

# Osnovna struktura HTML dokumenata

Kada kreiramo novi HTML dokument, različiti HTML elementi se pišu direktno unutar samog HTML dokumenta. Međutim, stvarnost je ipak takva da HTML jezik, i pored elemenata koji poseduju vizuelnu reprezentaciju, poznaje i nekoliko nevidljivih elemenata kojima se definiše osnovna struktura HTML stranice.

Baš kao i svaki drugi jezik, i HTML poseduje svoja semantička i sintakсна pravila, koja je neophodno poštovati. Jedno od osnovnih takvih pravila nalaže da svaki HTML dokument mora posedovati osnovnu strukturu sačinjenu iz nekoliko elemenata.

## Kreiranje HTML fajla

### ***Za Windows operativni sistem:***

Za kreiranje HTML fajla na Windows operativnim sistemima dovoljno je odabrati opciju za kreiranje novog tekstualnog (txt) fajla. Opcija **New -> Text Document** može se naći unutar kontekstnog menija, koji se dobija desnim klikom unutar nekog foldera ili desktopa. Aktiviranje spomenute opcije stvoriće novi fajl, kojem je potrebno postaviti naziv, a ekstenziju promeniti u html. Nakon definisanja naziva HTML dokumenta i promene ekstenzije u .html, operativni sistem Windows će nas pitati da li smo sigurni da želimo da izvršimo promenu ekstenzije fajla koji smo kreirali. Naravno, potrebno je odgovoriti potvrdno, čime je posao kreiranja HTML dokumenta na Windows operativnom sistemu završen.

Kao rezultat prethodnih koraka, dobija se jedan prazan HTML dokument. Da biste kreirani HTML fajl otvorili unutar editora koji koristite (npr. Visual Studio Code), dovoljno je da odaberete opciju **Open with -> Notepad**, koja se nalazi u kontekstnom meniju, koji se dobija desnim klikom na HTML fajl i tu izaberete editor.

### ***Za macOS operativni sistem:***

Kreiranje fajla sa ekstenzijom .html na macOS-u nešto je komplikovanije nego što je to bio slučaj kod Windowsa. Na macOS-u podrazumevana aplikacija za obradu teksta jeste TextEdit i ona će nam poslužiti kao primarni alat za obavljanje posla kreiranja prvog HTML dokumenta. Procedura će se sastojati iz kreiranja novog dokumenta korišćenjem ove aplikacije i njegovog čuvanja kao dokumenta sa ekstenzijom .html.

Aplikaciju TextEdit možete otvoriti odabirom opcije **Finder -> Applications -> TextEdit**. Nakon otvaranja aplikacije potrebno je napraviti određene izmene u podešavanjima kako bi ona na odgovarajući način obavila čuvanje fajla kao HTML dokumenta. Prvo je potrebno promeniti format dokumenta na Plain Text. To se obavlja iz prozora sa podešavanjima, koji se dobija pokretanjem opcije **TextEdit -> Preferences**

Unutar *Preferences* prozora i njegovog prvog taba *New Document* potrebno je čekirati opciju **Plain Text**. Unutar drugog taba (*Open and Save*) *Preferences* prozora potrebno je čekirati opciju **Display HTML files as HTML code instead of formatted text**. Nakon opisanih izmena u podešavanjima *TextEdit* aplikacije, fajl je moguće sačuvati kao HTML dokument pokretanjem opcije **File -> Save** i izmenom naziva i ekstenzije u prozoru za čuvanje. Na kraju, unesemo naziv stranice i preostaje samo da se potvrdi promena ekstenzije klikom na dugme **Use .html**

Prethodni koraci ilustrovali su kreiranje prvog HTML dokumenta na Windows i macOS operativnim sistemima. Takvi dokumenti su trenutno potpuno prazni, tako da je naš sledeći korak da unutar njih unesemo odgovarajući HTML kod, a u narednim lekcijama i CSS kod.

## Struktura HTML dokumenta

Osnova struktura HTML dokumenta je sledeća:

