

HTML5 i CSS3



Novi Sad, April 2017.

Sadržaj:

Dan 1	1
Kratka istorija interneta	1
Hypertext Markup Language (HTML)	1
Verzije HTML-a	1
Cascading Style Sheet (CSS)	2
Nekompatibilnost browsera	2
Rad sa html dokumentima	2
Programi koji se mogu koristiti za pisanje koda	2
Čuvanje html dokumenta	2
Pregledanje html dokumenta u Google Chrome/Mozilla Firefox	3
Struktura stranice (HTML tagovi)	3
Kako da dobijem naša slova u Browseru?	5
Podaci iz zaglavlja dokumenata	5
Šta sve može da sadrži <body> tag?	6
HTML headings (naslovi)	6
Vežba 1	6
HTML paragraphs (paragrafi)	7
Vežba 2	7
HTML empty elements	8
Tag za oznaku prelaska sa jedne teme na drugu (thematic break)	8
Vežba 3	8
HTML images (slike)	8
Vežba 4	10
HTML links (hiperlinkovi)	10
Tekst kao link	10
Vežba 5	10
Apsolutno adresiranje	10
Relativno adresiranje	10
Slika kao link	12
Vežba 6	12
Linkovanje E-mail adrese	12
Vežba 8	12
HTML Lists (lista)	12
 unordered – nenumerisane liste	12
Vežba 8	13
 ordered – numerisane liste	14
Vežba 9	15
HTML Description Lists (definicione liste)	15
HTML Navigation Bar (meni)	15
Vežba 10	15
Komentari u HTML	16
Vežba 11	16
HTML Entities (specijalni znakovi)	16
Domaći zadatak	17
DAN 2	18
Celine u telu HTML dokumenta (HTML Layouts)	18
Element header	18

Element nav.....	18
Element main.....	18
Element section	19
Element article	19
Element aside	19
Element footer.....	19
Vežba 1	20
Tabele (HTML Tables)	20
Vežba 2	21
Kreiranje formulara – prosleđivanje podataka	21
Dodavanje kontrola u obrazac	21
Elementi forme	22
Input elementi <input>.....	22
Text Input – polje za unos teksta	23
Password Input – polje za unos šifre.....	23
Radio Button Input – radio dugme	23
Input Type Checkbox – dugme za štikliranje	24
The Submit Button – dugme za slanje sadržaja forme	24
Input type reset – dugme za resetovanje na default-ne vrednosti	24
Ostali Input tipovi	24
Input type email – polje za unos email adrese.....	25
Input type date I time – polja za unos datumai vremena	25
Input type number – polje za unos brojeva	26
Atributi kojima podešavamo veličinu polja za unos I postavljamo ograničenja... 26	
Atribut size	26
Atribut maxlength.....	26
Atributi cols I rows	26
Ostali važni atributi.....	27
Atribut name	27
The Action Attribute – atribut action	27
The Method Attribute – atribut metod get I post	27
Atribut placeholder – tekst koji će nestati kada krenemo da popunjavamo polje za unos teksta	28
Ostali tipovi formi	28
Grupisanje podataka <fildset>	28
Padajuća lista <select>	29
Višelinijaska tekstualna polja <textarea>	29
Vežba 3.....	30
Vežba 4.....	30
Dan 3	34
CSS - Cascading Style Sheets	34
Primena CSS-a.....	34
Inline (Inlajn stil)	34
Internal (Ugrađeni stil)	34
External (Uvezeni stil)	35
CSS sintaksa	35
CSS selektori	35
Selekcija pomoću naziva elementa	35
Selekcija pomoću identifikatora (ID)	36
Selekcija pomoću klase (class)	36
Razlika između CSS klase i ID-a	37

Nasleđivanje stilskih opisa.....	37
Kaskada stilskih opisa.....	38
Najčešće korišćena svojstva i njihove vrednosti	38
Stilizovanje teksta (text)	38
Slovni lik (fonts)	40
Pozadina (background)	41
Margine (margins).....	43
Padding	44
Visina I širina (height/width)	45
max-width, max-height, min-width, min-height	45
Model kutija (Box Model)	45
Okvir (border).....	46
Stilizovanje lista (lists).....	47
Stilizovanje tabela (tables).....	48
Da se ivice “prilepe” koristimo komandu border-collapse:collapse;.....	48
Display (Prikaz).....	50
Blok elementi.....	50
Linijski elementi (inline elements)	51
Uloga tagova DIV i SPAN.....	51
Display:none;	52
Linijski – blok elementi (Inline-block).....	52
Pozicioniranje.....	52
Relativno pozicioniranje (relative)	52
Apsolutno pozicioniranje (absolute).....	53
Fiksno pozicioniranje (fixed)	54
Plutajući elementi (float)	54
Kontrola ponašanja float elemenata (Clear).....	54
Komentari u CSS-u.....	54
Dan 4	55
CSS linkovi (CSS Links).....	55
Izrada dugmeta (Link buttons)	56
CSS ikone (CSS Icon).....	56
CSS navigacija (CSS Navigation Bar)	57
Horizontalna navigacija (Horizontal navigation bar)	59
Izrada sajta.....	63
Dan 5	71
Dodatni materijal za učenje HTML5 I CSS3	71

Dan 1

Kratka istorija interneta

Temelji Interneta postavljeni su tokom hladnog rata u američkom ministarstvu odbrane kao „rezervni” kanal komunikacije u slučaju nuklearnog napada. Osnovna zamisao bila je stvoriti mrežu računara koja će nesmetano nastaviti da funkcioniše i u slučaju da neki njen deo bude uništen. Kako bi ta tehnologija bila standardizovana, izmišljen je protokol razmene podataka koji zovemo Transmission Control Protocol / Internet Protocol, odnosno TCP/IP – „jezik interneta”.

Najveće dostignuće Interneta je definitivno Word Wide Web, odnosno onaj deo globalne mreže koji povezuje milione dokumenata – web stranica. WWW kakvog ga danas poznajemo, postoji najviše zahvaljujući švajcarskom naučniku po imenu Tim Berns-Lee. On je u želji da osmisli razmenu dokumenata unutar mreže, koristeći tehnologiju hiperteksta (hypertext) koji korisnicima omogućuju da aktiviraju linkove i tako se „kreću” kroz veliki broj međusobno povezanih dokumenata, razvio standard po imenu hypertext transfer protokol, odnosno http. Osim toga, on je razvio i standarde kojima se predstavljaju hipertekst dokumenti, i koji se zajedno zovu Hypertext Markup Language, odnosno HTML.

Iako se tehnologija koja „oprema” internet menja svakodnevno, daleko od toga da je HTML prevaziđena tehnologija na www. Ovaj jezik ostao je osnovni kôd novog medija i čini se da će se još dugo zadržati na tom položaju. Zbog toga je, ako želite da se bavite razvojem web aplikacija u bilo kojoj tehnologiji, najbolje i jedino prirodno da najpre savladate HTML.

Hypertext Markup Language (HTML)

HTML - Hypertext Markup Language, je jezik kojim se browseru objašnjava šta stranica treba da sadrži od sadržaja. Za opis sadržaja koriste se HTML elementi. O njima ćemo kasnije pričati.

Verzije HTML-a

Verzija	Godina
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

HTML5 je nova verzija HTML standarda i pre ili kasnije svaki web developer će se naći u situaciji da radi na istom i mnogi su shvatili da je mnogo lakše raditi od početka nego migrirati kod iz HTML4 u HTML5 zbog razlika koje postoje između verzija. Zapravo HTML5 nije velika prepravka HTML4 ima dosta sličnosti među njima. Međutim postoje neke esencijalne razlike, neki tagovi su menjali ime, neki se u HTML5 ne koriste uopšte.

Cascading Style Sheet (CSS)

Jezik CSS služi za opis stila (boja, fontova, rasporeda sadržaja) veb-stranica.

Nekompatibilnost browsera

Uprkos svim standardima koji su spomenuti, internet je i dan danas jedan „donekle uređen kaos”, čemu najviše doprinose kompanije bez čijih se proizvoda ne može: proizvođači web browsera.

Web browseri, ili čitači weba su programi koji traže web stranice na Internetu i prikazuju ih korisnicima onako kako ih je web developer (čovjek koji pravi sajtove - u našem slučaju mi) napravio.

Ono što je važno da znate, jeste da kada generišemo kod i pokrenemo ga preko browsera, mogu postajati izvesne razlike u izgledu našeg sajta. Browseri koje ćemo mi najviše koristiti su Google Chrome i Mozilla Firefox.

Rad sa html dokumentima

Programi koji se mogu koristiti za pisanje koda

Za pisanje HTML5 koda, mogu se koristiti različiti programi. Najpopularniji su Notepad++, Brackets, Sublime... Mi ćemo koristiti Notepad++.

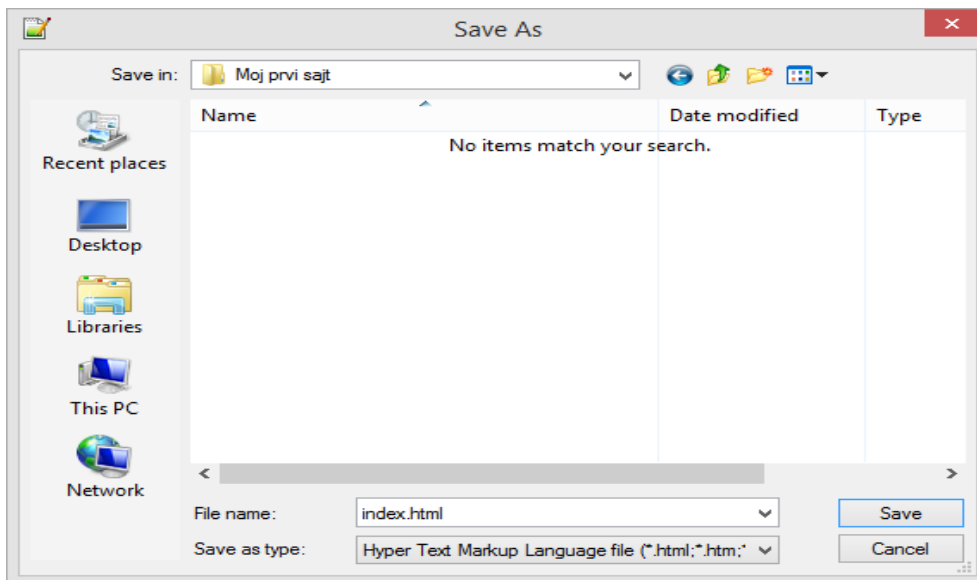
Notepad++

Notepad++ je besplatan alat koji predstavlja osnovni alat koji ćemo koristiti na ovom kursu. Može se besplatno preuzeti sa interneta.

Svaki HTML dokumenat se kreira i edituje u Notepad++.

Čuvanje html dokumenta

Pošto smo pravilno rasporedili osnovne tagove html dokumenta možemo da ga sačuvamo. U meniju odaberemo `File > Save As` i otvoriće nam se prozor u kome ćemo izabrati lokaciju, dati mu ime i što je najvažnije proveriti da li je ekstenzija našeg fajla podešena na `html`.



Pregledanje html dokumenta u Google Chrome/Mozilla Firefox

Da bi smo pogledali u nekom od browsera našu prvu stranu, to možemo uraditi na dva načina:

1. Dok nam je još otvoren program Notepad++, biramo meni Run/Launch in Firefox/Chrome/IE (neki od browsera) ili
2. Otvorimo folder u kome čuvamo našu .html stranicu, fajl označimo levim tasterom miša, pa potom desnim biramo opciju Open with/željeni browser.

Struktura stranice (HTML tagovi)

Svaki html dokument u sebi ima **tagove**. Tagovi predstavljaju osnovnu sintaksu html dokumenta. Imamo otvarajući `<tag>` koji browseru ukazuje da od te tačke primenjuje neku komandu i zatvarajući `</tag>` koji browseru ukazuje gde prestaje ta komada. Neki tagovi imaju zatvarajući tag, a neki ne. Kad se budemo upoznawali sa njima navešćemo da li treba taj tag zatvoriti.

`<tag>Sadržaj stranice</tag>`

Neki tagovi u svojoj strukturi imaju i **atribute**. Atributi se pišu unutar otvarajućeg taga i po pravilu imaju naziv i vrednost. Oni se nalaze unutar složenih oznaka i sadrže dodatne informacije o načinu prikazivanja i ponašanju označenog dela dokumenta. Atributi se smeštaju između znakova navoda.

`<ime_taga naziv_atributa="vrednost">`

Primer:

`<p>Tekst paragrafa.</p>`

`<p title="I'm a tooltip">This is a paragraph.</p>`

U prvom primeru se odabrani deo označava kao paragraf. U drugom primeru, koristi se atribut title koji omogućava da se kada se stavi kursor miša na paragraf ispiše tekst I'm a tooltip".

HTML dokumenti se sastoje iz dva osnovna dela: onog koji opisuje dokument i način na koji treba da bude predstavljen i dela koji predstavlja sadržaj dokumenta. Informacije koje opisuju dokument se smeštaju između oznaka za zaglavlje dokumenta `<head>`, `</head>`. U okviru zaglavlja postavlja se naslov dokumenta između oznaka `<title>`, `</title>`, dok se sam sadržaj smešta između oznaka `<body>`, `</body>`. Ova tri elementa zajedno sa oznakam `<html>`, `</html>` čine minimilanu strukturu HTML dokumenta:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Naslov dokumenta</title>
  </head>
  <body>
    <h1>Primer dokumenta</h1>
    <p>Ovo je primer jednog prostog HTML dokumenta.</p>
  </body>
</html>
```

1. `<!DOCTYPE html>`

Svaki HTML5 dokument počinje sa `<!DOCTYPE>` deklaracijom! Ona se piše samo jednom, na vrhu dokumenta. Ovaj tag je samozatvarajući.

2. `<html>` tag

Nakon `<!DOCTYPE html>` deklaracije sledi `<html>` tag, koji sadrži druge tagove. Ovaj tag se završava sa zatvarajućim tagom `</html>`.

```
<!DOCTYPE html>
<html>
</html>
```

`<html>` tag sadrži tagove `<head>` i `<body>`

3. `<head>` tag

HEAD tag označava zaglavlje dokumenta i obično sadrži sve informacije o samom dokumentu koje ga opisuju (o kodnom rasporedu, autoru, ključnim rečima....), ali se ne prikazuju na samoj stranici. Te informacije daju browseru „uputstva” kako da radi sa dokumentom.

```
<!DOCTYPE html>
<html>
  <head>
  </head>
</html>
```


4. <body> tag

Kada se stranica prikazuje u pretraživaču, on prikazuje samo informacije koje se nalaze unutar tagava body. Ovaj tag sadrži informacije koje su vidljive korisniku.

<body> tag mora biti posle <head> tag.

Svaka HTML stranica treba da ima sledeću strukturu.

```
<!DOCTYPE html>
<html>
  <head>
  </head>

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

5. <title> tag

Ovoj tag je izuzetak u <head> tagu čiji sadržaj se vidi u naslovnoj traci browsera. On ima zatvarajući tag </title>. Ono što napišemo unutar taga <title> biće prikazano kao naziv kartice (taba) u browseru.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Naslov stranice</title>
  </head>

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

Kako da dobijem naša slova u Browseru?

Potrebno je da u zaglavlju dokumenta (između <head> i </head>) imamo sledeći tag:

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

Prebacimo na našu ćiriličnu ili latiničnu tastaturu i kucamo, pa će se ti zapisi prikazati u Browseru onako kako mi želimo.

