

Tehnička škola Vranje
Područje rada: elektrotehnika

WEB DIZAJN

Upotreba HTML i CSS tehnologija pri izradi Web stranica

Autori:
Snežana Bencun
Dejan Trajković

Vranje, septembar 2013.

Sadržaj

Pojam Weba.....	3
Podela Web sadržaja na statički i dinamički, klijentske i serverske tehnologije.....	5
Web dizajn	7
Priprema za vežbe	8
Uvod u HTML i CSS	9
Elementi i tagovi.....	9
Osnovna struktura HTML dokumenta.....	10
Osnovna upotreba i sintaksa CSS-a.....	12
Tekst i formatiranje teksta	21
[HTML deo vežbe]	21
[CSS deo vežbe]	23
Upotreba linkova, slika i multimedijalnog sadržaja	26
[HTML deo vežbe]	26
[CSS deo vežbe]	29
Upotreba listi	32
[HTML deo vežbe]	32
[CSS deo vežbe]	33
Upotreba tabela	36
[HTML deo vežbe]	36
[CSS deo vežbe]	39
Dopunska vežba, fajlovi " <i>Vezba06_Dopunska.html</i> " i " <i>stil06_dopunski.css</i> "	42
HTML forme	47
[HTML deo vežbe]	47
[CSS deo vežbe]	54
Frejmovi u HTML-u.....	58

[HTML deo vežbe]	58
[CSS deo vežbe]	59
Upotreba WYSIWYG alata za izradu Web sajtova.....	61
Sistemi za upravljanje sadržajem - CMS.....	62
Instalacija WordPressa	62
Korišćena literatura.....	71

Pojam Weba

World Wide Web ili skraćeno samo Web predstavlja jedan deo globalne mreže računara poznatije kao Internet. Njega čini veliki broj računara i uređaja koji su međusobno povezani i koriste standardizovane načine za komunikaciju i predstavljanje komunikacija. Internet je medij koji raste iz dana u dan, ima sve veći broj korisnika, postaje sve obimniji i sveobuhvatniji, nudi ogroman broj usluga svojim korisnicima i nad internetom niko nema potpunu kontrolu niti je u nečijem vlasništvu. Delovi interneta kao što su oprema koja je u vlasništvu Internet provajdera ili serveri preduzeća i pojedinaca kao i linije za prenos podataka predstavljaju privatno vlasništvo lica i firmi koja ih koriste ali je Internet kao medij potpuno otvoren i dostupan svima. Povezivanje na Internet se vrši tako što Internet provajderi pružaju usluge povezivanja svojim klijentima u koje spadaju kako i kućni korisnici tako i velike firme. Svaki uređaj koji je direktno povezan sa Internetom ima svoju jedinstvenu IP adresu. Trenutno su u upotrebi IPv4 adrese (primer: 93.253.145.44 ili lokalne 192.168.1.1) ali je Internet toliko narastao da je taj opseg adresa postao nedovoljan i uskoro će biti zamenjen IPv6 opsegom adresa koji omogućava daleko veći broj jedinstvenih uređaja na Internetu. Na Internetu je, kako je on rastao, rastao i broj servisa koji su bili na raspolaganju korisnicima: e-pošta, WWW (World Wide Web), diskusione grupe, FTP, radio i TV prenos, onlajn kupovina i prodaja, oglašavanje, e-poslovanje itd.

WWW je servis koji omogućava pretraživanje interneta korišćenjem grafičkog korisničkog interfejsa. WWW ili W3 je najveći informacioni servis na internetu koji omogućava stvaranje, manipulaciju, organizaciju i pretraživanje multimedijalnih dokumenata. WWW je internet servis koji se najbrže razvija. Pomoću njega mogu da se pregledaju tekst, slike, video sekvence, audio zapisi ili pronađu informacije iz različitih područja. Služi za publikaciju, reklamu, prodaju i distribuciju proizvoda, učenju na daljinu itd.

Pregledanje Weba se zasniva na učitavanju i pregledanju Web stranica kao i kretanju sa jedne na drugu Web stranu. Web stranama i lokacijama se pristupa ukucavanjem adresa (URL - Uniform Resource Locator) u adresno polje nekog od Internet pregledača. U pregledače spadaju Internet Explorer, Mozilla Firefox, Google Chrome, Opera i slični. Transfer Web strana i Web sadržaja se vrši pomoću HTTP odnosno HyperText Transfer Protocol protokola. On predstavlja prefiks adresa svih Web strana (na primer: <http://www.google.com/>). Pored pomenutog prefiksa adresa se još sastoji i od prefiksa www (oznaka za World Wide Web) koji uglavnom nije obavezan, i od imena domena na kome se strana nalazi. Domen ne predstavlja ništa drugo do simboličko ime servera na kome se Web strana nalazi. Korišćenje ovih simboličkih adresa umesto ručnog ukucavanja IP adresa računara na kojima se nalaze Web strane omogućavaju DNS serveri o kojima ćemo kasnije pričati. Nakon navigacije na neku Web lokaciju uglavnom se učitava početna strana koja korisniku omogućava laku dalju navigaciju kroz Web strane na lokaciji na kojoj se nalazi. Kretanje kroz Web strane se vrši hipervezama (linkovima) koje se nalaze na njima. Web strane su napisane u HTML-u (HyperText Markup Language) koji predstavlja opisni jezik za kreiranje strukture i sadržaja Web stranica.

Kako Internet predstavlja ogroman izvor informacija postoje Web lokacije koje su namenjene pretraživanju Interneta. To su Web sajtovi kao što su Google, Yahoo i slični. Ti sajtovi imaju posebne servere na kojima se pokreću automatizovani programi (botovi) koji krstare Internetom i indeksiraju sadržaj u svojim bazama podataka kako bi omogućili svojim korisnicima brzu i efikasnu pretragu Interneta sa jedne lokacije. Bitno je razlikovati pretraživače od gore pomenutih pregledača. Pored ovih pretraživača većina Web lokacija sadrži integrisanu lokalnu pretragu koja omogućava da pretražite informacije na sajtu na kome se trenutno nalazite.

Web stranice pored klasičnih tekstualnih informacija od kojih je sve krenulo danas mogu da sadrže i slike, audio-vizuelne zapise, interaktivni sadržaj pa čak i kompletne softverske alate i programe.

Podela Web sadržaja na statički i dinamički, klijentske i serverske tehnologije

Na modernom Webu se za kreiranje strana (sadržaja i vizuelnog prikaza strane) koriste HTML i CSS (Cascading Style Sheets). HTML sadrži strukturu i sadržaj koji se prikazuje a CSS služi da rasporedi i ulepša elemente te strane.

Statičke strane se sastoje isključivo od HTML-a ili HTML+CSS kombinacije. Sadržaj ovakvih strana je potpuno statičan, sadržaj ostaje isti sve dok dizajner strane ne unese novi ili izmeni postojeći sadržaj i izgled strane, nema interaktivnog sadržaja i uglavnom pružaju gole informacije i omogućavaju osnovni nivo interakcije korisnika sa Web lokacijom i kreiranje nekih osnovnih servisa koje Web pruža i to na veoma niskom nivou. Statičke strane danas takoreći nisu u upotrebi i jako je teško pronaći potpuno statične Web sajtove. Uglavnom ih izrađuju amateri dizajneri koji su u ranoj fazi proučavanja Web dizajna. Lične prezentacije su često statične ali i one danas uglavnom sadrže dobru dozu dinamičkog sadržaja.

Dinamičke strane predstavljaju ono što danas čini Web sve popularnijim i moćnijim. Ovakve strane omogućavaju kreiranje ogromnih i obimnih Web sajtova sa visokim nivoom interaktivnosti u realnom vremenu kao što su društvene mreže (npr. Facebook), onlajn enciklopedije (Wikipedia), multimedijalni sajtovi kao što je YouTube, moderni sajtovi čiji se sadržaj konstantno osvežava i dopunjuje, webmail servisi kao što su Gmail, Yahoo mail i Live, onlajn igre koje se pokreću direktno u pregledaču, bezbedna trgovina, četovi, prenosi manifestacija u realnom vremenu itd. Dinamički Web pokreće i podržava ogroman broj klijentskih i serverskih tehnologija koje omogućavaju usklađen, stabilan i efikasan rad modernih Web sajtova i servisa.

Sa klijentske strane HTML i CSS se dopunjuju skripting jezicima koji omogućavaju dinamičku izmenu sadržaja i izgleda Web strane kao i pristup informacijama dostupnim na strani. Skripting jezik koji se koristi u ove svrhe jeste opšteprihvaćeni JavaScript koji programerima omogućava visok nivo fleksibilnosti i pisanje kompleksnih programa i aplikacija koje se izvršavaju direktno u pregledaču. Pored skriptinga mogućnosti sa klijentske strane takođe proširuju i tehnologije koje dopunjuju mogućnosti pregledača u vidu dodataka (plug-inova) za iste. Tako je moguće unutar bilo kog od modernih browsera pokrenuti Java plug-in koji omogućava pokretanje programa napisanih u Javi ili pokretanje Flash Playera koji omogućava interakciju sa sadržajem kreiranim u Flashu (igre, video -audio fajlovi, aplikacije i slično). Moderni pregledači su dostigli visok nivo razvijenosti pa tako danas većina njih sama po sebi omogućava reprodukciju audio i video fajlova, visok nivo performansi radi igranja igara i kreiranja kompleksnih aplikacija koje se pokreću direktno u pregledaču i sve u cilju oslobađanja Weba od potrebe za pomenutim plug-inovima tako da svi klijenti mogu da bez problema koriste dostupne usluge na bilo kom uređaju i bilo kom pregledaču. Web po tom pitanju teži fleksibilnosti i univerzalnosti tako da svi servisi budu dostupni svim klijentima bilo da je u pitanju jak desktop računar ili smartfon.

Što se serverske strane tiče priča je malo drugačija. Serveri su računari koji su posvećeni hostovanju Web sajtova, čuvanju baza podataka, fajlova, itd. Pored uloge jednostavnog smeštanja podataka oni takođe mogu upravljati njima i predstavljaju posrednika između zahteva klijenata i sirovih informacija koje se nalaze na njima. Oni opslužuju jako veliki broj korisnika i omogućavaju brzo i efikasno deljenje resursa na mreži. Mogu imati različite uloge pa se tako neki serveri koriste isključivo za hosting statičnog sadržaja, neki samo za baze podataka, neki za e-mail, dok neki mogu pružati i više usluga istovremeno. Na ovim serverima se pokreću programi koji vrše svu interakciju između klijenata i podataka na serveru i obrađuju

podatke i zahteve za podacima. Najpoznatije serverske tehnologije koje su danas u upotrebi jesu PHP, MySQL, ASP, ASP.NET, JSP itd. Ove tehnologije zapravo predstavljaju softverske alate koji se nalaze na serveru i prihvataju i obrađuju zahteve svojih klijenata i šalju obrađene informacije na uvid. Tako se PHP koristi za praktično sve moguće svrhe od prostog dinamičkog kreiranja HTML strane koja se dostavlja pregledaču do kompleksnih interakcija sa bazama podataka. MySQL predstavlja najpoznatiji sistem za menadžment baza podataka koji često radi u sprezi sa PHP-om ili sličnim skripting jezikom na serveru. Moć dinamičkog Weba leži u mogućnostima upravo ovih serverskih tehnologija koje omogućavaju brzu i efikasnu manipulaciju velikim brojem podataka i efikasno opsluživanje zahteva klijenata. One omogućavaju izradu i funkcionisanje apsolutno svih dinamičkih sajtova danas, od jednostavnih foruma i četova do velikih Web aplikacija i servisa za vesti.

Kako bismo mogli da objasnimo razliku između statičkog i dinamičkog Weba na primeru?

Recimo da postoji sajt sa vestima. Kod statičkog Weba postojale bi samo HTML i CSS datoteke koje bi server svaki put dostavljao klijentu i čiji bi sadržaj bio isti sve dok Web dizajner ručno ne izmeni sadržaj HTML datoteka kako bi klijentima dostavio najnovije vesti. Pored neophodne stručnosti da tako nešto uradi ovaj proces je i dugotrajan i zahtevan, podložan greškama, problemima itd.

Dinamički Web omogućava da ovakav sajt sadrži uprošćen grafički interfejs koji omogućava novinarima da se jednostavno prijave na sajt sa svojim nalogom i kroz forme na HTML stranicama i jednostavne grafičke obrasce unose vesti i sadržaj na svoj sajt bez ikakvog znanja o svim ovim tehnologijama. Server prihvata podatke, podatke obrađuje PHP i smešta ih u bazu podataka pomoću MySQL-a. Kada klijent zahteva učitavanje strane sa najnovijim vestima isti taj PHP vadi vesti iz baze podataka, kreira HTML stranu, dodaje sadržaj, prilagođava je i takvu dostavlja klijentu. Na ovaj način je sajt unapred isprogramiran da bude samoodrživ, maksimalno automatizovan i dostupan običnim korisnicima. Naravno, sve je to potrebno unapred programirati u PHP-u, HTML-u, CSS-u i JavaScriptu i upravo zato danas Web sajtovi predstavljaju kompleksne celine na čijoj izradi rade Web dizajneri i Web developeri. Dizajneri se uglavnom bave samo HTML-om i CSS-om dok Web developeri pišu klijentske i serverske skripte u JavaScriptu i PHP-u koji u zavisnosti od kompleksnosti mogu varirati od malih i brzih skripti koje jednostavno čitaju i pišu podatke do velikih programa koji vrše obimne i dugačke proračune.

Web dizajn

Web dizajn obuhvata sve radnje od trenutka kada dobijemo ideju da napravimo sajt pa do trenutka kad je sajt potpuno završen. Web dizajn obuhvata sledeće: osmišljavanje sadržaja sajta, odabir naziva sajta, registracija domena i zakupljivanje hosta, grafički web dizajn, programiranje sajta, optimizacija sajta za pretraživače, puštanje Web sajta u rad.

Sadržaj sajta je najbitniji element web sajta koji mora biti jedinstven, zanimljiv i često ažuriran aktuelnim i svežim informacijama. U zavisnosti od sadržaja sajta, mi definišemo kojoj grupi sajtova će on pripadati, a na osnovu grupe određujemo način dizajniranja sajta. Ime sajta je veoma važna stavka u web dizajnu (ime koje se lako pamti i često koristi u svakodnevnom govoru i ima veze sa temom sajta). Svaki sajt mora imati jedinstven domen sa ekstenzijom (com, net, org, sr itd.). U vezi imena sajta otvara se pitanje autorskih prava i zaštita licenci. Postoje pravila u pogledu registracije domena kada se radi o imenu nekog pravnog lica ili licenciranog brenda. Na kraju, kada se zakupi domen, zakupljuje se i hosting. Prilikom grafičkog dizajna, potrebno je izgled stranice pogledati u različitim okruženjima, i to u skoro svim internet pregledačima i u svim njihovim aktuelnim verzijama, u svim rezolucijama ekrana, i isprobati na svim brzinama interneta. Kada se završi sajt u smislu programiranja, grafičkog dizajna, zakupa domena i hosta, preostaje još da se fajlovi sajta prebace na host server kako bi sajt bio vidljiv na internetu. To prebacivanje fajlova sa kućnog računara na host obavlja se FTP konekcijom. FTP je protokol za prenos podataka između dva računara na mreži. Nakon aploadovanja fajlova na hosting server često je neophodno podesiti neke dodatne stavke kao što su pristup bazi podataka, kreiranja baza podataka, podešavanje serverskog softvera a nekada i instalacija i ažuriranje softvera na serveru ukoliko nam je tako nešto dostupno. Takođe se često vrši i podešavanje mail servera ukoliko ga naš sajt koristi.

U toku izrade, održavanja i korišćenja Web sajta potrebno je poštovati autorska prava. To znači da na našem sajtu ne smemo upotrebljavati niti kačiti bilo kakav autorskim pravima zaštićeni sadržaj bez prethodne dozvole autora. To podrazumeva slike, audio i video fajlove, softver, delove HTML i CSS dokumenata, skripte na sajtu i tako dalje. Sav sadržaj na Internetu podleže autorskim pravima i to što ga na Internetu možete pronaći ne znači da je preuzimanje i upotreba takvog sadržaja legalna. Trudite se da pri izradi sajta koristite besplatne fontove, grafiku i slično i obratite pažnju na to kakva je upotreba dozvoljena licencom koja ide uz takav sadržaj. Ukoliko vam je upotreba nekog kopirajtovanog sadržaja neophodna onda možete kontaktirati autore i zatražiti dozvolu za upotrebu tog sadržaja.

Priprema za vežbe

Pre početka vežbe neophodno je imati instaliran program Notepad++ i neki od internet pregledača. Kreirati jedan folder na desktopu pod imenom "*Web dizajn*" u kome će se nalaziti svi fajlovi i sav sadržaj i materijal koji ćete koristiti u toku ove vežbe.

Sada pokrenite Notepad++, idite u Settings meni, Preferences, New document/Default directory kartica i sa leve strane za Encoding odaberite UTF-8 da bi vam bila dostupna latinična i ćirilčna slova (ali UTF-8, nikako UTF-8 without BOM).

Uvod u HTML i CSS

HTML i CSS predstavljaju tehnologije koje se koriste pri dizajniranju web stranica.

HTML je skraćenica za *Hypertext Markup Language* i koristi se za struktuiranje web strane, odnosno predstavlja strukturni opis elemenata koji sačinjavaju stranu kao i sam sadržaj tih elemenata.

CSS (*Cascading Style Sheets*) se koristi za vizuelno stilizovanje HTML strana. Na primer, koristi se za poizicioniranje elemenata, bojenje, formatiranje teksta, pozadine, ivica, izgleda tabela, slika itd. CSS kod se može nalaziti unutar samog HTML dokumenta ili u posebnom fajlu. Najčešće se CSS piše u odvojenom fajlu pa se na način koji ćemo kasnije objasniti povezuje sa HTML dokumentom da bi stilovi koji se nalaze u njemu bili primenjivi.

Elementi i tagovi

HTML dokument sadrži obavezne elemente koji čine kostur strane. Elementi strane su opisani takozvanim tagovima koji se nalaze unutar znakova za veće i manje. Postoje tagovi koje je potrebno otvoriti i zatvoriti a neki su samozatvarajući. Primer taga koji se zatvara i način na koji se to radi:

```
1 <p>Neki tekst</p>
```

U datom primeru dat je tag koji označava element pasusa (eng. *paragraph*) a koji ćemo kasnije obraditi.

Primer samozatvarajućeg taga je **
** (ili **
**) koji služi kao *line break* odnosno koristi se za prelazak u novi red. Samozatvarajuće tagove je u HTML-u nepotrebno zatvarati pa su tako dozvoljeni i **
** i **
** tag.

Elementi unutar svojih tagova takođe mogu sadržati i neke atribute od kojih su neki obavezni a neki opcioni. Atributi bliže opisuju elemente i mogu i da utiču na izgled i ponašanje elemenata. Primer atributa koji služi da dodeli identifikator elementu pasusa je na slici (o identifikatorima ćemo kasnije):

```
1 <p id="prviPasus">Ovo je tekst pasusa</p>
```

Svi atributi imaju naziv i sadržaj koji se nalazi u navodnicima. U konkretnom slučaju atribut ima naziv **id** a njegov sadržaj je **prviPasus**. Atributi se nalaze samo u početnim oznakama elemenata.

