



VEB PROGRAMIRANJE

priručnik za treći i četvrti razred

radna verzija

UVOD	4
INTERNET	5
Prednosti interneta	5
Mane interneta	5
Svetska komunikaciona mreža	5
Veb pregledač (brauzer, engl. <i>web browser</i>)	5
Elektronsko poslovanje	6
VEB SAJT	7
Statički veb sajt	7
Dinamički veb sajt	7
VEB SERVER	8
Domain name server	8
Proxy server	8
FTP server	8
SKRIPT JEZICI	9
Serverski skript jezici	9
Klijentski skript jezici	9
HTML.....	10
Tagovi	10
Forme za unos i izbor	11
CSS	13
Selektori i svojstva	13
Pozicioniranje.....	15
Animacije	15
Fontovi	16
JAVASCRIPT	17
Osnove sintakse	17
Gde možemo pisati JavaScript kod?	17
Promenljive	17
Razlika između var i let.....	17
Konstanta – const.....	18
Tipovi promenljivih.....	18
Konverzija stringa u broj.....	18
Funkcije	18
Naredbe i petlje.....	19
Pristup html elementima	19
document.getElementById(id).....	19
document.getElementsByTagName(name)	19
document.getElementsByClassName(name)	19
Manipulisanje html elementima	19

element.innerHTML	19
element.atribut.....	20
element.style.svojstvo	20
Manipulisanje prozorom pretraživača	20
window.open.....	20
Manipulisanje zvukom	20
Upravljanje događajima	21
onclick	21
onmouseover	21
onchange	21
setTimeout i setInterval.....	22
Objekti	22
STRUKTURA FOLDERA SA DOKUMENTIMA SAJTA	23
VALIDATORI KODA	24
HTML.....	24
CSS.....	25
JavaScript	26

UVOD

Ovaj dokument za svrhu ima da pomogne pri savladavanju gradiva tokom izučavanja predmeta *veb programiranje* u trećem i četvrtom razredu smera *Elektrotehničar informacionih tehnologija*.

Trenutna verzija nije konačna i biće naknadno ispravljana i dopunjavana. Spisak izmena i dopuna tokom školske 2023/24. godine će se nalaziti na ovoj stranici.

INTERNET

Internet je svetski sistem umreženih računarskih mreža koji povezuje milijarde računara širom sveta na jedan nehijerarhijski način (što znači da nema glavnog računara od kog zavisi ceo internet). Za povezivanje računara i računarskih mreža koristi internet protokol (IP). Internet čini njena hardverska komponenta kao i sistem softverskih slojeva koji kontrolišu različite aspekte njene komunikacione infrastrukture.

Prednosti interneta

- Povezanost / komunikacija / deljenje – Mogućnost brze razmene poruka ili video poziva sa osobama na najrazličitijim mestima na planeti, kao i deljenje različitih tipova datoteka sa njima.
- Informacije / znanje / učenje – Dostupnost ogromnog broja informacija, što doprinosi mogućnostima usavršavanja i napredovanja u različitim disciplinama i veštinama.
- Trgovina / marketing – Pojavom interneta proširile su se mogućnosti za trgovinu s obzirom da kupac ne mora da bude fizički prisutan, a takođe su se proširile i mogućnosti reklamiranja sopstvenog biznisa ili proizvoda.
- Zabava – Pristup internetu pruža i mogućnost pristupa ogromnom broju zabavnog materijala, muzike, filmova, serija, knjiga...
- Rad od kuće – Zahvaljujući rasprostranjenosti interneta sve veći broj poslova se može obavljati i od kuće što otvara brojne šanse za pronalazak posla bez obzira na trenutnu lokaciju.

Mane interneta

- Zavisnost – Usled ogromnog sadržaja koji pruža i dostupnosti na svakom uređaju izuzetno je lako postati zavisan od korišćenja interneta što dalje može dovesti do različitih fizičkih i psihičkih problema.
- Sajber maltretiranje – Široka dostupnost i sloboda interneta omogućila je i da lažno predstavljanje, slanje neželjenih poruka, pretnji i ucenjivanja bude problem korisnicima.
- Sajber kriminal – Ostavljanjem svih svojih informacija na internetu otvara se mogućnost da do tih podataka dođu i oni koji imaju nameru da ih iskoriste u svrhe kriminalnih radnji.
- Virusi – U slučaju nepažljivog korišćenja interneta postoji mogućnost da korisnik na svoj računar preuzme neki od štetnih fajlova koji zatim dovode u opasnost sve podatke koji se na računaru nalaze.

Svetska komunikaciona mreža (engl. World Wide Web) poznata i samo kao veb (engl. the Web) je sistem međusobno povezanih hipertekstualnih dokumenata koji se nalaze na internetu. Veb je međunarodna mreža računarskih baza podataka koje koriste internet i njegov poseban sistem za povezivanje informacija. Veb i internet nisu isto, veb predstavlja samo jedan od servisa koje internet pruža.

Veb se sastoji od :

- URL (engl. Uniform Resource Locator) – poznat i kao veb adresa, odnosi se na http ili https URL šemu, tj. na složen iskaz nizova karaktera koji se koristi za lociranje nekog resursa na internetu.
- HTTP (engl. HyperText Transfer Protocol) – protokol kojim se određuju načini na koje server i brauzer međusobno komuniciraju.
- HTML (engl. HyperText Markup Language) – pomoću kog se šifrira poruka tako da može da se prikaže na različitim medijima.

S tim u vidu WWW ne predstavlja celokupan sadržaj na internetu, primeri upotrebe interneta nevezane za veb su razne striming platforme, bitcoin, torrent...

Veb pregledač (brauzer, engl. web browser)

Veb pregledač je alat koji korisniku omogućava pregledanje veb stranica i multimedijalnih sadržaja vezanih za iste. Služi za pronalaženje, pregledanje, preuzimanje i predstavljanje podataka na vebu, čije je odredište određeno veb adresom (URL-om), a osim same veb stranice može biti i slika, multimedijalnog sadržaja ili neke druge vrste datoteke (fajla). Iako su pregledači prvenstveno namenjeni za korišćenje veba, oni se mogu koristiti i za pristupanje podacima veb servera, privatnih mreža ili datotekama baza podataka. Primeri veb pregledača, odnosno brauzera, jesu Google Chrome, Mozilla Firefox, Opera, Microsoft Edge...

Elektronsko poslovanje

Elektronsko poslovanje jeste vođenje poslova na internetu, što ne podrazumeva samo kupovinu i prodaju, već organizaciju poslovanja firme u mrežnom okruženju, organizovanje komunikacije prema klijentima i brigu o klijentima.

Osnovni oblici su :

- e-trgovina – predstavlja poslovnu komunikaciju i prenos dobara i usluga (kupovina i prodaja)
- e-bankarstvo – predstavlja poslovanje na relaciji banka-klijent.
- e-uprava – predstavlja elektronsko poslovanje u upravi ili administraciji (države, preduzeća...)

