teksta poveznica što se može učiniti postavljanjem *text-decoration* svojstva:

```
a {
        text-decoration: none;
}
```

Osim postavki koje poveznice dijele s normalnim tekstom, poveznice se mogu urediti s obzirom na svoje stanje – jesu li prethodno pokrenute, prelazi li se pokazivačem preko njih i sl. Četiri stanja koja poveznice mogu imati su:

1) **a:link** – normalna, neposjećena poveznica
2) **a:visited** – prethodno posjećena poveznica
3) **a:hover** – poveznica u trenutku kada se pokazivačem prijeđe preko nje
4) **a:active** – poveznica u trenutku kada je pritisnuta

Primjerice, ukoliko želimo da poveznica izgleda kao ostatak teksta (crne boje, bez povlake), ali da bude podvučena i promijeni boju u crvenu kada se prijeđe mišem preko nje, moramo napisati sljedeći CSS kod:

```
a:link {
        text-decoration: none;
        color: black;
}

a:hover {
        text-decoration: underline;
        color: red;
}
```

Stanja link, visited, hover i active nazivaju se pseudo-klasama i mogu se kombinirati s normalnim klasama u CSS-u, na primjer:

```
a.naziv-klase:hover { /* svojstva */}
```

20) CSS liste

U HTML jeziku postoje dvije vrste lista: sortirane (*ordered*) i nesortirane (*unordered*). CSS nam omogućuje da liste dodatno uredimo.

Liste se mogu uređivati kao normalan tekst te sva spomenuta svojstva vrijede i za njih. Dodatna mogućnost su *list-style* svojstva koja nam omogućuju da promijenimo markere liste prema želji. Sljedeći kod:

```
ul.a {
        list-style-type: circle;
}
ul.b {
        list-style-type: square;
}
ol.c {
        list-style-type: upper-roman;
}
ol.d {
        list-style-type: lower-alpha;
}
```

dati će sljedeće rezultate:

- Kava
- Čaj
- Coca Cola

- Kava
- Čaj
- Coca Cola

I. Kava
II. Čaj
III. Coca Cola

a. Kava
b. Čaj
c. Coca Cola

Slika 62 - uređivanje sortiranih i nesortiranih listi

Osim unaprijed definiranih opcija, kao markere liste možemo koristiti slike po želji, tako da u CSS-u napišemo:

```
ul {
	list-style-image: url('ime_slike.gif');
}
```

Naziv „ime_slike.gif“ potrebno je zamijeniti sa punom ili relativnom adresom slikovne datoteke koju želimo koristiti kao marker liste. Primjer:

- Kava
- Čaj
- Coca Cola

Slika 63 - korištenje slike za markere liste

CSS-om možemo odrediti i položaj markera – hoće li se oni nalaziti izvan ili unutar sadržaja liste. Sljedećom slikom prikazana su oba slučaja, a oko stavaka liste (<li> elementi) postavljen je obrub kako bi se bolje uočila razlika:

list-style-position: inside:

- Stavak 1
- Stavak 2
- Stavak 3

list-style-position: outside:

- Stavak 1
- Stavak 2
- Stavak 3

Slika 64 - list-style-position

Ukoliko ne definiramo ovo svojstvo, koristit će se vrijednost *outside*.

21) CSS tablice

Izgled HTML tablica može se značajno poboljšati korištenjem CSS-a.

21. a Obrub tablice

```
table, th, td {
        border: 1px solid black;
}
```

Tablica kojoj je CSS stil definiran gornjim primjerom imat će dvostruke obrube (unutarnji elementi tablice + vanjski obrub tablice) zato što između ćelija postoji razmak.