Podaci iz zaglavlja dokumenata

HTML može da ima više META elemenata, ali svi META elementi moraju biti ugnježdjeni unutar HEAD elementa. Tačno značenje META taga određuju njegovi atributi i njihove vrednosti.

Primer

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="IT Courses and training ">
  <meta name="keywords" content="HTML, CSS, XML ,JavaScript">
  <meta name="author" content="John Doe">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Atribut **description** pruža sažeto objašnjenje o sadržaju veb stranice.

Atribut **keywords** predstavlja ključne reči po kojima će se tražiti veb stranica.

Atribut **author** predstavlja ime autora veb stranice.

Atribut **viewport** je važan kada je u pitanju responsive web stranica.

Šta sve može da sadrži <body> tag?

HTML headings (naslovi)

Naslovi su definisani tagovima od <h1> do <h6>.

<h1> tag označava glavni naslov – ovaj naslov je ispisan najvećim slovima,

<h2> tag označava nešto manje važan naslov I tako sve do <h6> koji predstavlja najmanje važan naslov.

Svaki <h_n> tag ima svoj zatvarajući tag </h_n>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
<h1>This is heading 1</h1>
```

```
<h2>This is heading 2</h2>
```

```
<h3>This is heading 3</h3>
```

```
<h4>This is heading 4</h4>
```

```
<h5>This is heading 5</h5>
```

```
<h6>This is heading 6</h6>
```

```
</body>
```

```
</html>
```

Vežba 1

Napisati sledeći tekst koristeći odgovarajući tag:

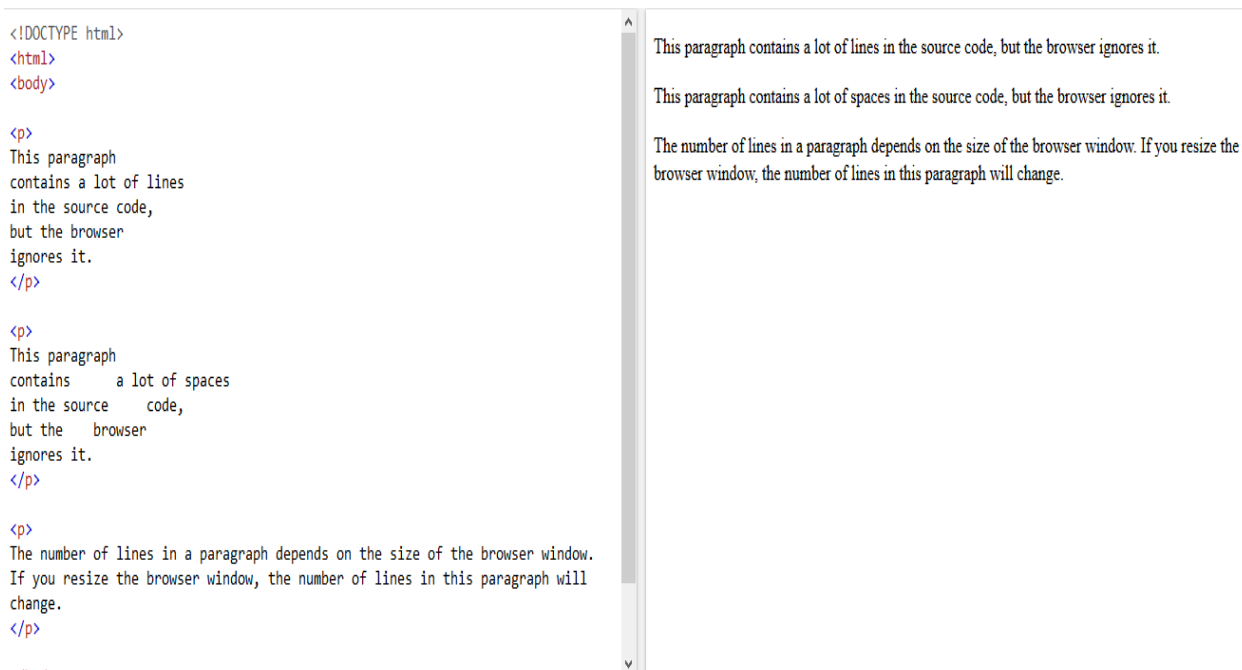
“Smart School” – naslov od najveće važnosti

“Trg Mladenaca 5” – sledeći naslov po važnosti

“Novi Sad” – naslov manje važan od naslova “Trg Mladenaca 5”

HTML paragraphs (paragrafi)

Kada je u pitanju pisanje teksta u okviru <p> taga, važno je da zapamtite da tamo gde vi napravite prelom reda, ne znači da će se taj prelom prikazati na web stranici. Za prelom teksta (novi red) koristi se
 tag.



<p> tag ima svoj zatvarajući tag </p>

Tagovi koji se takođe koriste u kombinaciji sa <p> tagom su i <i>. se koristi kada želite da podebljate određenu reč ili rečenicu (eng. bold), a <i> kada želite da iskosite određenu reč ili rečenicu (eng. italic).

Vežba 2

Dodati pasus "Hello World!" u dokument I upotrebiti određeni tag kojim će slova biti podebljana:

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
</body>
```

```
</html>
```

HTML empty elements

 prelom reda

Kad hoćemo da na browseru dobijemo novi red upisaćemo
. On nema zatvarajući tag.

```
<!DOCTYPE html>
<html>
<body>

<p>This is<br>a paragraph<br>with line breaks</p>

</body>
</html>
```

This is
a paragraph
with line breaks

Tag za oznaku prelaska sa jedne teme na drugu (thematic break)

<hr> horizontalna linija

Na mestu gde želimo da se pojavi horizontalna linija stavljamo tag <hr>. On nema zatvarajući tag.

```
<!DOCTYPE html>
<html>
<body>

<h1>London</h1>
<hr>
<p>London is the capital city of England. It is the most  
populous city in the United Kingdom, with a metropolitan area  
of over 13 million inhabitants.</p>

</body>
</html>
```

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Vežba 3

Napisati tekst:

Novi Sad

Novi Sad je glavni grad Vojvodine.

Novi Sad je posle Beograda, drugi grad u Srbiji po broju stanovnika.

Posle naslova Novi Sad, staviti horizontalnu liniju.

Koristeći odgovarajući tag, podesiti prelom reda u paragrafu.

HTML images (slike)

Pored teksta u HTML dokument mi možemo ubaciti i određenu vrstu grafike (fotografije, ilustracije, animirane ilustracije...), ali za internet prezentaciju koristimo tri formata koje browseri mogu tumačiti.

Format	Broj boja		Transparentnost	Animacija
.gif	256		Da	Da
.jpg	16 777	21	Ne	Ne
.png	16 777	21	Da	Ne

Optimalna rezolucija za slike na Webu je 72 dpi (piksela po inču).

** umetanje slike**

IMG tag nema zatvarajući tag i ima značenje tek ako sadrži SRC atribut.

Sintaksa za ubacivanje slike je sledeća:

src="URL adresa slike"

alt="alternativni tekst za sliku" (ako se slika ne pojavi iz nekog razloga, ispisaće se ovaj tekst)

Width – širina slike u pikselima

Height – visina slike u pikselima

Pr.

VAŽNO:

Podešavanje visine i širine slike može se vršiti na dva načina:

1. Uz pomoć atributa width i height:

Primer

2. Uz pomoć atributa style:

Primer

Preporučuje se drugi način sa atributom style!

Putanja do slika

Kada smo govorili o uvođenju slika u HTML dokument, koristili smo samo slučajeve kad su dokument i slika u istom folderu. Slike obično organizujemo u podfolderu, tako da moramo napraviti relativnu putanju do željene slike.

```

```

Ako se dokument nalazi u nekom podfolderu onda putanja do slike izgleda:

```

```

Vežba 4

Preuzeti jednu sliku sa interneta, promeniti joj ime u Logo I umetnuti je na index.html stranicu. Visina slike je 400, a širina 600

HTML links (hiperlinkovi)

Pritiskom na datu vezu vrši se skok na drugi dokument ili drugi deo istog dokumenta. Realizuje se pomoću oznake `<a>`. Najvažniji atribut za vezu je `href` u kome se određuje dokument na koju treba skočiti.

Moguće je postaviti i atribut `target` kojim se određuje da li će se dokument otvoriti u istom prozoru (`"_self"`) u novom prozoru (`"_blank"`), roditeljskom prozoru (`"_parent"` frame), datoteku glavnog prozora (`"_top"` - frame) ili u prozoru koji se navede u ovom atributu. Ako se ne postavi ni jedna vrednost atributa, otvara se u istom prozoru.

Tekst kao link

```
<a href="url">tekst koji linkujemo</a>
```

Pr.

```
<a href="https://www.google.com" target="_blank">Poseti google stranicu</a>
```

Vežba 5

Postaviti hiperlink sa tekстом "Poseti yahoo stranicu", koji će otvarati pomenutu stranicu u novom prozoru

Apsolutno adresiranje

Prethodni način zadavanja URL adrese predstavlja **apsolutno adresiranje**, gde je data celokupna adresa. Ovaj način se koristi u slučajevima kada želite da postavite vezu ka nekoj strani koju ste pronašli na Webu. Ovakva veza se još naziva i spoljna jer se odnosi na Web stranu koja se nalazi izvan vašeg servera, odnosno Web lokacije.

Relativno adresiranje

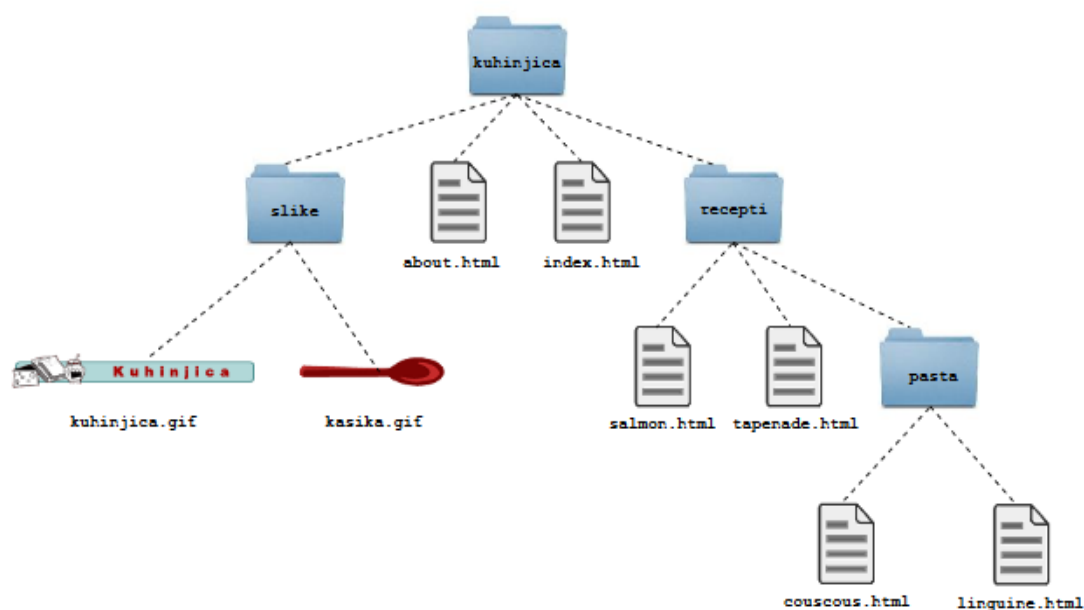
Osim apsolutnih koriste se i **relativne adrese**. Relativne adrese se koriste kada postavljate veze između strana na istoj web lokaciji. Kada se ne vidi „http://“, Web

čitač traži dokument na istom serveru. Relativno ime putanje opisuje kako se stiže od lokacije tekućeg dokumenta (na kojoj se trenutno nalazite) do dokumenta koji treba povezati.

Najlakše se pišu relativne adrese unutar direktorijuma. Kada se povezuju datoteke unutar istog direktorijuma dovoljno je da se samo napiše ime datoteke. Ako je URL samo ime datoteke, server ga traži u tekućem direktorijumu – onom u kojem je dokument sa vezom za tu datoteku.

U narednom primeru je postavljena veza između početne strane (`index.html`) ka strani sa opštim informacijama (`about.html`). Obe datoteke su u istom direktorijumu (`kuhinjica`). Dakle, dovoljno je da napišete sledeće:

```
<a href="about.html">O ovoj lokaciji...</a>
```



Slika 1. Dijagram strukture direktorijuma, za sajt „kuhinjica“

Često je potrebno da se postavi veza ka **direktorijumu nižeg nivoa**. U prethodnom primeru datoteke sa receptima su u poddirektorijumu recepti. Da bi postavili vezu od datoteke `index.html` ka datoteci u direktorijumu recepti koja se zove `salmon.html` treba napisati URL adresu koja govori Web čitaču da u tekućem direktorijumu potraži direktorijum recepti i u njemu datoteku `salmon.html` na sledeći način:

```
<a href="recepti/salmon.html">Garlic Salmon</a>
```

Da bi postavili vezu ka datoteci `couscous.html`, koja se nalazi u poddirektorijumu Pasta, treba napisati putanju kroz dva poddirektorijuma (recepti, zatim pasta) ka datoteci `couscous.html`

```
<a href="recepti/pasta/couscous.html">Couscous Recept</a>
```

Osim prethodnih primera moguće je i postavljanje veza ka **direktorijumu višeg nivoa**. Sledeći primer pokazuje kako se postavlja veza ka početnoj strani (`index.html`) od strane

salmon.html. Kako je salmon.html u poddirektorijumu recepti, moramo da se vratimo jedan nivo nagore, kako bismo našli index.html.

```
<a href="../index.html">Povratak na glavnu stranu </a>
```

Slika kao link

Kada slika predstavlja link, sintaksa je sledeća:

```
<a href="default.asp">
  
</a>
```

Vežba 6

Preuzeti sliku sa interneta I napraviti od nje hiperlink ka stranici www.google.com

Linkovanje E-mail adrese

Kada želimo da klikom na neki sadržaj (e-mail) na stranici pokrenemo program za e-mail koristimo isti atribut HREF, samo ne koristimo protokol http, nego mailto i potom navodimo e-mail adresu.

```
<a href="mailto:someone@example.com" target="_top">Send Mail</a>
```

Vežba 8

Linkovati svoju e-mail adresu. Tekst koji se linkuje je "Kontaktirajte me"

HTML Lists (lista)

Osnovna svrha Interneta i prezentacija na njemu jeste da pruži informacije.

Mnogo je upadljivije i upečatljivije ako se podaci prikazuju listama (pod tačkama – bullets, ili numerisani), umesto da se ređaju jedan za drugim odvojeni zarezom. HTML nam nudi tri načina prikazivanja lista:

Unordered (nenumerisane), ordered (numerisane) i definicione liste.

```
<li> stavka liste </li>
```

Sadržaj svakog pojedinačnog elementa liste definiše se tagovima , i uvek se nalazi ugnježđen unutar UL i OL tagova.

*** unordered – nenumerisane liste***

UL tag otvara i zatvara nenumerisanu listu, u kojoj ispred svakog elementa stoji neki simbol (bullet). Zatvarajući tag je obavezan.


```

<ul>
  <li>Prva stavka liste</li>
  <li>Druga stavka liste</li>
  <li>Treća stavka liste</li>
</ul>

```

List-style-type

List-style-type nudi mogućnosti izbora vrste tačke koja će biti dodeljena svakom elementu date liste. Ona može da bude:

Vrednost	Primer liste
disc	• (ispunjeni krug)
circle	○ (prazan)
square	■ (kvadrat)
none	Lista neće biti markirana

Pr.

```

<!DOCTYPE html>
<html>
<body>

<h2>Unordered List with Disc Bullets</h2>

<ul style="list-style-type:disc">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>

</body>
</html>

```

Unordered List with Disc Bullets

- Coffee
- Tea
- Milk

Vežba 8

Koristeći unordered lists, nabrojati tri vrste jela koja rado konzumirate. Za list-style-type postaviti square.