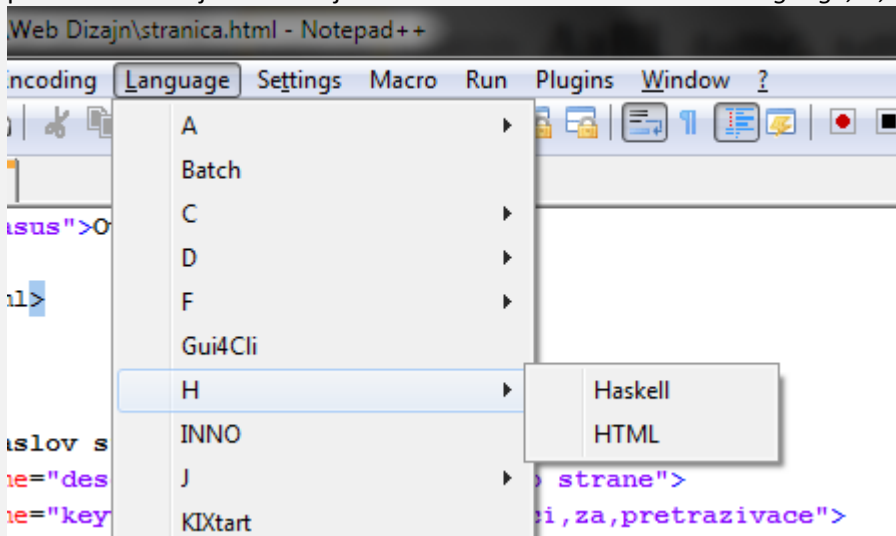
Način upotrebe tagova će postati mnogo jasniji kroz primere i vežbe koji slede.

Osnovna struktura HTML dokumenta

Osnovna struktura jednog HTML dokumenta je data na slici ispod:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5      <title>Naslov strane</title>
6      <meta name="description" content="Opis web strane">
7      <meta name="keywords" content="kljucne, reci, za, pretrazivace">
8      <meta charset="utf-8">
9  </head>
10
11 <body>
12
13 </body>
14
15 </html>
```

Otvorite Notepad i prekucajte kod sa slike a zatim sačuvajte taj fajl u folderu "Web dizajn" pod imenom "Vezba01.html". Notepad++ bi trebalo da detektuje da je u pitanju fajl pisan u jeziku HTML i shodno tome vizuelno prilagodi i oboji tekst kao na slici iznad. Ukoliko se to ne desi, potrebno da je odaberete HTML jezik kao podrazumevani jezik za ovaj dokument. To se radi kroz meni *Language, H, HTML*:



Ovo će omogućiti lakše čitanje dokumenta, detektovanje grešaka i slično jer će Notepad++ sam obeležavati elemente, atribute i njihove vrednosti odgovarajućim bojama. Mogućnosti kao što je ova ga izdvajaju od prostih tekst editora kao što je Windowsov Notepad.

Nakon što ste prekućali i sačuvali dokument otvorite ga u nekom Web pregledaču kao što je Internet Explorer, Mozilla Firefox ili Google Chrome. Web strana će biti potpuno prazna jer je **<body>** element prazan ali će naslov biti vidljiv u naslovnoj liniji pregledača ili taba u kome je strana otvorena jer je prisutan element **<title>**.

U prvoj liniji dokumenta nalazi se **definicija tipa dokumenta** (Document Type Declaration – DTD). DTD saopštava pretraživaču da očekuje HTML dokument.

Nakon DTD-a sledi **html element**. Svi ostali elementi HTML dokumenta nalaze se između početne **<html>** i krajnje **</html>** oznake HTML elementa.

U sekciji (ili odeljku) **zaglavlje strane**, čija je početna oznaka **<head>** a krajnja **</head>**, nalaze se informacije o Web strani kao što su naslov strane koji se prikazuje u naslovnoj liniji Web pretraživača ili na kartici Web pretraživača. Naslov se definiše između početne oznake **<title>** i krajnje oznake **</title>**. Ostali podaci koji se nalaze u odeljku zaglavlje strane, ne vide se u Web pretraživaču.

U zaglavlju strane nalaze se i metapodaci (podaci o podacima). Za definisanje metapodataka koristi se meta oznaka **<meta>**. Meta oznaka **description** daje detaljniji opis Web strane. Kada mašine za indeksiranje (Google, Yahoo) vrše indeksiranje, one najpre potražuju reči za indeksiranje koje se nalaze u meta oznaci **keywords** (ključne reči). Zato, za vrednost atributa **content** treba uzeti ključne reči koje su relevantne za ono što se nalazi na Web strani. Ključne reči je poželjno razdvojiti zarezom. Indeksiranje je proces ubacivanja strana u bazu (indeks) mašine za indeksiranje. Treća **<meta>** oznaka upotrebljena je da bi se preko njenog atributa **charset**, čija je vrednost **utf-8**, definisala upotreba **Unicode** standarda za snimanje i prikazivanje dokumenta u Web pretraživačima. Unicode standard omogućava prikazivanje znakova svih svetskih pisama, uključujući i znakove našeg ćiriličnog i latiničnog pisma.

U odeljku **telo**, čija je početna oznaka **<body>** a krajnja **</body>**, nalazi se sadržaj Web strane koji se prikazuje u Web pretraživaču.

Pored ovih obaveznih elemenata unutar **body** elementa se za struktuiranje koriste i neki od sledećih elemenata:

<code><section>...</section></code>	izrađuje odeljak u HTML dokumentu
<code><article>...</article></code>	deli sekciju ili deo strane na odeljke ili stavke, na primer može da odvaja poruke na forumu ili blogu
<code><nav>...</nav></code>	omogućava kreiranje navigacije na sajtu
<code><header>...</header></code>	omogućava kreiranje zaglavlja na strani ili unutar elemenata kao što su section ili article
<code><footer>...</footer></code>	omogućava kreiranje podnožja na strani ili unutar elemenata kao što su section ili article
<code><div>...</div></code>	predstavlja generički strukturni element koji može da podeli stranu ili bilo koji drugi element na delove ili sekcije
<code><h1>...</h1></code> <code><h2>...</h2></code> ... <code><h6>...</h6></code>	predstavljaju elemente koji se koriste za naslove i podnaslove; najveći je h1 a najmanji h6 , hijerarhijski, i ovu hijerarhiju treba poštovati i prilikom stilizovanja strane
<code><p>...</p></code>	predstavlja običan tekstualni pasus
<code></code>	element koji se koristi za umetanje slike na putanji određenoj atributom src ; kao što se i vidi ovaj tag je samozatvarajući
<code>Tekst</code>	anchor ili sidro (hiperveza, link), koristi se za povezivanje Web strane sa nekom drugom stranom, fajlom ili dokumentom ili sadržajem na istoj ili drugoj strani a koji se nalazi na lokaciji definisanoj atributom href

<pre>... <i>...</i> ... <sub>...</sub> ...</pre>	mnogobrojni elementi koji se koriste prilikom formatiranja teksta, biće detaljno objašnjeni kasnije
--	---

Jako je bitno razumeti dva generička atributa koja se jako puno koriste pri stilizovanju Web strana ali i prilikom pisanja skripti u jezicima kao što su JavaScript i slični. U našem slučaju je potrebno razumeti ih radi stilizovanja jer se pisanjem skripti nećemo baviti. Ovi atributi su **id** i **class**. Njih može posedovati bilo koji element.

Atribut **id** predstavlja jedinstveni identifikator nekog elementa na strani. Služi za jednostavnu selekciju tog elementa radi stilizovanja prilikom pisanja CSS-a. Samim tim što je jedinstven podrazumeva se da nijedan drugi element ne sme imati atribut **id** koji ima istu vrednost.

Sa atributom **class** je situacija suprotna. Takođe se koristi za selekciju elemenata radi stilizovanja ali ne predstavlja jedinstveni identifikator već klasu odnosno grupu elemenata na koje je primenjen određeni stil. To nam omogućava da primenimo istovetni CSS stil na više elemenata odjednom. Kasnije ćete na primerima videti kako se ova dva atributa koriste i kada je koji efikasnije primeniti.

Primer: imamo tri bicikla različitih boja: plavi, crveni i zeleni. Njihov **id** bi redom bio "plavi", "crveni" i "zeleni" a **klasa** bi bila "bicikl". Rečju "bicikl" koja predstavlja klasu se obraćamo svim biciklima dok se korišćenjem boja kao jedinstvenih identifikatora obraćamo samo jednom biciklu. Kada budemo koristili CSS biće vam jasnije kada se koriste klase a kada identifikatori.

Osnovna upotreba i sintaksa CSS-a

Za ovaj zadatak će nam biti potreban novi HTML dokument u koji ćemo unositi sadržaj a potom ga i stilizovati. Napravite kopiju fajla "Vezba01.html" i nazovite ga "Vezba02.html". Ovo smo uradili da ne bismo ponovo prekucavali kostur strane. Sada otvorite fajl "Vezba02.html" u Notepadu++ kako bismo krenuli sa dodavanjem sadržaja u dokument.

O sintaksi CSS-a je jako teško diskutovati bez primene iste i zato ćemo u ovoj vežbi pre svega izraditi jednostavnu HTML stranu sa dva novinska članka i potom stilizovati ta dva članka korišćenjem CSS-a.

Novinski članci će sadržati naslov, pasus teksta i jednu fotografiju. HTML kod nećemo detaljno objašnjavati kao ni CSS stilove, kasnije ćemo se baviti time, cilj je da samo uočite strukturu i sintaksu oba jezika kao i način njihove upotrebe.

Struktura **tela** bez vidljivog sadržaja će izgledati ovako:

```
11 <body>
12   <article class="clanak" id="prvi">
13     <h1></h1>
14     <p></p>
15     
16   </article>
17
18   <article class="clanak" id="drugi">
19     <h1></h1>
20     <p></p>
21     
22   </article>
23 </body>
```

Prekucajte strukturu elementa sa slike iznad u otvoreni dokument "Vezba02.html".

Body dokumenta nakon umetanja sadržaja izgleda ovako:

```
11 <body>
12   <article class="clanak" id="prvi">
13     <h1>Misli globalno, sadi lokalno</h1>
14     <p>Akciji ozelenjavanja zelenih površina koje organizuje samorganizovana grupa "gerila
15     baštovana" priključili su se i zaposleni u Infostudu koji su posadili 215 sadnica
16     cveća na 7 lokacija u Subotici i na taj način dodatno ulepšali i uredili svoj grad.</p>
17     
18   </article>
19
20   <article class="clanak" id="drugi">
21     <h1>Stiže i novi Nexus 10 tablet</h1>
22     <p>Google će opet ove godine imati pun mobilni portfolio jer nam poznati nalog
23     @evleaks poručuje da je novi Nexus 10 tablet izrađen od strane Asusa, blizu objave.
24     Ovim će Google uz Nexus 7 tablet i Nexus 5 smartfon, upotpuniti čitavu paletu novih
25     proizvoda za 2013.</p>
26     
27   </article>
28 </body>
```

Sadržaj za oba članka možete kopirati odavde (naslov za element **h1** i pasus za element **p**)

Misli globalno, sadi lokalno

Akciji ozelenjavanja zelenih površina koje organizuje samorganizovana grupa "gerila baštovana" priključili su se i zaposleni u Infostudu koji su posadili 215 sadnica cveća na 7 lokacija u Subotici i na taj način dodatno ulepšali i uredili svoj grad.

Stiže i novi Nexus 10 tablet

Google će opet ove godine imati pun mobilni portfolio jer nam poznati nalog @evleaks poručuje da je novi Nexus 10 tablet izrađen od strane Asusa, blizu objave. Ovim će Google uz Nexus 7 tablet i Nexus 5 smartfon, upotpuniti čitavu paletu novih proizvoda za 2013.

Slike koje koriste elementi **img** u oba članka nalaze se u folderu "Materija" na desktopu i potrebno ih je kopirati u folder "Web dizajn" u kome se nalazi i dokument "Vezba02.html" da bi pregledač mogao da ih pronađe i učita.

Nakon što ste umetnuli sav sadržaj sačuvajte dokument i otvorite ga u pregledaču, trebalo bi da izgleda ovako:

Webpage Screenshot

Misli globalno, sadi lokalno

Akciji ozelenjavanja zelenih površina koje organizuje samorganizovana grupa "gerila baštovana" priključili su se i zaposleni u Infostudu koji su posadili 215 sadnica cveća na 7 lokacija u Subotici i na taj način dodatno ulepšali uređili svoj grad.



Stiže i novi Nexus 10 tablet

Google će opet ove godine imati pun mobilni portfolio jer nam poznati nalog @evleaks poručuje da je novi Nexus 10 tablet izrađen od strane Asusa, blizu objave. Ovim će Google uz Nexus 7 tablet i Nexus 5 smartfon, upotpuniti čitavu paletu novih proizvoda za 2013.



file:///C:/Users/Bencun/Desktop/Web%20Dizajn/Vezba01.html

Dokument koji trenutno imamo je pravilno struktuiran, sadržaj je ubačen ali kao što vidite sav sadržaj se ređa jedan ispod drugog, tekst nije formatiran, slike su prevelike itd. Dakle, sada nam sledi stilizovanje, ovde ćemo videti kako se biraju elementi koje želimo da stilizujemo i kako se pravilno primenjuju stilovi nad njima. Nećemo zalaziti u detaljna objašnjenja jer je poenta ove vežbe samo objasniti sintaksu i način upotrebe CSS-a a ne njegove mogućnosti, o njima ćemo kasnije.

U Notepadu++ kreirajte novi fajl i sačuvajte pod nazivom "stil02.css". U ovom fajlu će se nalaziti stilovi kojima ćemo stilizovati elemente dokumenta "Vezba02.html".

CSS stilove je moguće pisati na tri načina:

- unutar početnog taga elementa kao vrednost atributa **style** (*inline stilovi-redni*)
- unutar elementa **head** dokumenta (*interni stilovi*)
- u posebnom fajlu koji se vezuje za neki HTML dokument (*eksterni stilovi*) - trenutno koristimo ovaj način pisanja stilova jer je najčešći i najlakši za upotrebu i održavanje

Redosled stilova u CSS dokumentu je proizvoljan.

Pošto koristimo eksterni stil potrebno je dodati jednu liniju koda u **head** dokumenta koja će povezati naš dokument sa dokumentom koji sadrži CSS stilove za tu stranu. Ta linija koda izgleda ovako:

```

4  <head>
5      <title>Naslov strane</title>
6      <meta name="description" content="Opis web strane">
7      <meta name="keywords" content="kljucne, reci, za, pretrazivace">
8      <meta charset="utf-8">
9      <link rel="stylesheet" href="stil02.css">
10 </head>

```

Element **link** služi da vezuje druge dokumente kao što su CSS stilovi ili neke skripte sa HTML dokumentom. Atribut **rel** elementa **link** služi da odredi tip veze (u konkretnom slučaju mi pregledaču govorimo da je u pitanju CSS dokument sa stilovima) a atribut **href** određuje putanju do dokumenta koji vezujemo sa HTML dokumentom (u konkretnom slučaju u pitanju je "stil02.css").

Dodati liniju koda sa slike u HTML dokument "Vezba02.html" da biste ga povezali sa prethodno kreiranim dokumentom "stil02.css" i sačuvati fajl.

Kada stilizujemo elemente strane prvo odabiramo element ili više elemenata na koje primenjujemo neki stil a zatim pišemo stil. Recimo da imamo element **h1** kojem bismo želeli da povećamo veličinu slova na 40 i promenimo boju teksta u zelenu. CSS stil za ovakvu radnju bi izgledao ovako:

```

1  h1
2  {
3      font-size: 40px;
4      color: green;
5  }

```

Kucanjem **h1** smo selektovali element na koji će se primeniti stil, nakon toga otvaramo vitičaste zagrade. Unutar njih se piše stil i to ređanjem različitih osobina koje želimo da promenimo. Sintaksa je ovakva:

```

3  |   osobina: vrednost;

```

Mi smo podesili dve osobine naslova **h1**: veličinu fonta na 40 piksela i boju slova na zelenu.

Pošto smo objasnili način na koji se pristupa elementima na strani radi stilizovanja, preći ćemo na stilizovanje HTML strane koju imamo. Sada ćemo upotrebiti klase i identifikatore koje smo uneli u HTML kod da bismo odabrali elemente koje želimo da stilizujemo.

Dakle, oba elementa **article** u dokumentu pripadaju klasi proizvoljno imenovanoj **clanak**. Kada budemo pristupili stilizovanju klase **clanak** to će značiti da se ovaj stil primenjuje i na prvi i na drugi članak jer oba pripadaju istoj klasi. Međutim, u našem slučaju pristupanjem klasi **clanak** mi ne pristupamo nijednom elementu koji možemo stilizovati već elementu **article** unutar koga su elementi **h1**, **p** i **img** (imajte na umu da se ovo opet odnosi na oba članka!) koji nas interesuju i koje želimo da stilizujemo. Njima ne pristupamo (odnosno ne selektujemo ih) direktno kao što smo

to činili u prethodnom primeru već na sledeći način:

```
1  .clanak h1
2  {
3  }
4
5  .clanak p
6  {
7  }
8
9  .clanak img
10 {
11 }
```

Sva tri stila su trenutno prazna.

Prekucajte sva tri stila u "stil02.css".

Dakle, klasu elemenata selektujemo kucanjem tačke a zatim naziva klase. Ukoliko pristupamo elementu koji se nalazi unutar elementa te klase, odnosno na hijerarhijski i strukturno nižem nivou u HTML dokumentu (u našem slučaju su sva tri elementa unutar **article** elemenata koji su klase **clanak**), onda ih jednostavno navodimo jedan za drugim.

Primer: ukoliko imamo element **article** unutar koga se nalazi **header** a unutar koga su elementi **p**, **h1** i **h2** za selekciju elementa **p** kucamo jednostavno:

article header p{...}

...dakle potreban je razmak između svakog elementa.

Ukoliko želimo da selektujemo **h1** i **h2** istovremeno radi primenjivanja istovetnog stila na njih, onda između njih stavljamo zarez:

article header h1, article header h2 {...}.

Sada ćemo podesiti širinu oba članka na 600 piksela. Svaki pasus će zauzimati 70% širine članka dok će svaka slika zauzimati preostalih 30% širine članka i nalaziće se desno od pasusa. Nakon izmena, kod izgleda ovako (dok formatiranje **h1** naslova još nećemo dirati):

```
1  .clanak
2  {
3      width: 600px;
4  }
5
6  .clanak h1
7  {
8  }
9
10 .clanak p
11 {
12     width: 70%;
13 }
14
15 .clanak img
16 {
17     width: 30%;
18 }
```

Prekucajte stilove sa slike u CSS dokument. Nakon toga sačuvajte i HTML i CSS dokument i otvorite HTML dokument u pregledaču da vidite efekte primenjenog stila.

Sada ćemo selektovati pojedinačno naslove **h1** u oba članka da bismo objasnili kako funkcionišu identifikatori. Prvi članak ima identifikator **prvi** a drugi članak ima identifikator **drugi**. Prvi naslov ćemo obojiti u zeleno a drugi u plavo. Selekcija identifikatora se vrši znakom "#". Takođe ćemo iskoristiti to što pripadaju istoj klasi koju smo već koristili i upotrebiti jedan isti stil nad njima da bismo promenili izgled fonta u kurziv. Evo kako izgleda kod nakon izmene:

```
6  .clanak h1
7  {
8      font-style: italic;
9  }
10
11  #prvi h1
12  {
13      color: green;
14  }
15
16  #drugi h1
17  {
18      color: blue;
19  }
```

Imajte na umu da su ostali stilovi koje smo prethodno napisali i dalje tu samo nisu prikazani na slici radi uštede prostora.

Prekucajte dodatne stilove. Sačuvajte CSS dokument i osvežite HTML dokument u browseru.

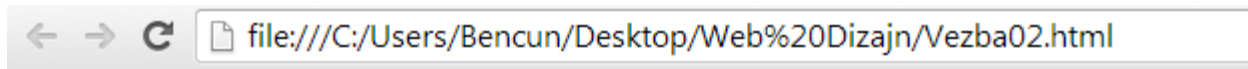
Nakon ovoga primenjujemo na oba **p** i **img** elementa dodatni stil koji omogućava da elementi sami "plutaju" ulevo i ređaju se jedan pored drugog. Slike će se same prilagoditi i smestiti u 30% prostora koji im je dodeljen od širine

članka (koja je inače, podsetimo se, 600 piksela). Kompletan CSS kod nakon svih izmena izgleda ovako:

```
1  .clanak
2  {
3      width: 600px;
4  }
5  .clanak h1
6  {
7      font-style: italic;
8  }
9  #prvi h1
10 {
11     color: green;
12 }
13 #drugi h1
14 {
15     color: blue;
16 }
17 .clanak p
18 {
19     float: left;
20     width: 70%;
21 }
22 .clanak img
23 {
24     width: 30%;
25 }
```

Prekucajte i proverite CSS kod, nakon toga sačuvajte dokument i osvežite HTML stranicu u browseru da biste videli sve promene do kojih je došlo prilikom stilizovanja dokumenta: članci su se podesili na širinu od 600 piksela od kojih je 70% širine dodeljeno pasusima a 30% širine dodeljeno slikama koje su se same prilagodile dostupnom prostoru. Promenjeni su naslovi, koristili smo klase i identifikatore kao i same nazive elemenata da bismo ih selektovali za stilizovanje. Cilj ove vežbe je bio da naučite kako da pravilno odaberete elemente na strani korišćenjem samih naziva elemenata ili korišćenjem naziva klase kojoj pripadaju ili pak korišćenjem jedinstvenog identifikatora elementa.

Stranica bi nakon stilizovanja trebala da izgleda ovako:



Misli globalno, sadi lokalno

Akciji ozelenjavanja zelenih površina koje organizuje samorganizovana grupa "gerila baštovana" priključili su se i zaposleni u Infostudu koji su posadili 215 sadnica cveća na 7 lokacija u Subotici i na taj način dodatno ulepšali i uredili svoj grad.



Stiže i novi Nexus 10 tablet

Google će opet ove godine imati pun mobilni portfolio jer nam poznati nalog @evleaks poručuje da je novi Nexus 10 tablet izrađen od strane Asusa, blizu objave. Ovim će Google uz Nexus 7 tablet i Nexus 5 smartfon, upotpuniti čitavu paletu novih proizvoda za 2013.



Ukoliko bismo, na primer, želeli da u drugom pasusu slika bude levo od teksta onda bi ovo bile neophodne promene:

```
17 .clanak p
18 {
19     width: 70%;
20 }
21
22 .clanak img
23 {
24     width: 30%;
25 }
26 #prvi p
27 {
28     float: left;
29 }
30 #drugi p
31 {
32     float: right;
33 }
```

...nakon čega će strana izgledati ovako:



Misli globalno, sadi lokalno

Akciji ozelenjavanja zelenih površina koje organizuje samorganizovana grupa "gerila baštovana" priključili su se i zaposleni u Infostudu koji su posadili 215 sadnica cveća na 7 lokacija u Subotici i na taj način dodatno ulepšali i uredili svoj grad.



Stiže i novi Nexus 10 tablet



Google će opet ove godine imati pun mobilni portfolio jer nam poznati nalog @evleaks poručuje da je novi Nexus 10 tablet izrađen od strane Asusa, blizu objave. Ovim će Google uz Nexus 7 tablet i Nexus 5 smartfon, upotpuniti čitavu paletu novih proizvoda za 2013.

Domaći zadatak za vežbu: napraviti stranu identične strukture sa drugim tekstom i slikama i podesiti CSS kod tako da obe slike budu levo od teksta a oba naslova žute boje (yellow).

Tekst i formatiranje teksta

Za početak napravite još jednu kopiju fajla "Vežba01.html" i nazovite je "Vežba03.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu. Takođe kreirajte jedan prazan CSS dokument pod nazivom "stil03.css" koji ćemo koristiti za stilizovanje dokumenta na kome budemo radili u ovoj vežbi.

U ovoj vežbi ćemo raditi sa tekстом i naučiti kako se formatira tekst u CSS-u.

[HTML deo vežbe]

Za početak, potrebno je poznavati osnovne elemente koji se koriste za rad sa tekстом u HTML-u. Osnovni elementi i njihov opis slede u tabeli:

Element	Opis
<p>...</p>	element koji definiše jedan pasus, podrazumeva se slobodan prostor pre i posle pasusa
<h1>...</h1> ... <h6>...</h6>	definiše naslove i podnaslove
...	definiše sadržaj koji će biti istaknut ili naglašen
<i>...</i>	služi za pisanje teksta kurzivom
...	služi za pisanje boldovanog teksta
...	definiše tekst koji značajno utiče na sadržaj (snažno naglašeni tekst)
<small>...</small>	definiše tekst koji će biti prikazan malim slovima u odnosu na okolni tekst
_{...}	definiše tekst koji će biti prikazan u indeksu
^{...}	definiše tekst koji će biti prikazan u eksponentu
...	definiše tekst koji će biti prikazan sa horizontalnom linijom preko njega (precrtan tekst)
<ins>...</ins>	definiše umetnuti sadržaj
<code>...</code>	definiše sadržaj kao računarski kod
 (ili)	prelom reda u tekstu (novi red)
 	običan <i>Space</i> karakter, funkcioniše kao dugme <i>Space</i> u tekst editorima, pravi razmak
...	element za opšte izdvajanje sadržaja iz linije teksta radi stilizovanja


```
atribut2: vrednost2;  
}
```

Kao što vidite, elementi kao što su `<i>` ili `` već unose neko podrazumevano formatiranje, dok elementi kao što je `` nemaju vidljivog uticaja na dokument sve dok ne upotrebimo CSS da stilizujemo taj deo teksta.

[CSS deo vežbe]

Pre nego što krenemo sa stilizovanjem dokumenta bilo bi lepo da objasnimo RGB sistem boja koji se u CSS-u esktenzivno koristi pored klasičnih naziva boja kao što su *blue*, *red*, *orange*, *yellow* itd. RGB sistemom pisanja boja se mogu napisati sve nijanse osnovnih boja koje ovaj sistem koristi: **Red**, **Green** i **Blue**. Boje se pišu u heksadecimalnom sistemu (svaka cifra je u rasponu od 0 do F) i sa tarabom ispred boje. Primer čisto zelene boje: `#00FF00` - prve dve cifre određuju intenzitet crvene, druge dve intenzitet zelene a treći par cifara određuje intenzitet plave boje.

Takođe, ne zaboravite da ubacite element **link** kojim ćemo povezati HTML dokument sa CSS dokumentom.

```
Linkujte HTML dokument sa CSS dokumentom na ranije objašnjen način odnosno dodavanjem ove linije u head  
HTML dokumenta:  
<link rel="stylesheet" href="stil03.css">
```

Sada ćemo stilizovati redom:

- telo HTML dokumenta **body**: pozadinu podešavamo na svetlo sivu korišćenjem RGB sistema
- naslov **h1**: podešavamo veličinu fonta na 32px i porodicu fonta na Verdana
- naslovi **h2**: podešavamo veličinu fonta na 20px i boju u plavu
- pasuse **p**: poravnanje teksta podešavamo na **justify** (poravnanje sa obe strane) i podešavamo veličinu fonta na 16px
- deo teksta izdvojen elementom **span** podvlačimo
- deo teksta izdvojen elementom **del** ćemo iskositi
- deo teksta izdvojen elementom **code**: podešavamo veličinu fonta na 12px

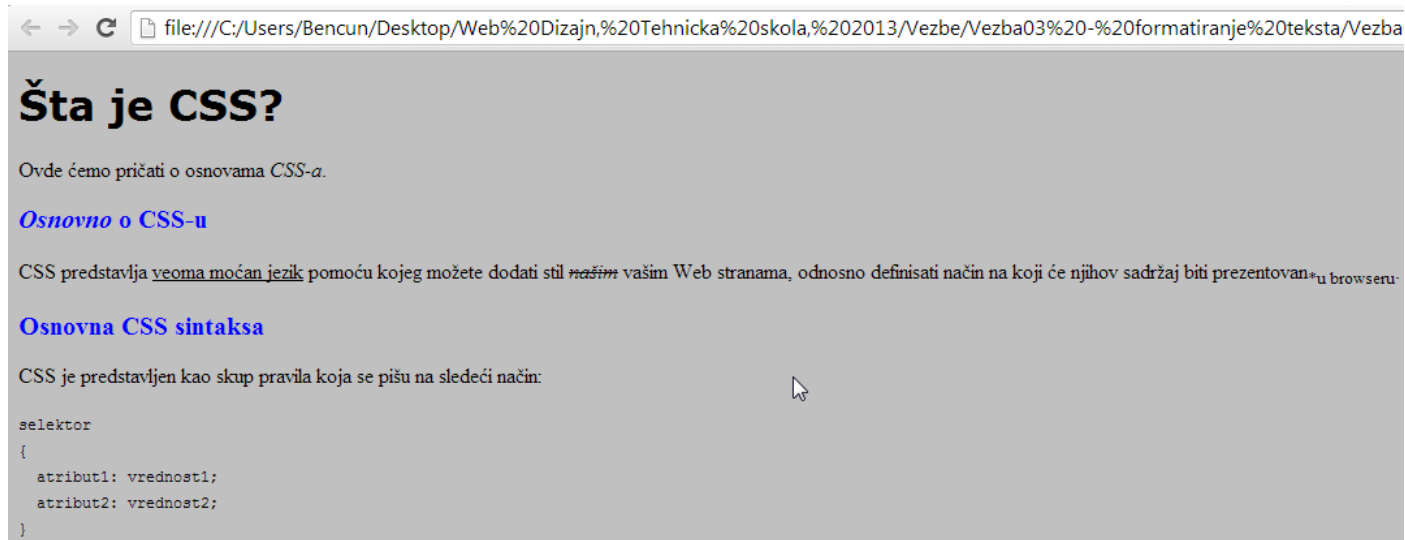
Izvršiti stilizovanje sa liste iznad.

Kompletan CSS kod izgleda ovako:

```

1  body
2  {
3      background-color: #C0C0C0;
4  }
5
6  h1
7  {
8      font-size: 32px;
9      font-family: Verdana;
10 }
11
12 h2
13 {
14     font-size: 20px;
15     color: blue;
16 }
17
18 p
19 {
20     text-align: justify;
21     font-size: 16px;
22 }
23
24 span
25 {
26     text-decoration: underline;
27 }
28
29 del
30 {
31     font-style: italic;
32 }
33
34 code
35 {
36     font-size: 12px;
37 }
    
```

Stilizovani dokument će izgledati ovako:



CSS atributi koji se najčešće koriste za formatiranje teksta (ali i još nekih elemenata na stranama) kao i primer njihove upotrebe su dati u tabeli:

Atribut	Primeri upotrebe	Opis atributa
font-family	font-family: Verdana;	definisanje tipa fonta

font-size	font-size: 20px;	definisane velicine (visine) slova
font-style	font-style: italic; font-style: normal;	definisane stila teksta
font-weight	font-weight: bold;	definisane debljine teksta
color	color: blue; color: #45DF4C	definisane boje teksta ili nekog drugog elementa
background-color	isto kao za color	definisane boje pozadine elementa ili teksta
text-align	text-align: left; text-align: justify;	ravnane teksta
text-decoration	text-decoration: underline; text-decoration: line-through;	ukrasavanje teksta
text-indent	text-indent: 18px;	uvlacenje pasusa

Upotreba linkova, slika i multimedijalnog sadržaja

Za početak napravite još tri kopije fajla "Vezba01.html" i nazovite ih redom "Vezba04.html", "Vezba041.html" i "Vezba042.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu svaki put. Takođe kreirajte jedan prazan CSS dokument pod nazivom "stil04.css" koji ćemo koristiti za stilizovanje dokumenta na kome budemo radili u ovoj vežbi. Kreirajte novi direktorijum pod imenom "folder" unutar foldera "Web design" i premestite dokument "Vezba042.html" u taj folder. U sva tri dokumenta odmah dodajte **link** do stila koji ste prethodno kreirali kako biste ih povezali. Jedan fajl će služiti za stilizaciju sva tri dokumenta.

Valja odmah napomenuti da dokument "Vezba042.html" koji se nalazi u direktorijumu "folder" nije moguće linkovati direktno jer se CSS dokument i HTML dokument u ovom slučaju ne nalaze u istom direktorijumu. Za sada prekopirajte ovu liniju koda u **head** HTML dokumenta pa ćemo kasnije objasniti kako smo formirali ovakvu putanju do CSS dokumenta:

```
<link rel="stylesheet" href="../stil04.css">
```

U ovoj vežbi radićemo sa linkovima i multimedijalnim sadržajem na HTML stranama. Kreiraćemo jednu glavnu stranu i dve dodatne strane od kojih se jedna ne nalazi u istom direktorijumu kao glavna strana već u zasebnom poddirektorijumu.

[HTML deo vežbe]

Linkovi u HTML dokumentima imaju višestruku ulogu:

- da povežu više različitih dokumenata i omoguće navigaciju između njih
- da omoguće navigaciju kroz sam HTML dokument u kome se nalaze (korišćenjem drugih linkova ili identifikatora)
- da omoguće povezivanje sa eksternim dokumentima, programima, omoguće download fajlova itd.

Reč "link" predstavlja opšteprihvaćen naziv koji se koristi za ovaj HTML element međutim pravi naziv ovog elementa je **anchor** što u bukvalnom prevodu znači sidro a njegova oznaka (tj. tag) je `<a>...`. Njegov najbitniji atribut je **href** koji zapravo predstavlja putanju do dokumenta ili bilo koji drugi objekat do koga ovaj link vodi tj. za koji je vezan (fajl, identifikator, identifikator u drugom fajlu, drugo sidro itd.).

Primer upotrebe:

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

Za učitavanje i prikazivanje slika u HTML dokumentima koristi se element `...` sa atributom **src** koji predstavlja putanju do slike:

Primer upotrebe:

```
1 
```

Što se tiče reprodukcije audio i video fajlova koriste se istoimeni elementi `<audio controls>...</audio>` i `<video controls>...</video>` koji se koriste na praktično istovetan način pa će u ovoj vežbi biti dat primer upotrebe elementa **video**.

Primer upotrebe:

```
1 <video controls>
2   <source src="video.mp4" type="video/mp4">
3 </video>
```

Sada ćemo u HTML dokumentu "Vezba04.html" kreirati link koji vodi do strane "Vezba041.html" i link koji vodi do strane "Vezba042.html" kao i dva linka od kojih jedan vodi do dna strane a drugi nas vraća na početak strane. U slučaju poslednja dva linka umesto putanje dokumenta jednostavno koristimo identifikator elementa na strani do koga želimo da "skočimo". Podsetimo se, atributi **id** i **class** se mogu dodeliti bilo kom elementu HTML-a i ovo je odličan primer još jednog načina na koji se upotrebljavaju pored toga što se koriste kao selektori za stilizovanje u CSS-u.

Nakon izmena telo HTML dokumenta izgleda ovako (mnogobrojni **
** elementi su tu da bi omogućili veštačko uvećanje strane po vertikali):

```
12 <body>
13   <p id="pocetak">Ovo je pocetak strane</p>
14   <a href="Vezba041.html">Link do prve strane</a>
15   <br>
16   <a href="folder/Vezba042.html">Link do druge strane</a>
17   <br>
18   <a href="#kraj">Kraj strane</a>
19   <br>
20   <br>
21   <br>
22   <br>
...
...
...
63   <br>
64   <br>
65   <br>
66   <p id="kraj">Ovo je kraj strane</p>
67   <a href="#pocetak">Povratak na pocetak strane</a>
68 </body>
```

Prekucajte HTML kod u dokument, sačuvajte ga.

Zatim ćemo popuniti telo dokumenta "Vezba041.html":

```
12 <body>
13   <p>Ovo je prva strana koja se nalazi u istom folderu kao i pocetna</p>
14   
15   <a href="Vezba04.html">Nazad na pocetnu stranu</a>
16 </body>
```

...a zatim i telo dokumenta "Vezba042.html":

```
12 <body>
13   <p>Ovo je druga strana koja se nalazi u podfolderu i sadrzi video klip koji se reprodukuje
14   direktno u browseru</p>
15   <video controls>
16     <source src="video.mp4" type="video/mp4">
17   </video>
18   <br>
19   <a href=" ../Vezba04.html">Nazad na pocetnu stranu</a>
20 </body>
```

Prekucajte HTML kod u oba dokumenta i sačuvajte ih.

Nakon toga iz foldera "Materijal" kopirajte fajl "slika.jpg" u folder "Web dizajn" i fajl "video.mp4" u folder "folder". Sada možete da otvorite glavni HTML dokument i proverite da li funkcionišu svi linkovi, da li prva strana prikazuje sliku i da li druga strana reprodukuje video fajl.

Kao što vidite, href atribut elementa a predstavlja putanju do fajla na koji link ukazuje ili predstavlja neki identifikator na strani na kojoj se sam link nalazi i data su oba primera upotrebe. Međutim, kada pristupamo fajlovima ili HTML dokumentima koji se ne nalaze u istom direktorijumu kao i HTML fajl sa kojim radimo, ne možemo jednostavno upotrebiti nazive fajlova već moramo upotrebiti relativnu putanju do fajla.

Kad smo već kod toga: upotreba apsolutnih putanja je besmislena, na primer "C:/Users/Bencun/Desktop/Web design/Vezba041.html" predstavlja apsolutnu putanju do dokumenta.

Primer pristupanja drugom dokumentu koji se nalazi u folderu ispod onog u kome smo trenutno dat je u HTML dokumentu koji predstavlja glavnu stranu:

```
16 | <a href="folder/Vezba042.html">Link do druge strane</a>
```

Kada je u pitanju pristupanje fajlovima i HTML dokumentima koji su hijerarhijski u folderu iznad onog u kome se nalazi HTML dokument način rada je drugačiji. Za pristup direktorijumu koji se nalazi jedan nivo iznad u hijerarhiji u odnosu na onaj u kome je HTML dokument koriste se dve tačke "../". Ukoliko želimo da pristupimo direktorijumu koji je dva nivoa iznad onda to pišemo ovako: "../../"; isto ovo pravilo važi za sve naredne direktorijume iznad, samo dodajemo dve tačke za svaki.

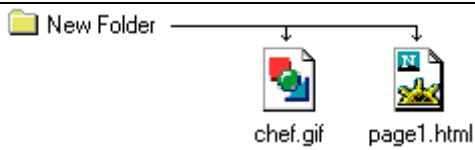
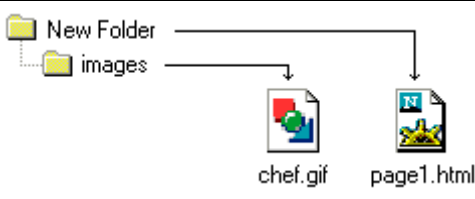
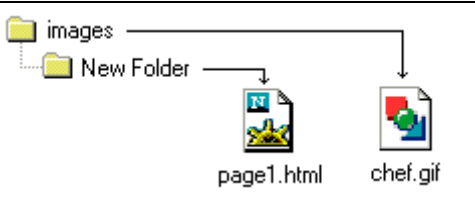
Primer upotrebe pristupa fajlu koji je u direktorijumu iznad dat je u dokumentu "Vezba042.html" kod linkovanja dokumenta sa CSS stilom koji nije prisutan u istom folderu u kome se dokument nalazi i kod formiranja linka koji vodi nazad na početnu stranu a od kojih se oba nalaze u jednom direktorijumu iznad:

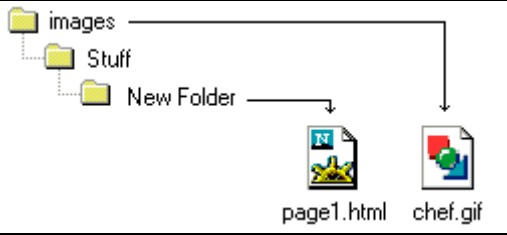
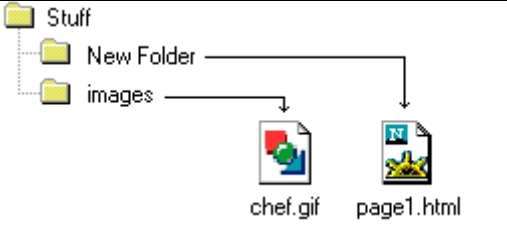
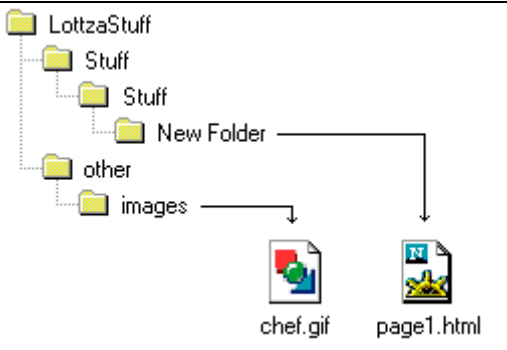
```
9 | <link rel="stylesheet" href="../stil04.css">
```

...