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 65 - Tablica s dvostrukim obrubom

Kako bismo taj razmak uklonili, možemo koristiti *border-collapse* svojstvo:

```
table {
   border-collapse: collapse;
}
```

te na taj način dobivamo sljedeće:

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 66 - Tablica - border-collapse

21. b Dimenzije tablica

Tablicama se CSS-om mogu definirati dimenzije. Ukoliko se ne definiraju, veličina ćelija prilagođavat će se sadržaju i tablica može izgledati nečitko, na primjer:

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 67 - Tablica bez postavki dimenzija

Širina i visina elemenata mijenja se uz pomoć *height* i *width* svojstava, dok se prazan prostor oko sadržaja u ćelijama definira *padding* svojstvom:

```
th, td {
        width: 100px;
        height: 30px;
        padding: 5px;
}
```

Gore napisanim kodom dobivamo sljedeće:

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 68 - Tablica - određena širina, visina i padding

Uočite da se promjenom *padding* svojstva mijenjaju i vanjske dimenzije tablice a ne samo položaj teksta unutar ćelija (premda se *padding* odnosi na ćeliju). Više o tome bit će rečeno u poglavlju **CSS box model**.

21. c Tekst u tablicama

CSS-om se, naravno, može mijenjati izgled i položaj teksta u tablicama, a pritom za sami tekst vrijede sve prethodno spomenute postavke (poglavlje **CSS tekst**). U ovom primjeru pozabavit ćemo se samo s položajem teksta koristeći *text-align* i *vertical-align* svojstva:

```
th {
        text-align: right;
        vertical-align: top;
}
td {
        text-align: center;
}
```

čime dobivamo:

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 69 - Tablica - položaj teksta

21. d Boje tablice

Elementima tablice možemo dodijeliti različite boje pozadine, teksta, obruba i sl. Osim toga, različiti elementi tablice mogu imati različite obrube.

```
table {
        border: 2px solid green;
}
td {
        border: 1px dashed green;
}
th {
        border: 2px solid green;
        background-color: #A7C942;
        color: white;
}
```

Ovaj kod prikazat će sljedeće:

Naslov	Naslov
Ivan	Horvat
Podatak	Kava

Slika 70 - Tablica s određenim bojama

Ćelije ili cijela tablica za pozadinu mogu imati i sliku koja se postavlja na način spomenut u poglavlju **CSS pozadine**.

22) CSS box model

HTML elemente možemo shvatiti kao kućice, odnosno kutije unutar koji se nalazi neki sadržaj. Te kutije sastoje se od sadržaja, *paddinga*, obruba i margina:

Slika 71 - CSS box model

- ***Content*** – sadržaj kutije gdje se pojavljuju tekst i slike.
- ***Padding*** – prostor oko sadržaja, do ruba kutije, koji se postavlja zbog čitljivosti
- ***Border*** – standardni obrub (kao kod tablica)
- ***Margin*** – margine – prozirni prostor oko obruba koji čini razmak između HTML elemenata

Kada nekom HTML elementu definiramo dimenzije (visinu i širinu), zapravo definiramo samo dimenzije *content* kutije. Primjerice, ukoliko napišemo neku kutiju sa sadržajem (recimo element) širine 100px i visine 100px, te zadamo *padding* od 10px sa svake strane, obrub od 2px, i margine od 20px sa svake strane, stvarne dimenzije tog elementa biti će 132x132 piksela (100 + 10 + 2 + 20).

Kada izrađujemo Internet stranicu, potrebno je pravila *box* modela uvijek imati na umu te vrijednosti *paddinga*, margina i obruba uvijek uzimati u obzir!

22. a Sadržaj

Dimenzije sadržaja mogu se odrediti apsolutnim ili relativnim vrijednostima. Apsolutne vrijednosti uvijek će dati iste dimenzije neke HTML „kutije“ sa sadržajem, dok će relativne dimenzije mijenjati veličinu te kutije u ovisnosti o veličini ekrana (ili prozora) Internet preglednika.

CSS svojstva kojima se podešavaju dimenzije su *width* i *height*, a vrijednosti mogu biti u px i em (apsolutne) ili postocima (relativne).

Osim *width* i *height* svojstava, elementi mogu imati zadane vrijednosti minimalnih i maksimalnih dimenzija s prefiksima *min-* i *max-*, na primjer *max-width: 560px*. Te vrijednosti koriste se kod responzivnog CSS dizajna gdje se izgled stranice automatski prilagođava veličini ekrana ili prozora preglednika te ej potrebno odrediti granične dimenzije. Responzivni dizajn izlazi iz opsega ove skripte pa će se koristiti samo apsolutno zadane vrijednosti *width* i *height*.