** ordered – numerisane liste**

OL tagovi otvara i zatvara numerisanu listu, u kojoj su elementi liste redom numerisani.

Primer

```
<ol>
  <li>Prva stavka liste</li>
  <li>Druga stavka liste</li>
  <li>Treća stavka liste</li>
</ol>
```

Type

Atribut koji nudi mogućnosti izbora numeracije koja će prethoditi svakom elementu liste. Defaultna vrednost je 1.

Vrednost	Primer liste
A	A, B, C,...
a	a, b, c,...
I	I, II, III,...
i	i, ii, iii,...
1	1, 2, 3,...

Primer:

```
<!DOCTYPE html>
<html>
<body>

<h2>Ordered List with Letters</h2>

<ol type="A">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

</body>
</html>
```

Ordered List with Letters

A. Coffee
B. Tea
C. Milk

Start

Određuje redni broj prvog elementa u listi. Pošto OL nudi mogućnosti numerisanja i rimskim brojevima i slovima, u start atribut se i u takvim listama stavlja ceo broj, koji označava koji je redni znaka u listi kojim želimo da počnemo nabrojanje. Na primer, ukoliko su elementi poređani pod A, B, C,...

Da bi takva lista krenula od C, u START atributu dodeljujemo vrednost 3.

Vežba 9

Koristeći `order lists`, nabrojati tri vrste pića koja rado konzumirate. Za `type` atribut postaviti A. Neka nabranje krene od slova D.

HTML Description Lists (definizione liste)

Ovi tagovi služe za takozvane definicione liste (DL). Ovde treba razlikovati termine koji su ugnježdeni u `<dt>` tag i opis tih termina ugnježdenih u `<dd>` tag. Ukoliko u DL tag dodamo njegov jedini atribut `compact` (nema vrednost), termin i opis biće ispisani u jednom redu.

Primer

```
<!DOCTYPE html>
<html>
<body>

<h2>A Description List</h2>

<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>

</body>
</html>
```

A Description List

Coffee
- black hot drink
Milk
- white cold drink

HTML Navigation Bar (meni)

Sada kada znamo kako se radi sa listama i hiperlinkovima, možemo da kreiramo meni.

Primer

```
<body>
  <ul>
    <li><a href="index.html">Početna</a></li>
    <li><a href="about.html">O nama</a></li>
    <li><a href="kontakt.html">Kontakt</a></li>
  </ul>
</body>
```

Kada budemo prešli CSS, znaćemo da pravimo horizontalne i vertikalne menije, da im zadajemo boju i sl.

Vežba 10

Napraviti meni sa stavkama Home, About Us, Contact.

Komentari u HTML

Pravila lepog kodiranja nalažu da se eventualnim čitaocima koda objasni šta se dešava u pojedinim delovima koda. Osim toga, i samom autoru koda korisno je da zabeleži šta se dešava, jer je se ponekad teško snaći kada se posle nekoliko meseci vratite da nešto prepravite u kodu. Za to nam služe komentari – delovi koda koji su vidljivi za čoveka, a nevidljivi za browsere.

U komentar možete staviti bilo šta, pošto browser kada naiđe na početak „<!--” komentara sadržaj do zatvarajućih zagrada „-->” bukvalno preskoči.

```
<!--Ovde ide komentar.-->
```

Vežba 11

Napraviti dva pasusa I pored svakog napisati komentar (na mestima gde treba) “Početak prvog pasusa”, “Kraj prvog pasusa”, “Početak drugog pasusa”, “Kraj drugog pasusa”.

HTML Entities (specijalni znakovi)

Specijalni znaci obezbeđuju metod da se unesu karakteri koji ne mogu da se unesu sa tastature ili su zauzeti npr. znak manje < nemože da se koristi pri unosu sadržaja pošto se taj znak koristi za tagove.

Kada je u pitanju upotreba specijalnih znakova, za njihovo ispisivanje koristimo ili imena ili brojeve.

Rezultat	Opis	Entity Name (ime)	Entity Number (broj)
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

Primer

<p>Škola računara © SMART</p>

Domaći zadatak

1. Odabrati temu (npr: grad, država, biljka, životinja ili hobi...)
2. Prikupiti slike i tekst za odabranu temu
3. Kreirajte jedan html dokument na odabranu temu
4. Dokument mora da ima glavni naslov (h1)
5. Dve podceline (h2) sa pratećim tekstom i svaka celina da ima sliku
6. U prvoj podcelini napraviti nenumeričko nabarajnje, a u drugoj numeričko

DAN 2

Celine u telu HTML dokumenta (HTML Layouts)

Sadržaj tela HTML dokumenta se obično može podeliti na neke manje logičke celine. Na primer, na većini veb-stranica se može identifikovati deo stranice koji sadrži linkove za kretanje unutar veb-sajta, može se identifikovati neko zaglavlje koje često sadrži logotip veb-sajta, glavni naslov, može se identifikovati neki deo, obično postavljen sa strane koji sadrži sporedne informacije (na primer, neke reklame) i slično. Počevši od verzije 5, jezik HTML podržava nekoliko različitih elemenata koji se koriste za opis podele veb-stranica na ovakve celine. Naglasimo da su ti elementi logičke (semantičke) prirode i da nemaju nikakvog direktnog uticaja na način kako će njihov sadržaj biti prikazan, niti na poziciju na ekranu gde će se taj sadržaj postavljati. Označavanje sadržaja uvek treba da bude po tome šta taj sadržaj predstavlja (njegove semantike), a ne po tome kako taj sadržaj treba da izgleda. Raspoređivanje i stilizovanje sadržaja radi se naknadno, pomoću jezika CSS.

Elementi header, footer, nav, main, section, article, aside...



Element header predstavlja zaglavlje (engl. header) veb-stranice ili zaglavlje određene manje sekcije u okviru stranice (o takvim sekcijama će biti reči u nastavku). Zaglavlje stranice često sadrži uvodnih elemenata poput naslova veb-sajta, logotipa, grupe navigacionih linkova i slično.

Element nav predstavlja deo veb-stranice koji sadrži grupisane linkove ka drugim stranicama (obično u okviru istog veb-sajta) ili linkove ka nekom sadržaju unutar iste te veb-stranice. Element nav je obično deo zaglavlja stranice, ali to nije obavezno.

Element main se koristi da označi centralni sadržaj tela veb-stranice. Očekuje se da je sadržaj ovog elementa u direktnoj vezi sa centralnom temom veb-stranice ili centralnom funkcionalnošću veb-aplikacije. Sadržaj elementa main treba da bude jedinstven za tu specifičnu stranicu i element main ne bi trebalo da sadrži elemente koji se ponavljaju na svim stranicama istog veb-sajta (zaglavlja stranice, navigacione linkove i slično). Svaka stranica može da sadrži najviše jedan element main.

Element section predstavlja odeljak u dokumentu. Ako veb-stranica ima dosta materijala, pogodno je taj materijal podeliti u zasebne sekcije (engl. sections). Sekcije bi mogle da odgovaraju poglavljima neke knjige, a neka veb-stranica bi mogla da ima svoju uvodnu sekciju, sekciju sa vestima i sekciju sa kontakt-podacima. Za označavanje sekcija koristi se element section. Svaka sekcija predstavlja neko tematsko grupisanje sadržaja i trebalo bi da počne nekim naslovom koji ilustruje o čemu se u toj sekciji govori. Pošto sekcije mogu predstavljati i krupnije celine, one u sebi mogu sadržati i zaglavlja, podnožja, manje podsekcije i slično.

Element article definiše samostalan članak u dokumentu. Potpuno, zaokružene celine, nezavisne od ostalog sadržaja nazivaju se članci (engl. Articles). Članak bi mogao predstavljati članak u nekim elektronskim novinama, ali i svaki pojedinačni post (objava) na nekom forumu ili blogu kao i svaki pojedinačni komentar čitalaca. Članci se obeležavaju se elementom article. Članak može da sadrži manje članke (na primer, objava na blogu je članak, koji u sebi može da sadrži komentare čitalaca koji su takođe članci). Duži članci mogu biti podeljeni na sekcije (i sadržati elemente section), a sekcije u sebi mogu sadržati članke. Članci mogu imati naslove (ali ne moraju). Sekcije i članci imaju svoju jasnu semantiku i ne bi ih trebalo upotrebljavati veštački, tj. samo da bi se grupisao neki sadržaj da bi se stilizovao (za to se može koristiti generički element div).

Element aside definiše sadržaj, pored sadržaja same stranice.

Element footer predstavlja podnožje (engl. footer) veb-stranice ili podnožje određene manje sekcije u okviru stranice. Podnožje obično sadrži podatke o tome ko je napisao stranicu, informacije o zaštiti autorskih prava (engl. copyright), linkove ka srodnim dokumentima i slično.

Struktura stranica bi mogla da izgleda ovako:

```
<body>
<header>
  <!-- naslov i logotip stranice -->
  <nav>
    <!-- linkovi za navigaciju unutar sajta -->
  </nav>
</header>
<main>
  <!-- centralni deo stranice -->
</main>
<aside>
  <!-- vesti i aktuelnosti -->
</aside>
<footer>
  <h3>Autor: Pera Perić</h3>
</footer>
</body>
```

Vežba 1

Napraviti stranicu koja sadrži celine Header, nav, 2 x article I footer.

U header staviti naslov HTML Layouts, nav treba da sadrži listu sa menijima Home, About us, Contact, oba artiola sadrže po jedan h2 tag I jedan p tag. U prvi h2 tag napisati Heading 1, a u drugi Heading 2. Prvi p tag sadrži tekst Paragraph 1, a drugi Paragraph 2. U footer staviti svoje ime kao ime autora u okviru h3 taga.

Tabele (HTML Tables)

Tabele u HTML dokumentu imaju dvostruku funkciju: za tabelarni prikaz nekih podataka (rezultati nekog istraživanja...), i za pozicioniranje sadržaja unutar neke stranice.

Tabele se sastoje iz redova i kolona u čijem preseku se nalaze ćelije tabele. Najmanja tabela ima jedan red i jednu kolonu, odnosno samo jednu ćeliju.

<table> tabela

Ovaj tag sadrži celu tabelu i njegovi atributi zadaju osobine cele tabele. Te osobine se mogu promeniti za svaku ćeliju pojedinačno. Zatvarajući tag mu je obavezan.

```
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Pera </td>
    <td>Perić</td>
    <td>50</td>
  </tr>
  <tr>
    <td>Eva</td>
    <td>Petrović</td>
    <td>94</td>
  </tr>
</table>
```

<tr> red

Ovaj tag sadrži jedan red unutar tabele. Njegovi atributi daju osobine celom redu, mada se preporučuje da se zadaju osobine posebno za svaku ćeliju.

<th> table header

Označava prvi red u tabeli (zaglavlje tabele)

<td> ćelija

Tagovi koji označavaju početak i kraj jedne ćelije.

Vežba 2

Napraviti tabelu sa slike:

Pozicija	Zemlja	Udeo akcionog tržišta
1.	Francuska	22%
2.	Italija	20%
3.	Španija	16%
4.	Australija	8%

Kreiranje formulara – prosleđivanje podataka

Iako se većina stranica sastoji od teksta, slika i hiperveza, skoro sve Web lokacije imaju bar po jednu stranicu na kojoj se popunjavaju podaci i te informacije se vraćaju operateru WEB lokacije.

Obrazac (eng. form) je skup polja koja se koriste za prikupljanje podataka ili interakciju sa korisnicima. Nakon popunjavanja formulara, posetilac Web stranice vrši potvrdu unetih podataka i ti podaci se zatim šalju serveru na obradu. Forme sadrže različite vrste objekata (polja ili kontrola) kao što su tekst polja, padajuće liste, polja za potvrdu, opciona dugmad, dugmad za slanje podataka, dugmad za brisanje unetih podataka i sl. Kada se serveru proslede podaci iz formulara, vrši se njihova obrada odgovarajućom skriptom ili aplikacijom i kao potvrda uspešnosti popunjenog formulara pojavljuje se naredna strana sa rezultatima obrade.

Prilikom kreiranja stranice sa obrascima, najpre se definiše koje podatke od korisnika treba prikupiti. Zatim se dodaje forma kojoj pripadaju sva polja i u nju se dodaju pojedinač na polja. Moguće je postaviti pravila unosa podataka u polja (dopuštene vrednosti, format unosa, obavezna polja,...). Za obradu podataka u formama potrebno je da postoji određen skript na serverskoj strani (npr. CGI script –Common Gateway Interface), a na najjednostavniji način obrade je prosleđivanje podataka na određenu e-mail adresu.

Takođe, kontrola popunjenih polja se može izvršiti i na klijentskoj strani, pre slanja na server pomoću JavaScript-a.

Kreiranje obrasca se vrši pomoću elementa `form`:

```
<form>
```

form elements

```
</form>
```

Dodavanje kontrola u obrazac

Većina kontrola se u obrazac dodaje pomoću taga `input`. Atributi koji su neophodni da se definišu su:

type–tip polja i

name–jedinstveno ime polja

value - kojim se određuje vrednost podatka

Atribut `type` uzima jednu od vrednosti: `button`, `checkbox`, `radio`, `text`, itd.

Na primer:

```
<input type="text" name="ImePolaznika">
```

umeće tekst polje koje ima naziv `ImePolaznika`. Kada korisnik na Web stranici upiše adresu u ovo tekst polje, i izvrši potvrdu formulara promenljiva `ImePolaznika` će sadržati tekst koji je korisnik upisao u ovo polje. Takav par atribut – vrednost (`ImePolaznika = Marko`) se šalje serveru na obradu.

Elementi forme mogu biti:

Input elementi `<input>`

Input elementi se razlikuju u odnosu na atribut `type`.

<code><input type="text"></code>	jednolinijska tekst polja u koja se unose kratki stringovi
<code><input type="password"></code>	Polje za unos lozinke (umesto unetih karaktera prikazuju se tačkice)
<code><input type="radio"></code>	Radio dugmad (za odabir jednog od više mogućnosti)
<code><input type="checkbox"></code>	Polje za štikliranje
<code><input type="button"></code>	Obično (akciono dugme)
<code><input type="submit"></code>	Dugme za potvrdu (za slanje sadržaja forme)

Input je oznaka za ulazno polje opšteg tipa, polju se dodeljuje ime atributom `name`, `type` je oznaka tipa ulaznog polja, a ostali parametri zavise od tipa ulaznog polja.

Pored atributa **`type`**, obično se zadaju atributi **`name`** kojim se zadaje ime podatka i **`value`** kojim se određuje vrednost podatka. Da bi više **`radio`**-dugmadi bilo sinhronizovano, potrebno je da sva imaju istu vrednost atributa `name`. Sadržaj elementa **`input`** je prazan.