```
18 | <a href="../Vezba04.html">Nazad na pocetnu stranu</a>
```

Radi lakšeg shvatanja načina na koji funkcionišu relativne putanje evo vizuelizovanog primera:

	<p>SRC="chef.gif" znači da se slika nalazi u istom folderu u kome je i HTML dokument, koji je poziva.</p>
	<p>SRC="images/chef.gif" znači da se slika nalazi u folderu ispod HTML dokumenta koji je poziva. Pitanje je: koliko nivoa ispod? Onoliko koliko je potrebno (a u našem slučaju - jedan)!</p>
	<p>SRC="../../chef.gif" znači da je slika u folderu iznad HTML dokumenta koji je poziva.</p>

	<p>SRC="../../chef.gif" znači da je slika dva foldera iznad HTML dokumenta koji je poziva.</p>
	<p>SRC="../images/chef.gif" znači da je slika jedan folder gore, pa jedan dole u odnosu na HTML dokument.</p>
	<p>SRC="../../other/images/chef.gif"</p>

Linkovi (sidra) se takođe mogu koristiti za slanje e-mail pošte odnosno mogu izvršiti pokretanje e-mail klijenta na računaru i automatsko popunjavanje adresnog polja. Primer linka koji bi otvorio e-mail klijent i kreirao novu poruku koja će biti poslata na adresu *ucenik@skola.com* izgledao bi ovako:

```
21 | <a href="mailto:ucenik@skola.com">Slanje e-maila</a>
```

Vredi i spomenuti opcioni atribut elementa **a** koji omogućava da definišemo da li će se link otvoriti u novom tabu/prozoru, u prozoru svog roditeljskog dokumenta itd. Ovaj atribut se naziva **target** i vrednosti koje može imati su: **"_blank"**, **"_parent"**, **"_self"**, **"_top"** od kojih ćemo spomenuti samo **"_blank"** koji dokument otvara u novom tabu ili prozoru.

[CSS deo vežbe]

Sada ćemo preći na stilizovanje dokumenta.

Za stilizaciju linkova koriste se pseudo klase koje definišu izgled linkova u

- normalnom stanju - **a**
- u stanju kada miš prelazi preko njih - **a:hover**
- kada je link aktivan (odnosno kada se na njega klikne) - **a:active**
- i kada je već posećen - **a:visited**

Mi ćemo naše linkove stilizovati na sledeći način: linkovi će izgledati isto kada su neaktivni i kada su posećeni dok će se njihov izgled menjati prilikom prelaska miša i klika na njih. Takođe ćemo izmeniti veličinu fotografije na prvoj

strani.

Pre upuštanja u stilizovanje spomenućemo nekoliko osobina CSS-a koje ranije nismo koristili:

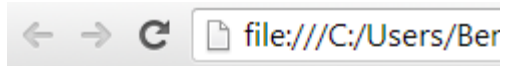
- **width** - koristi se za podešavanje širine elementa (u pikselima ili procentima)
- **height** - isto kao **width** samo za visinu
- **max-height** i **max-width** - ograničava visinu i širinu elementa bez obzira na ostala podešavanja širine i visine
- **display** - menja način na koji su elementi prikazani na ekranu a samim tim može uticati i na njihovo ponašanje; moguće vrednosti koje se najčešće koriste su: *inline* - prikazuje element kao jednolinijski (kao element ``), *block* (prikazuje element kao `<p>`), *inline-block* (blok u jednoj liniji); u datom primeru korišćićemo `display: block`; kako bi kompletna oblast koju linkovi zauzimaju reagovala na kursor miša i klikove a ne samo tekst linka

Evo kako izgleda CSS dokument nakon stilizovanja:

```
1  a, a:visited
2  {
3      text-decoration: none;
4      display: block;
5      background-color: #CCCCCC;
6      color: blue;
7      width: 150px;
8  }
9
10 a:hover
11 {
12     color: red;
13 }
14
15 a:active
16 {
17     color: yellow;
18 }
19
20 img
21 {
22     height: 150px;
23 }
```

Prekucajte stilove u CSS dokument, sačuvajte ga i sačuvajte izmene u svim ostalim dokumentima ako već niste i nakon toga otvorite glavnu stranu i krećite se po ostalim dokumentima. Uočite manje dimenzije slike i izgled linkova.

Nakon primene stilova početna strana bi trebalo da izgleda ovako:



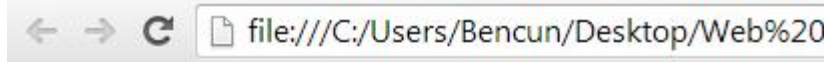
Ovo je pocetak strane

[Link do prve strane](#)

[Link do druge strane](#)

[Kraj strane](#)

Dok će prva strana na kojoj smo ograničili širinu slike sada izgledati ovako:



Ovo je prva strana koja se nalazi u istom folderu kao i pocetna



[Nazad na pocetnu stranu](#)

Upotreba listi

Kao i uvek pre svega napravite još jednu kopiju fajla "Vežba01.html" i nazovite je "Vežba05.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu. Takođe kreirajte jedan prazan CSS dokument pod nazivom "sti/05.css" koji ćemo koristiti za stilizovanje dokumenta na kome budemo radili u ovoj vežbi. Odmah linkujte dotični dokument sa HTML dokumentom korišćenjem elementa **link**.

[HTML deo vežbe]

HTML podržava tri osnovna tipa listi:

- **Neuređene liste** - njihove stavke nemaju specifičan redosled
 - ovakva lista se nalazi u elementu `...` a njene stavke unutar `` taga u elementu `...`; podrazumevani marker za stavke je popunjeni kružić
- **Numerisane ili uređene liste** - stavke imaju specifičan redosled, mogu biti numerisane korišćenjem brojeva ili uređene korišćenjem slova
 - ove liste se nalaze u elementu `...` a njihove stavke unutar `` taga u elementu `...`; podrazumevani marker za stavke su decimalne cifre
- **Definicione liste** - sadrže termine i po jednu ili više definicija za svaki termin
 - nalaze se u elementu `<dl>...</dl>` unutar kojeg se nalaze elementi `<dt>...</dt>` (koji označava termin koji se definiše) i `<dd>...</dd>` (kojeg može biti jedan ili više od jednog i koji definišu termin označen sa `<dt>`)

Liste se takođe mogu ugnežđavati jedna u drugu i to na neograničen broj nivoa tako što se svaka nova pod-lista stavlja unutar `` elementa liste u kojoj treba da se nalazi.

U HTML dokumentu ćemo kreirati četiri liste koje slede, naravno u telu dokumenta (u okviru elementa **body**):

Neuređena lista:

```
13 <h1>Primer neuredjene liste</h1>
14 <ul id="neuredjena">
15   <li>Prva stavka</li>
16   <li>Druga stavka</li>
17   <li>Treca stavka</li>
18 </ul>
```

Uređena lista:

```
20 <h1>Primer uredjene liste</h2>
21 <ol id="uredjena">
22   <li>Crveno</li>
23   <li>Zeleno</li>
24   <li>Plavo</li>
25 </ol>
```

Definiciona lista:

```

27 <h1>Primer liste sa definicijama</h1>
28 <dl>
29     <dt>HTML</dt>
30     <dd>HyperText Markup Language</dd>
31     <dd>HTML jezik je opisni jezik za HTML dokumenta</dd>
32 <dt>CSS</dt>
33     <dd>Cascading Style Sheets jezik se koristi za stilizovanje HTML dokumenata</dd>
34 </dl>
    
```

Primer ugnježdenih listi:

```

36 <h1>Primer ugnjezdenih lista</h1>
37 <p>U ovom primeru imamo dve numerisane liste unutar jedne nenumerisane</p>
38 <ul>
39     <li>Prva stavka
40     <ol>
41         <li>Podstavka prve stavke</li>
42         <li>Podstavka prve stavke</li>
43         <li>Podstavka prve stavke</li>
44     </ol>
45 </li>
46
47     <li>Druga stavka
48     <ol>
49         <li>Podstavka druge stavke</li>
50         <li>Podstavka druge stavke</li>
51         <li>Podstavka druge stavke</li>
52     </ol>
53 </li>
54 </ul>
    
```

Prepišite HTML kod sa slika u telo dokumenta, sačuvajte ga i otvorite u browseru. Primitite da određene liste sadrže identifikatore koji će nam kasnije omogućiti da primenimo određeni stil samo na njih a ne na sve liste u dokumentu. Takođe primetite kako liste izgledaju u browseru: neuređene liste podrazumevano imaju kružić za marker a uređene imaju decimalne brojeve.

[CSS deo vežbe]

Pre nego što pređemo na stilizovanje dokumenta spomenućemo CSS svojstvo koje ćemo koristiti prilikom stilizovanja liste a to je **list-style-type** a koje može imati veliki broj vrednosti pa ćemo mi spomenuti samo najkorišćenije:

Vrednost	Opis markera
none	bez markera
disc	ispunjeni kružić (podrazumevano za ul)
circle	prazan kružić
square	kvadratić
decimal	decimalni brojevi (podrazumevano za ol)
upper-alpha	velika slova abecede
lower-alpha	mala slova abecede

upper-roman	veliki rimski brojevi
lower-roman	mali rimski brojevi

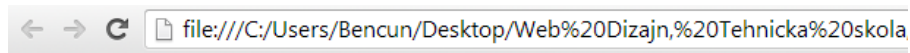
Neuređena lista sa identifikatorom istog imena će za marker ispred stavki imati kvadratić dok će uređena imati mala slova abecede.

Dokument je stilizovan na ovaj način:

```
1 #neuredjena
2 {
3     list-style-type: square;
4 }
5
6 #uredjena
7 {
8     list-style-type: lower-alpha;
9 }
10
11 #uredjena li
12 {
13     color: red;
14 }
15
16 dl dt
17 {
18     font-weight: bold;
19 }
20
21 dl dd
22 {
23     font-style: italic;
24 }
```

Prepišite stilove u CSS dokument, sačuvajte ga a onda otvorite HTML dokument u browseru i uočite nastale promene.

Nakon stilizovanja stranica izgleda ovako:



Primer neuredjene liste

- Prva stavka
- Druga stavka
- Treća stavka

Primer uredjene liste

- a. Crveno
- b. Zeleno
- c. Plavo

Primer liste sa definicijama

HTML

HyperText Markup Language

HTML jezik je opisni jezik za HTML dokumenta

CSS

Cascading Style Sheets jezik se koristi za stilizovanje HTML dokumenata

Primer ugnjezenih lista

U ovom primeru imamo dve numerisane liste unutar jedne nenumerisane

- Prva stavka
 1. Podstavka prve stavke
 2. Podstavka prve stavke
 3. Podstavka prve stavke
- Druga stavka
 1. Podstavka druge stavke

Upotreba tabela

Kao i uvek pre svega napravite još jednu kopiju fajla "Vezba01.html" i nazovite je "Vezba06.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu. Takođe kreirajte jedan prazan CSS dokument pod nazivom "sti06.css" koji ćemo koristiti za stilizovanje dokumenta na kome ćemo raditi u ovoj vežbi. Odmah linkujte dotični dokument sa HTML dokumentom korišćenjem elementa **link**.

Tabele se u HTML-u koriste za prikaz tabelarnih podataka. Ranije su mnogi Web dizajneri koristili tabele da bi njima definisali vizuelni raspored elemenata na strani što je pogrešan pristup rešenju problema rasporeda elemenata - za to se koristi CSS. Upamtite, HTML određuje strukturu i sadržaj, a CSS vizuelni izgled dokumenta.

[HTML deo vežbe]

Kreiranje tabela u HTML-u je jako jednostavan proces. Za kreiranje tabele po standardima se koristi element **<table>...</table>** unutar koga se nalaze elementi **caption** (naslov tabele), **thead** (zaglavlje tabele), **tfoot** (podnožje tabele), i **tbody** (telo tabele, podaci), tim redosledom, s tim što element za podnožje nije obavezan. Za kreiranje redova u zaglavlju, podnožju i telu tabele koristi se element **<tr>...</tr>** dok se za kreiranje polja odnosno ćelija sa podacima unutar ovih redova koristi element **<td>...</td>** s tim što se u zaglavlju tabele umesto njega koristi **<th>...</th>**. Kostur jedne tabele izgleda otprilike ovako:

```

87 <table>
88     <caption>...</caption>
89     <thead>
90         <tr>
91             <th>...</th>
92             ...
93         </tr>
94     </thead>
95     <tfoot>
96         <tr>
97             <td>...</td>
98             ...
99         </tr>
100    </tfoot>
101    <tbody>
102        <tr>
103            <td>...</td>
104            ...
105        </tr>
106    </tbody>
107 </table>

```

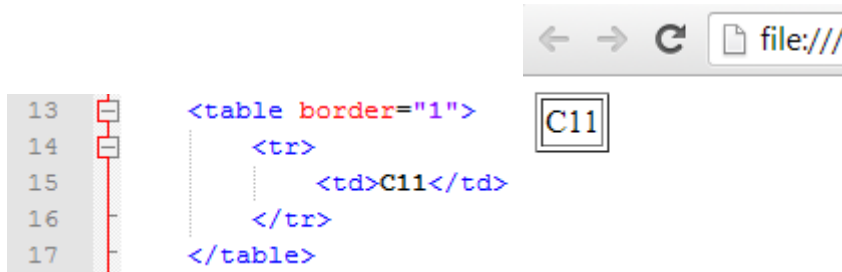
Broj kolona tabele određen je brojem kolona u prvom redu tabele odnosno u prvom redu zaglavlja tabele. Ukoliko u narednim redovima u telu ili podnožju postoji kolona koja se nalazi van ovog opsega ona će štrčati iz tabele.

Podelu tabela možemo izvršiti na simetrične i asimetrične. Simetrične tabele sadrže isti broj ćelija u svakom redu dok asimetrične imaju varijacije u svojoj strukturi a te varijacije zapravo predstavljaju formiranje većih ćelija objedinjavanjem više ćelija po kolonama ili redovima.

U ovoj vežbi ćemo krenuti od jednostavne simetrične tabele koju ćemo postepeno dopunjavati elementima.

U ovoj vežbi je potrebno da nakon svakog koraka i unošenja novog koda sačuvate HTML dokument i otvorite ga u browseru da biste uočili promene.

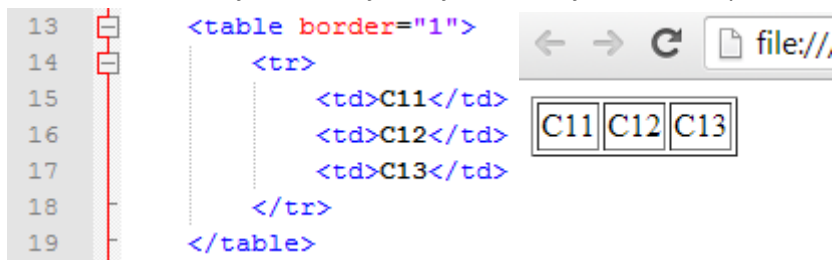
Prvo ćemo (naravno u telu dokumenta) kreirati osnovnu tabelu sa jednim redom i jednom ćelijom. Dodaćemo atribut **border** u okviru elementa **table**, ovaj metod se inače ne primenjuje već se kreiranje ivica tabele vrši u CSS-u ali ćemo radi jednostavnije demonstracije upotrebiti atribut **border** da bi se videle ivice ćelija:



```

13 <table border="1">
14 <tr>
15 <td>C11</td>
16 </tr>
17 </table>
    
```

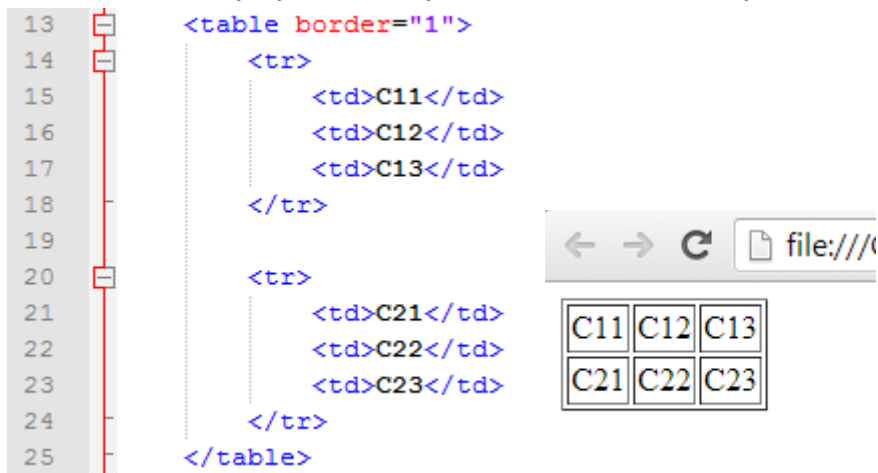
Zatim ćemo dodati još dve ćelije u taj isti red koji smo već napravili:



```

13 <table border="1">
14 <tr>
15 <td>C11</td>
16 <td>C12</td>
17 <td>C13</td>
18 </tr>
19 </table>
    
```

Zatim ćemo dodati još jedan red koji će takođe sadržati tri ćelije:



```

13 <table border="1">
14 <tr>
15 <td>C11</td>
16 <td>C12</td>
17 <td>C13</td>
18 </tr>
19 <tr>
20 <td>C21</td>
21 <td>C22</td>
22 <td>C23</td>
23 </tr>
24 </table>
25
    