22. b Obrubi

Obrubi se za HTML elemente definiraju svojstvom *border*, tako da prva vrijednost daje debljinu obruba, druga stil, a treća boju. Vrijednosti se odvajaju razmakom.

```
div.klasa {
        border: 2px dotted green;
}
```

Mogu se zasebno definirati obrubi za gornji, donji, lijevi i desni dio nekog elementa te oni mogu imati različite stilove. To se radi tako da se svojstvu border doda oznaka sufiksa (*–top*, *-bottom*, *-left* ili *–right*), na primjer:

```
border-bottom: 2px solid black;
border-top: 5px dashed green;
```

22. c Padding i margine

Padding je prostor koji dijeli sadržaj od obruba i povećava preglednost i čitljivost na Internet stranici. Margine su prozirni prostor oko elementa (izvan obruba) koji odvaja elemente HTML dokumenta. Mogu se mijenjati odjednom za sve strane elementa korištenjem svojstva *padding* ili *margin*, ili im se može dodati sufiks (isto kao za obrub) kako bi se postavili na jednu stranu elementa, ili kako bi se za različite strane elementa definirao drugačiji *padding* ili margina.

Kako bi oko sadržaja ostavili 15px prostora sa svake strane, pišemo sljedeće:

```
padding: 15px;
```

Kako bi postavili različite vrijednosti za različite stranice HTML elementa, možemo pisati sljedeće:

```
padding-top: 15px;
padding-bottom: 20px;
padding-left: 10px;
padding-right: 5px;
```

Skraćeno, gornji primjer možemo pisati i ovako:

```
padding: 15px 5px 20px 10px;
```

gdje vrijednosti redom označavaju gornji, desnu, donju i lijevu vrijednost (u smjeru kazaljke na satu, počevši od gornje strane).

Ista pravila vrijede i za margine!

22. d Box-sizing

Box-sizing je svojstvo CSS-a koje govori Internet pregledniku kako treba prikazivati kutije sa sadržajem. Kao što je spomenuto, dimenzije *paddinga* i obruba utječu na dimenzije html elementa, dok *height* i *width* postavke mijenjaju samo dimenzije unutarnje kutije sa sadržajem. Ukoliko želimo moći fiksno odrediti dimenzije *border*-kutije (unutar koje se nalaze *padding* i sadržaj), moramo promijeniti *box-sizing*.

box-sizing: border-box;

Kada to odredimo za neki element, onda mijenjanjem *height* i *width* svojstava zapravo mijenjamo dimenzije kutije obruba. Primjerice, ako zadamo:

```
div.klasa1 {
        box-sizing: border-box;
        border: 2px solid black;
        padding: 10px;
        height: 100px;
        width: 100px;
}
```

nacrtani element klase „klasa1“ će imati dimenzije 100x100 px, ali sadržaj će imati dimenzije 76x76 px (100px – 2*10px – 2*2px).

Ako se ne definira da je element tipa *border-box*, vrijedit će standardna postavka *content-box*. Razlika između ta dva načina prikaza, sa istim postavkama dimenzija, prikazana je na slici:

Slika 72 - Razlika između border-boxa i content-boxa

23) CSS pozicioniranje i prikaz elemenata

CSS-om možemo odrediti kako će se HTML elementi prikazivati i kako će biti raspoređeni po ekranu.

23. a *Inline*, *block* i *none*

```
display: inline;
display: block;
display: none;
```

Inline elementi zauzimaju onoliko prostora koliko im je potrebno te se sljedeći element istog tipa prikazuje odmah iz prethodnog *inline* elementa. Primjerice, sljedeći elementi:

```
<span>Inline elementi prikazuju se</span>
<span> jedan za drugim.</span>
```

će se sa CSS kodom

```
span {
	display: inline;
	border: 1px dotted black;
}
```

Prikazati ovako:

Inline elementi prikazuju se jedan za drugim.