- `select` – (padajuća) lista. Pojedinačne stavke zadaju se elementom `option` čiji sadržaj predstavlja stavku koja se vidi u listi. Element `option` se opet karakteriše atributima `name` i `value` koji su relevantni prilikom slanja podataka.
- `textarea` – polje za unos teksta u više redova. Sadržaj ovog elementa predstavlja tekst koji se inicijalno upisuje u polje (a koji korisnik može da menja). Atribut `name` imenuje podatke koji se upisuju, dok se atributima `width` i `height` određuju dimenzije polja.
- `label` – predstavlja natpis čiji se sadržaj vidi na ekranu. Atribut `for` može da sadrži identifikator neke kontrole (obično `input`) i tada pregledači uspostavljaju određenu vezu između kontrole i natpisa (na primer, kada korisnik klikne na natpis, kontrola se stavlja u žižu tj. fokus).

Text Input – polje za unos teksta

`<input type="text">`

Primer:

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

First name:

Last name:

Password Input – polje za unos šifre

`<input type="password">`

Primer:

```
<form>
User name:<br>
<input type="text" name="userid">
<br>
User password:<br>
<input type="password" name="psw">
</form>
```

User name:

User password:

Lozinke `type="password"` podržavaju isti tip unosa kao i *text* polja s tim što se sadržaj unosa ne vidi na ekranu.

Radio Button Input – radio dugme

`<input type="radio">`

Primer:

```
<form>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other
</form>
```

☒ Male
☐ Female
☐ Other

Value – inicijalna vrednost polja (ono što se ispisuje)

Sva radio dugmad u okviru iste grupe treba da imaju isto ime. Samo izabrano radio dugme generiše par *ime/vrednost* koje se prosleđuje. Sa parametrom *CHECKED* se postavlja predefinisana vrednost.

Input Type Checkbox – dugme za štikliranje

`<input type="checkbox">`

Primer:

```
<form>  
<input type="checkbox" name="vehicle1" value="Bike">Ja imam bicikl  
<br>  
<input type="checkbox" name="vehicle2" value="Car">Ja imam auto  
</form>
```

☐ Ja imam bicikl
☐ Ja imam auto

The Submit Button – dugme za slanje sadržaja forme

`<input type="submit">`

Primer:

```
<form action="/action_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey"><br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse"><br><br>  
<input type="submit" value="Submit">  
</form>
```

First name:

Last name:

Input type reset – dugme za resetovanje na default-ne vrednosti

Primer

```
<form action="/action_page.php">  
First name:<br>  
<input type="text" name="firstname" value="Mickey"><br>  
Last name:<br>  
<input type="text" name="lastname" value="Mouse"><br><br>  
<input type="submit" value="Submit">  
<input type="reset">  
</form>
```

First name:

Last name:

Ostali Input tipovi

HTML5 je uvela još 1 tipove:

- email
- color
- date
- datetime-local
- month
- number
- range
- search
- tel
- time
- url
- week

Input type email – polje za unos email adrese

Pomoću oznake `<input type="email">` Web čitaču se saopštava da ne bi smeo da dozvoli da se obrazac pošalje ako korisnik nije upisao nešto što liči na valjanu elektronsku adresu. Kao sa svim ulaznim tipovima, korisnik može da pošalje obrazac s praznim poljem, sem ako postoji atribut `required`.

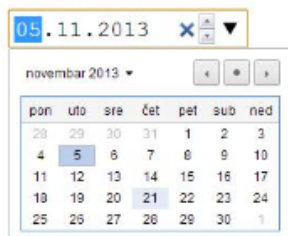


Atribut `multiple` ukazuje na to da vrednost polja može biti lista ispravnih elektronskih adresa razdvojenih zarezom.

Input type date I time – polja za unos datumai vremena

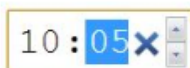
Ulazni tip date

Često je neophodno da se u obrascu unese datum, ali to može biti i problematično zbog postojanja različitih formata (DD-MM-YYYY, MM-DD-YYYY, YYYY-MM-DD,...). Mnogi programeri pišu na JavaScript-u birače datuma koji se razlikuju po izgledu, upotrebljivosti i prisrupočnosti od jedne strane do druge. Pomoću oznake `<input type="date">` nalažemo Web čitaču da prikaže kontrolu za izbor datuma.



Ulazni tip time

Pomoću oznake `<input type="time">` može se korisniku omogućiti unos vremena u 24-časovnom formatu i provera valjanosti upisanog podatka. I ovom slučaju zavisi od čitača kako će kontrola izgledati.



Važno:

`type="date"` kao i `type="time"`, `type="month"`, nisu podržani u browser-ima Firefox, i Internet Explorer 11 i ranijim verzijama.

Input type number – polje za unos brojeva

Oznaka `<input type="number">` omogućava proveru valjanosti numeričkog unosa. Nije predviđena za brojeve telefona jer se oni često pišu sa razmacima, zagradama, znakom plus.



Pomoću atributa `min` i `max` vrši se ograničavanje vrednosti koje se mogu uneti i ne može se poslati obrazac ako je u njega unet broj koji nije iz definisanog opsega.

```
<input type="number" min="2" max="10">
```

Atributi kojima podešavamo veličinu polja za unos i postavljamo ograničenja

Atribut size

- `size` – veličina odnosno širina tekst polja u karakterima

```
<input type="text" name="txtIme" size="10" >  
<input type="text" name="txtIme" size="20" >  
<input type="text" name="txtIme" size="30" >
```

Rezultat je:



Atribut maxlength

- `maxlength` – maksimalan broj karaktera koji se može uneti u polje. Na primer za poštanski broj nekog mesta potrebno je max 5 cifara:

```
<input type="text" name="txtIme" maxlength="5" >
```

Atributi cols i rows

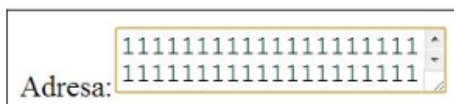
Ovi atributi se koriste kod višerednog tekstualnog polja `<textarea>`.

`cols` i `rows` ukazuju koliko će kolona odnosno redova zauzeti polje za unos podataka *textarea*.

Primer:

```
<textarea name="txArAdresa" >
```

Tekstualna oblast ima određenu širinu i visinu, i ako se posebno ne navedu one iznose 20 karaktera širine i dve linije teksta po visini:



Ostali važni atributi

Atribut name

Atribut name služi da podatke unete u polje input proslede dalje . Ako je atribut name izostavljen, podaci koji su uneti u to polje se ne prosledjuju.

Primer:

```
<form action="/action_page.php">  
  First name:<br>  
  <input type="text" value="Mickey">  
  <br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse">  
  <br><br>  
  <input type="submit" value="Submit">  
</form>
```

Kada se klikne na Submit dugme, uneti podaci se prosleđuju na stranicu pod nazivom action_page.php.

Međutim, "First name" nema atribut name, zbog čega podaci uneti u ovo polje neće biti prosledjeni dalje.

The Action Attribute – atribut action

Atribut action definiše šta se dešava sa formom kada se klikne na dugme za slanje sadržaja forme.

Normalno, podaci forma se šalju na veb stranice na serveru kada korisnik klikne na dugme za prijavu.

U gornjem primeru, podaci sa obrazca se šalje na stranicu na serveru pod nazivom "/action_page.php". Ako je atribut action izostavljen, podaci sa forme se šalju na trenutnu stranu.

The Method Attribute – atribut metod get i post

Postoje dve metode prosleđivanja podataka HTML forme PHP dokumentu, i to su **POST** i **GET** metoda

```
<form action="/action_page.php" method="get">  
ili  
<form action="/action_page.php" method="post">
```

Odabir metode prosleđivanja podataka forme se vrši navođenjem u method argumentu <form> taga. Moguće vrednosti method argumenta zato su kao što smo rekli "post" ili "get".

Ako odaberemo metodu GET podaci forme se šalju kroz komandnu liniju query string, tj. iza znaka ? u adresi baru brauzera. Zato se ova metoda ne koristi za slanje osetljivih informacija.

Izgled address bar-a, primer:

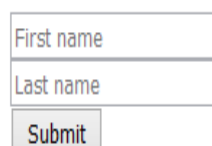
/action_page.php?firstname=Mickey&lastname=Mouse

Odabirom metode POST podaci nisu vidljivi u komandnoj liniji već se šalju kroz header HTTP requesta, i time se na podatke ne može uticati izmenom linka u address baru brauzera kao što to može biti slučaj sa GET metodom. Koristi se za slanje personalnih (poverljivih) informacija.

Atribut placeholder – tekst koji će nestati kada krenemo da popunjavamo polje za unos teksta

Primer

```
<form action="/action_page.php">  
  <input type="text" name="fname" placeholder="First name"><br>  
  <input type="text" name="lname" placeholder="Last name"><br>  
  <input type="submit" value="Submit">  
</form>
```

A simple web form with two text input fields. The first field has the placeholder text "First name" and the second field has the placeholder text "Last name". Below the fields is a "Submit" button.

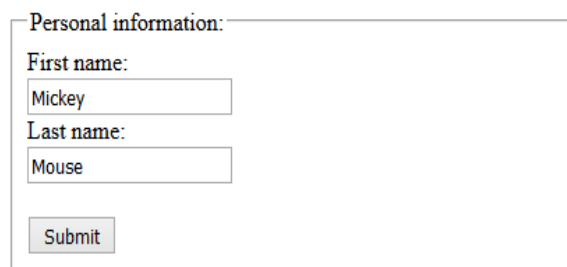
Ostali tipovi formi

Grupisanje podataka <fieldset>

<fieldset> element se koristi za grupisanje podataka u formu, a **<legend>** predstavlja naslov grupe.

Primer

```
<form action="/action_page.php">  
  <fieldset>  
    <legend>Personal information:</legend>  
    First name:<br>  
    <input type="text" name="firstname" value="Mickey">  
    <br>  
    Last name:<br>  
    <input type="text" name="lastname" value="Mouse">  
    <br><br>  
    <input type="submit" value="Submit">  
  </fieldset>  
</form>
```

A web form titled "Personal information:" enclosed in a fieldset. It contains two text input fields. The first field is labeled "First name:" and contains the text "Mickey". The second field is labeled "Last name:" and contains the text "Mouse". Below the fields is a "Submit" button.

Padajuća lista <select>

<select> elementi definišu padajuću listu.

Primer

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```



<option> elementi nam nude opcije koje mogu biti selektovane u okviru padajuće liste.

Vrednost koju želimo da stoji na vrhu padajuće liste označavamo atributom selected.

Primer:

```
<option value="fiat" selected>Fiat</option>
```

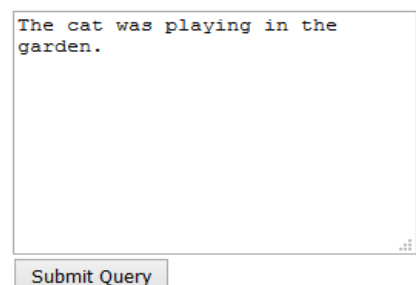
Atributom `size` određuje broj vidljivih stavki u listi.

Višelinijaska tekstualna polja <textarea>

Višelinijaska tekstualna polja se obeležavaju kodom TEXTAREA. Za menije sa jednostrukim ili višestrukim izborom koristimo element SELECT odnosno MULTIPLE.

Sadržaj tekst polja možete predefinisati atributom VALUE. Vrednost atributa je željeni sadržaj tekst polja.

```
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
<br>
<input type="submit">
```



Vežba 3

Napraviti formular koji izgleda kao na slici.

- Postaviti tekstualnu oblast za unos **Adrese** sa 3 reda I 35 karaktera širine
- Za Grad staviti: Čačak, Beograd, Niš I Novi Sad. Inicijalno je izabran Niš I prikazuju se 3 grada
- Za Godine staviti opcije 0-15, 16-30, 31-45, preko 45.

Izgled formulara:

Ime:

Prezime:

Adresa:

Grad:

Pol: ☒ muški ☐ ženski

Godine:

☐ Da, želim da mi pošaljete informacije na e-mail adresu

Vežba 4

- Grupisati podatke u formu, a za naslov grupe staviti "Personalne informacije"
- Za e-mail koristiti određeni Input type (ako se ne unese e-mail adresa, korisnik će biti upozoren da nešto nije u redu)
- Za Datum početka obuke koristiti određeni Input type
- Ukupan broj pohađanih kurseva koristiti određeni Input type I podesiti da broj pohađanih kurseva ne bude manji od 0 I veći od 20
- Za nazove kursa staviti Web dizajn, C#, Napredna škola programiranja za decu, Java. Inicijano je izabran C#.
- Postaviti tekstualnu oblast za unos **komentara** sa 4 reda I 50 karaktera širine. Tekst koji je ispisan u tekstualnoj oblasti nestaje kada se počne sa unosom teksta.

Personalne informacije:

Ime:

Prezime:

E-mail:

Datum početka obuke:

Broj pohađanih kurseva (između 0 i 20):

Odaberite naziv kursa:

Da li vam se dopada sadržaj kursa: ☒ da ☐ ne

Pomozite nam da unapredimo kurs koji pohađate:

Unesite komentar...

Potvrdi

Vežba 3 rešenje:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formular</title>
  </head>
  <body>
    <form name="frmPodaci" method="get" action="">
      Ime: <input type="text" name="txtIme"> <br><br>
      Prezime: <input type="text" name="txtPrezime"> <br><br>

      Adresa:
      <textarea name="txtAdresa" cols="35" rows="3"></textarea>

      Grad:
      <select name="selGrad" size="3" id="selGrad">
        <option>Čačak</option>
        <option>Beograd</option>
        <option selected>Niš</option>
        <option>Novi Sad</option>
      </select><br><br>

      Pol:
      <input type="radio" name="rbPol" value="M" checked> muški
      <input type="radio" name="rbPol" value="Z"> ženski <br><br>

      Godine:
      <select name="selGodine">
        <option>0-15</option>
        <option>16-30</option>
        <option>31-45</option>
        <option>preko 45</option>
      </select><br><br>

      <input type="checkbox" name="chToMail">Da, želim da mi
      pošaljete informacije na e-mail adresu<br><br>

      <input type="submit" name="btnPotvrđi" value="Potvrđi">
      <input type="reset" name="btnObrisi" value="Obrisi">

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

Vežba 4 rešenje:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Formular</title>
  </head>
  <body>
    <form name="frmPodaci" method="get" action="/action_page.php">
      <fieldset>
        <legend>Personalne informacije:</legend>
        Ime:<br>
        <input type="text" name="firstname">
        <br>
        Prezime:<br>
        <input type="text" name="lastname">
        <br><br>
        Odaberite naziv kursa:
        <select name="course" id="selGrad">
          <option>Web dizajn</option>
          <option>Napredna škola programiranja za decu</option>
          <option selected>C#</option>
          <option>Java</option>
        </select><br><br>
        Da li vam se dopada sadržaj kursa:
        <input type="radio" name="rbPol" value="M" checked> da
        <input type="radio" name="rbPol" value="Z"> ne<br><br>
        Pomozite nam da unapredimo kurs koji pohađate:<br>
        <textarea    rows="4"    cols="50"    placeholder="Unesite
komentar..."></textarea>
        <br><br>
        <input type="submit" value="Submit">
      </fieldset>
    </form>
  </body>
</html>
```

Dan 3

CSS - Cascading Style Sheets

HTML određuje sadržaj odnosno strukturu dokumenta, kojom se opisuju deo sadržaja dokumenta: naslov, paragraf, slika, listu, link na drugu stranicu...

CSS određuje izgled dokumenta, odnosno definisanje margina, boja, fontova, razmaka, pozicija objekata na stranici. Atributi se mogu definisati sa znatno većom preciznošću nego što to dopušta html. Možete pozicionirati sadržaj stranice sa preciznošću reda piksela. I to na jednom mestu vrlo jednostavno, a ne da komande neprestano ponavljamo na celoj strani.

W3C -standardizuje primenu stilova u HTML dokumentu, tako što umesto promene ili dogradnje samog HTML jezika koji definiše sadržaj, dodaje CSS koji daje punu kontrolu nad izgledom stranice ili celog Web sajta.

Primena CSS-a

Postoje tri načina na koji CSS može biti ugrađen u HTML I to:

Inline (Inlajn stil) – Ovaj stil se ugrađuje u HTML upotrebom atributa style. Ovo je stil koji se koristi kod pojedinačnih elemenata. Može biti vrlo koristan ako je potrebno da elementu samo jednom promenite stil, ali se ne smatra idealnim rešenjem.

Primer