Elektronsko poslovanje ima sledeće prednosti :

- smanjenje troškova poslovanja i to prevashodno vezane za izradu papirnih dokumenata,
- smanjenje grešaka, pogotovu gde je tačnost informacija od značaja,
- ušteda vremena, posebno u prenosu informacija,
- smanjenje obima ljudskog rada,
- pristupačnost i razmenljivost informacija.

VEB SAJT

Veb sajt, poznat i kao veb lokacija, predstavlja mesto na internetu, skup veb stranica koje mogu da sadrže tekst, slike, video snimke i drugi multimedijalan sadržaj sastavljen u jednu celinu. Sajt može biti smešten na lokalnom računaru ili na internet ili veb serveru. Neki sajtovi su dostupni svim korisnicima, a neki imaju ograničenja u pogledu pristupa. Veb sajtovi imaju dosta funkcija i mogu biti korišćeni u različitim prilikama, veb sajt može biti lični, komercijalni, neprofitne organizacije...

Statički veb sajt

Statički veb sajtovi su oni koji sadrže neke informacije koje se ne menjaju i ne osvežavaju na duže vreme (sajtovi sa arhivom dokumenata, prezentacije poslovanja samo sa osnovnim podacima...). Oni se korisniku prikazuju onako kako su kreirani i smešteni na veb server, nema personalizovanog sadržaja, mogućnosti ulogovanja ili kupovine i slično.

Redosled izvršavanja akcija kod statičkog veb sajta :

1. Autor je kreirao stranu koja se sastoji od HTML koda i stavio je na veb server.
2. Korisnik zahteva da vidi veb stranu (klikom na link, ukucavanjem adrese u liniju veb pretraživača i slično).
3. Veb server pronalazi HTML stranu koju je korisnik zahtevao.
4. Veb server šalje pronađenu stranu klijentu – veb pretraživaču.
5. Veb pretraživač obrađuje dobijeni HTML kod i prikazuje korisniku uredno formatiranu stranicu sa svim elementima (slikama, linkovima, tabelama...).

Dinamički veb sajt

Dinamički veb sajtovi su oni čiji sadržaj se često osvežava i dopunjuje novim informacijama (vesti, sportski rezultati, oglasi...). Baziraju se na bazama podataka jer se sadržaj dinamičkog veb sajta čuva u bazi podataka sa kojom server komunicira pre nego što prosledi veb stranicu korisniku. Ovo omogućuje da korisnik dobije personalizovani sadržaj u zavisnosti od većeg broja mogućih faktora kao što su npr. koje je doba dana, gde se korisnik nalazi, koji veb pretraživač koristi, informacije koje se nalaze na njegovom privatnom profilu tog veb sajta i slično. Stranice se izvršavaju na veb serveru a rezultat ovog izvršavanja predstavlja HTML kod.

Redosled izvršavanja akcija kod dinamičkog veb sajta :

1. Autor je kreirao stranu koja se sastoji od serverskih kontrola i instrukcija u nekom programskom jeziku i stavio je na veb server.
2. Korisnik kreira zahtev za pregled dinamičke veb strane. Zahtev se prosleđuje od klijenta (veb pretraživača) do veb servera na kome se nalazi zahtevana strana.
3. Veb server obrađuje zahtev korisnika i pronalazi dinamičku stranu koju je korisnik zahtevao.
4. Veb server izvršava programski kod koji je sastavni deo strane i kreira HTML kod.
5. Veb server šalje preko Interneta generisani HTML kod veb pretraživaču.
6. Veb pretraživač obrađuje dobijeni HTML kod i prikazuje korisniku stranicu sa svim elementima.

VEB SERVER

Veb server predstavlja skup programa koji korisnicima omogućavaju pristup informacijama i eventualno njihovu izmenu, dok administratorima omogućavaju dodavanje, brisanje i izmenu datoteka. Termin veb server se može odnositi na kompletan sistem (računar) ili specifičnije na softver koji prihvata i obrađuje HTTP zahteve. Dok god nema zahteva veb server miruje i praktično ne radi ništa. Tek po dobijanju zahteva veb server ga preuzima i obrađuje, a zatim po potrebi daje povratne informacije korisniku koji je zahtev prosledio. Da bi veb server bio svrsishodan mora imati neprekidan pristup internetu.

Primarna funkcija veb servera je da čuva, obrađuje i isporučuje veb stranice korisnicima. Funkcija veb servera je i da prima sadržaj od strane korisnika, na ovaj način se izvodi prihvatanje podataka na serveru putem veb formulara i postavljanje datoteka na veb.

Neki od specijalizovanih programa sa funkcijom veb servera su : Microsoft Internet Information Services, Apache Web Server, Nginx, Sun Java System Web Server...

Domain name server

DNS server je sistem koji pretvara imena domena u IP adrese. Naime, reč je o distribuiranom hijerarhijskom sastavu Internet poslužitelja u kojem se nalaze informacije povezane sa domenskim nazivima. Može se reći da je DNS server jedna baza podataka u kojoj su pisana sva imena i IP adrese pojedinih računara, kao i grupe funkcija koje omogućavaju prevođenje istih.

Kada u vaš pretraživač ukucate veb adresu, vaš računar će uz pomoć DNS servera to ime pretvori u odgovarajuću adresu numerički zapisanu, što predstavlja IP adresu računara na kojem se nalazi navedeni sajt.

Proxy server

Veb proxy serveri su oni koji omogućavaju pristup veb sadržajima svojim klijentima. Pored toga što poboljšavaju performanse konekcije, filtriraju zahteve prosleđujući ih na pravi server oni i posredno obavljaju više značajnih funkcija :

- blokiraju neželjeni sadržaj sa spoljašnjih mreža,
- omogućavaju kontrolisanje i evidenciju pristupa Internetu u zavisnosti od klijenta,
- kontrolišu kojim sajtovima se može pristupiti,
- kontrolišu ko može pristupiti,
- modifikuju sadržaj koji preuzimaju sa Interneta tako da on bude prilagođen klijentima i njihovom uređaju.

FTP server

FTP server je računarski program koji je napravljen da bi rukovao razmenom podataka između računara. FTP server čeka korisnika da se konektuje i da izda komande koje serveru daju do znanja da li je potrebno da aploaduje ili preuzme podatke, ili da izlista sve direktorijume koji su dostupni na tom serveru. FTP (engl. *File Transfer Protocol*) je protokol pomoću kog FTP server uspeva da izvrši sve navedeno. FTP je brz, efikasan i posebno pogodan za prenos velikih fajlova.

SKRIPT JEZICI

Skripta je grupa programa ili instrukcija koje treba da budu izvršene u svrhu određenog programa ili aplikacije. Serverska skripta se izvršava na serverskoj strani i samim tim nije vidljiva korisniku, dok se klijentska skripta izvršava na strani klijenta i može biti vidljiva korisniku.