Slika 73 - Inline elementi

Inline elementi ne podržavaju postavku visine budući da im je primarna namjena korištenje unutar retka teksta gdje ne želimo da nam neki element mijenja visinu tog retka. Kada ipak poželimo mijenjati visinu *inline* elementa, potrebno ih je definirati kao *inline-block* elemente.

```
display: inline-block;
```

Block elementi, s druge strane, zauzimaju cijelu dostupnu širinu reda i prikazuju se jedan ispod drugog:

```
<span>Block elementi prikazuju se</span><span>jedan ispod drugog.</span>
```

```
span {
	display: block;
	border: 1px dotted black;
}
```

Block elementi prikazuju se
jedan ispod drugog.

Slika 74 - Block elementi

Ukoliko želimo neki element privremeno sakriti bez da ga obrišemo, možemo koristiti svojstvo:

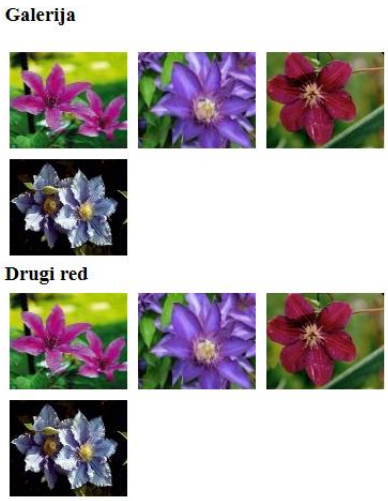
```
display: none;
```

23. b Float

Korištenje *float* svojstva omogućava HTML elementima da budu „zalijepljeni“ na lijevu ili desnu stranu, do drugog elementa sa definiranim *floatom*, a da se pritom ostali elementi obaviju oko njih. Ovo svojstvo korisno je, na primjer, pri pisanju članaka gdje tekst mora obaviti sliku.

Za *float* svojstvo postoje opcije *left* i *right* (lijevo i desno) te je ono naročito korisno kada se sadržaj stranice mora prilagoditi veličini prozora. Ukoliko se, primjerice, za slike koristi svojstvo float: left; te ukoliko one nemaju mjesta biti u jednom retku, automatski će se poravnavati po lijevom rubu jedna uz drugu kroz više redova.

Slika 75 - float: left

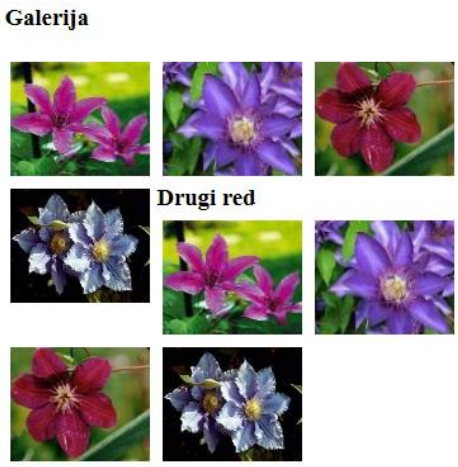


Slika 76 - float: left uz smanjeni prozor

Prilikom korištenja *float* svojstva, valja imati na umu da elementi prije onih koji imaju definiran *float* ostaju „slobodni“, tj. za njih ne vrijedi to poravnanje (npr. tekst „Galerija“), dok se za ostale elemente koje ne želimo poravnavati na taj način mora koristiti svojstvo

```
clear: both;
```

kako bi se oni oslobodili poravnavanja po lijevom rubu uz prethodni element (npr. tekst „Drugi red“). Kada ne bismo koristili *clear: both*, pri smanjenju prozora dogodilo bi se sljedeće:



Slika 77 - Izostavljen clear: both;

Događa se to da se i daljnji elementi, npr. tekst „Drugi red“ i popratne slike, nastoje poravnati prema ranije zadanom svojstvu *float: left;* uz prethodni element (sliku u ovom slučaju).