```
<h1 style="color:blue;">Ovo je plavi naslov</h1>
```

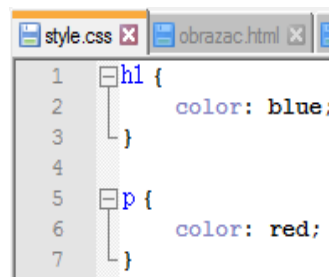
Internal (Ugrađeni stil) - Ovo je stil koji kontroliše jedan dokument i koji se postavlja unutar taga <style> u <head> sekciju HTML dokumenta.

Primer:

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      h1 {color:blue;}
    </style>
  </head>
  <body>
    <h1>Ovo je plavi naslov.</h1>
  </body>
</html>
```

Stranica stilova se sa HTML dokumentom povezujemo pomoću taga `<link>` unutar `<head>` sekcije. Oba dokumenta treba čuvati u istom folderu.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <h1>Ovo je plavi naslov.</h1>
    <p>Ovo je paragraf.</p>
  </body>
</html>
```



Selektor deklaracija deklaracija

h1 {color:blue; font-size:12px;}

 Svojstvo vrednost svojstvo vrednost

CSS selektori služe da pronađu (ili selektuju) HTML elemente pomoću naziva elementa, identifikatora (id), klase (class) itd.

p { color: red;}

tada se svim pasusima (svi elementima p) u dokumentu boja teksta menja u crvenu.

Selekcija pomoću identifikatora (ID)

Ponekad želimo da se podešavanje stila odnosi samo na jedan konkretan element. Jedna mogućnost je da se sva stilska podešavanja navedu u okviru atributa style tog elementa. Na primer, plavu boju teksta u nekom pasusu možemo postaviti na sledeći način.

```
<p style="color:blue">U ovom pasusu biće opisana glavna svojstva ... </p>
```

Ipak, važan nedostatak ovog pristupa je to što su stilska podešavanja isprepletana sa obeležavanjem strukture i sadržaja dokumenta. Mnogo bolji pristup je taj da se elementu pridruži neki identifikator (navođenjem atributa id) i da se kasnije taj identifikator iskoristi kao CSS selektor (oblika #id) da bi se stilska podešavanja primenila na taj konkretan element. Identifikator mora biti jedinstven i to na nivou čitave veb-stranice (HTML dokumenta), tj. u čitavoj stranici ne sme da postoji drugi element čiji bi identifikator bio jednak datom. Na primer, ako bismo želeli da stilizujemo pasus koji sadrži neki opis, u telu dokumenta taj pasus bi smo obeležili na sledeći način.

```
<p id="opis">U ovom pasusu biće opisana glavna svojstva ... </p>
```

Podešavanje boje bi se onda uradilo ili u okviru elementa style u zaglavlju dokumenta ili u zasebnoj datoteci .css.

```
p#opis {  
    color: blue;  
}
```

Selektor p#opis označava da je u pitanju pasus (element p) sa identifikatorom opis. Pošto je identifikator jedinstven, ime elementa nije neophodno navoditi.

```
#opis {  
    color: blue;  
}
```

Ipak, ako se navede i ime elementa čitaocu CSS pravila može biti malo jasnije šta se zapravo stilizuje, pa ćemo ime elementa uvek navoditi uz identifikator.

Selekcija pomoću klase (class)

Ponekad se želi stilizovanje nekoliko konkretnih elemenata istovremeno. Jedan način je da se svakom elementu koji se želi stilizovati na odabrani način dodeli neka klasa postavljanjem atributa class), a zatim da se ta klasa upotrebi kao CSS selektor (oblika .class). Na primer, u dokumentu možemo imati puno pasusa, od kojih se u nekim od njih rezimiraju osnovne stvari navedene u prethodnim pasusima. Ako želimo da pasuse koji predstavljaju rezimee uokvirimo i ispišemo plavom bojom, to možemo uraditi na sledeći način. Prvo, u HTML dokumentu pasusima koji predstavljaju rezimee dodelićemo klasu rezime. Klasa se dodeljuje postavljanjem atributa class.

<p class="rezime">Jezik HTML je ...</p>

...

<p class="rezime">Jezik CSS je ... </p>

U okviru stilskog opisa (bilo u elementu style, bilo u zasebnom stilskom listiću) izvršili bismo sledeće podešavanje.

```
p.rezime { color: blue; border: 1px solid black; }
```

Opet je ime elementa moguće izostaviti.

```
.rezime { color: blue; border: 1px solid black; }
```

Međutim, pošto više različitih elemenata mogu pripadati istoj klasi (na primer, moguće je istovremeno definisati i neki element div klase rezime, tj. Element oblika <div class="rezime">...</div>), prethodna dva CSS pravila ne moraju biti identična. U prvom se vrši stilizovanje svih pasusa klase rezime , a u drugom se vrši stilizovanje svih elemenata klase rezime (u našem primeru, bili bi stilizovani elementi p , ali i element div).

Razlika između CSS klase i ID-a

- Glavni korisnici klasa i ID-a su CSS i Javascript, preko kojih možemo upravljati određenim elementom unutar HTML-a. Ali koje su razlike?
- ID je jedinstven:
- element može imati samo jedan definisan ID
- na stranici ne može (ustvari ne sme) biti više elemenata sa istim ID-em Klase nisu jedinstvene:
- jedan element može imati više klasa
- na stranici može biti bezbroj elemenata sa istom klasom

Nasleđivanje stilskih opisa

Neka svojstva se nasleđuju kroz celo stablo dokumenta, tj. u nekim slučajevima kada se

deklaracija stila pridruži nekom elementu, istu deklaraciju automatski nasleđuju i svi elementi sadržani u tom elementu. Na primer, ako se podesi boja slova tela dokumenta, svi sadržani elementi nasleđuju navedenu boju.

```
body {  
    color: red;  
}
```

Sa druge strane, nasleđeno svojstvo se može promeniti i eksplicitno podešena vrednost nekog svojstva uvek ima prednost u odnosu na nasleđenu. Na primer, ako posle podešavanje boje

teksta celog tela dokumenta na crvenu podesimo boju teksta u svim pasusima na plavu

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

tada će boja teksta u svim pasusima biti plava, dok će boja teksta van pasusa biti crvena.

Neka svojstva se ne nasleđuju. Na primer, ako se podesi margina telu dokumenta, to ne znači da će i svi sadržani elementi imati istu marginu.

Kaskada stilskih opisa

Stilski opis za neki element može biti istovremeno naveden i na nekoliko različitih mesta. Autori dokumenta mogu svojim dokumentima da pridruže stilske opise na tri prethodno opisana načina (u spoljašnjim stilskim listovima, u okviru elementa `style` i atributa `style`). Korisnik može svojim podešavanjima pregledača da utiče na stil i postoji stilski opis na nivou korisnika. Ukoliko ni autor dokumenta ni korisnik ne navedu stilski opis za neki element, onda se koristi podrazumevani stil tog elementa zadat na nivou korisničkog agenta (tj. pregledača veba). Različiti opisi za neki element se „sabiraju” tj. kombinuju kako bi se dobio konačan stil prikaza elementa. U slučaju da dođe do konflikta, prednost se daje stilskim listovima autora, zatim stilskim listovima korisnika, dok podrazumevani stilski listovi korisničkih agenata imaju najmanji prioritet. U slučaju da do konflikta dođe u okviru stilskog lista autora, prioritet imaju pravila navedena na nivou elementa (u okviru atributa `style`), zatim pravila navedena u na nivou dokumenta (u okviru elementa `style`) i tek na kraju pravila navedena u spoljašnjim listovima (uključene elementom [link](#)).

Ova osobina se često u praksi koristi, da bi se promenila neka ranija podešavanja. Na primer, ako se na nivou celog veb-sajta promene margine pasusa, to podešavanje bi bilo uneto na nivou spoljašnjeg lista koji se uključuje u sve stranice sajta. Ako u nekoj stranici želimo da promenimo to podešavanje, samo je potrebno da u njenom zaglavlju promenimo podešavanje margine za sve pasuse i to novo podešavanje će imati prednost u odnosu na ona podešavanja učitana iz spoljašnjeg lista.

Najčešće korišćena svojstva i njihove vrednosti

U ovom poglavlju biće navedena neka najčešće korišćena CSS svojstva i njihove najčešće korišćene vrednosti. Potpun pregled svojstava i vrednosti moguće je naći u zvaničnoj CSS specifikaciji.

Stilizovanje teksta (text)

Određeni broj svojstava posvećen je podešavanju teksta – moguće je podešavati uvlačenje prve linije, poravnavanje teksta, razmak između reči i slova i slično. Za podešavanje teksta koriste se naredna CSS svojstva.

- **text-indent** – Ovo svojstvo definiše uvlačenje prve linije teksta u okviru elemenata. Vrednost se najčešće navodi bilo u obliku dužine ili u obliku procenta tj. jedinice em.

```
p {
  text-indent: 50px;
}
```

- **text-align** - Ovo svojstvo definiše poravnavanje teksta u okviru elementa. Moguće su vrednosti left (poravnavanje na levu stranu), right (poravnavanje na desnu stranu), center (centriran tekst) ili justify (tekst poravnat na obe strane).

```
p {text-align: left;}
```

- **text-decoration** – Ovim svojstvom se mogu zadati različite dekoracije teksta. Moguće vrednosti su:

none (tekst nema dekoracija), underline (tekst je podvučen), overline (tekst je nadvučen), line-through (tekst je precrtan) I blink (tekst „trepće”). Na primer, linkovi su obično podrazumevano podvučeni i dizajneri često isključuju taj efekat, osim kada je miš iznad linka.

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

- **text-transform** – Ovim svojstvom se zadaju različite automatske transformacije teksta. Vrednost uppercase prouzrokuje da se ceo tekst, bez obzira na to kako je otkucan, prikaže velikim slovima (na primer, OVO JE NEKI TEKST), lowercase (prouzrokuje da se ceo tekst prikaže malim slovima (na primer, ovo je neki tekst), a capitalize da prva slova reči prikazuju kao velika (na primer, Ovo Je Neki Tekst). Na primer, ako želimo da sva slova u svim glavnim naslovima na stranici budu velika, to možemo postići na sledeći način.

```
h1 { text-transform: uppercase; }
```

- **letter-spacing** – Ovim svojstvom se podешава razmak између слова (vrednost je dimenzija obično zadata u pikselima, tачкама или јединици em).

- **word-spacing** – Ovim svojstvom se подешава размак између речи (vrednost je dimenzija obično zadata u pikselima, tачкама или јединици em).

- **line-height** - Ово својство одређује висину једне линије текста. Обично користи да би се задао размак између суседних линија текста. Вредност се обично задаје као децимални број или проценат при чему се размак између линија израчунава као задати умножак величине фонта. На пример,

```
p { line-height: 1.5; }
```

поставља такозвани проред један-и-по, тј. одређује ће висина једне линије текста бити 150% величине текста унутар ње.

- **color** – Ovo svojstvo određuje boju teksta koji se nalazi u elementu. Vrednost ovog svojstva se nasleđuje (na primer, ako se postavi boja teksta na nivou celog tela veb-stranice, tu boju će imati i tekst u svim pasusima koji se nalaze u

telu).

Svojstva kojima se podešava boja pozadine elemenata ili boja okvira nekog elementa biće prikazana kasnije.

Boje se mogu zadati na nekoliko načina. Najkorišćeniji načini da se zada boja tj. nijansa boje su sledeći:

- **ime boje** – boja se zadaje preko imena boje (na primer, red, yellow, black , ...). Na primer, naredno pravilo postavlja boju teksta u pasusima na crvenu:
`p { color: red; }`

- **heksadekadni kôd oblika #rrggbb** – boja se zadaje zadavanjem tri dvocifrena heksadekadna broja koji određuju redom crvenu, zelenu i plavu komponentu boje. Vrednost 00 je najmanja, a vrednost ff je najveća. Svedeno je da li se pišu velika ili mala heksadekadna slova. Na primer, vrednost #00ff00 označava zelenu boju (vrednost 0 kod crvene komponente i zelene komponente i maksimalna moguća vrednost kod plave komponente). Stoga, naredno pravilo postavlja sav tekst u pasusima na zelenu boju:
`p {color: #00ff00;}`

- **heksadekadni kôd oblika #rgb** – boja se može zadati i zadavanjem tri heksadekadne cifre. Zapis #rgb predstavlja skraćeni oblik zapisa #rrggbb. Na primer, zapis #27c je skraćeni zapis za boju #2277cc.

- **Dekadna specifikacija oblika rgb(r, g, b)** – boja se zadaje zadavanjem tri dekadna broja (između 0 i 255) koji određuju redom crvenu, zelenu i plavu komponentu boje. Ovaj način prilično odgovara prethodnom, jedino što se vrednosti zadaju u dekadnom, umesto u heksadekadnom brojevnom sistemu. Na primer, narednim pravilom se boja teksta u svim pasusima postavlja na plavu:
`p { color: rgb(0, 0, 255); }`

Slovni lik (fonts)

Naredna svojstva definišu izgled slovnih likova tj. fontova (engl. fonts) koji se koriste prilikom ispisa teksta. Sva ova svojstva se mogu primeniti na bilo koji element i automatski se nasleđuju na sve sadržane elemente.

- **font-family** – Ovo svojstvo određuje font ili familiju fontova koji će se koristiti prilikom ispisa teksta. Moguće je navesti precizan naziv fonta, pri čemu se imena fontova koja imaju više reči obično navode pod navodnicima (npr. "Times New Roman", Arial, "Courier New"), ime familije fontova (npr. Times) ili ime vrste fontova (npr. serif, sans-serif, monospace, cursive, fantasy). Moguće je navesti i više opisa u opadajućem prioritetu. Tako, na primer,

```
p {  
  font-family: "New Century Schoolbook", Times, serif;  
}
```

označava da se koristi New Century Schoolbook font ako postoji na sistemu. Ako ne postoji, koristi se bilo koji Times font, a ako ne postoji, koristi se bilo koji serifni font.

Ko želi da zna više?

Veliki broj fontova koji se mogu koristiti na vebu dostupan je na stranici <https://www.google.com/fonts>.

- **font-style** – Ovo svojstvo određuje iskošenost teksta. Moguće vrednosti su normal, italic i oblique.

- **font-variant** – Ovo svojstvo daje mogućnost korišćenja „MALIH-VELIKIH” slova. Moguće vrednosti su normal i small-caps.

- **font-weight** – Ovo svojstvo određuje „debljinu” slova. Najčešće vrednosti su normal, bold i lighter.

- **font-size** – Ovo svojstvo određuje veličinu fonta (to nije veličina nijednog konkretnog karaktera, već je obično veličina malo veća od visine koja se izračunava kao razmak između najniže tačke koja se može dostići i najviše tačke koja se može dostići u nekom karakteru (ne obavezno istom). Najčešće se vrednosti zadaju u apsolutnim jedinicama mere kakve su pt ili px. Tako, na primer, pravilo