Skripte mogu biti pisane na dva načina, na strani servera ili na strani klijenta. Osnovna razlika između serverskih i klijentskih skript jezika je to što serverski koriste server za procesuiranje same skripte. Sa druge strane klijentskim skript jezicima je potreban internet pregledač (brauzer) da pokrene skriptu na mašini klijenta (korisnika) i nema interakciju sa serverom dok procesuiraju klijentsku skriptu.

Serverski skript jezici se koriste u programiranju u svrhu proizvodnje koda koji može da pokreće softver na strani servera. Bilo koji rad sa skriptama koji se može pokretati na veb serveru spada u serversko skriptovanje. Operacije poput menjanja/prilagođavanja veb sajta, dinamičkih promena u sadržaju veb sajta, generisanje odgovora na korisnikove upite, pristup bazama podataka i slično, se izvršavaju na serverskoj strani.

Serverska strana podrazumeva tri dela :

- server,
- bazu podataka,
- API-je i bekind veb softver kreiran serverskim skript jezicima.

API (engl. *Application Programming Interface*) je softver koji omogućava dvema aplikacijama da komuniciraju između sebe. Svaki put kada koristite aplikaciju na mobilnom telefonu aplikacija se povezuje na internet i šalje podatke serveru, server preuzima te podatke, interpretira ih i zatim izvodi potrebne akcije pre vraćanja na mobilni telefon. API je aplikacija koja interpretira podatke i prezentuje informacije tako da budu čitljive.

Kada brauzer prosledi korisnikov zahtev serveru, ukoliko je zahtev vezan za veb stranicu koja se sastoji od skriptovanja sa serverske strane, veb server prvo procesuiraju skriptu pa tek onda dostavlja veb stranicu brauzeru. Pod pomenuto procesuiranje skripte može da spada izvlačenje informacija iz baze podataka, izvršavanje jednostavnih računskih radnji ili samo biranje prikladnog sadržaja koji će se prikazati na strani klijenta. Nakon procesuiranja skripte njena izlazna vrednost se prosleđuje brauzeru. Veb server skriptu drži skrivenom od korisnika i samim tim pruža veću sigurnost kako podacima tako i izvornom kodu.

Primeri serverskih skript jezika su :

- **ASP** – (engl. *Active Server Pages*) odličan za kreiranje dinamičkih veb sajtova sa pristupom bazama podataka. Koriste ga sajtovi Microsoft-a, StackOverflow, Svetske banke...
- **PHP** – koristi se zajedno sa SQL jezikom radi funkcionisanja i konekcija sa bazom podataka. Korišćen je u veb sajtovima Facebook, Wordpress, Wikipedia...
- **Python** – brz je i koristi kraći kod. Funkcije Python-a se koriste u sajtovima Google, Youtube...
- **Ruby** – poseduje kompleksnu logiku i visoku funkcionalnost sa bazama podataka. Koriste ga veb sajtovi Twitch, Goodreads, GitHub...

Klijentski skript jezici se koriste za skriptovanje sa klijentske strane. Ta vrsta skriptovanja se izvodi da bi generisala kod koji može da se izvršava na strani klijenta (u brauzeru) bez potrebe da se procesuiraju sa serverske strane. Ove vrste skripti su umetnute unutar HTML dokumenta. Skriptovanje sa klijentske strane može biti korišćeno da ispita eventualne greške u formi koju popuni korisnik pre nego što je prosledi kao i da menja određeni sadržaj u zavisnosti od toga šta korisnik unese.

Efektivno skriptovanje sa klijentske strane može u mnogome da smanji opterećenje servera. Dizajnirano je tako da funkcioniše unutar brauzera. Kada korisnik pošalje zahtev putem brauzera serveru za određeni veb sajt, brauzer preuzima HTML i CSS kod u vidu običnog teksta i prilagođava ga tako da korisnik dobije sadržaj koji korisnik vidi bude potpun.

Primeri klijentskih skript jezika su :

- **HTML** – organizuje celokupan sadržaj u celine i raspoređuje ih.
- **CSS** – služi za stilizovanje veb stranica i celokupnog sadržaja.
- **JavaScript** – koristi se najviše u dinamičkim veb sajtovima za dodavanje raznih interakcija i slično.

HTML

HTML je klijentski skript jezik kojim se definiše struktura i sadržaj veb stranice. Može se pisati unutar svakog tekst editora dok god se sačuva sa ekstenzijom .html. Preporučuje se da se koriste neki od tekst editora za kreiranje različitih skript dokumenata. Na primer :

Brackets – [Link za download](#)

Pre početka pisanja dokument sačuvati sa željenom ekstenzijom kako bi editor znao o kom tipu dokumenta će se raditi i samim tim ponudio pomoć pri pisanju.

Komentari u HTML-u se pišu između znakova `<!--` i `-->` .

Tagovi

Tagovima se nazivaju delovi koda unutar kojih smeštamo meta podatke, opis i sadržaj veb stranice

`<p>` - otvarajući tag za paragraf teksta

`</p>` - zatvarajući tag za paragraf teksta (neki tagovi nemaju zatvarajući tag)

Svaki HTML dokument ima sledeće tagove, odnosno sledeću strukturu :