Dakle, svi elementi koji moraju stajati samostalno, u svom redu, moraju imati CSS-om definirano svojstvo *clear: both;*.

24) Popis slika

Slika 1 - Prikaz strukture HTML dokumenta ... 3
Slika 2 - Google Chrome - Inspect element ... 7
Slika 3 - Mozilla Firefox - Inspect Element ... 8
Slika 4 - Konzola za pregled elemenata na stranici ... 8
Slika 5 - Pregled svojstava elemenata ... 9
Slika 6 - Naslov članka za uređivanje ... 9
Slika 7 - Odabir elementa u konzoli ... 9
Slika 8 - Pregled boje elementa ... 9
Slika 9 - Promjena boje u crvenu ... 10
Slika 10 - Promijenjena boja naslova u crvenu ... 10
Slika 11 - Naslov članka ... 10
Slika 12 - Promjena teksta naslova 1 ... 10
Slika 13 - Promjena teksta naslova 2 ... 10
Slika 14 - Izmijenjeni naslov ... 11
Slika 15 - Odlomci 1 ... 12
Slika 16 - Odlomci 2 ... 12
Slika 17 - Odlomci 3 ... 13
Slika 18 - Prijelom linije ... 14
Slika 19 – Headings ... 14
Slika 20 - Character encoding problem ... 16
Slika 21 - Character encoding ispravno ... 16
Slika 22 – Komentari 1 ... 18
Slika 23 - Komentari 2 ... 18
Slika 24 - Stvaranje tablice ... 20
Slika 25 - Stvaranje naslova stupaca ... 21
Slika 26 – Colspan ... 22
Slika 27 - Border tablice ... 23
Slika 28 - Uređivaje širine tablice 1 ... 23
Slika 29 - Uređivanje širine tablice 2 ... 24
Slika 30 - Cellpadding 10px ... 25
Slika 31 - Cellpadding 2px ... 25
Slika 32 - Cell width ... 25
Slika 33 – Cellspacing ... 26
Slika 34 - Nesortirana lista ... 27
Slika 35 - Sortirana lista ... 27
Slika 36 - DIV 1 ... 28
Slika 37 - Podebljani tekst ... 29
Slika 38 - Nakonešni tekst ... 30
Slika 39 - Podvučeni tekst ... 30
Slika 40 - Povećani tekst ... 30
Slika 41 - Smanjeni tekst ... 30
Slika 42 - Obrisani tekst ... 31
Slika 43 - Nevažeći tekst ... 31
Slika 44 - Input elementi ... 32

Slika 45 - Input type = Text 33
Slika 46 - Input type = password 33
Slika 47 - input type = email 33
Slika 48 - input type = radio 34
Slika 49 - Input type = checkbox 34
Slika 50 - input type = button 35
Slika 51 - input type = submit 35
Slika 52 - Otvaranje stranice nakon Submit akcije 35
Slika 53 - input type = reset 36
Slika 54 - input type = color 36
Slika 55 - Color picker 36
Slika 56 - input type = file 37
Slika 57 - Odabir datoteke 37
Slika 58 - input type = datetime 38
Slika 59 - input type = date 39
Slika 60 - Input type = time 39
Slika 46- Usporedba serif i sans-serif familije fontova 50
Slika 47 - uređivanje sortiranih i nesortiranih listi 52
Slika 48 - korištenje slike za markere liste 53
Slika 49 - list-style-position 53
Slika 50 - Tablica s dvostrukim obrubom 54
Slika 51 - Tablica - border-collapse 54
Slika 52 - Tablica bez postavki dimenzija 55
Slika 53 - Tablica - određena širina, visina i padding 55
Slika 54 - Tablica - položaj teksta 56
Slika 55 - Tablica s određenim bojama 57
Slika 56 - CSS box model 58
Slika 57 - Razlika između border-boxa i content-boxa 61
Slika 58 - Inline elementi 62
Slika 59 - Block elementi 63
Slika 60 - float: left 64
Slika 61 - float: left uz smanjeni prozor 64
Slika 62 - Izostavljen clear: both; 65