```
p {  
font-size: 12pt;  
}
```

određuje da će veličina teksta u svim pasusima biti 12 tipografskih tačaka. Pored ovoga, moguće je navoditi i relativne mere (u procentima) ili u jedinicama kakva je em koja predstavlja umnožak trenutne (podrazumevane ili nasleđene) velične fonta. Tako, na primer, pravila

```
p {  
font-size: 120%;  
}
```

```
p {  
font-size: 1.2em;  
}
```

oba određuju da će veličina teksta biti 120% trenutne.

Pozadina (background)

Svakom elementu je moguće modifikovati pozadinu korišćenjem CSS-a. Moguće je postaviti boju pozadine, ali i u pozadinu postaviti neku sliku i dodatno podesiti mnoga njena svojstva. Pozadina može da obuhvata samo sadržaj elementa, sadržaj i punjenje ili sadržaj, punjenje i okvir (što je podrazumevano).

- **background-color** – Ovo svojstvo određuje boju pozadine elementa. Svojstvo se nasleđuje. Osim specifikacije boje, vrednost ovog svojstva može da bude i transparent čime se element čini providnim. Osim zadavanja boje, pozadina nekog elementa može biti i slika.

Primer:

```
body {
  background-color: lightblue;
}
```

• **background-image** – Ovo svojstvo određuje sliku koja će biti prikazana kao pozadina elementa. Svojstvo se ne nasleđuje. Moguće je dodatno podesiti neke parametre prikaza slika (npr. poziciju, način ponavljanja). Moguće vrednosti su none kada se ne koristi slika ili URL slike u obliku url(...). Preporučuje se da se prilikom korišćenja ovog svojstva postavi i boja pozadine (korišćenjem svojstva background-color). Boja se prikazuje u slučaju da slika nije dostupna, i prikazuje se na onim delovima elementa gde slika nije postavljena ili je providna.

Primer:

```
body {
  background-image: url("paper.gif");
}
```

• **background-repeat** – U slučaju kada je elementu postavljena pozadinska slika (korišćenjem svojstva background-image), ovim svojstvom se podešava da li će slika biti prikazana samo jednom ili će biti ponavljana dok ne ispuni celu širinu i/ili visinu elementa. Moguće vrednosti su:

- repeat – slika se ponavlja i horizontalno i vertikalno.
- repeat-x – slika se ponavlja samo horizontalno.
- repeat-y – slika se ponavlja samo vertikalno.
- no-repeat – slika se ne ponavlja (prikazuje se samo jedna kopija)

Primer:

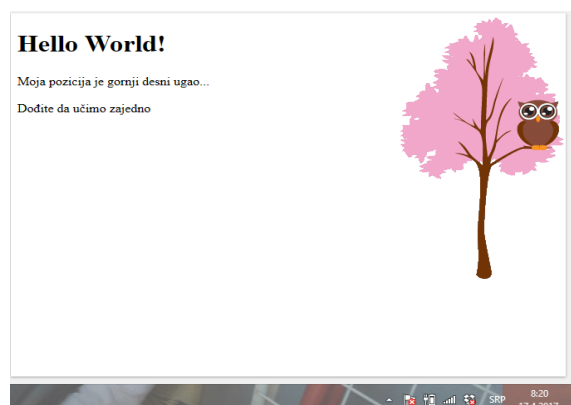
```
body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
```

• **background-position** – U slučaju kada je elementu postavljena pozadinska slika (korišćenjem svojstva background-image), ovim svojstvom se određuje njegova početna pozicija. Ukoliko se navedu dve vrednosti, prva određuje horizontalnu, a druga vertikalnu poziciju. Ukoliko se navede samo jedna vrednost, za drugu se podrazumeva center. Vrednosti mogu biti navedene na sledeći način:

- procenat – Vrednost od x% poravnava tačku koja se nalazi na x% širine (dužine) slike sa tačkom koja se nalazi na x% širine (dužine) elementa.

Tako npr. 0% 0% poravnava gornja leva temena, dok 100% 100% poravnava donja desna.

- rečima –
- left top
- left center
- left bottom
- right top
- right center
- right bottom



center top
center center
center bottom

Primer:

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

• **background-attachment** – U slučaju kada je elementu postavljena pozadinska slika (korišćenjem svojstva background-image), ovim svojstvom se određuje da li će se prilikom pomeranja sadržaja stranice (tzv. skrolovanja) slika pomerati zajedno sa sadržajem ili će ostati fiksna u odnosu na prozor I posmatrača. U prvom slučaju navodi se vrednost scroll , a u drugom fixed).

Primer:

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
    background-attachment: fixed;  
}
```

• **background (skraćeni zapis)** – Ovo svojstvo omogućava da se istovremeno navede više aspekata pozadine.

Na primer, umesto:

```
body {  
    background-color: #aaaaaa;  
    background-image: url('background.png');  
    background-repeat: repeat-x;  
    background-attachment: scroll;  
    background-position: right top;  
}
```

moгуće je koristiti:

```
body { background: #aaaaaa url("background.png") repeat-x scroll right top; }
```

Margine (margins)

Margine predstavljaju prazan prostor oko elemenata. Kada se kao vrednost svojstva margin navede jedna dimenzija (na primer, margin: 10px) podrazumeva se da se ona odnosi na sve četiri margine (levu, desnu, gornju i donju).

Ako se navedu dve dimenzije (na primer, margin: 10px 20px) prva od njih se odnosi na gornju i donju marginu, a druga na levu i desnu.

Ako se navedu četiri dimenzije (na primer, margin: 10px 20px 30px 40px) one se odnose redom na gornju, desnu, donju I levu marginu (navode se, dakle, u pravcu kazaljke na satu).

Ako se navedu tri dimenzije (na primer, margin: 10px 20px 30px), one se odnose redom na gornju, levu I desnu, donju

Vrednost auto prouzrokuje da se margine automatski podjednako rasporede što se jako često koristi u centriranju elemenata.

Moguće je zasebno podešavati svaku od četiri margine i to atributima:

margin-top,
margin-right,
margin-bottom i
margin-left.

Primer:

```
p {  
    margin-top:20px;  
    margin-right:30px;  
    margin-bottom: 40px;  
    margin-left: 20px;  
}
```

ili se može pisati skraćeni oblik

```
p {margin: 20px 30px 40px 20px;}
```

Padding

Ovim svojstvo podešava se unutrašnja margina (tj. punjenje) i ono se koristi analogno svojstvu margin. Pojedinačne dimenzije mogu se podešavati svojstvima

padding-top,
padding-right,
padding-bottom I
padding-left.

Sva ostala pravila su kao I kod margina (skraćeni oblik pisanja I dr.)

Primer:

```
p {  
    padding-top: 50px;  
    padding-right: 30px;  
    padding-bottom: 50px;  
    padding-left: 80px;  
}
```

ili se može pisati skraćeni oblik


```
p {padding: 50px 30px 50px 80px;}
```


Visina I širina (height/width)

Svojstva `width` i `height` služe za postavljanje širine i visine elementa. Podrazumevana vrednost visine I širine je `auto` (browser izračunava širinu I visinu). Takođe, visinu I širinu možemo podesiti u sledećim jedinicama: `px`, `%`, `cm`...

Primer:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  height: 200px;
  width: 50%;
  background-color: powderblue;
}
</style>
</head>
<body>
  <p>Ovaj div element ima visinu 200px i širinu
50%:</p>
  <div></div>
</body>
</html>
```

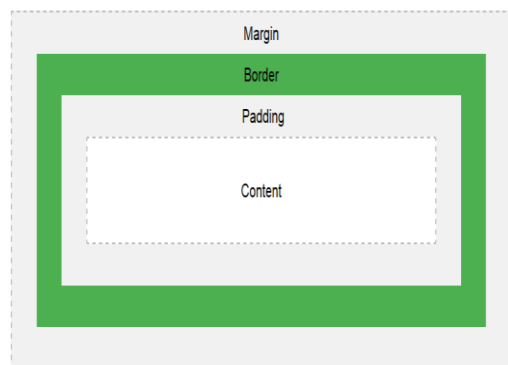
Ovaj div element ima visinu 200px i širinu 50%:


max-width, max-height, min-width, min-height – Ova svojstva određuju maksimalnu odnosno minimalnu visinu i širinu elementa. Šta se u širinu i visinu uračunava, zavisi od vrednosti svojstva `box-sizing` (isto kao u slučaju svojstava `width` i `height`). Ova svojstva mogu da imaju neke prednosti u odnosu na eksplicitno postavljanje širine i visine (svojstvima `width` i `height`). Na primer, kada se ekran suzi, kutijice kojima je postavljena maksimalna širina se takođe sužavaju, za razliku od onih kojima je širina fiksna.

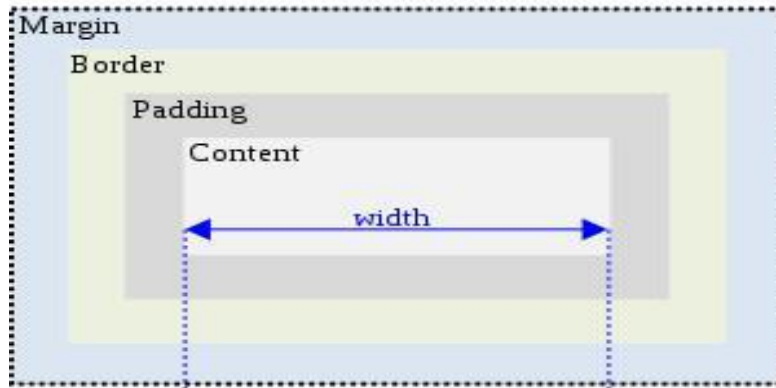
Model kutija (Box Model)

Svi HTML elementi mogu da se smatraju pravougaonim površinama — kutijama, (engl. boxes). Svaka kutija ima svoj sadržaj (engl. content) i može da ima svoj okvir (eng. Border). Okvir je razdvojen od sadržaja unutrašnjom marginom tj. punjenjem (engl. padding), a od okolnih elemenata spoljašnjom marginom (engl. margin).

- **margin** – prazan prostor oko bordure, to je potpuno transparentno,
- **border** - granica koja ide oko sadržaja
- **padding** – prazan prostor između sadržaja i bordure, transparentan,
- **content** – sadržaj, gde se tekst i slike pojavljuju.



W3C box model

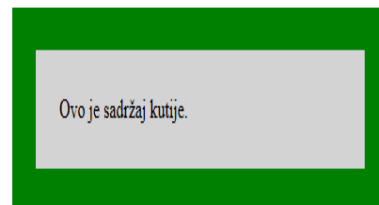


Internet Explorer box model



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
background-color: lightgrey;
width: 300px;
border: 25px solid green;
padding: 25px;
margin: 25px;
}
</style>
</head>
<body>
<h2>Demonstracija Box Modela.</h2>
<div>Ovo je sadržaj kutije.</div>
</body>
</html>
```

Demonstracija Box Modela.



Okvir (border)

- **border-width** – Ovim svojstvom se podešava debljina okvira (na primer, border-width: 3px). Pojedinačni okviri (levi, gornji, desni i donji) se mogu podešavati svojstvima border-top-width, border-right-width, border-bottom-width I border-left-width.
- **border-style** – Ovim se svojstvom podešava tip linije okvira (vrednosti su solid za okvir iscrtan punom linijom, dashed za isprekidane linije, dotted za tačkasti okvir itd.). Opet se mogu podešavati i pojedinačni okviri i to svojstvima border-top-style, border-right-style, border-bottom-style I border-left-style.

Primer:

```
p {  
  border-top-style: dotted;  
  border-right-style: solid;  
  border-bottom-style: dotted;  
  border-left-style: solid;  
}
```

Različite vrste okvira.

- **border-color** – Ovim svojstvom se podešava boja okvira. Opet se mogu podešavati i pojedinačni okviri i to svojstvima border-top-color I border-right-color, border-bottom-color I border-left-color.

- **border** – Ovo svojstvo kombinuje sva tri aspekta okvira (debljinu, vrstu linije i boju).

Na primer,

border: 1px solid black;

kreira tanki, crni okvir, iscrtan punom linijom. Pojedinačni okviri se mogu podešavati svojstvima border-top, border-right, border-bottom I border-left.

- **border-radius** – Ovim svojstvom određuje se zaobljenost okvira. Vrednost se zadaje obično u pikselima i određuje poluprečnik zamišljenih krugova upisanih u svaki od 4 temena pravougaonog polja. Što je veći poluprečnik, to je okvir zaobljeniji.

Stilizovanje lista (lists)

I nenumerisane i numerisane liste se mogu stilizovati. Najčešća podešavanja odnose se na podešavanje oznake stavki liste (tačkica, tj. crtica u nenumerisanim i brojki tj. slova u numerisanim listama). Za to se koriste naredna CSS svojstva.

- **list-style-type** – Ovo svojstvo definiše oblik „crtice” u nenumerisanoj listi ili broja u numerisanoj listi (najčešće vrednosti su disc (ispunjen krug), circle (prazan kružić), square (kvadrat), decimal (broj), lower-alpha (mala latinička slova a, b, c, ..), lower-roman (mali rimski brojevi i, ii, iii, iv, ...), upper-alpha (velika latinička slova A, B, C, ...), upper-roman (veliki rimski brojevi I, II, III, IV, ...) itd.).

- **list-style-image** – Umesto „crtice” može se prikazati i neka slika, koja se zadaje postavljanjem ovog svojstva. Na primer, ako želimo da se slika slika.png snimljena u isti direktorijum kao i HTML dokument prikaže umesto tačkica u nenumerisanim listama, to možemo uraditi na sledeći način.

```
ul { list-style-image: url("slika.png"); }
```

- **list-style-position** – Ovo svojstvo određuje da li će se crtice, brojevi ili slike prikazivati unutar liste (tj. unutar njenog okvira) ili van nje. Vrednosti ovog svojstva su inside i outside (defaultno podešavanje).

- **list-style** – Ovo svojstvo objedinjuje prethodna svojstva, tako što se prvo navede vrednost svojstva list-style-type, list-style-position i list-style-image.

Stilizovanje tabela (tables)

- **border, background-color, padding, margin** – ova svojstva su veoma važna za stilizovanje tabela, a koriste se na potpuno isti način kao i kod svih drugih elemenata. Naglasimo da je ove attribute moguće postavljati na nivou tabele (elementa table), na nivou pojedinačnih redova (elementa tr) i pojedinačnih ćelija (elementa td). Takođe je važno da zapamtite da se ivice u HTML5 zadaju putem CSS-a.

Kada se u CSS-u zadaju sl. vrednosti:

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

Ivice tabele izgledaju ovako:

Ime	Prezime
Peter	Petrović
Marko	Matić

Da se ivice „prilepe” koristimo komandu border-collapse:collapse;

- **border-collapse** – Ovo veoma često korišćeno svojstvo uz vrednost collapse prouzrokuje da se susedne ćelije „prilepe” tako da između tako da između njih nema prostora i tako da se između susednih postavlja jedinstveni okvir (u suprotnom svaka ćelija ima svoj posebni okvir).

Primer:

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

Ime	Prezime
Peter	Petrović
Marko	Matić

- border – ako želimo da zadamo samo okvir oko tabele, to radimo na sl. način:

```
table {
  border-collapse: collapse;
  border: 1px solid black;
}
```

Ime	Prezime
Petar	Petrović
Marko	Matić

- **width, height** – zadavanje visine i širine može se vršiti u različitim jedinicama.