`<html>`

`<head>`

Postavljamo naslov naše stranice, koji set karaktera će se koristiti i slično

`</head>`

`<body>`

Ovde stavljamo sve što će se videti na samoj stranici

`</body>`

`</html>`

```
<!-- Tag unutar kog se nalazi cela struktura veb stranice. -->
<html>
<!-- Unutar head taga se smeštaju stvari koje se neće videti na samoj stranici. -->
<head>
  <!-- Metadata predstavlja podatke o podacima.
        U prvom meta tagu omogućavamo korišćenje svih karaktera koji pripadaju UTF-8 setu (č,ć,ž...).
        U drugom meta tagu prilagođavamo sadržaj širini ekrana uređaja na kom se učitava. -->
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Ubacivanje eksternog css-a se vrši pomoću linka.
        rel="stylesheet" je obavezno nakon čega se u href stavlja putanja do css dokumenta. -->
  <link rel="stylesheet" href="mystyle.css">
  <!-- Title se ne prikazuje na stranici ali se prikazuje u tabu brauzera. -->
  <title>Primer HTML-a</title>
  <!-- Unutar style-a pišemo CSS kod za uređivanje izgleda stranice. -->
  <style>
  </style>
  <!-- Tag zatvaramo kosom crtom (/) i zatim nazivom taga koji želimo da zatvorimo. -->
</head>
<!-- Unutar body-ja se smeštaju svi elementi vidljivi na veb stranici. -->
<body>
  <!-- Prva sekcija je header i u njoj se uglavnom nalazi naslov/logo.
        Deljenje u sekcije nije obavezno ali je preporučljivo. -->
  <header>
    <!-- Tagovi za naslove idu od h1 do h6. -->
    <h1>HTML - Osnove</h1>
  </header>
  <!-- Sledeća sekcija je navigacija sajta.
        Unutar navigacije se smeštaju linkovi koji uglavnom vode ka ostalim stranicama veb sajta. -->
  <nav>
    <!-- Unutar a taga se postavlja ono što će označavati link (tekst/slika).
        Unutar navodnika se stavlja putanja ka onome što link treba da otvori. -->
```

```

<a href="Pocetna.html">Link ka početnoj</a>
<a href="O autoru.html">O autoru</a>
</nav>
<!-- Unutar div taga se smešta sadržaj koji će se nalaziti ispod navigacije i iznad footer-a.
Stranica može sadržati više div tagova. -->
<div>
  <!-- Unutar p taga se smešta tekst paragrafa.
  Title nije obavezan, a označava tekst koji se prikazuje
  kada se pređe mišem preko elementa za koji smo definisali title.
  Da tekst bude zakrivljen koristimo tag i.
  Da tekst ima crtu kroz sebe koristimo tag s.
  Da odvojimo deo teksta u istom redu možemo ga umetnuti unutar span taga. -->
  <p title="Uvod u HTML">HTML je klijentski <s>serverski</s> <i>skript</i> <span>jezik</span></p>
  <!-- sliku postavljamo preko img src="nazivSlike.ekstenzija" ukoliko se slika nalazi u istom
  folderu.
  Ukoliko se slika nalazi u drugom folderu ili na internetu stavlja se cela putanja do slike.
  alt="Opis" označava šta će se ispisati ukoliko se slika ne učitava u brauzeru.
  Na primeru ispod je slika umetnuta unutar linka pa će se taj link otvarati klikom na sliku.
  target="_blank" označava da želimo da se link otvori u novom tabu.
  target ne moramo da postavimo kada želimo da se link otvori u tabu u kojem je i veb stranica
  koju pregledamo. -->
  <a href="https://www.youtube.com" target="_blank"></a>
  <!-- Sa tagom b se tekst bolduje.
  Sa tagom strong se tekst i bolduje i označava kao bitnijim tekstom stranice. -->
  <p>Deo teksta je <b>boldovan</b>, a deo teksta je <strong>obeležen strong tagom.</strong></p>
  <!-- Neuređena lista započinje tagom ul (uređena sa tagom ol).
  Nakon toga se članovi liste pišu unutar tagova li.
  Moguće je isti ili različiti tip liste umetati unutar već kreirane liste. -->
  <ul>
    <li>Prvi član liste</li>
    <li>Drugi član liste</li>
  </ul>
  <!-- Unutar taga table se kreira tabela. -->
  <table>
    <!-- Unutar taga tr se smeštaju članovi reda tabele. -->
    <tr>
      <!-- th označava naslovno polje u tabeli. -->
      <th>Prvi naslov kolone</th>
      <th>Drugi naslov kolone</th>
    </tr>
    <tr>
      <!-- td označava polje sa podatkom u tabeli. -->
      <td>Prvi podatak</td>
      <td>Drugi podatak</td>
    </tr>
  </table>
  <!-- iframe tag koristimo da unutar njega postavimo stranicu/dokument unutar naše stranice. -->
  <iframe src="https://tehnickaso.edu.rs" title="STŠ Sombor"></iframe>
</div>
<!-- footer se nalazi ispod div dela, na samom kraju veb stranice.
U footer se obično smeštaju podaci o autoru veb stranice/sajta. -->
<footer>
  <p>Podaci o autoru/autorskim pravima/godini izrade</p>
</footer>
</body>
</html>

```

Forme za unos i izbor

```

<html>
<head>
  <meta charset="UTF-8">
</head>
<body>
  <form>
    <label>Unesite ime :</label>

```

```

<!-- unutar input taga biramo tip input-a
      text znači da će polje za unos služiti za unos teksta -->
<input type="text"><br><br>
<label>Dani kojima idete u školu :</label>
<!-- checkbox koristimo da omogućimo izbor više opcija -->
<input type="checkbox">
<label>Ponedjeljak</label>
<input type="checkbox">
<label>Utorak</label>
<input type="checkbox">
<label>Sreda</label>
<input type="checkbox">
<label>Četvrtak</label>
<input type="checkbox">
<label>Petak</label><br><br>
<label>Omiljena smena :</label>
<!-- radio koristimo da omogućimo izbor samo jedne od više opcija
      preko atributa name vezujemo opcije za jedan izbor -->
<input type="radio" name="smene">
<label>Prepodnevna</label>
<input type="radio" name="smene">
<label>Poslepodnevna</label><br><br>
<label>Omiljeni predmet :</label>
<select>
  <option>Programiranje</option>
  <option>Veb programiranje</option>
  <option>Primenjene IT</option>
</select><br><br>
<label>Unesite šifru :</label>
<!-- password služi za polje čiji se karakteri neće videti, odnosno polje za unos šifre -->
<input type="password"><br><br>
<!-- dugme unutar forme se koristi preko inputa tipa button
      za tekst unutar dugmeta se koristi atribut value -->
<input type="button" value="Potvrdi">
</form>
</body>
</html>

```

Unesite ime :

Dani kojima idete u školu : ☐ Ponedjeljak ☐ Utorak ☐ Sreda ☐ Četvrtak ☐ Petak

Omiljena smena : ☒ Prepodnevna ☐ Poslepodnevna

Omiljeni predmet :

Unesite šifru :

Potvrdi

Slika 1. Izgled stranice kreirane pomoću koda iznad

CSS

CSS je klijentski skript jezik kojim se definiše izgled veb stranice. Može se pisati unutar svakog tekst editora dok god se sačuva sa ekstenzijom .css. Preporučuje se da se koriste neki od tekst editora za kreiranje različitih skript dokumenata. Na primer :

Brackets – [Link za download](#)

Pre početka pisanja dokument sačuvati sa željenom ekstenzijom kako bi editor znao o kom tipu dokumenta će se raditi i samim tim ponudio pomoć pri pisanju.

Komentari u CSS-u se pišu između znakova `/* i */` .