```

Ova tabela predstavlja primer jednostavne simetrične tabele sa dva reda i tri kolone. Sada ćemo od ove tabele načiniti asimetričnu tabelu. Objedinićemo obe ćelije u prvoj koloni i zadnje dve ćelije u prvom redu.

Za proširenje ćelija koriste se atributi **rowspan** i **colspan**.

Atribut **rowspan** proširuje ćeliju za onoliko redova kolika je njegova vrednost. Primer: **rowspan="6"** znači da će se ćelija protezati preko šest redova a sadržaj te ćelije će biti ono što se nalazi između početne i krajnje oznake elementa **td**.

Atribut **colspan** proširuje ćeliju za onoliko kolona kolika je njegova vrednost. Primer: **colspan="3"** znači da će se ćelija protezati preko tri naredne kolone.

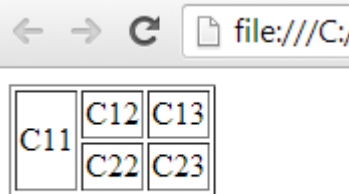
Sada ćemo prvo proširiti prvu ćeliju u prvom redu tako da se proširi i u drugi red, ali ovo znači da ćemo sada u drugom redu imati jednu ćeliju višak koju moramo ukloniti inače će ona pokvariti izgled tabele.

Dodaćemo atribut **rowspan** ćeliji C11 dok ćemo ćeliju C21 ukloniti jer će njeno mesto morati da zauzme ćelija C11:

```

13 <table border="1">
14 <tr>
15 <td rowspan="2">C11</td>
16 <td>C12</td>
17 <td>C13</td>
18 </tr>
19
20 <tr>
21 <td>C22</td>
22 <td>C23</td>
23 </tr>
24 </table>

```

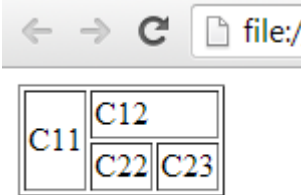


Sada ćemo takođe proširiti drugu ćeliju prvog reda tako da zauzima dve kolone. Ovo će ponovo uslovljavati uklanjanje suvišne ćelije. Ovog puta C12 treba da se proširi za jednu kolonu i zauzme mesto ćelije C13 pa ćemo C13 ukloniti:

```

13 <table border="1">
14 <tr>
15 <td rowspan="2">C11</td>
16 <td colspan="2">C12</td>
17 </tr>
18
19 <tr>
20 <td>C22</td>
21 <td>C23</td>
22 </tr>
23 </table>

```



Da bi tabela bila kompletna i izrađena po najnovijim HTML standardima neophodno je da ima ranije spomenute elemente: naslov, zaglavlje, podnožje i telo.

Kod koji već imamo predstavlja sadržaj tabele što znači da on predstavlja njeno telo **tbody** i zato ćemo ga uokviriti tim elementom:

```

15 <tbody>
16 <tr>
17 <td rowspan="2">C11</td>
18 <td colspan="2">C12</td>
19 </tr>
20
21 <tr>
22 <td>C22</td>
23 <td>C23</td>
24 </tr>
25 </tbody>

```

Ova promena nema uticaja na izgled tabele samo utiče na logičku strukturu dokumenta i same tabele.

Sada ćemo dodati elemente **caption**, **thead** i **tfoot** koji, po HTML standardima, moraju da se nalaze ispred **tbody** elementa. Element **tfoot** nije obavezan ali ćemo ga dodati da bismo videli kako se upotrebljava:

```

13 <table border="1">
14 <caption>Naslov tabele</caption>
15
16 <thead>
17 </thead>
18
19 <tfoot>
20 </tfoot>
21
22 <tbody>
23 <tr>
24 <td colspan="2">C11</td>

```

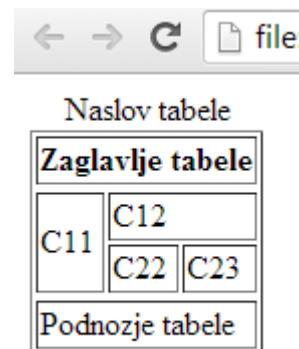
Na izgled tabele utiče samo dodavanje naslova a zaglavlje i podnožje su trenutno bez sadržaja pa ne utiču na prikaz tabele.

Sada ćemo dodati sadržaj u zaglavlje i podnožje. Postoji samo jedna razlika u odnosu na uobičajen način dodavanja sadržaja u tabelu a to je da se u elementu **thead** za kreiranje ćelija ne koristi element **td** već **th**. Podesićemo **colspan** atribut u zaglavlju i podnožju tako da oba elementa imaju samo po jednu ćeliju koja se proteže preko sve tri kolone.

```

13 <table border="1">
14 <caption>Naslov tabele</caption>
15
16 <thead>
17 <tr>
18 <th colspan="3">Zaglavlje tabele</th>
19 </tr>
20 </thead>
21
22 <tfoot>
23 <tr>
24 <td colspan="3">Podnozje tabele</td>
25 </tr>
26 </tfoot>
27
28 <tbody>
29 <tr>

```



[CSS deo vežbe]

Izvršićemo prosto stilizovanje tabele:

1. Promenićemo stil teksta naslova u kursiv

```

1 caption
2 {
3     font-style: italic;
4 }

```

2. Obojicemo pozadinu zaglavlja u svetlo sivu boju

```
6 th
7 {
8     background-color: #DDDDDD;
9 }
```

3. Podesićemo ivice tabele tako da između ćelija nema slobodnog prostora

```
11 table
12 {
13     border-collapse: collapse;
14 }
```

Svojstvo **border-collapse** tabeli govori da li treba da ukloni razmak između pojedinačnih ivica njenih ćelija i njene celokupne spoljašnje ivice. Upotreba ovog svojstva izgleda ovako:

border-collapse: collapse;

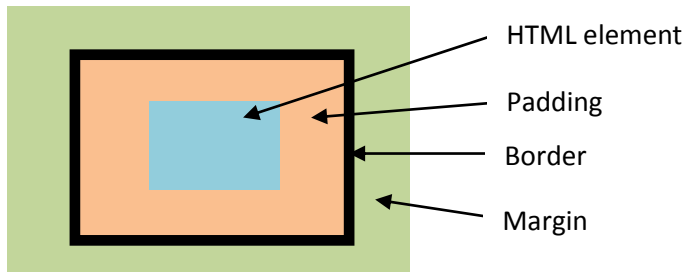
4. Podesićemo sve ćelije tako da sadržaj ćelija bude odvojen od ivica i tekst centriran

```
16 th,td
17 {
18     padding: 4px;
19     text-align: center;
20 }
```

Svojstvo **padding** definiše unutrašnje rastojanje između sadržaja (u našem slučaju teksta) i **bordera** (ivice).

Spomenućemo i svojstvo **margin** koje predstavlja prazan prostor između **bordera** i okolnih elemenata dokumenta.

Vrednost se daje takođe u pikselima. Ilustrovano to izgleda ovako:



5. Obojicemo slova u podnožju u crvenu boju

```
22 tfoot
23 {
24     color: red;
25 }
```

Ovim je stilizovanje završeno. Međutim, moramo spomenuti još jednu stvar. Mi smo u HTML dokument u elementu **table** uključili atribut **border** da bi ivice ćelija bile vidljive. Ovo se inače radi u CSS-u i ukoliko bismo ivice implementirali preko CSS-a onda bi bilo potrebno da obrišemo atribut **border** iz HTML dokumenta a da u stil za **th** i **td** dodamo **border: 1px solid black;** što znači da sintaksa za ovo svojstvo glasi ovako:

border: debljina_ivice izgled_linije boja_linije;

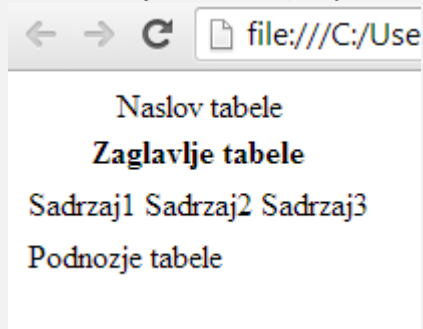
Za izgled linije možemo koristiti vrednosti: *none, solid, dashed, dotted*, itd.

Dovoljno je primeniti svojstvo **border** na **th** i **td** elemente da bismo dobili oivičene ćelije, međutim ukoliko želimo i okvir oko cele tabele (npr. kada imamo ćelije koje štrče i slično) potrebno je primeniti ovo svojstvo i na element **table** odnosno na celu tabelu koju stilizujemo.

Za podešavanje širine i visine tabele i pojedinačnih ćelija koriste se svojstva **width** i **height** (u pikselima, naravno) a za centriranje tabele u okviru elementa u kome se nalazi (u našem slučaju u telu HTML dokumenta) je dovoljno podesiti margine tabele na vrednost *auto* a sintaksno ta naredba izgleda ovako:

margin: *auto*;

Za domaći zadatak pokušajte da oformite ovakvu tabelu u HTML-u koristeći sve elemente: **table**, **caption**, **thead**, **tfoot**, **tbody**, **tr**, **th** i **td** (imajte na umu da bez stilova tabela nema ivice!):



I da je zatim stilizujete da izgleda ovako (centrirana je na strani):

Dizajn,%20Tehnicka%20skola,%202013/Domaci/Tabele/tabela.html



A screenshot of a web browser window showing a styled table. The address bar shows 'Dizajn,%20Tehnicka%20skola,%202013/Domaci/Tabele/tabela.html'. The table is centered on the page and has a black border. The title 'Naslov tabele' is in green. The header 'Zaglavlje tabele' is in bold. The content cells 'Sadrzaj1', 'Sadrzaj2', and 'Sadrzaj3' are in the middle. The footer 'Podnozje tabele' is in bold and underlined.

Zaglavlje tabele		
Sadrzaj1	Sadrzaj2	Sadrzaj3
Podnozje tabele		

Dopunska vežba, fajlovi "Vezba06_Dopunska.html" i "stil06_dopunski.css"

U ovoj vežbi ćemo kreirati tabelu koja sadrži sve pomenute elemente i potom ćemo je stilizovati. Prvo ćemo kreirati element **table** a unutar njega element **caption** koji će sadržati naslov tabele "Naslov tabele".

Zatim kreirajte element **thead** koji će sadržati jedan red **tr** a koji će unutar sebe sadržati jednu praznu kolonu zaglavlja i dve kolone od kojih jedna sadrži reč "Ime" a druga "Prezime". Ovim smo širinu ostatka tabele automatski ograničili na 3 kolone.

Nakon zatvaranja elementa **thead** otvaramo element **tfoot** i u njemu kreiramo takođe jedan red **tr** u kome ćemo kreirati samo jedan element **td** koji će činiti podnožje tabele. Naš cilj je da podnožje tabele čini samo jedna ćelija pa stoga ovom elementu **td** dodeljujemo atribut **colspan** i vrednost atributa podešavamo na "3" čime smo ovoj ćeliji rekli da se proširi tj. zauzme prostor od tri kolone.

Proširenje jedne ćelije tako da zauzima dva ili više redova ili dve ili više kolona postižemo korišćenjem atributa **rowspan** i **colspan** kojima određujemo koliko redova i kolona ćelija može zauzeti u tabeli.

Nakon zatvaranja elementa **tfoot** otvaramo element **tbody** i započinjemo unos podataka. Kreirajte dva prazna reda. Prvi red neka sadrži tri ćelije sa sledećim vrednostima: 1. , Jovan, Jovanović.

Drugi red neka sadrži takođe tri ćelije sa sledećim vrednostima: 2. , Petar, Petrović.

Drugom redu dodelite atribut **class** sa vrednošću "parni".

Sada imate dva formirana reda sa rednim brojevima, imenima i prezimenima. Kopirajte ih i pejstujte u telo tabele dva puta tako da imate ukupno šest osoba na listi i u skladu sa time izmenite redne brojeve tih osoba, imena ne morate menjati.

Zatvorite **tbody** tabele, sačuvajte dokument i otvorite tabelu u pregledaču. Tabela nema ivice ali bi trebalo da sadrži zaglavlje, šest imena i prezimena sa rednim brojevima i podnožje.

Sada ćemo, radi demonstracije, dodati sedmi i osmi red u tabelu i pokazati kako funkcionišu atributi **colspan** i **rowspan**.

Nakon što ste dodali dva reda koji slede iza imena i prezimena koje već imate, u prvom kreirajte jednu ćeliju sa atributima **rowspan** i **colspan** pri čemu ćemo oba podesiti na vrednost "2", a to znači da će ta ćelija zauzimati prostor od dva reda i dve kolone. Sada u ovaj red dodajte još dve ćelije sa bilo kojim sadržajem.

U drugi dodatni red dodajte samo jednu ćeliju sa takođe proizvoljnim sadržajem.

Sačuvajte i otvorite dokument u pregledaču. Primetićete da suvišna ćelija u sedmom redu štrči jer je prva ćelija u tom redu podešena da zauzme prostor od dve kolone. Takođe ćete primetiti da je osmi red, iako sadrži samo jednu ćeliju, potpuno popunjen iz razloga što je ćelija iz prethodnog reda podešena da zauzme još jedan red ispod sebe i dve kolone u njemu.

Telo HTML dokumenta izgleda ovako:

```
13 <table id="kompleksna">
14 <caption>Naslov tabele</caption>
15 <thead>
16 <tr>
17 <th></th>
18 <th>Ime</th>
19 <th>Prezime</th>
20 </tr>
21 </thead>
22 <tfoot>
23 <tr>
24 <td colspan="3">Kraj tabele u 1 redu i 3 kolone</td>
25 </tr>
26 </tfoot>
27 <tbody>
28 <tr>
29 <td>1.</td>
30 <td>Jovan</td>
31 <td>Jovanović</td>
32 </tr>
33 <tr class="parni">
34 <td>2.</td>
35 <td>Petar</td>
36 <td>Petrović</td>
37 </tr>
38 <tr>
39 <td>3.</td>
40 <td>Jovan</td>
41 <td>Jovanović</td>
42 </tr>
```

```

43 <tr class="parni">
44 <td>4.</td>
45 <td>Petar</td>
46 <td>Petrović</td>
47 </tr>
48 <tr>
49 <td>5.</td>
50 <td>Jovan</td>
51 <td>Jovanović</td>
52 </tr>
53 <tr class="parni">
54 <td>6.</td>
55 <td>Petar</td>
56 <td>Petrović</td>
57 </tr>
58 <tr>
59 <td rowspan="2" colspan="2">Ćelija koja zauzima<br>
60 i naredni red i kolonu</td>
61 <td>7. red, 3. kol</td>
62 <td>7. red, 4. kol</td>
63 </tr>
64 <tr>
65 <td>8. red 3. kol</td>
66 </tr>
67 </tbody>
68 </table>

```

Sada ćemo preći na stilizovanje tabele.

Prvo CSS svojstvo koje je jako bitno jeste **border** i ono definiše izgled ivice tabele. Sintaksa glasi ovako:

border: debljina_ivice izgled_linije boja_linije;

Za izgled linije možemo koristiti vrednosti: *none, solid, dashed, dotted*, itd.

Dovoljno je primeniti svojstvo **border** na **th** i **td** elemente da bismo dobili oivičene ćelije, međutim ukoliko želimo i okvir oko tabele potrebno je primeniti ovo svojstvo i na element **table** odnosno na celu tabelu koju stilizujemo.

Sledeće bitno svojstvo jeste **border-collapse** koje tabeli govori da li treba da ukloni razmak između pojedinačnih ivica njenih ćelija i njene celokupne spoljašnje ivice. Upotreba ovog svojstva izgleda ovako:

border-collapse: collapse;

U ovoj vežbi takođe koristimo svojstvo **padding** koje definiše unutrašnje rastojanje između sadržaja (u našem slučaju teksta) i **bordera** (ivice). Spomenućemo i svojstvo **margin** koje predstavlja prazan prostor između **bordera** i okolnih elemenata dokumenta. Vrednost se daje takođe u pikselima.

U ovoj vežbi ćemo uraditi sledeće:

- podesiti **border-collapse: collapse;** za ceo element **table**
- podesiti elementima **table, th, i td** ivicu na debljinu od jednog piksela, punu liniju i boju linije na crnu
- podesiti visinu svih elemenata **tr** na 30px;
- podesiti svojstvo **text-align** na *left* za sve **th** i **td** elemente (takoreći sve ćelije u tabeli) i podesiti **padding** svojstvo na 4 piksela
- podesiti pozadinsku boju zaglavlja na sivu a boju slova na belu

- podesiti boju teksta podnožja na plavu
- svakom redu koji je klase "*parni*" ćemo pozadinsku boju podesiti na svetlo sivu: #CCCCCC.

Nakon editovanja CSS dokumenta isti izgleda ovako:

```
1  table
2  {
3      border-collapse: collapse;
4  }
5  table, td, th
6  {
7      border: 1px solid black;
8  }
9  tr
10 {
11     height: 30px;
12 }
13 td, th
14 {
15     text-align: left;
16     padding: 4px;
17 }
18 thead
19 {
20     background-color: gray;
21     color: white;
22 }
23 tfoot
24 {
25     color: blue;
26 }
27 .parni
28 {
29     background-color: #CCCCCC;
30 }
```

Sačuvajte i HTML i CSS dokument i otvorite stranicu u pregledaču kako biste videli efekte stilizovanja. Menjajte neke vrednosti u CSS fajlu da biste videli kako utiču na izgled tabele.

Tabela koju biste trebali da dobijete na kraju vežbe izgleda ovako:

file:///C:/Users/Bencun/Deskto

Naslov tabele

	Ime	Prezime
1.	Jovan	Jovanović
2.	Petar	Petrović
3.	Jovan	Jovanović
4.	Petar	Petrović
5.	Jovan	Jovanović
6.	Petar	Petrović
Ćelija koja zauzima i naredni red i kolonu	7. red, 3. kol	7. red, 4. kol
	8. red 3. kol	
Kraj tabele u 1 redu i 3 kolone		

HTML forme

Kao i uvek pre svega napravite još jednu kopiju fajla "Vezba01.html" i nazovite je "Vezba07.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu. Takođe kreirajte jedan prazan CSS dokument pod nazivom "sti107.css" koji ćemo koristiti za stilizovanje dokumenta na kome budemo radili u ovoj vežbi. Odmah linkujte dotični dokument sa HTML dokumentom korišćenjem elementa **link**.

HTML forme predstavljaju obrasce za unos podataka. Sastoje se od polja za unos teksta, brojeva, datuma, dugmadi itd, čije se vrednosti nakon popunjavanja forme šalju na server na kome se nalazi Web sajt radi obrade podataka.

[HTML deo vežbe]

Forma se kreira upotrebom elementa **<form>...</form>** koji unutar sebe sadrži sve elemente forme. Najbitniji atributi ovog elementa su:

- **method** - može imati vrednosti **POST** ili **GET** i definiše način slanja podataka na server
- **action** - definiše koja se akcija poziva kada se klikne na dugme za slanje podataka koje forma sadrži
- **enctype** - definiše tip enkripcije odnosno zaštite podataka koji se šalju sa forme

Grupisanje srodnih elemenata forme unutar same forme se vrši upotrebom elementa **<fieldset>...</fieldset>** koji unutar sebe sadrži element **<legend>...</legend>** koji služi da opiše fieldset u kome se nalazi i polja za unos podataka. Element fieldset nije neophodan kod jednostavnih formi.

Element **<label>...</label>** služi da pojedinačno opiše svako polje za unos podataka i obično se u HTML dokumentu nalazi uz element koji opisuje, neposredno pre ili posle njega u dokumentu. Ne predstavlja ništa drugo do običan tekst koji se ispisuje na ekranu.

Kreiranje osnovnih polja za unos podataka vrši se pomoću više različitih elemenata od kojih ćemo mi razmotriti samo osnovne. Svaki od ovih elemenata mora sadržati atribut **name** koji definiše ime ovog elementa pri slanju podataka na server i atribut **value** koji definiše vrednost koju element ima pri slanju.

Element **<input>** se koristi za kreiranje jednolinijskih tekstualnih polja, dugmadi, čekboksova, radio dugmadi itd. Atributom **type** se određuje koji će tip ulaznog polja biti kreiran na ekranu pa tako za vrednosti ovog atributa imamo (nabrajaćemo samo osnovne tipove polja za unos podataka):

- **text** - kreira klasično jednolinijsko tekstualno polje za unos kratkih tekstualnih podataka
- **password** - kreira polje za unos šifre, u ovakvom polju slovni karakteri nisu prikazani već browser umesto njih prikazuje kružice ili zvezdice ali ovo polje za vrednost atributa **value** ipak ima unešen tekst
- **checkbox** - kreira čekboks koji može da se štiklira i omogućava biranje jedne ili više ponuđenih opcija odjednom, kao i mogućnost da se nijedna opcija ne izabere; kada koristimo čekboksove možemo koristiti atribut **checked** da ga unapred štikliramo
- **radio** - kreira radio dugme; radio dugmad predstavljaju grupu dugmadi od kojih samo jedno može biti odabrano u jednom trenutku; grupu radio dugmadi kreiramo tako što im svima dodelimo isti atribut **name** i time postizemo to da se dugmad te grupe međusobno isključuju; i nad ovom dugmadi važi upotreba atributa **checked**
- **submit** - kreira dugme koje šalje podatke sa forme na server; atributom **value** određujemo tekst koji je prikazan na dugmetu
- **reset** - kreira dugme koje resetuje sve podatke na formi u početno stanje
- **button** - kreira dugme za opštu namenu

Element **<select>** se koristi za kreiranje padajućih listi, unutar sebe sadrži elemente **<option>...</option>** od kojih svaki sadrži po jednu opciju koju ova padajuća lista nudi a podrazumevani izbor možemo označiti korišćenjem atributa **selected** unutar taga te opcije. Mogu se i kreirati grupe opcija korišćenjem taga **<optgroup>...</optgroup>** kojim se nećemo baviti u ovoj vežbi.

Element **<textarea>...</textarea>** služi za kreiranje velikih tekstualnih polja koja omogućavaju unos velike količine tekstualnih podataka.

Vredi spomenuti i element **<button>...</button>** koji se koristi za kreiranje dugmadi ali za razliku od elementa **input** sa atributom **type="button"** ovaj element nije samozatvarajući pa je tako moguće smestiti dodatne HTML elemente unutar dugmeta. Nećemo ga upotrebljavati u vežbi.

Korišćenjem ovih elemenata kreiraćemo stranicu koja sadrži formu koja služi za prijavu rada. Forma će sadržati dva fieldseta, prvi će sadržati polja koja se tiču ličnih podataka o učeniku a druga će se sastojati od polja koja se tiču informacija o radu. Izvan fieldsetova nalaziće se polje za unos šifre i dva dugmeta od kojih jedno šalje podatke a drugo resetuje formu u početno stanje.

Prvo ćemo kreirati formu:

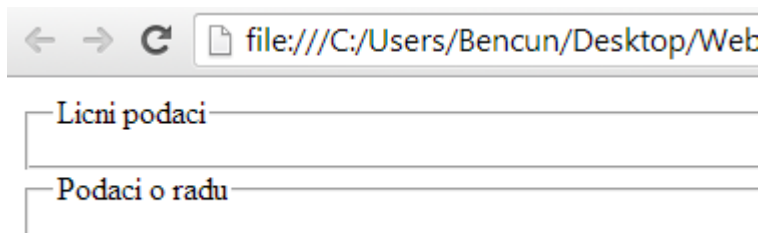
```
12 <body>
13 <form method="POST" action="mailto:mej1@mej1.com?subject=Prijava rada" enctype="text/plain">
14
15 </form>
16 </body>
```

U browseru je strana prazna jer forma nema sadržaj.

Nakon toga unutar forme dodajemo dva fieldseta, jedan će služiti za polja koja se tiču ličnih podataka a drugi fieldset za polja koja se tiču podataka o radu koji učenik dostavlja:

```
13 <form method="POST" action="mailto:mej1@mej1.com?subject=Prijava rada" enctype="text/plain">
14 <fieldset>
15 <legend>Licni podaci</legend>
16
17 </fieldset>
18
19 <fieldset>
20 <legend>Podaci o radu</legend>
21
22 </fieldset>
23 </form>
```

U browseru bi stranica sada trebala da izgleda ovako:



file:///C:/Users/Bencun/Desktop/Web

Licni podaci

Podaci o radu

Zatim ćemo dodati tekstualna polja i labele koje ih opisuju u prvi **fieldset** koji se tiče ličnih podataka. Labele nećemo objašnjavati jer je njihova upotreba očigledna i jedostavna, dok ćemo za element **input** samo prikazati osnovnu

sintaksu:

```
<input type="tip_polja" name="ime_polja" value="vrednost_polja">
```

Atribut **value** kod tekstualnih polja nije obavezan jer tekst u polje ionako unosi korisnik osim ukoliko ne želimo da polje ima predefinisanu vrednost.

Evo kako izgleda kreiranje tri labele i tri tekstualna polja za ime, prezime i temu rada:

Napomena: elementima smo unapred dodelili klase koje ćemo kasnije koristiti kod stilizovanja forme, klase možete ukloniti u slučaju da ne radite stilizovanje dokumenta na ovaj način. Pomeranje elemenata u novi red u tom slučaju možete lako postići i dodavanjem elementa **
** nakon svakog polja za unos podataka (ovaj metod se inače ne koristi jer se pozicioniranje elemenata uvek vrši kroz CSS stilove).

```

14 <fieldset>
15 <legend>Licni podaci</legend>
16 <label class="ceored">Ime:</label>
17 <input class="tekst" type="text" name="Ime"/>
18 <label class="ceored">Prezime:</label>
19 <input class="tekst" type="text" name="Prezime"/>
20 <label class="ceored">Tema:</label>
21 <input class="tekst" type="text" name="Tema"/>
22 </fieldset>

```

U browseru dokument izgleda ovako nakon ove izmene:

file:///C:/Users/Bencun/Desktop/Web%20Dizajn,%20Tehnicka%20skola,%20

Licni podaci

Ime: Prezime: Tema:

Podaci o radu

Dalje ćemo dva radio dugmeta i naravno labele koje ih opisuju. Obratite pažnju na to da smo ih grupisali tako što smo im dodelili isto ime (**name**) tako da se međusobno isključuju:

```

14 <fieldset>
15 <legend>Licni podaci</legend>
16 <label class="ceored">Ime:</label>
17 <input class="tekst" type="text" name="Ime"/>
18 <label class="ceored">Prezime:</label>
19 <input class="tekst" type="text" name="Prezime"/>
20 <label class="ceored">Tema:</label>
21 <input class="tekst" type="text" name="Tema"/>
22 <label>Učenik:</label>
23 <input type="radio" name="zvanje" value="Ucenik" checked />
24 <label>Student:</label>
25 <input type="radio" name="zvanje" value="Student"/>
26 </fieldset>

```

Dokument u browseru sada izgleda ovako:

Dizajn,%20Tehnicka%20skola,%202013/Vezbe/Vezba07%20-%

Tema: Učenik: Student:

Nakon toga prelazimo na drugi **fieldset** koji se tiče podataka o radu. Prvo ćemo dodati tri čekboksa koji služe za izbor oblika maturalskog rada. Labelama i čekboksovima ponovo unapred dodeljujemo klase koje će se koristiti pri stilizovanju dokumenta:

```
26 </fieldset>
27
28 <fieldset>
29 <legend>Podaci o radu</legend>
30 <input class="cek" type="checkbox" name="Strad" value="Strad" checked />
31 <label class="cek">Stampani rad</label>
32 <input class="cek" type="checkbox" name="CD" value="CD"/>
33 <label class="cek">CD</label>
34 <input class="cek" type="checkbox" name="Slajdovi" value="Slajdovi"/>
35 <label class="cek">Slajdovi</label>
36 </fieldset>
```

Dokument u browseru sada izgleda ovako:

file:///C:/Users/Bencun/Desktop

Licni podaci

Ime: Prezime:

Podaci o radu

Stampani rad CD Slajdovi

Sada ćemo dodati padajuću listu koja omogućava izbor predmeta.

```

28 <fieldset>
29 <legend>Podaci o radu</legend>
30 <input class="cek" type="checkbox" name="Strad" value="Strad" checked />
31 <label class="cek">Stampani rad</label>
32 <input class="cek" type="checkbox" name="CD" value="CD"/>
33 <label class="cek">CD</label>
34 <input class="cek" type="checkbox" name="Slajdovi" value="Slajdovi"/>
35 <label class="cek">Slajdovi</label>
36 <label>Predmet:</label>
37 <select name="Predmet">
38 <option value="Matematika" selected>Matematika</option>
39 <option value="Fizika">Fizika</option>
40 <option value="Hemija">Hemija</option>
41 </select>
42 </fieldset>

```

Dokument sada izgleda ovako:

The screenshot shows a web browser window with the address bar displaying 'file:///C:/Users/Bencun/Desktop/Web%20Di'. The form contains two sections: 'Licni podaci' with input fields for 'Ime:' and 'Prezime:', and 'Podaci o radu' with three checkboxes for 'Stampani rad', 'CD', and 'Slajdovi', and a dropdown menu for 'Predmet:' currently showing 'Matematika'.

Poslednja stavka u ovom **fieldsetu** je **textarea**. Ovaj element služi za unos dugačkog višelinijskog teksta.

```

28 <fieldset>
29 <legend>Podaci o radu</legend>
30 <input class="cek" type="checkbox" name="Strad" value="Strad" checked />
31 <label class="cek">Stampani rad</label>
32 <input class="cek" type="checkbox" name="CD" value="CD"/>
33 <label class="cek">CD</label>
34 <input class="cek" type="checkbox" name="Slajdovi" value="Slajdovi"/>
35 <label class="cek">Slajdovi</label>
36 <label>Predmet:</label>
37 <select name="Predmet">
38 <option value="Matematika" selected>Matematika</option>
39 <option value="Fizika">Fizika</option>
40 <option value="Hemija">Hemija</option>
41 </select>
42 <label>Opis rada:</label>
43 <textarea name="opisRada"></textarea>
44 </fieldset>

```

Dokument nakon ove izmene izgleda ovako:

ktop/Web%20Dizajn,%20Tehnicka%20skola,%202013/Vezbe/Vezba

Tema: Učenik: Student:

Opis rada:

Posle oba fieldseta dodaćemo polje za lozinku i dva dugmeta od kojih jedno šalje formu a drugo je resetuje. Atribut **action** dugmeta *Submit* je podešen tako da kreira lažni e-mail koji će nam omogućiti da pogledamo u kom se obliku podaci nakon unosa šalju na server. Evo poslednjeg dela koda:

```

43         <textarea name="opisRada"></textarea>
44     </fieldset>
45
46     <label id="labela">Lozinka:</label>
47     <input id="lozinka" type="password" name="Lozinka"/>
48     <input type="submit" value="Prijavi"/>
49     <input type="reset" value="Poništi"/>
50 </form>
51 </body>
    
```

...koji u browseru izgleda ovako:

file:///C:/Users/Bencun/Desktop/Web%20Dizajn

Licni podaci

Ime: Prezime: T:

Podaci o radu

Stampani rad CD Slajdovi Predmet: Opi

Lozinka:

Kompletan HTML kod tela dokumenta na kraju izgleda ovako:

```

12 <body>
13 <form method="POST" action="mailto:mej1@mej1.com?subject=Prijava rada" enctype="text/plain">
14 <fieldset>
15 <legend>Licni podaci</legend>
16 <label class="ceored">Ime:</label>
17 <input class="tekst" type="text" name="Ime"/>
18 <label class="ceored">Prezime:</label>
19 <input class="tekst" type="text" name="Prezime"/>
20 <label class="ceored">Tema:</label>
21 <input class="tekst" type="text" name="Tema"/>
22 <label>Učenik:</label>
23 <input type="radio" name="zvanje" value="Ucenik" checked />
24 <label>Student:</label>
25 <input type="radio" name="zvanje" value="Student"/>
26 </fieldset>
27
28 <fieldset>
29 <legend>Podaci o radu</legend>
30 <input class="cek" type="checkbox" name="Strad" value="Strad" checked />
31 <label class="cek">Stampani rad</label>
32 <input class="cek" type="checkbox" name="CD" value="CD"/>
33 <label class="cek">CD</label>
34 <input class="cek" type="checkbox" name="Slajdovi" value="Slajdovi"/>
35 <label class="cek">Slajdovi</label>
36 <label>Predmet:</label>
37 <select name="Predmet">
38 <option value="Matematika" selected>Matematika</option>
39 <option value="Fizika">Fizika</option>
40 <option value="Hemija">Hemija</option>
41 </select>
42 <label>Opis rada:</label>
43 <textarea name="opisRada"></textarea>
44 </fieldset>
45
46 <label id="labela">Lozinka:</label>
47 <input id="lozinka" type="password" name="Lozinka"/>
48 <input type="submit" value="Prijavi"/>
49 <input type="reset" value="Poništi"/>
50 </form>
51 </body>

```

Dok u browseru ceo dokument izgleda ovako:

[CSS deo vežbe]

Prepišite stilove koji slede u CSS dokument i nakon svakog dodatog stila, sačuvajte CSS fajl i osvežite HTML dokument u browseru da biste uočili nastale promene.

Prelazimo na stilizovanje dokumenta.

1. Bojimo pozadinu **tela** dokumenta u tamno-sivu boju

```
1  body
2  {
3      background-color: #555555;
4  }
```

2. Podešavamo širinu cele **forme** na 300 piksela i **margin** na *auto* kako bismo centrirali formu unutar **body** elementa

```
6  form
7  {
8      width: 300px;
9      margin:auto;
10 }
```

3. **Fieldsetovima** podešavamo pozadinsku boju na svetlo-sivu i isključujemo im ivice

```
12 fieldset
13 {
14     background-color: #EEEEEE;
15     border: none;
16 }
```

4. **Legendama** podešavamo boju fonta na svetlo-sivu (skoro bela boja) i podešavamo **padding** donje strane na 20 piksela kako bismo odvojili **legendu** od samog **fieldseta** (podizemo je)

```
18 legend
19 {
20     color: #DDDDDD;
21     padding-bottom: 20px;
22 }
```

5. **Labele** koje su klase **.ceored** podešavamo tako plutaju ulevo i širinu podešavamo tako da zauzmu celu širinu **fieldseta** u svom redu.

```
24 .ceored
25 {
26     float:left;
27     width:100%;
28 }
```

6. Polja za unos teksta koja su klase **.tekst** podešavamo tako da takođe plutaju ulevo, međutim, kako labela zauzimaju kompletan prethodni red ova tekstualna polja su prisiljena da se smeste ispod svake labela pojedinačno jer za njih nema mesta u prethodnom redu.

Širinu polja podešavamo na 80% širine **fieldseta** a desnu marginu podešavamo na 20% širine **fieldseta** kako

bismo postigli takav efekat da tekstualno polje zauzme ceo red ali smo prostor koji zauzima ovaj element podelili na 80% prostora u korist samog tekstualnog polja i 20% smo dodelili njegovoj desnoj margini koja je tu samo da zauzme prostor do kraja **fieldseta** i ne dozvoli nekom drugom elementu da se smesti iza tekstualnog polja već da tu ostane prazan prostor.

```

29  .tekst
30  {
31      float:left;
32      width: 80%;
33      margin-right:20%;
34  }
```

7. Klasi **.cek** koja je dodeljena svim čekboksovima i njihovim labelama rekli smo da plutaju ulevo i dodelili desnu marginu širine 10 piksela svima da bismo ih međusobno razdvojili. Takođe smo svima dodelili i marginu na dnu širine 10 piksela kako bismo napravili malo prostora između čekboksova i elemenata koji slede posle njih.

```

36  .cek
37  {
38      float:left;
39      margin-right: 10px;
40      margin-bottom: 10px;
41  }
```

8. Elementu **select**, odnosno padajućoj listi smo širinu podesili na kompletnu širinu **fieldseta** tako da ona zauzima ceo red. Njena labela se sama smešta u prostor između čekboksova i padajuće liste pošto se u HTML dokumentu strukturno nalazi pre padajuće liste (odnosno iznad nje).

```

43  select
44  {
45      width:100%;
46      margin-bottom: 10px;
47  }
```

9. Elementu **textarea** podešavamo širinu na 100% širine **fieldseta** u kome se nalazi i visinu na 100 piksela.

```

49  textarea
50  {
51      width: 100%;
52      height:100px;
53  }
```

10. Identifikatoru **#labela** koji pripada labeli koja stoji uz polje za unos šifre smo dodelili boju teksta **#DDDDDD** (svetlo-siva).

```

60  #labela
61  {
62      color: #DDDDDD;
63  }
```

11. Identifikatoru `#lozinka` koji je dodeljen polju za unos šifre smo podesili širinu na 118 piksela metodom pokušaja i pogrešaka kako bi se polje najbolje uklopilo u formu.