Primer:

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

Ime	Prezime	Cena proizvoda
Petar	Petrović	\$100
Marko	Matić	\$150
Milica	Nikolić	\$300
Cleveland	Brown	\$250

```
table {
  border-collapse: collapse;
  width: 100%;
}
```

```
th {
  height: 50px;
}
```

• **text-align, vertical-align** – ova svojstva se koriste da bi se odredilo poravnavanje sadržaja ćelija tabele.

Horizontalno poravnanje se vrši putem komande text-align (left, right, center).

Primer:

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

```
table {
  border-collapse: collapse;
  width: 100%;
}
```

```
th {
  height: 50px;
  text-align: right;
}
```

Ime	Prezime	Cena proizvoda
Petar	Petrović	\$100
Marko	Matić	\$150
Milica	Nikolić	\$300
Cleveland	Brown	\$250

Vertikalno poravnanje se vrši uz pomoć svojstva vertical-align čije su najčešće korišćene vrednosti top, middle i bottom.

Primer:

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

```
table {
  border-collapse: collapse;
  width: 100%;
}
```

```
td {
  height: 50px;
  vertical-align: bottom;}
}
```

Ime	Prezime	Cena proizvoda
Petar	Petrović	\$100
Marko	Matić	\$150
Milica	Nikolić	\$300
Cleveland	Brown	\$250

- **padding** – ovo svojstvo koristimo kako bismo uvukli tekst u odnosu na ivice tabele.

Primer:

```
table, td, th {
    border: 1px solid #ddd;
    text-align: left;
}

table {
    border-collapse: collapse;
    width: 100%;
}

th, td {
    padding: 15px;
}
```

Ime	Prezime
Petar	Petrović
Marko	Matić

- **table color** – boju u tabeli zadajemo pomoću svojstva **background-color:color;**

Primer:

```
th {
    background-color: #4CAF50;
    color: white;
}
```

Ime	Prezime
Petar	Petrović
Marko	Matić

Display (Prikaz)

Svi elementi zauzimaju određeni prostor pravougaonog oblika na ekranu, ali postoje razlike u tome kako se ti pravougaonici formiraju (na primer, koliki prostor zauzimaju) i kako se slažu jedan uz drugi. Izuzetno važno svojstvo, koje određuje način na koji će neki element biti prikazan je svojstvo **display**. Na osnovu njihovog podrazumevanog načina prikaza, elementi se uglavnom dele na blok-elemente (engl. block elements) čija je podrazumevana deklaracija **display: block** i na linijske elemente (engl. inline elements) čija je podrazumevana deklaracija prikaza **display:inline**.

Blok elementi (block- level elements)

Blok elementi uvek počinju u novom redu I zauzimaju red od početka do kraja (full width). To izgleda ovako:

<div>

Blok elementi su:

- <div>
- <h1> - <h6>
- <p>
- <form>
- <header>
- <footer>
- <section>

Linijski elementi (inline elements)

Linijski elementi ne počinju u novom redu i zauzimaju samo onoliko mesta koliko je neophodno. To izgleda ovako:

```
<span>
```

Linijski elementi su:

-
- <a>
-

Tabele nisu ni blok ni linijski elementi.

Uloga tagova DIV i SPAN

Veća fleksibilnost se dobija za izradu dizajna Web strane sa dva nova taga DIV i SPAN u okviru HTML. Tagovi <div> i </div> služe da deo HTML koda kojim se nalazi između njih odredimo CSS stilom. Ako se unutar tagova <div> i </div> treba primeniti neki drugi CSS stil na samo jedan HTML red ili deo reda (html naredbe), tada se kao „nosač“ tog stila koristi par tagova i . U okviru jednog para tagova <div> i </div> može biti ugnježđen drugi par tagova <div> i </div>.

Primer:

```
<html>
<head>
<style>
.b1 {
  width:300px;
  padding:20px;
  border:5px solid red;
  margin:10px;
}
</style>
</head>
<body>

<div class="b1">Tekst u Box modelu.</div>
<p><b>Napomena:</b> Tekst izvan Box modela.</p>
</body>

</html>
```

Display:none;

Ova vrednost označava da element ne treba da se prikaže uopšte, već da se potpuno izostavi iz prikaza (samim tim, ne zauzima nikakav prostor na stranici i okolni elementi se raspoređuju kao da on nije tu).

Ko želi da zna više?

Deklaracija slična deklaraciji `display: none` je deklaracija `visibility:hidden`, međutim, u tom slučaju se element ne prikazuje, ali zauzima prostor na stranici.

Linijski – blok elementi (Inline-block)

Ova vrednost daje prikaz u obliku linijskog bloka – elementa koji se donekle ponaša kao linijski element, jer se ne prostire celom širinom i slaže se uz, a ne ispod i iznad svojih susednih elemenata, a donekle kao blok element jer mu je moguće postavljati širinu, visinu i margine na potpuno isti način kao kod blok elemenata.

Pozicioniranje

Pozicioniranje elemenata vrši se postavljanjem njihovog svojstva `position`. Moguće su četiri vrednosti: `static`, `relative`, `absolute` i `fixed`.

Statičko pozicioniranje (static)

je podrazumevano (elementi kojima vrednost atributa `position` nije postavljena imaju statičko pozicioniranje). To podrazumeva da se oni uklapaju u takozvani normalni tok (engl. `normal flow`). Elementi nivoa bloka se postavljaju jedan ispod drugog, dok se linijski elementi slažu jedan uz drugi.

Relativno pozicioniranje (relative)

omogućava da se elementi pomere u odnosu na svoju statičku poziciju, tj. da se izmeste iz normalnog toka, pri čemu to nikako ne utiče na pozicioniranje ostalih elemenata. Koordinate se zadaju svojstvima `top`, `left`, `bottom` i `right`. Na primer,

```
p {  
  position: relative;  
  top: -10px;  
  left: 20px;  
}
```

prouzrokuje da se svi pasusi pomere 10 piksela gore i 20 piksela desno u odnosu na njihovu normalnu poziciju.

Promenom pozicioniranja elementi mogu i preklopiti, a koji će biti iznad, a koji ispod određuje se svojstvom ***z-index*** – veće vrednosti određuju elemente koji će biti iznad tj. koji će se videti. Naglasimo i da se relativno pozicioniranje često koristi u kombinaciji sa apsolutnim i da se deklaracija `position: relative` bez promene koordinata često koristi kao deo apsolutnog pozicioniranja.

Apsolutno pozicioniranje (*absolute*)

dovodi do toga da se element pozicionira u okviru nekog šireg elementa i to tako što se zadaju njegove koordinate u odnosu na taj širi element. Po definiciji, to je najbliži element koji ga sadrži, a pozicioniran je nekako tj. ima vrednost svojstva `position` različitu od podrazumevane vrednosti `static`. Ako takav element ne postoji, apsolutno pozicioniranje se vrši u odnosu na element `body`. Da bi se naglasilo da će se neki element koristiti kao element u kome će se sadržaj pozicionirati apsolutno, njemu se obično samo vrednost svojstva `position` postavi na `relative`. Apsolutno pozicionirani element se izbacuje iz normalnog toka i ostali elementi se pozicioniraju kao da on ne postoji. Koordinate apsolutno pozicioniranih elemenata se opet zadaju svojstvima `top`, `left`, `bottom` i `right`. Razmotrimo jedan primer. Neka u HTML kodu postoje sledeći elementi:

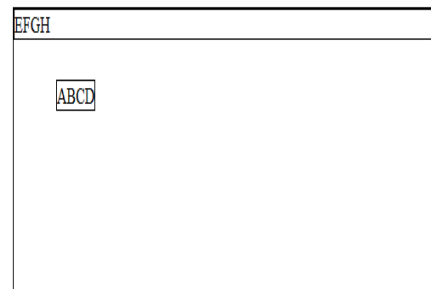
```
<div id="container">
  <div id="content">ABCD</div>
  <div>EFGH</div>
</div>
```

I neka je:

```
div {border: 1px solid black;}
```

```
#container {
  position: relative;
  width: 500px;
  height: 200px;
  top: 50;
  left: 70;
}
```

```
#content{
  position: absolute;
  top: 50px;
  left: 50px;
}
```

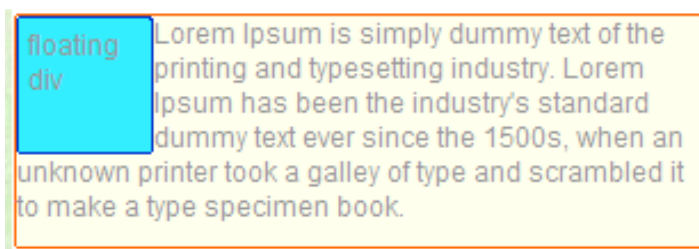


Tada se element sa identifikatorom `content` pozicionira tako da mu je gornji levi ugao 50 piksela ispod i 70 piksela levo od gornjeg levog ugla elementa `container`. Element u kome piše `EFGH` nalazi se u vrhu elementa sa identifikatorom `container` jer je element sa identifikatorom `content` izbačen iz normalnog toka.

Fiksno pozicioniranje (fixed)

je donekle slično apsolutnom, ali se element pozicionira u odnosu na ekran pregledača i ne pomera se prilikom skrolovanja stranice. Najčešće se koristi za fiksiranje naslova ili potpisa stranice. Pošto se i fiksno pozicionirani elementi ibacuju iz normalnog toka, ako se želi obezbediti da ne preklapaju druge elemente, potrebno je tim drugim elementima postaviti odgovarajuće margine.

Plutajući elementi (float)

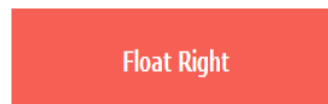
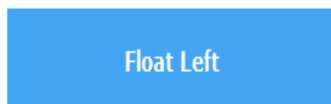


Neki elementi mogu da budu „plutajući” tj. „lebdeći”, tj. da se podese tako da se pomeraju u levi ili desni kraj okružujućeg elementa, a da se ostali sadržaj „obmotava” oko njih (slično kao što se

tekst obmotava oko slika u novinama).

Ovo se postiže svojstvom float, čije su vrednosti left i right. Kada dva susedna elementa imaju istu vrednost svojstva float, oni se pozicioniraju jedan do drugoga (na primer, ako je vrednost left prvi element će se pomeriti nalevo, do početka okružujućeg elementa, a drugi će odlebdeti nalevo, ali samo do kraja prvog lebdećeg elementa).

Ono što je važno da zadamo elementima koji float-uju jeste širina.



Kontrola ponašanja float elemenata (Clear)

Ako se želi da se lebdeći element složi ispod, a ne pored drugog lebdećeg elementa, onda je potrebno navesti mu svojstvo clear (moguće vrednosti su left, right i both).

Komentari u CSS-u

Komentari se u CSS-u pišu na sledeći način:

```
/*Ovo je komentar.*/
```

Dan 4

CSS linkovi (CSS Links)

Linkovi se u CSS dekorišu najčešće pomoću 4 pseudo klase. Te 4 klase se međusobno najčešće definišu različitim atributima da bi link imao vizuelno lep efekat, ali mogu i dva po dva imati iste attribute ili tri pa jedan itd. Evo kako se prave pseudo klase za link:

a:link { } - klasa koja definiše izgled linka koji nikad nije posećen od strane posetioca;
a:visited { } - izgled linka nakon klika na njega;
a:hover { } - izgled linka kad se predje mišem preko njega;
a:active { } - izgled linka u trenutku klika.

Pseudo klase moramo navoditi zadanim redosledom. Primer ispravnog redosleda pseudo klasa:

```
a:link {  
    text-decoration: none;  
    color: red;  
}  
a:visited {  
    text-decoration: none;  
    color: green;  
}  
a:hover {  
    text-decoration: underline;  
    color: blue;  
}  
a:active {  
    text-decoration: underline;  
}
```

text-decoration se najčešće koristi kod linkova kako bi se uklonila linija koja nam je u prvom momentu pokazatelj da se radi o hiperlinku. Da bi se ta linije uklonila, potrebno je zadati svojstvo **text-decoration: none**

Linija se može ponovo vratiti svojstvom **text-decoration: underline;**

Izrada dugmeta (Link buttons)


Primer:

```
<!DOCTYPE html>
<html>
<head>
<style>
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
</style>
</head>
<body>

<a href="default.asp" target="_blank">This is a link</a>

</body>
</html>
```



CSS ikone (CSS Icon)

Najjednostavniji način da dodate ikone na vašu HTML stranicu, jeste upotrebom biblioteke Font Awesome. Bilo koji inline element može da sadrži ikonu (pr. `<i>`, ``). Svim ikonama se može menjati boja, veličina, zadavati senka I dr. Da bi koristili Font Awesome, potrebno je da u `<head>` dodate sledeće:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
```

Primer upotrebe ikona:

```
<i class="fa fa-cloud"></i>
<i class="fa fa-heart"></i>
<i class="fa fa-car"></i>
<i class="fa fa-file"></i>
<i class="fa fa-bars"></i>
```



CSS navigacija (CSS Navigation Bar)

1. Da bismo napravili navigaciju I podesili njen izgled, potrebno je najpre da u HTML-u napravimo listu koja sadrži linkove.

Primer:

```
<ul>
  <li><a href="home.html">Početna</a></li>
  <li><a href="news.html">Novosti</a></li>
  <li><a href="contact.html">Kontakt</a></li>
  <li><a href="about.html">O nama</a></li>
</ul>
```

- [Početna](#)
- [Novosti](#)
- [Kontakt](#)
- [O nama](#)

2. Zatim treba da ukonimo tačkice (bullets), margine I padding. To podešavamo preko CSS-a:

```
<style>
ul {
  list-style-type: none;    /*za uklanjanje tačkica*/
  margin: 0;
  padding: 0;              /*za uklanjanje default-nih margina I padding-a koje ima svaki browser */
}
</style>
```

[Početna](#)
[Novosti](#)
[Kontakt](#)
[O nama](#)

Navigacija sada izgleda ovako!

Ova dva koraka su obavezna, a potom se odlučujemo da li ćemo izrađivati vertikalni ili horizontalni Navigation bar.

3. Vertikalna navigacija (Vertical Navigation Bar)

Kada je u pitanju vertikalna navigacija, u CSS dodajemo sledeće:

```
li a {
  display: block; /*čitavo polje koje sadrži linkove je klikabilno*/
  width: 60px;    /*block elementi zauzimaju čitavu širinu po default-u, ali kako mi želimo da
                  ograničimo prostor koji će zauzimati, zadajemo širinu od 60px*/
  background-color: #dddddd; /*pozadinska boja*/
}
```

4. Dodatno ulepšavanje vertikalne navigacije

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  width: 200px;
  background-color: #f1f1f1;
}

li a {
  display: block;
  color: #000;
  padding: 8px 16px;
  text-decoration: none;
}

/* Change the link color on hover */
li a:hover {
  background-color: #555;
  color: white;
}
</style>
</head>
<body>
<h2>Vertical Navigation Bar</h2>
<ul>
  <li><a href="home.html">Početna</a></li>
  <li><a href="news.html">Novosti</a></li>
  <li><a href="contact.html">Kontakt</a></li>
  <li><a href="about.html">O nama</a></li>
</ul>

</body>
</html>
```

Vertical Navigation Bar

Početna

Novosti

Kontakt

O nama

5. Ukoliko želimo da podesimo da se npr. **Prvi od menija, razlikuje od ostalih**, to radimo uz pomoć **klase active** na sl. način:

• U HTML dodamo klasu:

```
<ul>
  <li><a class="active" href="home.html">Početna</a></li>
  <li><a href="news.html">Novosti</a></li>
  <li><a href="contact.html">Kontakt</a></li>
  <li><a href="about.html">O nama</a></li>
</ul>
```

• A potom u CSS svojstva:

```
li a.active {  
    background-color: #4CAF50;  
    color: white;  
}  
  
li a:hover:not(.active) {  
    background-color: #555;  
    color: white;  
}
```

Početna

Novosti

Kontakt

O nama

6. Ukoliko želimo vertikalnu **navigaciju koja će zauzimati čitavu visinu stranice (full-height)**, potrebno da za ul podesimo sledeće:

```
ul {  
    list-style-type: none;  
    margin: 0;  
    padding: 0;  
    width: 25%;  
    background-color: #f1f1f1;  
    height: 100%; /* Full height */  
    position: fixed; /* Make it stick, even on scroll */  
    overflow: auto; /* Enable scrolling if the sidenav has too much content */  
}  
  
li a {  
    display: block;  
    color: #000;  
    padding: 8px 16px;  
    text-decoration: none;}
```

Horizontalna navigacija (Horizontal navigation bar)

Postoje dva načina za kreiranje horizontalne navigacije: **inline lista** ili **float lista**.