Selektori i svojstva

Element koji pišemo pre otvaranja vitičastih zagrada se naziva selektor. Selektor može biti naziv HTML elementa, naziv klase ili naziv identifikatora. Osobine koje pišemo unutar vitičastih zagrada se nazivaju svojstvima i imaju svoje vrednosti (width: 100px; - width je svojstvo, a 100 piksela vrednost).

```
/* Pre otvaranja vitičastih zagrada se postavlja ono na šta će se svojstva unutar zagrada odnositi.
   Ukoliko želimo ista pravila za više elemenata možemo ih razdvojiti zarezom (npr. p, h2, a { CSS kod }).
   Ukoliko želimo pravila samo za elemente unutar određenih drugih elemenata razdvajamo ih bez zareza
   (npr. div p { CSS kod } ) */
element / element, element2 / elementRoditelj elementDete {
    /* width označava širinu elementa, može se izraziti u pikselima i procentima. */
    width: 100%;
    /* height označava visinu elementa. */
    height: 350px;
    /* border predstavlja okvir elementa.
       Ispisuje se u formatu: veličina tip boja. */
    border: 20px solid black;
    /* border-style označava samo tip okvira. */
    border-style: solid;
    /* border-color označava samo boju okvira. */
    border-color: white;
    /* border-width označava samo veličinu okvira (gore desno dole levo). */
    border-width: 20px 10px 15px 5px;
    /* Mogu se koristiti i : border-top:,border-right:,border-bottom:,border-left:
       kako bi se određena vrednost dodelila samo za određeni deo okvira. */
    border-right: 5px solid blue;
    /* Ukoliko želimo da okvir bude zaobljen koristimo border-radius. */
    border-radius: 30px;
    /* Ukoliko želimo da se okvir redova i podataka tabele stopi sa okvirom tabele koristimo border-
    collapse i vrednost collapse. */
    border-collapse: collapse;
    /* padding predstavlja razmak između sadržaja i okvira (ili krajnjih ivica elementa ako nema okvira).
       Npr. ukoliko element ima podešenu pozadinsku boju ili sliku ona se odnosi i na padding deo.
       Može se koristiti zasebno za određeni deo padding-top:, padding-right:, padding-bottom:, padding-
    left:. */
    padding: 25px;
    /* margin predstavlja razmak između elemenata.
       Može se koristiti zasebno za određeni deo margin-top:, margin-right:, margin-bottom:, margin-left:
    */
    margin: 25px;
    /* color menja boju teksta. */
    color: lightblue;
    /* background-color menja boju pozadine. */
    background-color: chocolate;
    /* opacity menja nivo transparentnosti (od 0 do 1 - 0.5 predstavlja 50%). */
    opacity: 0.5;
    /* background-image nam omogućava da za pozadinu postavimo sliku.
       Ukoliko se slika nalazi u folderu izvan foldera u kom je css kod koristimo
    ../FolderUKomJeSlika/NazivSlike.ekstenzija */
    background-image: url(nazivSlike.ekstenzija);
```

```

/* Ukoliko slika nije dovoljno velika da pokrije prozor ona će se ponavljati ukoliko ne koristimo
background-repeat: no-repeat; */
background-repeat: no-repeat;
/* text-align služi da postavimo tekst kako želimo (right, left, center, justify). */
text-align: center;
/* text-decoration se odnosi na to da li će tekst biti podvučen, nadvučen sa linijom kroz njega ili
bez ičega. */
text-decoration: none;
/* font-family služi da se izabere font kojim će biti ispisan tekst elementa.
Ukoliko se font sastoji od više od jedne reči potrebno je staviti naziv pod navodnike ("Times New
Roman") */
font-family: Arial;
/* font-size služi da označi veličinu teksta elementa. */
font-size: 30px;
}

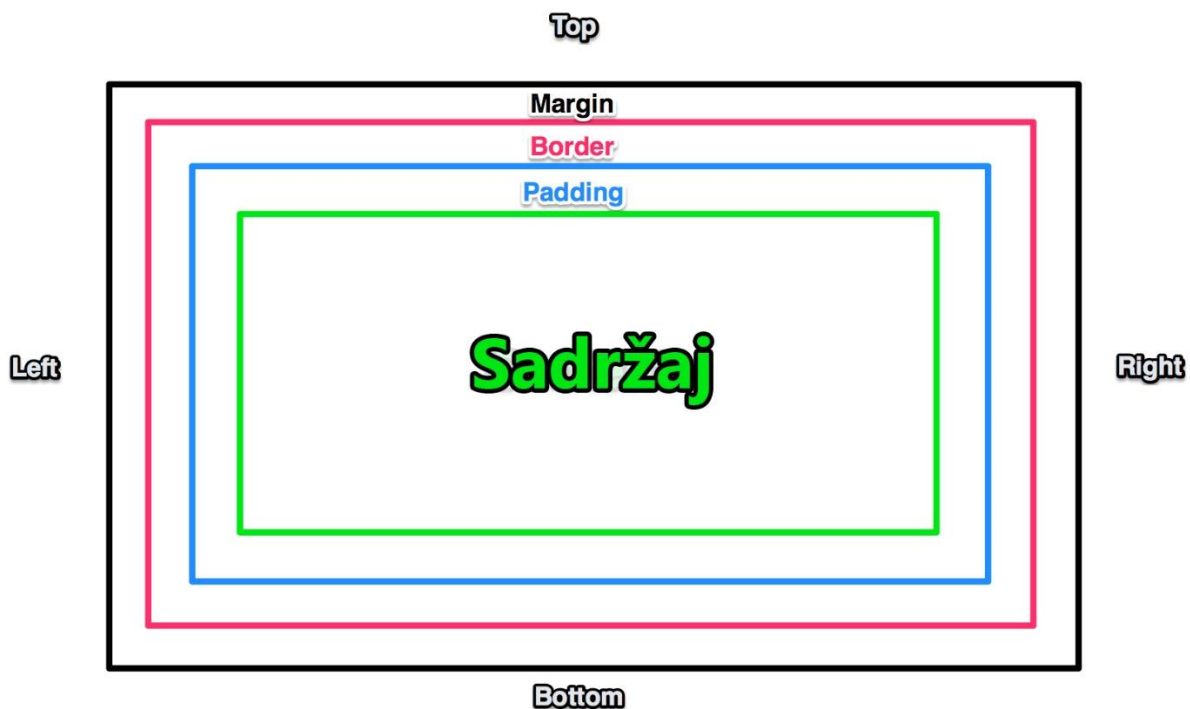
/* Tačka pre naziva označava da će se raditi o svojstvima i vrednostima za celu klasu.
Klasa se poziva unutar html dokumenta, unutar otvarajućeg taga - class="NazivKlase".
Ukoliko koristimo više klasa za isti element unutar HTML-a pišemo njihova imena jedno za drugim
class="prva druga treća" */
.NazivKlase { CSS kod }

/* # pre naziva označava da će se raditi o svojstvima i vrednostima za određeni ID.
ID se poziva unutar html dokumenta, unutar otvarajućeg taga - id="NazivId".
Preporučljivo je da se jedan ID koristi samo jednom unutar html dokumenta. */
#NazivId { CSS kod }

/* :hover stavljamo ispred elementa kako bi se određena svojstva i vrednosti odnosila za prelaz mišem
preko tog elementa.
Za pisanje koda za posećeni element koristimo :visited */
element:hover { CSS kod }

/* Ukoliko želimo da centriramo određeni element (u ovom primeru div) postavljamo margin-left i margin-
right na auto. (npr. iframe možemo centrirati i pomoću text-align: center). */
div {
margin-left: auto;
margin-right: auto;
}

```



Slika 2. Objašnjenje margina i padding-a

Pozicioniranje

Korisni linkovi za razumevanje pozicioniranja HTML elemenata pomoću svojstava *position*, *top*, *left*, *right* i *bottom* : [LINK 1](#) , [LINK 2](#)

Animacije

Animacija omogućuje da element postepeno menja stil iz jednog u drugi. Moguće je menjati bilo koji broj CSS svojstava, koliko god puta želimo.