```

55     #lozinka
56     {
57         width: 118px;
58     }

```

12. Dugmad sama okupiraju preostali prostor u redu u kome se nalaze.

Kompletan CSS dokument na kraju izgleda ovako:

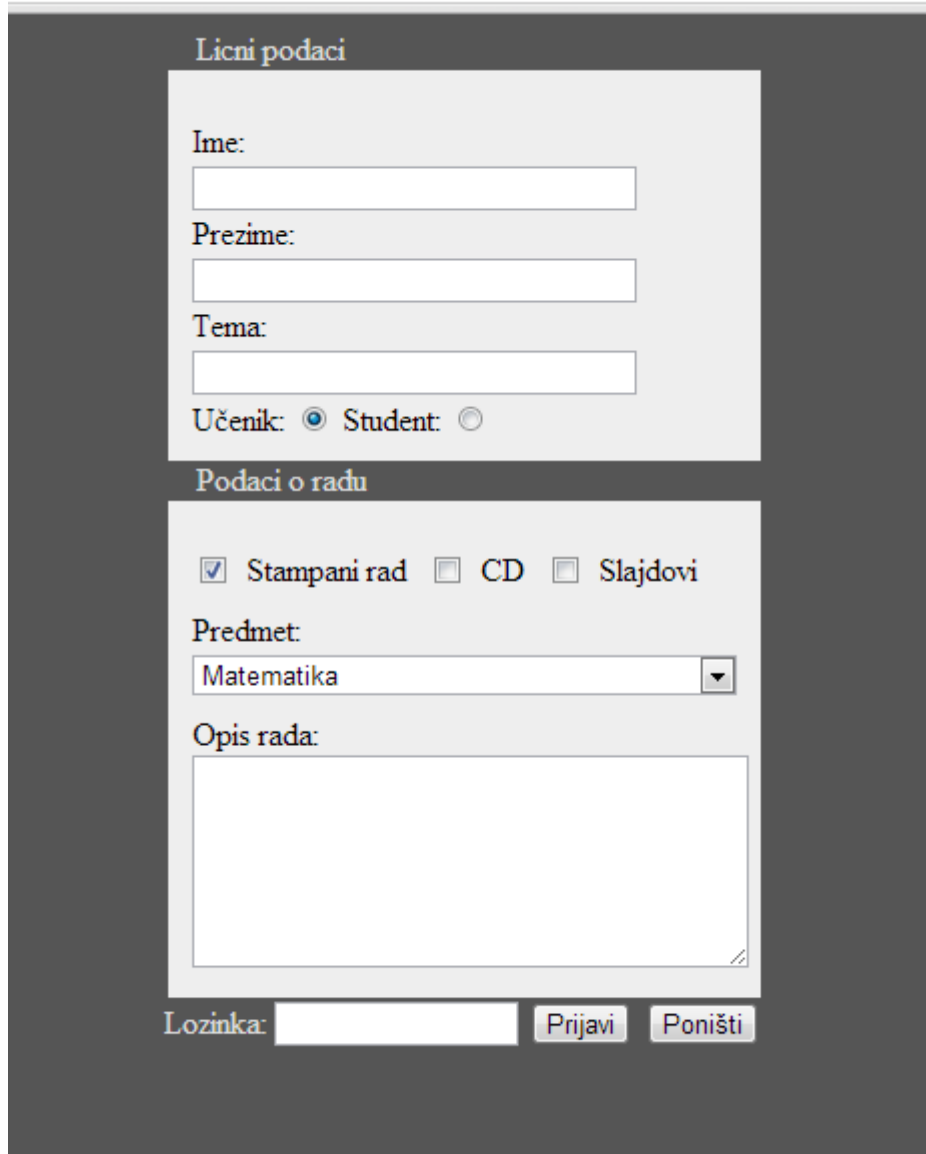
```

1     body
2     {
3         background-color: #555555;
4     }
5
6     form
7     {
8         width: 300px;
9         margin:auto;
10    }
11
12    fieldset
13    {
14        background-color: #EEEEEE;
15        border: none;
16    }
17
18    legend
19    {
20        color: #DDDDDD;
21        padding-bottom: 20px;
22    }
23
24    .ceored
25    {
26        float:left;
27        width:100%;
28    }
29    .tekst
30    {
31        float:left;
32        width: 80%;
33        margin-right:20%;
34    }
35
36    .cek
37    {
38        float:left;
39        margin-right: 10px;
40        margin-bottom: 10px;
41    }
42
43    select
44    {
45        width:100%;
46        margin-bottom: 10px;
47    }
48
49    textarea
50    {
51        width: 100%;
52        height:100px;
53    }
54
55    #lozinka
56    {
57        width: 118px;
58    }
59
60    #labela
61    {
62        color: #DDDDDD;
63    }

```

Dokument nakon stilizovanja u browseru izgleda kao na slici (dokument je centriran, nalazi se na sredini tela dokumenta tj. sredini strane):

%20Tehnicka%20skola,%202013/Vezbe/Vezba07%20-%20forme/\



Licni podaci

Ime:

Prezime:

Tema:

Učenik: Student:

Podaci o radu

Stampani rad CD Slajdovi

Predmet:

Opis rada:

Lozinka:

Napomena: u pregledaču Internet Exploreru će forma izgledati malo drugačije, smanjiti width na 100px za #lozinka identifikator. Isto važi i za Mozilla Firefox.

Frejmovi u HTML-u

Kao i uvek pre svega napravite kopiju fajla "Vezba01.html" i nazovite ga "Vezba08.html" kako ne biste morali da iznova pišete kostur HTML dokumenta za ovu vežbu. Takođe kreirajte jedan prazan CSS dokument pod nazivom "sti/08.css" koji ćemo koristiti za stilizovanje dokumenta na kome budemo radili u ovoj vežbi. Odmah linkujte HTML dokument sa CSS dokumentom korišćenjem elementa **link**.

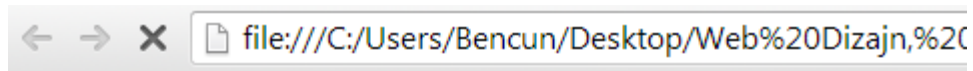
[HTML deo vežbe]

Frejmovi se u HTML-u koriste za učitavanje drugih HTML strana unutar postojeće strane. Za kreiranje frejmova koristi se element `<iframe>...</iframe>` pri čemu se u atribut **src** smešta lokacija HTML dokumenta koji želimo da učitamo a između tagova se smešta alternativni sadržaj koji se prikazuje u slučaju da element **iframe** nije podržan od strane browsera. Napravićemo jedan prost primer sa dva frejma od kojih jedan učitava stranu koju smo sami napravili u prethodnoj vežbi a drugi učitava (ukoliko je prisutna Internet konekcija) početnu stranu sajta W3Schools.

HTML dokument izgleda ovako:

```
12 <body>
13   <h1>Prvi frejm <i>(Vezba07.html)</i></h1>
14   <iframe src="Vezba07.html" id="prvi"><p>Iframe nije dostupan</p></iframe>
15   <h1>Drugi frejm <i>(www.w3schools.com)</i></h1>
16   <iframe src="http://www.w3schools.com/" id="drugi"><p>Iframe nije dostupan</p></iframe>
17 </body>
18
```

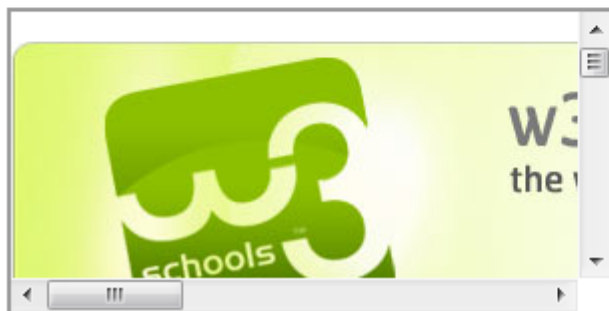
U browseru izgleda ovako:



Prvi frejm (*Vezba07.html*)



Drugi frejm (*www.w3schools.com*)

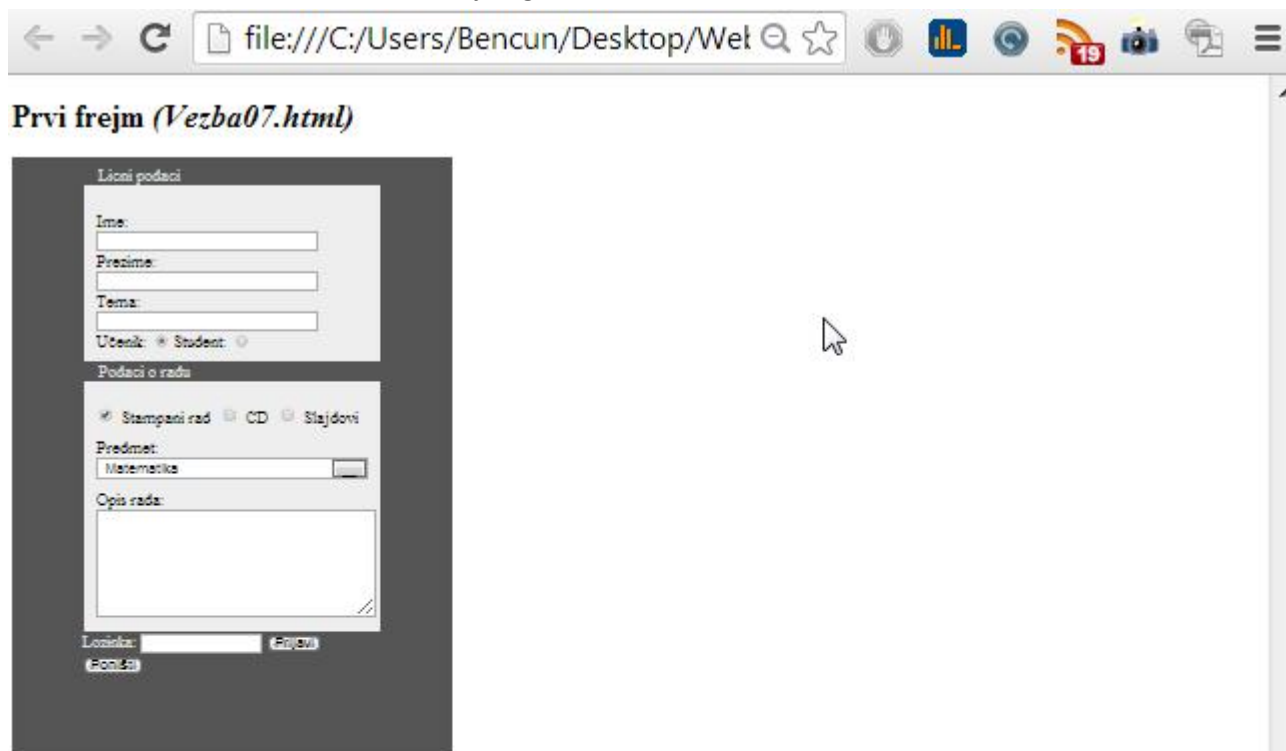


[CSS deo vežbe]

A CSS dokument kojim smo podesili širinu i visinu frejmova i uklonili ivice prvog frejma izgleda ovako:

```
1 #prvi
2 {
3     width: 440px;
4     height: 600px;
5     border: none;
6 }
7
8 #drugi
9 {
10    width: 100%;
11    height: 400px;
12 }
```

U browseru dokument nakon stilizovanja izgleda ovako:



Drugi frejm (www.w3schools.com)



Prekucajte HTML dokument, sačuvajte ga i otvorite u browseru da biste videli kako izgleda bez stilova. Zatim prekucajte i CSS stilove, sačuvajte dokument i ponovo učitajte HTML dokument u browser da vidite kako izgleda sa stilovima.

Upotreba WYSIWYG alata za izradu Web sajtova

Za izradu Web sajtova se takođe jako često upotrebljavaju WYSIWYG - What You See Is What You Get alati koji omogućavaju vizuelno dizajniranje sajtova u grafičkom okruženju bez pisanja koda gde program vrši svo prevođenje vizuelnog dizajna u HTML/CSS/JS kod. Dizajniranje sajta se vrši ubacivanjem potrebnih elemenata na stranu kroz intuitivni grafički interfejs, a vizuelni dizajn i funkcionalni i vizuelni atributi su isto tako lako dostupni i menjaju se kroz interfejs programa. Prednost kodiranja sajtova direktnom upotrebom HTML i CSS tehnologija je daleko veća kontrola nad izgledom i sadržajem u odnosu na upotrebu WYSIWYG alata. Čak i ako se dizajner odluči za upotrebu nekog od ovih alata poznavanje spomenutih jezika je apsolutni imperativ jer je praktično uvek neophodno zaviriti u kod kada dođe do problema pri dizajniranju sajta.

NAPOMENA: Vežba koja sledi preuzeta je sa sajta Elektronskog fakulteta u Nišu (laboratorijska vežba iz predmeta Uvod u računarstvo, autor Mr Bratislav Predić), i predviđena je za rad kod kuće. Ova vežba nije predviđena za štampanje i moguće joj je pristupiti samo elektronski i to duplim klikom na ikonicu ispod ovog pasusa.



Adobe
Dreamweaver.doc

Sistemi za upravljanje sadržajem - CMS

CMS ili *Content Management System* predstavlja softver koji omogućava lak razvoj, kreiranje i održavanje Web sajtova bez poznavanja bilo koje od Web tehnologija koje se koriste u razvijanju sajtova i aplikacija. CMS-ovi su najčešće pisani u nekom od popularnih skripting jezika kao što su PHP i Perl što automatski znači da se oni izvršavaju na serveru. Najpribližniji opis CMS-ova bi bio da su oni samoodrživi Web sajtovi za čiju izradu i održavanje nije potreban neki poseban nivo naprednog znanja.

Što se tiče popularnijih CMS-ova tu su Joomla, WordPress, Drupal, DotNetNuke, TinyCMS i mnogi drugi.

Kako ovi sistemi predstavljaju velike i kompleksne Web aplikacije mi ćemo se pozabaviti samo instalacijom i osnovnom konfiguracijom jednog od njih a to je WordPress.

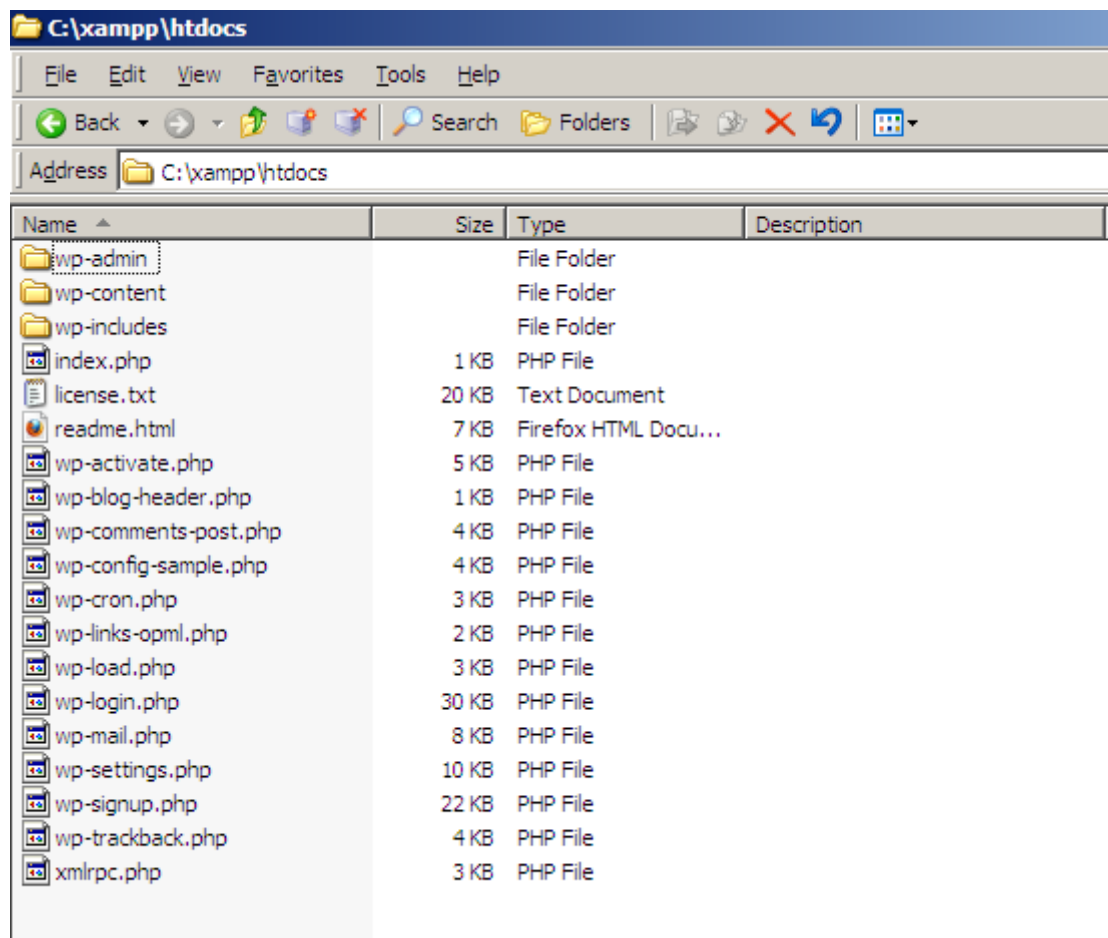
Instalacija WordPressa

WordPress je trenutno najpopularniji CMS na tržištu, koristi open-source licencu, ima ogromnu zajednicu, veliki broj tema i dodataka i konstantno napreduje. Pogodan je za kreiranje blogova i manjih do srednjih Web sajtova mada se uz prilagođavanje može koristiti za virtuelno bilo kakav Web sajt.

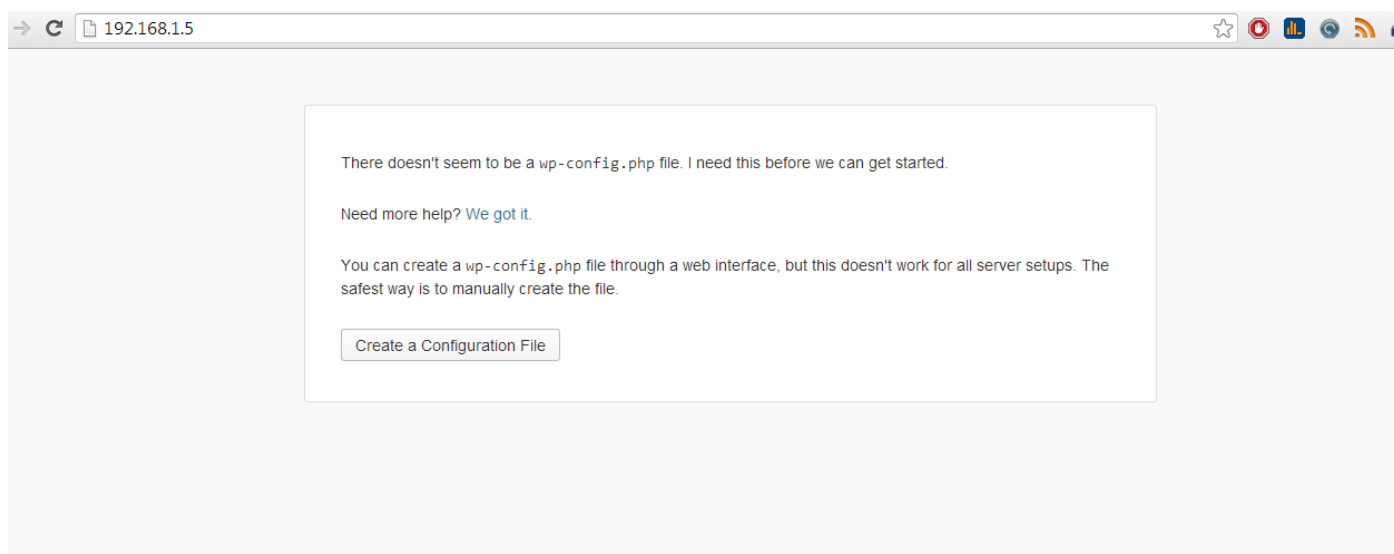
Za početak potrebno je iznajmiti hosting prostor i domen. U ovoj vežbi mi smo radi demonstracije kreirali sopstveni Web server sa PHP-om kao skripting jezikom (u kome je WP napisan) i MySQL-om kao sistemom za upravljanje bazama podataka. Kada zakupite hosting dobićete korisničko ime i šifru za pristup fajlovima na serveru preko FTP protokola i korisničko ime i šifru za pristup vašoj bazi podataka kao i ime baze podataka. U datom primeru mi smo sami kreirali bazu i korisničko ime i šifru za pristup istoj no to nije predmet ove vežbe.

Pre svega je potrebno preuzeti najnoviji WordPress sa zvaničnog Web sajta ovog CMS-a: <http://wordpress.org/>

Nakon što preuzmete arhivu potrebno je raspakovati i postaviti na Web server putem FTP transfera fajlova. Sajt se na Web server (u ovom slučaju naš sajt je WP a ne nešto što smo sami napravili) postavlja u folder koji se najčešće zove "*public_html*", "*www*" ili "*htdocs*". Ovi folderi su jedini dostupni javnosti odnosno to su jedini folderi kome posetioци imaju pristup. Početna strana sajta najčešće ima naziv "*index.html*" ili "*index.php*". Evo kako izgleda raspakovana instalacija WP-a u folderu "*htdocs*" na našem serveru:



Nakon što smo postavili fajlove na server dovoljno je da pristupimo domenu našeg Web sajta sa bilo koje lokacije da bismo započeli instalaciju. Nakon što pristupimo našem sajtu (u konkretnom primeru smo koristili IP adresu a ne domen jer je ova vežba rađena u lokalnoj mreži a ne na Internetu) dočekuje nas ovakva početna strana koja nas informiše da konfiguracioni fajl WP-a nije pronađen i da možemo krenuti sa instalacijom:



Nakon klika na dugme "Create a configuration file" dočekuje nas prvi ekran gde počinje instalacija WordPressa. Dovoljno je kliknuti "Let's Go!" za nastavak instalacije:

nin/setup-config.php



Welcome to WordPress. Before getting started, we need some information on the database. You will need to know the following items before proceeding.

1. Database name
2. Database username
3. Database password
4. Database host
5. Table prefix (if you want to run more than one WordPress in a single database)

If for any reason this automatic file creation doesn't work, don't worry. All this does is fill in the database information to a configuration file. You may also simply open `wp-config-sample.php` in a text editor, fill in your information, and save it as `wp-config.php`.

In all likelihood, these items were supplied to you by your Web Host. If you do not have this information, then you will need to contact them before you can continue. If you're all ready...

Let's go!

Na sledećem ekranu potrebno je uneti ime baze podataka, korisničko ime za pristup istoj, šifru, server na kome se nalazi baza (*localhost* je podrazumevano i najčešće podešavanje) dok zadnje polje ne treba dirati osim ako ne instaliramo WP više puta i sve instalacije koriste istu bazu podataka.



Below you should enter your database connection details. If you're not sure about these, contact your host.

Database Name	<input type="text" value="korisnik"/>	The name of the database you want to run WP in.
User Name	<input type="text" value="korisnik"/>	Your MySQL username
Password	<input type="text" value="sifra"/>	...and your MySQL password.
Database Host	<input type="text" value="localhost"/>	You should be able to get this info from your web host, if localhost does not work.
Table Prefix	<input type="text" value="wp_"/>	If you want to run multiple WordPress installations in a single database, change this.



Sledeći ekran nas obaveštava da smo prošli kroz osnovnu konfiguraciju i da je sada potrebno instalirati WordPress. Kliknućemo na dugme "Run the install":



All right, sparky! You've made it through this part of the installation. WordPress can now communicate with your database. If you are ready, time now to...

Ovde je potrebno uneti naziv Web sajta i korisničko ime i šifru koje želimo da dodelimo administratorskom nalogu (ono nema veze sa korisničkim imenom za naš hosting). Takođe treba uneti e-mail adresu za administratora. Štiklirati poslednje polje jer ono omogućava da pretraživači indeksiraju naš Web sajt.

Information needed

Please provide the following information. Don't worry, you can always change these settings later.

Site Title

Username
Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods and the @ symbol.

Password, twice
A password will be automatically generated for you if you leave this blank.

Strength indicator

Hint: The password should be at least seven characters long. To make it stronger, use upper and lower case letters, numbers and symbols like ! " ? \$ % ^ &).

Your E-mail
Double-check your email address before continuing.

Privacy Allow search engines to index this site.

Sledeći ekran predstavlja završetak instalacije i dovoljno je kliknuti "Log in" kako bismo dobili pristup našem Web sajtu i počeli da dodajemo sadržaj.



Success!

WordPress has been installed. Were you expecting more steps? Sorry to disappoint.

Username admin

Password *Your chosen password.*

Log In



Username

admin

Password

Remember Me

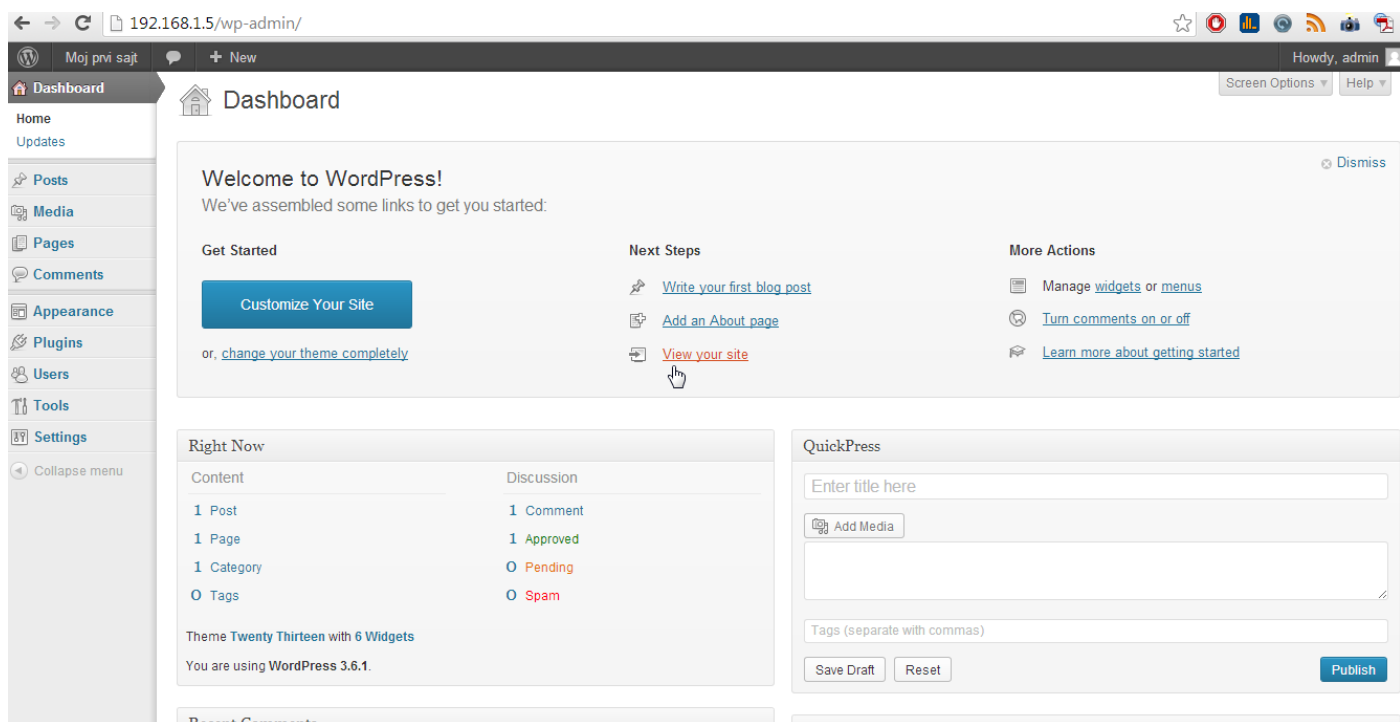
Log In

[Lost your password?](#)

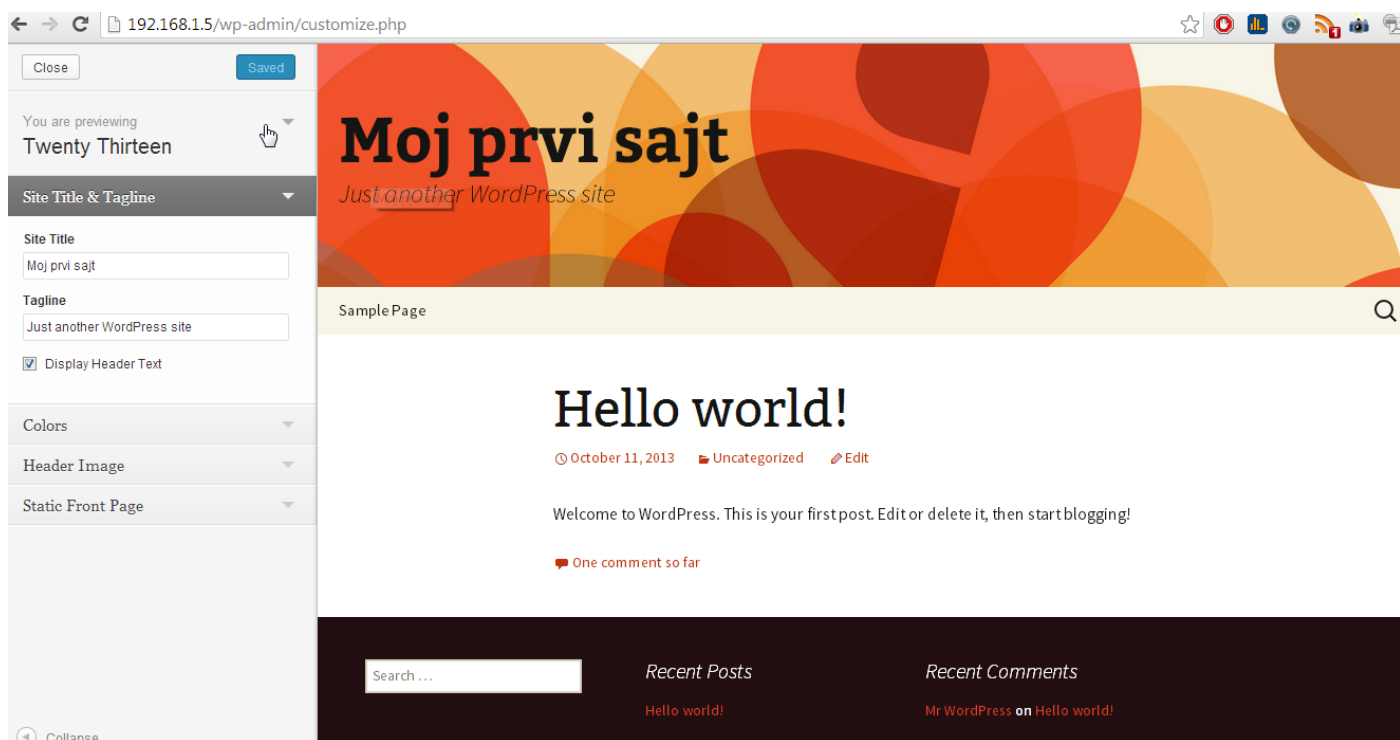
[← Back to Moj prvi sajt](#)

Primena računara u elektrotehnici, Web dizajn

Nakon logovanja dočekuje nas interfejs administratorskog panela koji omogućava menadžment Web sajta. Pored ovog administratorskog naloga moguće je kreirati naloga za obične korisnike koji mogu da dodaju sadržaj na sajt ali ne mogu da menjaju formu i strukturu sajta.



Sljedeći skrinšot predstavlja deo konfiguracionog panela koji omogućava modifikovanje vizuelnih elemenata sajta.



Sljedeći skrinšot pokazuje mogućnost promene vizuelne teme Web sajta.

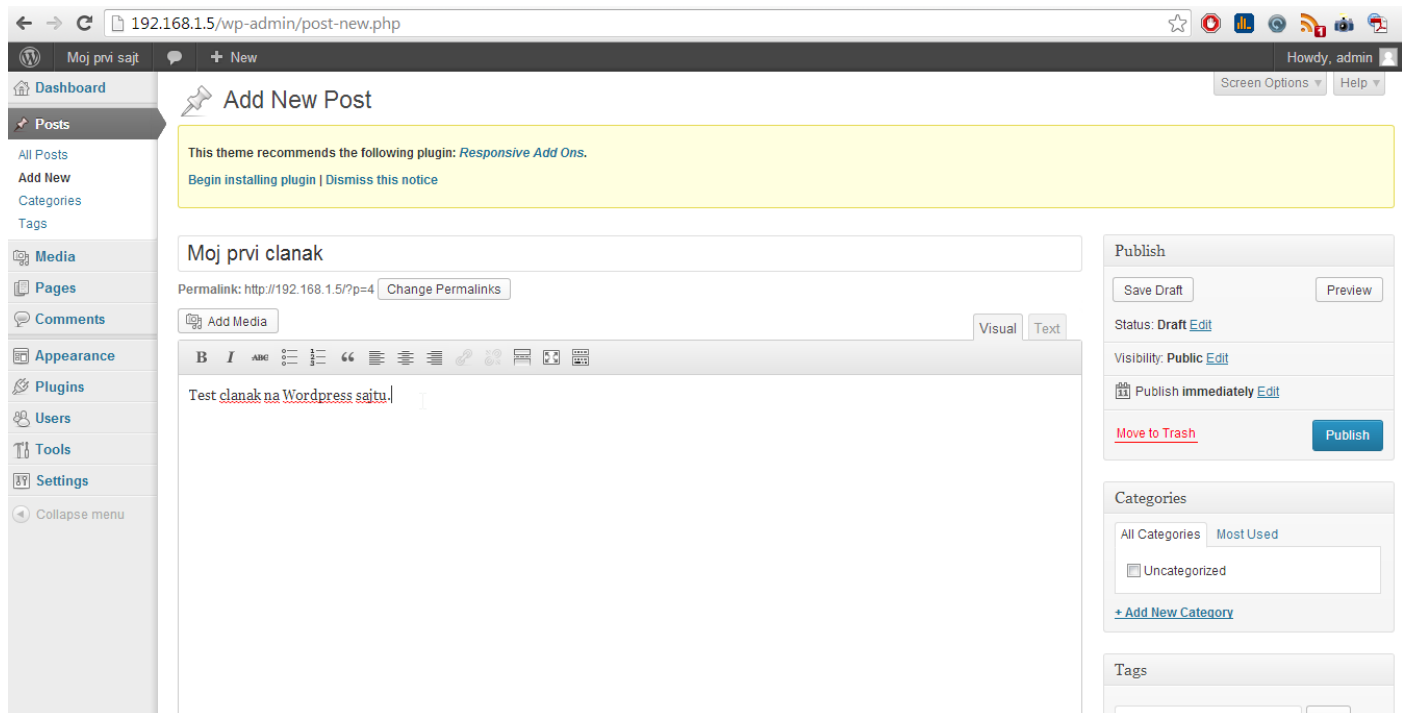
Primena računara u elektrotehnici, Web dizajn

The screenshot shows the WordPress Admin interface at the URL `192.168.1.5/wp-admin/themes.php?activated=true`. The left sidebar contains navigation menus for Dashboard, Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. The main content area is titled 'Manage Themes' and shows the 'Responsive' theme is currently active. A yellow notification banner at the top states: 'This theme recommends the following plugin: Responsive Add Ons. Begin installing plugin | Dismiss this notice'. Below this, another banner says 'New theme activated. Visit site'. The 'Current Theme' section displays 'Responsive' by CyberChimps.com, version 1.9.3.8. A description of the theme follows, highlighting its fluid grid system and various features like 9 page templates and 11 widget areas. At the bottom, there are links for 'Customize' and 'Options' (Widgets, Menus, Theme Options, CyberChimps Themes, Activate Add Ons, Header, Background). The 'Available Themes' section at the bottom shows thumbnails for 'Twenty Thirteen' and 'Twenty Twelve'.

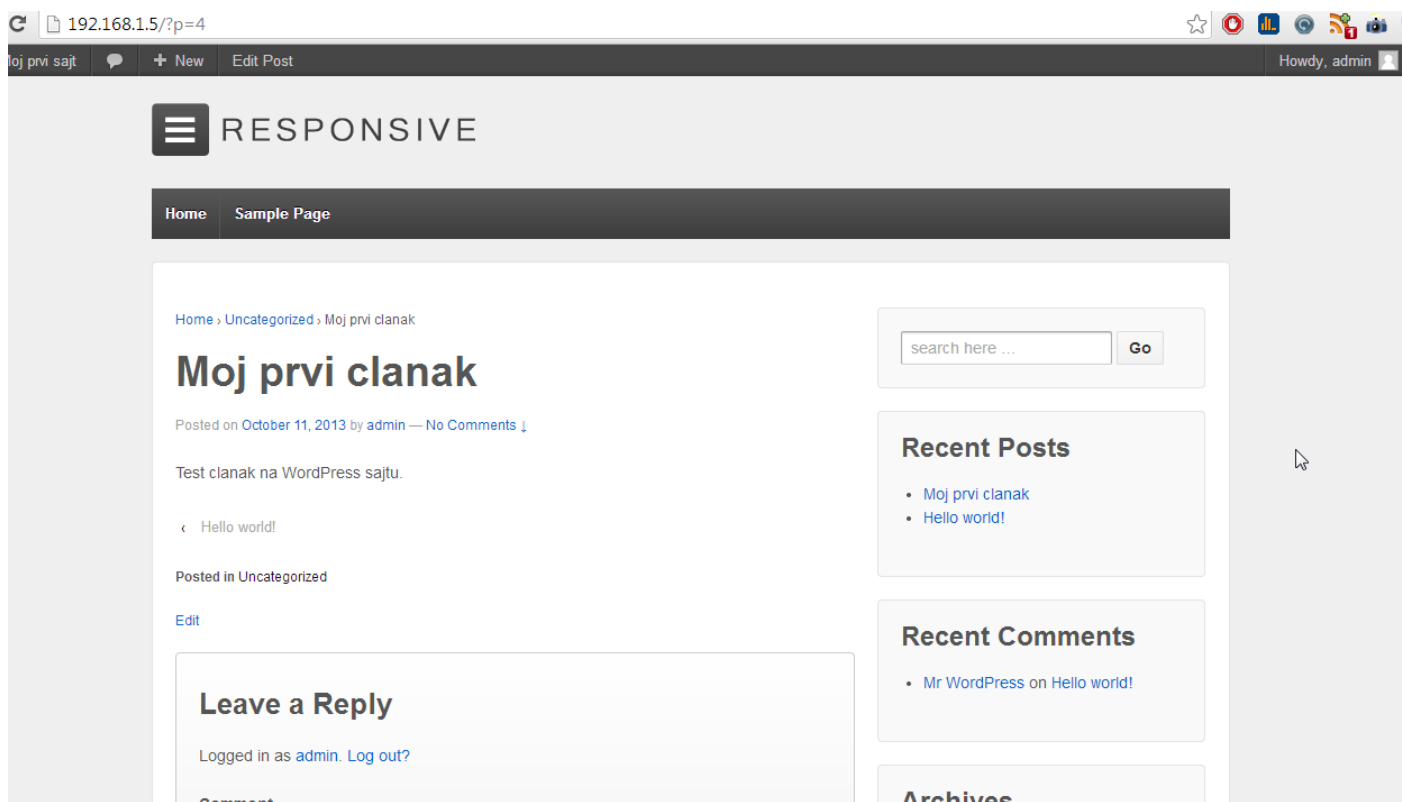
Sledeći skinšot prikazuje mogućnost dodavanja i uklanjanja widgeta na Web sajtu:

The screenshot shows the WordPress Admin 'Widgets' page. The left sidebar is the same as in the previous image. The main content area is titled 'Widgets' and features the same yellow notification banner. Below the banner, there are two main sections: 'Available Widgets' and 'Inactive Widgets'. The 'Available Widgets' section contains a grid of widget boxes, each with a title and a brief description. A mouse cursor is hovering over the 'Recent Posts' widget. The 'Inactive Widgets' section is currently empty. On the right side of the page, there are three sidebar areas: 'Main Sidebar', 'Right Sidebar', and 'Left Sidebar'. The 'Main Sidebar' area is titled 'Area 1 - sidebar.php' and contains a list of widgets that are currently active on the site, including Search, Recent Posts, Recent Comments, Archives, Categories, and Meta.

Skrinšot ispod prikazuje formu za unos sadržaja na sajt odnosno kreiranje tzv. postova:



Evo kako izgleda naš sajt nakon izmene teme u temu "Responsive" i dodavanja jednog *posta* na sajt:



Korišćena literatura

1. Josh Hill, James A. Brannen: *Briljantno HTML5 i CSS3*, CET Computer Equipment and Trade, Beograd, 2011.
2. Web sajt: <http://www.w3schools.com/>
3. Razni izvori informacija širom Interneta