1. Inline List

 elementi su po default-u block elementi (zauzimaju čitavu širinu stranice). Da bi napravili horizontalni navigaciju, potrebno je da elementima dodelimo način prikazivanja inline - **display: inline;**

Primer:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
ul {
```

```
    list-style-type: none;
```

```
    margin: 0;
```

```
    padding: 0;
```

```
}
```

Horizontal Navigation Bar

[Početna](#) [Novosti](#) [Kontakt](#) [O nama](#)

```

li {
    display: inline;
}
</style>
</head>
<body>
<h2>Horizontal Navigation Bar</h2>
<ul>
  <li><a href="home.html">Početna</a></li>
  <li><a href="news.html">Novosti</a></li>
  <li><a href="contact.html">Kontakt</a></li>
  <li><a href="about.html">O nama</a></li>
</ul>

</body>
</html>

```

2. Floating List

Drugi način za pravljenje horizontalnog menija jeste podešavanje svojstva float za li elemente.

```

li {
    float: left;
}

a {
    display: block;
    padding: 8px;
    background-color: #dddddd;
}

```

Primer:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden; /*ovo svojstvo se dodaje u ul kako bi se sprečilo da li elementi idu
van liste*/
}

li {
    float: left;
}

li a {

```



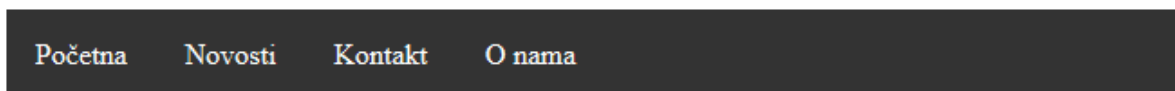

```

display: block;
padding: 8px;
background-color: #dddddd;
}
</style>
</head>
<body>
<h2>Horizontal Navigation Bar</h2>
<ul>
  <li><a class="active" href="home.html">Početna</a></li>
  <li><a href="news.html">Novosti</a></li>
  <li><a href="contact.html">Kontakt</a></li>
  <li><a href="about.html">O nama</a></li>
</ul>

</body>
</html>

```

Dodatno ulepšavanje horizontalne navigacije



Za izradu navigacije kao na slici, kod je sledeći:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
  list-style-type: none;
  margin: 0;
  padding: 0;
  overflow: hidden;
  background-color: #333;
}

li {
  float: left;
}

li a {
  display: block;
  color: white;
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}

```

```

li a:hover {
    background-color: #111;
}
</style>
</head>
<body>

<ul>
    <li><a class="active" href="home.html">Početna</a></li>
    <li><a href="news.html">Novosti</a></li>
    <li><a href="contact.html">Kontakt</a></li>
    <li><a href="about.html">O nama</a></li>
</ul>

</body>
</html>

```

Ukoliko želimo da jedan od menija bude pozicioniran sa desne strane I da u isto vreme bude npr. druge boje koristimo svojstvo float:right; I zadajemo mu class="active";



Primer:

```

<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}

li {
    float: left;
}

li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

```

```

li a:hover:not(.active) {
    background-color: #111;
}

.active {
    background-color: #4CAF50;
}
</style>
</head>
<body>

<ul>
    <li><a href="home.html">Početna</a></li>
    <li><a href="news.html">Novosti</a></li>
    <li><a href="contact.html">Kontakt</a></li>
    <li style="float:right"><a class="active" href="about.html">O nama</a></li>
</ul>

</body>
</html>

```

Izrada sajta

Sliku sajta, HTML I CSS kod možete pogledati na stranicama koje slede.



NOVA MOKKA X

SA INOVACIJAMA IZ PREMIUM KLASSE.

Vozila

Ponuda i usluge

Za vlasnike vozila

Novosti

Kontakt

Vozila

Familija	Model	Cena
Adam	Rocks	od 15 750 €
Corsa	3 vrata	od 10 500 €
Astra	GTC	od 21 100 €

Opel (nem. Adam Opel AG, AG od Aktiengesellschaft - Akcionarsko društvo) je nemački proizvođač automobila sa sedištem u Rislshajmu kojeg je osnovao Adam Opel 1862. godine. Kompanija je započela s proizvodnjom automobila 1899. Nastavio je da radi pod starim imenom „Adam Opel AG“ i predstavlja najveći Dženeral motorsov brend u Evropi, zajedno sa Voksolom (u Ujedinjenom Kraljevstvu).

U 2010. godini, kompanija je prodala preko 1,1 miliona putničkih automobila i lakih komercijalnih vozila u Evropi, čime je ostvarila tržišni udeo od 6,2 odsto. Opel je jedan od najvećih nemačkih proizvođača automobila i pored glavne fabrike u sedištu u Rislshajmu u Nemačkoj ima još tri fabrike u Bohumu, Kajzerslauternu i Ajzenahu, i dodatnih šest u drugim zemljama Evrope. Godine 2011. „Adam Opel GmbH“ ponovo postaje akcionarsko društvo „Adam Opel AG“ (GmbH od 2005. do kraja 2010).

Sponzor



Ponuda i usluge



Vaš pravi izbor po sajamskoj ceni uz opremu iz snova! Opel Meriva Drive, model sa 5 vrata, benzinskim 1.4 motorom i 100KS, dolazi sa opremom iz snova i specijalnom cenom od 12.900 EUR ili uz specijalne uslove Opel finansiranja uz ratu od 127€ mesečno!

Vazdušni jastuci za vozača i suvozača, bočni vazdušni jastuci + vazdušne zavese, ESP + ABS + TPMS + HSA, automatski dvozonski klima uređaj, aluminijumske felne 16", elektro podizači prednjih i zadnjih stakala, elektro podesivi spoljni retrovizori u boji karoserije, grejači spoljnih retrovizora, ručice vrata u boji karoserije, trokraki progresivni servo upravljač, podešavanje stuba upravljača po visini i dubini, multimedijalni sistem za info-zabavu IntelliLink (7" kolor ekran osjetljiv na dodir, bluetooth, USB, podrška za pametne telefone, 6 zvučnika, multimedijalni sadržaji...), upravljač presvučen kožom sa Bluetooth i audio komandama, automatski tempomat sa limiterom brzine, putni računar, senzor za kišu i još puno dodatne opreme...

Novosti

Geneva Motor Show 2017 | Opel highlight show-reel | Best of ...



HTML kod:

```
<!DOCTYPE html>
<html lang="sr">
  <head>
    <title>Opel Srbija</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="stylesheets/style.css" type="text/css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
  </head>

  <body>
    <div id="wrapper" >
      <header>
        <div class="fl_left">
          <h1><a href="index.html">Opel Srbija</a></h1>
        </div>
        <div class="fl_right">
          <ul>
            <li><i class="fa fa-phone"></i> +00 (123) 456
7890</li>
            <li><i class="fa fa-envelope-o"></i>
info@domain.com</li>
          </ul>
        </div>
        
        <nav>
          <ul>
            <li><a class="active"
href="#cars">Vozila</a></li>
            <li><a href="#offer">Ponuda i
usluge</a></li>
            <li><a href="#owner">Za vlasnike
vozila</a></li>
            <li><a href="#news">Novosti</a></li>
            <li class="right"><a
href="contact.html">Kontakt</a></li>
          </ul>
        </nav>
      </header>
      <div id="content">
        <article>
          <h2 id="cars">Vozila</h2>
          <table style="width:100%">
            <tr>
              <th>Familija</th>
              <th>Model</th>
              <th>Cena</th>
            </tr>
```

<tr>
<td>Adam</td>
<td>Rocks</td>
<td>od 15 750 €</td>
</tr>
<tr>
<td>Corsa</td>
<td>3 vrata</td>
<td>od 10 500 €</td>
</tr>
<tr>
<td>Astra</td>
<td>GTC</td>
<td>od 21 100 €</td>
</tr>
</table>

<p>Opel (nem. Adam Opel AG, AG od Aktiengesellschaft - Akcionarsko društvo) je nemački proizvođač automobila sa sedištem u Riselshajmu kojeg je osnovao Adam Opel 1862. godine. Kompanija je započela s proizvodnjom automobila 1899. Nastavio je da radi pod starim imenom „Adam Opel AG“ i predstavlja najveći Dženeral motorsov brend u Evropi, zajedno sa Voksolom (u Ujedinjenom Kraljevstvu).

U 2010. godini, kompanija je prodala preko 1,1 miliona putničkih automobila i lakih komercijalnih vozila u Evropi, čime je ostvarila tržišni udeo od 6,2 odsto. Opel je jedan od najvećih nemačkih proizvođača automobila i pored glavne fabrike u sedištu u Riselshajmu u Nemačkoj ima još tri fabrike u Bohumu, Kajzerslauternu i Ajzenahu, i dodatnih šest u drugim zemljama Evrope. Godine 2011. „Adam Opel GmbH“ ponovo postaje akcionarsko društvo „Adam Opel AG“ (GmbH od 2005. do kraja 2010). </p>

</article>

<article>

<h2 id="offer">Ponuda i usluge</h2>

<p> Vaš pravi izbor po sajamskoj ceni uz opremu iz

snoval

Opel Meriva Drive, model sa 5 vrata, benzinskim 1.4 motorom i 100KS, dolazi sa opremom iz snova i specijalnom cenom od 12.900 EUR ili uz specijalne uslove Opel finansiranja uz ratu od 127€ mesečno! Vazdušni jastuci za vozača i suvozača, bočni vazdušni jastuci + vazdušne zavese, ESP + ABS + TPMS + HSA, automatski dvozonski klima uređaj, aluminijumske felne 16", elektro podizači prednjih i zadnjih stakala, elektro podesivi spoljni retrovizori u boji karoserije, grejači spoljnih retrovizora, ručice vrata u boji karoserije, trokraki progresivni servo upravljač , podešavanje stuba upravljača po visini i dubini, multimedijalni sistem za info-zabavu IntelliLink (7" kolor ekran osetljiv na dodir, bluetooth, USB, podrška za pametne telefone, 6 zvučnika, multimedijalni sadržaji...), upravljač presvučen kožom sa Bluetooth i audio komandama, automatski tempomat sa limiterom brzine, putni računar, senzor za kišu i još puno dodatne opreme... </p>

</article>

```

<article>
    <h2 id="owner">Za vlasnike vozila</h2>
    
        <p>Vaše zadovoljstvo je prioritet broj jedan u Opel
servisu. Naš široki asortiman usluga je usmeren na to da obezbedi da dobijete što više
užitka od svog vozila. Mi to činimo tako što koristimo originalne Opel delove,
primenjujemo najviše standarde kvaliteta, zapošljavamo kvalifikovane tehničare i
naplaćujemo fer cene. Dodajte do ovlaštenog Opelovog servisera i uverite se sami, mi vas
očekujemo.

        Opel provere i pregledi<br>
        Svi automobili trebaju redovne preglede i provere.
Od sezonskih provera do generalnih pregleda, Opelovi servis partneri nude vam priuštiv
kvalitet i pouzdanost.</p>
    </article>
    <article>
        <h2 id="news">Novosti</h2>
        <iframe
src="https://www.youtube.com/embed/w4wwuTjR2ZI"></iframe>
    </article>
</div>
<aside>
    <h3>Sponzor</h3>
    
</aside>
<footer>
    <h3>Copyright &copy; Opel Serbia 2017</h3>
</footer>
</div>
</body>

```

CSS

```
#wrapper {
    width:940px;
    margin: 20px auto;
    background-color: #000;
    border-radius:40px;
}

body {
    background-image:url(../images/bkg.jpg);
    font-family: "Helvetica", "Arial";
    line-height: 25px;
    color: #fff;
}

header {
    padding: 25px;
}

.fl_left {
    float: left;
}

.fl_left a{
    text-decoration: none;
    color: #ffa31a;
}

.fl_right {
    float: right;
}

.fl_right ul {
    list-style-type: none;
}

img {
    width: auto;
    margin: 0 auto;
    display: block;
}

iframe {
    width: 608px;
    height: 342px;
    margin: 0 auto;
    display: block;
    padding-bottom: 10px;
    border: none;
}
```



```

nav ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #333;
}

nav li {
    float: left;
}

nav li a {
    display: block;
    color: white;
    text-align: center;
    padding: 14px 16px;
    text-decoration: none;
}

nav li a:hover {
    background-color: #111;
}

h2 {
    font-size: 25px;
    padding: 10px 0;
    text-align: center;
    font-weight: 700;
}

h3 {
    font-size: 15px;
    padding: 10px 0;
    text-align: center;
    color: #fff;
    font-weight: bold;
}

p {
    text-align: justify;
    margin: 20px;
    font-size: 18px;
}

#content {
    width: 700px;
    float: left;
    margin-left: 20px;
}

```

```
article{
    border: 2px solid #e68a00;
    margin: 20px;
}

table, th, td {
    border: 1px solid #e68a00;
    border-collapse: collapse;
}
th, td {
    padding: 5px;
}

th {
    text-align: center;
    background-color: #e68a00;
}

aside {
    float:right;
    width:220px;
}

footer {
    clear:both;
    background-color: #333;
    height: 50px;
    border-radius: 10px;
}
```

Dan 5

Poslednji dan predviđen je za testiranje I izradu sajta na osnovu slike. Svi polaznici koji pokažu zadovoljavajući stepen znanja biće preporučeni IT firmi sa kojom naša škola ima skopljjen ugovor, a koja vam pruža mogućnost zaposlenja.

Dodatni materijal za učenje HTML5 I CSS3

Teorija i praksa:

Codecademy HTML / CSS	http://www.codecademy.com/en/tracks/web
Codecademy Make a Website	http://www.codecademy.com/en/skills/make-a-website
Code School HTML / CSS	https://www.codeschool.com/paths/html-css
W3 Schools CSS	http://www.w3schools.com/css/
W3 Schools HTML	http://www.w3schools.com/html/default.asp

Tutorijale:

Channel9 - HTML5 and CSS3 Fundamentals	https://channel9.msdn.com/Series/HTML5-CSS3-Fundamentals-Development-for-Absolute-Beginners
YouTube- HTML5 and CSS3 beginners tutorial	https://www.youtube.com/watch?v=LqvFluVlyP8&list=PL41IfR-6DnOrugMacTfff1zrEcqtm7Fv