Da bi se koristila CSS animacija potrebno je prvo specificirati ključne kadrove (keyframes).

Keyframe-ovi sadrže vrednost stilova u određenim momentima.

animation-name: imeAnimacije; - pišemo unutar CSS koda za html element (ili elemente) za koje želimo da se animacija izvršava.

animation-duration: vremeUsekundama; - predstavlja vremenski period u kom će se animacija izvršiti, pišemo unutar CSS koda za html element (ili elemente) za koje želimo da se animacija izvršava, ispod animation-name svojstva.

@keyframes imeAnimacije {

from {cssSvojstvo: vrednost1;}

to {cssSvojstvo: vrednost2;} } – pišemo unutar CSS koda i predstavlja promenu koja će se izvršavati za svojstva koja izaberemo.

```
#tekst {
  color: black;
  font-size: 20px;
  animation-name: promenaBoje;
  animation-duration: 5s;
}

@keyframes promenaBoje {
  from { color: black; }
  to { color: white; }
}
```

@keyframes imeAnimacije {

procenti {cssSvojstvo: vrednost1;}

procenti {cssSvojstvo: vrednost2;}

procenti {cssSvojstvo: vrednost3;}

procenti {cssSvojstvo: vrednost4;} } – animacija menja vrednost svojstva nakon prolaska određenog procenta od vremena (animation-duration) animacije.

```
#tekst2 {
  color: black;
  font-size: 20px;
  animation-name: promenaBoje2;
  animation-duration: 5s;
}

@keyframes promenaBoje2 {
  0% { color: black; }
  20% { color: white; }
  40% { color: red; }
  60% { color: blue; }
  80% { color: grey; }
}
```

Podrazumevane (default) postavke animacije vraćaju sve vrednosti CSS svojstava na početne vrednosti nakon isteka vremena animacije.

Ukoliko želimo da se animacija ne izvršava odmah po učitavanju stranice nego nakon određenog vremena koristimo **animation-delay: vremeUsekundama;** .

Ukoliko želimo da zadržimo nove vrednosti svojstava koristimo **animation-fill-mode: forwards;** i to postavljamo ispod naziva animacije i njene dužine trajanja.

```
#slika {  
  animation-name: povecaj;  
  animation-duration: 3s;  
  animation-delay: 10s;  
  animation-fill-mode: forwards;  
}  
  
@keyframes povecaj {  
  from { width: 40px; }  
  to { width: 360px; }  
}
```

Fontovi

Fontovi koji se mogu koristiti pri definisanju fonta u css kodu :

- Arial
- Verdana
- Helvetica
- Tahoma
- Trebuchet MS
- Times New Roman
- Georgia
- Garamond
- Courier New
- *Brush Script MT*

Korišćenje drugih fontova može dovesti do toga da ih brauzer ne prepozna.

JAVASCRIPT

JavaScript je klijentski skriptni jezik. U pitanju je takozvani objektni skriptni jezik, čija je osnovna svrha korišćenja vezana za poboljšanje dinamike određene stranice. Kada se koristi HTML kod, njegova uloga je samo da oblikuje određene elemente stranice, kao što su recimo tekst, tabele, linkovi ili forma, te da ih nakon toga uredi, ali ovaj kod nema ulogu da odredi na koji način će se konkretni elementi ponašati. Upravo tada se u čitav proces uključuje JavaScript, koji ima značajnu ulogu u kreiranju ponašanja ovih elemenata. Može se pisati unutar svakog tekst editora dok god se sačuva sa ekstenzijom **.js**. Preporučuje se da se koriste neki od tekst editora za kreiranje različitih skript dokumenata. Na primer :

Brackets – [Link za download](#)

Visual Studio Code - [Link za download](#)

Pre početka pisanja dokument sačuvati sa željenom ekstenzijom kako bi editor znao o kom tipu dokumenta će se raditi i samim tim ponudio pomoć pri pisanju.

Osnove sintakse

Jezik je case sensitive (pravi razliku između velikih i malih slova).

Nazivi promenljivih moraju počinjati ili sa slovom ili sa donjom crtom `_`.

Komentari se pišu nakon `//` ili između `/*` i `*/`.

Gde možemo pisati JavaScript kod?

JavaScript kod možemo pisati unutar **<script>** taga u head ili body sekciji html dokumenta.

```
<script>
    // JavaScript kod
</script>
```

Eksterni JavaScript učitavamo na sledeći način :

```
<script src="nazivSkripte.js"></script>
```

I eksternu skriptu možemo pozivati unutar head i body sekcije html dokumenta. S obzirom da se html dokument učitava s vrha na dole skriptu možemo učitati pre zatvaranja body taga kako bi svi html elementi već bili učitani pre izvršavanja JavaScript koda. Kao i kod CSS-a, kada pišemo eksternu skriptu nije potrebno da u nju stavljamo tag script.

Unutar jednog html dokumenta se može nalaziti više skripti.

Promenljive

Razlika između var i let

Kada koristimo **var** promenljiva postoji u celoj skripti i ona je globalna (ukoliko je definisana van funkcije) ili unutar cele funkcije ukoliko je definisana unutar nje.

Kada koristimo **let** promenljiva postoji samo unutar bloka (između vitičastih zagrada) unutar kog smo je definisali.

```
function Test1() {
    var x = 1;
    {
        var x = 2; // ista promenljiva sa onom iznad
        console.log(x); // vrednost x je 2
    }
    console.log(x); // vrednost x je 2
}
```

```
function Test2() {
  let x = 1;
  {
    let x = 2; // različita promenljiva od one iznad
    console.log(x); // vrednost x je 2
  }
  console.log(x); // vrednost x je 1
}
```

Konstanta – const

Vrednost konstante ne može da se menja unutar skripte ili bloka unutar kog je definisana. Koristi se kada želimo promenljivu koja će uvek imati istu vrednost.

```
// definiše promenljivu pi kao konstantu i daje joj vrednost 3.14
const pi = 3.14;

// pokušaj dodeljivanja vrednosti konstanti će izbaciti grešku (error)
pi = 5;
```

Tipovi promenljivih

Tipovi promenljivih se mogu menjati tokom izvršavanja programa.

```
let x = 21; // x sada ima numeričku vrednost
x = 'iks'; // x je sada niz karaktera
x = true; // x je sada logičkog tipa
```

Konverzija stringa u broj

Za konverziju stringa u broj, koriste se funkcije:

- **EVAL** - ocenjuje string i ako je moguće pretvara ga u broj;
- **parseInt** - konvertuje string u integer, ako je moguće;
- **parseFloat** - konvertuje string u floating-point broj, ako je moguće.

Funkcije

Funkcije (ili metode) se započinju službenom rečju **function** nakon čega sledi naziv funkcije i zagrade sa parametrima (ako ih ima, ako ne zagrade ostaju prazne) nakon čega se između vitičastih zagrada pišu naredbe.

```
function nazivFunkcije() {
  // JavaScript kod
}
```

U slučaju da funkcija ima parametre nije potrebno pisati tip podataka već samo naziv parametara.

```
function nazivFunkcije(x, y) {
  // JavaScript kod
}
```

Funkcija može a ne mora da vraća vrednost. U oba slučaja se funkcija piše na isti način (nema tipa podataka niti reči void), dok se za vraćanje vrednosti koristi reč return.

Funkciju možemo smestiti unutar promenljive (tada nam nije potreban naziv funkcije) :

```
let x = function () {
  // JavaScript kod
}
```

Funkciju pozivamo preko njenog naziva ili naziva promenljive u kojoj je skladištena.

Naredbe i petlje

U JavaScript-u možemo koristiti :

- if – else
- switch
- for
- while
- do – while

na isti način kao i u C#.

Pristup html elementima

document.getElementById(id)

Pristupa elementu na osnovu id-ja koji je dodeljen unutar html dokumenta.

```
<!-- html kod -->
<p id="opis">Tekst paragrafa.</p>
```

```
// JavaScript kod
let opis = document.getElementById("opis");
```

document.getElementsByTagName(name)

Pristupa elementima sa istim tagom unutar html dokumenta.

```
<!-- html kod -->
<p>Tekst paragrafa.</p>
<p>Tekst drugog paragrafa.</p>
<p>Tekst trećeg paragrafa.</p>
```

```
// JavaScript kod, odnosi se na sve elemente sa tagom p
let paragrafi = document.getElementsByTagName("p");
```

document.getElementsByClassName(name)

Pristupa elementima html dokumenta koji pripadaju istoj klasi.

```
<!-- html kod -->
<p class="specijalni">Tekst paragrafa.</p>
<p>Tekst drugog paragrafa.</p>
<p class="specijalni">Tekst trećeg paragrafa.</p>
```

```
// JavaScript kod, odnosi se na elemente koji pripadaju klasi specijalni (prvi i treći paragraf)
let specijalniP = document.getElementsByClassName("specijalni");
```

Manipulisanje html elementima

element.innerHTML

Menja sadržaj html elementa, odnosno ono što se nalazi između otvarajućeg i zatvarajućeg taga.

```
<!-- html kod -->
<p id="opis">Tekst paragrafa.</p>
```

```
// JavaScript kod
let opis = document.getElementById("opis");
opis.innerHTML = "Novi tekst";
```

Nakon izvršenja skripte tekst u paragrafu sa id-jem opis će se promeniti u Novi tekst.

element.atribut

JavaScript nam daje mogućnost i da menjamo attribute određenog elementa.

```
<!-- html kod -->

```

```
// JavaScript kod
let slika = document.getElementById("pozadina");
slika.src = "macka.jpg";
```

Nakon izvršenja skripte slika na veb stranici će se promeniti iz pas.jpg u macka.jpg.

element.style.svojstvo

Menja css svojstvo određenog html elementa.

```
<!-- html kod -->
<p id="opis">Tekst paragrafa.</p>
```

```
// JavaScript kod
let opis = document.getElementById("opis");
opis.style.color = "blue";
```

Nakon izvršenja skripte boja teksta paragrafa sa id-jem opis će se promeniti u plavu.

Izmene se mogu praviti i bez dodeljivanja elementa promenljivoj :

```
document.getElementById("test").innerHTML = ...
document.getElementById("test").atribut = ...
document.getElementById("test").style.property = ...

<p id="test" title="paragraf" style="color: blue;">Tekst paragrafa.</p>
```

Elementu pristupamo preko id-ja i zatim možemo da mu menjamo sadržaj, attribute i stil.

Manipulisanje prozorom pretraživača

window.open

Koristimo za otvaranje novog prozora pretraživača.

```
window.open("link ka sajtu", "naziv prozora (ako se ostavi prazno otvaraće uvek novi prozor)", "svojstva prozora (npr. height i width)");
```

Manipulisanje zvukom

Unutar JavaScript-a možemo koristiti i klasu Audio za puštanje zvuka. Najbolje je koristiti zvučne fajlove sa .mp3 ekstenzijom.

```
let zvuk = new Audio('Putanja do zvuka');
zvuk.play();
```

Iskaćuci prozor

alert

```
alert("Tekst koji se pojavljuje u iskaćucem prozoru");
```

Upravljanje događajima

Promene html elementa možemo vršiti i u odnosu na odigravanje nekog događaja.

onclick

Izvršavanje koda nakon klika na određeni element.

Prvi način :

```
<!-- html kod -->
<p id="opis">Tekst paragrafa.</p>
```

```
// JavaScript kod
document.getElementById("opis").onclick = function () {
document.getElementById("opis").innerHTML = "Novi tekst"; }
```

Drugi način :

```
<!-- html kod -->
<p id="opis" onclick="promeni()">Tekst paragrafa.</p>
```

```
// JavaScript kod
function promeni() {
document.getElementById("opis").innerHTML = "Novi tekst"; }
```

Klikom na paragraf poziva se funkcija unutar skripte i tekst paragrafa se menja.

onmouseover

Izvršavanje određenog koda nakon prelaska mišem preko određenog elementa.

Prvi način :

```
<!-- html kod -->
<p id="opis">Tekst paragrafa.</p>
```

```
// JavaScript kod
document.getElementById("opis").onmouseover = function () {
document.getElementById("opis").innerHTML = "Novi tekst"; }
```

Drugi način :

```
<!-- html kod -->
<p id="opis" onmouseover="promeni()">Tekst paragrafa.</p>
```

```
// JavaScript kod
function promeni() {
document.getElementById("opis").innerHTML = "Novi tekst";
}
```

Prelaskom mišem preko paragrafa poziva se funkcija unutar skripte i tekst paragrafa se menja. Za izvršavanje koda nakon sklanjanja miša sa elementa koristimo ***onmouseout***.

onchange

Koristi se unutar forme za izbor kako bi se funkcija pozivala pri promeni izbora.

```
<!-- html kod -->
<form>
```

```
<select id="izbor" onchange="Zameni()">
  <option value="orasac">Збор у Орашцу</option>
  <option value="misar">Бој на Мишару</option>
</select>
</form>

```

```
// JavaScript kod
function Zameni() {
  let izbor = document.getElementById("izbor").value;
  let slika = document.getElementById("slika");

  if (izbor == "orasac") {
    slika.src = "Slike/Orasac.png";
  }
  else {
    slika.src = "Slike/Misar.png";
  }
}
```

Nakon svake promene od dve opcije za izbor menjaće se i slika koja se prikazuje na stranici.

setTimeout i setInterval

setTimeout(nazivFunkcije,vremeUmilisekundama) koristimo kada želimo da se funkcija izvrši jednom nakon izabranog vremenskog perioda.

setInterval(nazivFunkcije,vremeUmilisekundama) koristimo kada želimo da se funkcija poziva nakon svakog novog isteka izabranog vremenskog perioda.

Ukoliko funkcija koju pozivamo ima parametre njih pišemo nakon vremena setInterval(nazivFunkcije,vremeUmilisekundama,parametar1,parametar2) (isto važi i za setTimeout).

clearInterval(nazivIntervala) i clearTimeout(nazivTimeouta) služe da prekinu/zaustave izvršavanje Intervala ili Timeouta.

Objekti

U JavaScript jeziku svaka veb stranica ima određene objekte :

- **window** – objekat najvišeg nivoa koji sadrži sve druge objekte i predstavlja prozor pretraživača.
- **document** – predstavlja trenutno učitani dokument i služi kao pristupna tačka sadržaju stranice.
- **location** – sadrži informacije o URL adresi dokumenta koji je trenutno prikazan u posmatranom prozoru.
- **history** – sadrži URL adrese koje su prethodno bile posećene, kao i URL adrese koje su posećene nakon posete trenutnoj stranici.

U JavaScript jeziku mnogi objekti imaju ugrađene funkcije (metode) koje simuliraju događaje nastale usled akcija korisnika. Neke od njih su :

- **focus** – za ulazak u fokus nekog elementa forme.
- **blur** – za izlazak iz fokusa nekog elementa forme.
- **load** – za učitavanje stranice u pregledač.
- **submit** – za izvršavanje slanja forme.

STRUKTURA FOLDERA SA DOKUMENTIMA SAJTA



U folder sa nazivom koji bi trebao da označava šta veb sajt predstavlja se smeštaju html stranice **i u odvojene foldere** slike, css, JavaScript i sve ostalo što čini sajt. Početna stranica ima naziv **index.html** (ovo je praksa iz razloga što pojedini veb serveri automatski prepoznaju index.html kao početnu stranicu veb sajta) dok se ostale (ukoliko ih ima) nazivaju u skladu sa onim šta predstavljaju. Da se ne bi poremetili eventualne povezanosti imena html stranica, slika, css i JavaScript dokumenata ne treba menjati kada se jednom naprave.

Eksterni css dokument (ili dokumenti) i JavaScript skripta (ili skripte) se nakon toga povezuju unutar html-a.

VALIDATORI KODA

HTML

Link - <https://validator.w3.org/nu/#textarea>

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Ready to check

Checker Input

Show ☒ source ☐ outline ☐ image report [Options...](#)

Check by ☐ css

```
<!DOCTYPE html>
<html lang="">
<head>
<title>Test</title>
</head>
<body>
<p></p>
</body>
</html>
```

[Check](#)

Nakon kopiranja svog html koda unutar polja za tekst treba izabrati opciju *Check*.

1. **Error** The `center` element is obsolete. [Use CSS instead.](#)
[From line 28, column 9 to line 28, column 16](#)
`><center>`

2. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
[From line 36, column 65 to line 36, column 114](#)
`= "blank"></a></`

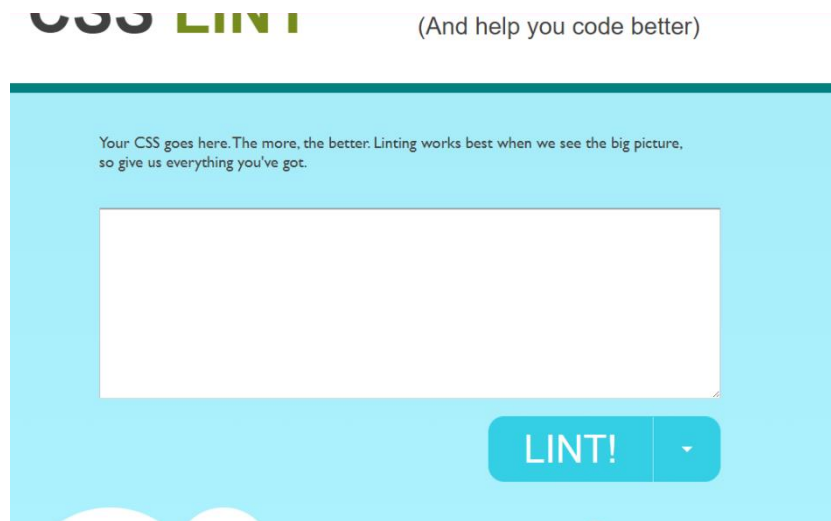
3. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
[From line 42, column 21 to line 42, column 71](#)
`</a></`

4. **Error** An `img` element must have an `alt` attribute, except under certain conditions. For details, consult [guidance on providing text alternatives for images](#).
[From line 47, column 65 to line 47, column 114](#)
`= "blank"></a></`

Ispod polja za unos teksta će se pojaviti opis eventualnih grešaka (*Error*) i upozorenja (*Warning*).

CSS

Link - <http://csslint.net>



Kopiranjem css koda u polje za tekst i izborom opcije *LINT!* uneseni kod se proverava i otvara se veb stranica sa rezultatima. Ispisuje se broj grešaka (*errors*) i upozorenja (*warnings*). Klikom na njih u listi stranica se spušta na mesto gde se te greške ili upozorenja nalaze u kodu.



[About](#) | [Contribute](#) | [Source](#) | [Node/Rhino CLI](#) | [Documentation](#)

Search:

	line	column	title	description	browser
	66	9	Parsing Errors	Expected COLON at line 66, col 9. header{	All
	49	3	Disallow IDs in selectors	Don't use IDs in selectors. #velikaTabela td{	All
	68	3	Disallow duplicate properties	Duplicate property 'color' found. color: white;	All

JavaScript

Link - <https://beautifytools.com/javascript-validator.php>

Javascript Validator

Use Online Javascript Validator to validate javascript code and find errors and warnings of your code that can be fixed. See real time code validation and fix errors and warnings of your code as you type. After coding is complete, format your javascript code.

1

Validate Code

Options

Format Code

Clear

Line	Col	Errors
No syntax errors!		

Nakon unosa JavaScript koda u polje za tekst bira se opcija *Validate Code*. Sa desne strane će se pojaviti eventualne greške.

1 // Povezujemo promenljivu ime sa elementom koji ima id ime
2 let ime = document.getElementById("ime");
3 // Povezujemo promenljivu prezime sa elementom koji ima id prezime
4 let prezime = document.getElementById("prezime");
5
6 function Pozdravi() {
7 // funkcija alert ispisuje tekst unutar zagrada u iskačućem prozoru
8 alert("Dobrodošli " + ime.value + " " + prezime.value + " !");
9 }

Line	Col	Errors
6	10	'Pozdravi' is defined but never used.

Najčešće upozorenje koje će se pojavljivati jeste *is defined but never used* pošto u primerima koje radimo funkcije pozivamo u html-u i ne koristimo ih u samoj JS skripti. To upozorenje se može zanemariti.