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

</head>
<body>

</body>
</html>
```

U prikazanoj strukturi možemo razaznati nekoliko otvarajućih i zatvarajućih tagova, odnosno, drugim rečima, nekoliko HTML elemenata. Može se reći da je osnovna struktura HTML dokumenta sačinjena iz sledećih delova:

- **DOCTYPE deklaracija**
- **html element**
- **head element**
- **body element**

U nastavku ove lekcije detaljno će biti obrađena ova 4 osnovna gradivna sastojka svakog HTML dokumenta.

### Napomena

*Naime, današnji web browseri sposobni su da u određenoj meri koriguju kod HTML dokumenta. Drugim rečima, ukoliko autor dokumenta napravi neku grešku, web browser će automatski pokušati da je ispravi. Upravo je ovo razlog zbog kojeg je moguće napisati HTML dokument bez prikazane osnovne strukture. Jednostavno, web browser će takvu osnovnu strukturu uspostaviti umesto nas.*

*Veoma je bitno u ovom trenutku reći da ovu pogodnost web browsera nikako nije dobro svesno koristiti. Drugim rečima, nikako nije dobro izostavljati osnovnu HTML strukturu, samo zato što znamo da će tako nešto browser uraditi za nas. Pisanje nevalidnog HTML koda može stvoriti brojne probleme prilikom funkcionisanja web sajta, odnosno web aplikacije (greške prilikom indeksiranja od strane web crawlera, neodgovarajući prikaz u različitim browserima itd...).*

## DOCTYPE

Prvi element osnovne HTML strukture jeste DOCTYPE deklaracija. DOCTYPE deklaracija (*Document Type Declaration*, skraćeno DTD) smeštena je na sam početak svake HTML strane pod „tagom” DOCTYPE. Reč *tag* je sa razlogom stavljena pod navodnike. Zapravo, DOCTYPE nije pravi HTML tag, već instrukcija upućena browseru (zbog oznake ! nakon otvaranja taga).

DOCTYPE deklaracija govori browseru koja se verzija HTML-a koristi u dokumentu. Iz prethodne lekcije poznato je da je trenutna verzija HTML jezika 5, a ona definiše deklaraciju sledećeg oblika:

```
<!DOCTYPE html>
```

Postoje i deklaracije u raznim drugim oblicima u zavisnosti od verzije HTML jezika koja se koristi (tabela 2.1).

Verzija	DOCTYPE
HTML5	<!DOCTYPE html>
HTML4	<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
XHTML	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

Tabela 2.1. Različiti oblici DOCTYPE deklaracije u zavisnosti od verzije HTML jezika

Sve ovo što je do sada rečeno o **DOCTYPE** deklaraciji praktično znači da možete sami da odaberete koju ćete varijantu HTML jezika koristiti za pisanje dokumenta. Ipak, danas ne postoji nijedan valjani razlog za korišćenje bilo koje verzije koja nije HTML5. Stoga vam ova priča o različitim DOCTYPE deklaracijama možete koristiti ukoliko se ikada susretnete sa nekim web sajtom koji je napisan veoma davno kako biste mogli da utvrdite koja je verzija HTML jezika korišćena za pisanje.

### DOCTYPE i validnost HTML dokumenta

S obzirom da DOCTYPE deklaracija govori browseru koja je verzija HTML-a korišćena za pisanje koda, ovaj pojam se veoma često vezuje i sa validnošću HTML dokumenta. Jednostavno, svaka verzija HTML-a posedovala je svoje specifičnosti kada su u pitanju pravila pisanja koda i dostupnost različitih tagova, elemenata i atributa. Stoga, definisani DOCTYPE igra veliku ulogu u tome da li će neki dokument biti validan ili ne.

Inače, **jedan HTML dokument validan je ukoliko unutar njega nema grešaka**. Greške se mogu ogledati u upotrebi nekog pravila iz druge verzije jezika koja nije podržana u verziji koja se koristi. Na primer, u XHTML-u može se koristiti atribut `cellspacing` nad tagom tabele, dok je taj konkretan atribut u HTML5 izbačen iz upotrebe. S toga, ukoliko se atribut `cellspacing` upotrebi u XHTML varijanti jezika, kôd će biti validan, dok u HTML5 neće (bez obzira na to da li ga je browser prepoznao ili nije).

Validnost dokumenta može se proveriti korišćenjem W3C validatora na sledećoj adresi:

<http://validator.w3.org/>

## HTML element

Nakon DOCTYPE deklaracije, osnovna struktura HTML dokumenata definiše HTML element. S obzirom da se DOCTYPE ne može nazvati pravim elementom, može se reći da je koreni element svakog HTML dokumenta `html` element. On počinje otvarajućim `<html>` tagom i uvek se završava zatvarajućim `</html>` tagom:

```
<html>
...
</html>
```

Neke od najznačajnijih osobina `html` elementa su sledeće:

- predstavlja koreni element HTML dokumenta;
- obavezni element, bez koga HTML dokument nije validan;
- obuhvata sve ostale elemente na stranici.

Element `html` je roditelj za dva direktna potomka. To su elementi:

- `head` – zaglavlje dokumenta;
- `body` – telo dokumenta.

`head` i `body` elementi se drugačije nazivaju zaglavlje i telo dokumenta, respektivno, i neizostavni su delovi svakog HTML dokumenta. Svi ostali elementi koje HTML poznaje dodaju se ili unutar `head` ili unutar `body` dela. Takođe, `head` i `body` elementi unutar HTML dokumenta mogu se pojaviti samo jednom, i to uvek u redosledu prikazanim slikom 2.1.

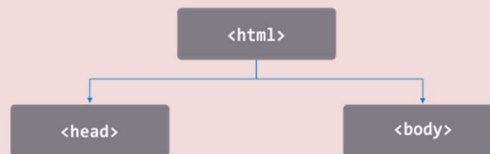


Slika 2.1. Redosled osnovnih HTML elemenata

### Stablo HTML elemenata

U ovom trenutku se po prvi put susrećemo sa jednom od osnovnih karakteristika HTML dokumenata. Reč je o načinu na koji su podaci unutar HTML dokumenta strukturirani. Naime, unutar HTML dokumenata podaci su organizovani u formi stabla.

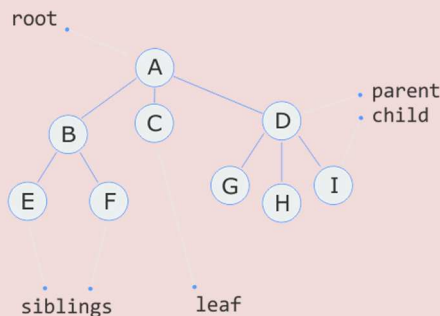
**Forma stabla** jeste jedan veoma čest način strukturiranja podataka u kompjuterskim naukama. Njegova osnova zasniva se na smeštanju jednog elementa u neki drugi, pritom se poštuju određena pravila. Upravo takvu situaciju imali smo prilike da vidimo kod strukture HTML dokumenata. Elementi `head` i `body` smeštaju se unutar elementa `html`. Tako se ova struktura može prikazati i kao na slici 2.2.



*Slika 2.2. Stablo osnovnih HTML elemenata*

Osnovna struktura HTML dokumenata idealni je primer forme stabla. Unutar elementa `html` nalaze se elementi `head` i `body`. Reč je o osnovnoj HTML strukturi, dok je u praksi broj elemenata u ovakvom stablu mnogo veći, kao što ćemo uskoro imati prilike da vidimo.

Forma stabla poseduje određena ustrojstva, sa kojima ćemo se mi upoznati u narednim redovima. Sve to će nam pomoći u lakšem razumevanju HTML dokumenata, koje ćemo kreirati u narednim lekcijama i koji će imati znatno razgranatiju strukturu. Za početak, slikom 2.3. biće prikazana jednostavna forma stabla, sa svim karakterističnim elementima.



*Slika 2.3. Primer strukture u obliku stabla*

Element na najvišem nivou forme stabla drugačije se naziva koreni (engl. **root**) element. Forma stabla može imati samo jedan koreni element i to je ujedno element unutar koga se nalaze svi ostali elementi. Na slici 2.3. koreni element obeležen je slovom A.

Koreni element može imati proizvoljan broj potomaka (engl. **descendants**). Na slici 2.3. elementi B, C i D potomci su (naslednici) elementa A. Takođe, elementi E i F naslednici su elementa B, a elementi G, H, i I potomci su elementa D. Element C nema potomaka.

Dva elementa u formi stabla mogu imati odnos roditelj–dete (engl. **parent–child**).

Takav odnos je na slici 2.3. predstavljen između elemenata D i I, pri čemu je element D roditelj, a element I dete. Ipak i između mnogih drugih elemenata prikazanih slikom postoji ova relacija roditelj–dete. Na primer, element B je roditelj elementima E i F, odnosno oni su njegova deca. Takođe, element A je roditelj elementima B, C i D, pa i između njih postoji relacija roditelj–dete.

Unutar forme stabla jedan element može biti i roditelj i dete. Na slici 2.3. takvi su elementi B i D. Oni su potomci elementa A, a svaki od njih poseduje i svoje potomke.

Elementi koji u formi stabla imaju istog roditelja jesu takozvani srodni elementi, odnosno **siblings**. Elementi koji u formi stabla imaju istog roditelja takozvani su srodni elementi, odnosno siblings. Na slici 2.3. su kao srodni obeleženi elementi E i F. Srodni elementi su i:

- B, C i D
- G, H i I

Na kraju, elementi u formi stabla ne moraju imati potomke. Na slici 2.3. takvih je nekoliko elemenata: E, F, C, G, H, I. Element koji nema potomke drugačije se naziva list (engl. **leaf**).

Sve ove osobine o formi stabla, koje su iznete u redovima za nama, u potpunosti su primenljive i na HTML dokumente, čiji elementi međusobnim gneždenjem formiraju upravo ovakvu strukturu. Zato je razumevanje forme stabla veoma značajno za razumevanje same strukture HTML dokumenata.

Nakon upoznavanja korenog elementa (`html` element), prelazimo na element koji predstavlja telo HTML dokumenta.

## Telo HTML dokumenta – `body` element

Element `body` predstavlja telo HTML dokumenta. Unutar `body` dela smeštaju se svi oni elementi za koje želimo da budu vidljivi na samoj stranici nakon njenog parsiranja unutar browsera. To praktično znači da svi oni HTML elementi koje smo do sada koristili za realizaciju uvodnih primera pripadaju telu dokumenta i da ih je po pravilu potrebno smestiti unutar `body` elementa. Na primer, u prethodnoj lekciji korišćen je jedan ovakav HTML dokument:

```
<h1>This is a heading</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec
luctus sapien magna, eu sollicitudin augue porttitor nec. Nulla
pharetra, mi eget maximus posuere, quam quam malesuada metus, a volutpat
mauris arcu id lorem.</p>
<p>Vivamus neque felis, pharetra sed odio a, mattis lacinia neque.
Donec id urna quam. Fusce ac risus vel diam vestibulum maximus vel sit
amet erat.</p>
```

Tri prikazana HTML elementa (naslov i dva paragrafa), poštujući pravila HTML strukture, potrebno je smestiti unutar tela dokumenta:

```
<body>
<h1>This is a heading</h1>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec
luctus sapien magna, eu sollicitudin augue porttitor nec. Nulla
pharetra, mi eget maximus posuere, quam quam malesuada metus, a volutpat
mauris arcu id lorem.</p>
<p>Vivamus neque felis, pharetra sed odio a, mattis lacinia neque.
Donec id urna quam. Fusce ac risus vel diam vestibulum maximus vel sit
amet erat.</p>
</body>
```

## Zaglavlje HTML dokumenta – head element

Unutar `head` elementa, koji se drugačije naziva zaglavlje HTML dokumenta, postavljaju se elementi koji nisu direktno vidljivi na stranici. Na primer, u `head` delu postavljaju se metatagovi, veze (linkovi) ka spoljnim CSS i drugim fajlovima i tako dalje.

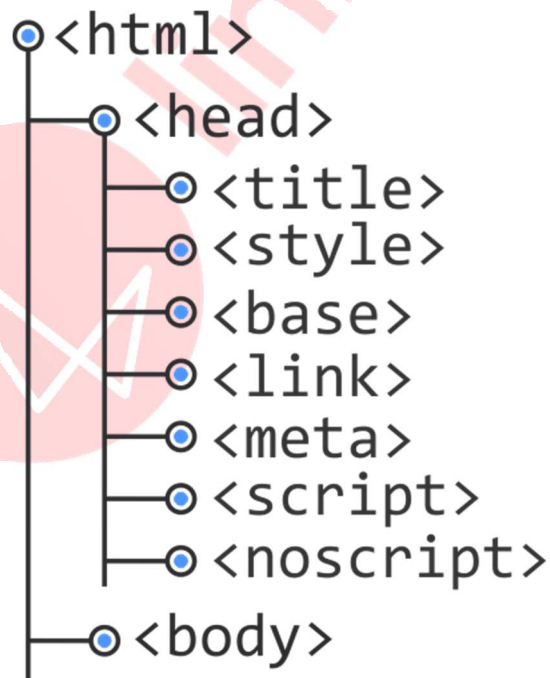
Element `head` počinje tagom `<head>`, a završava se tagom `</head>`. To je koreni element za sve podatke zaglavlja jednog HTML dokumenta. Element `<head>` je roditeljski element sledećim HTML elementima:

- `<title>`
- `<style>`
- `<base>`
- `<link>`
- `<meta>`

### Napomena

*U HTML5 jeziku `<head>` element nije obavezno definisati kako bi stranica bila validna, ali se u tom slučaju `<title>` element mora definisati, s obzirom na to da je neophodno da svaki HTML dokument poseduje naziv. To nije slučaj i sa `html` i `body` elementima, koji su obavezni kako bi dokument bio validan.*

Osnovna struktura zaglavlja HTML dokumenta prikazana je slikom 2.4.



Slika 2.4. Osnovna struktura zaglavlja HTML dokumenta

Na slici 2.4. može se videti da zaglavlje HTML dokumenta može sadržati nekoliko HTML elemenata. Ipak, jedini obavezni element jeste `title`.

U ovom trenutku većina prikazanih elementa zaglavlja HTML dokumenta za nas nema preteranu važnost, s obzirom na to da smo još na početku upoznavanja HTML jezika. Jednostavno, sa različitim elementima koji pripadaju zaglavlju HTML dokumenta detaljno ćemo se upoznavati u nastavku ovog kursa, ali i u narednim kursevima, kada se za tako nešto javi potreba. Ipak, u nastavku će ukratko biti objašnjena uloga svakog od elemenata koji se mogu naći unutar HTML zaglavlja.

### **<title> element**

Element `title` je obavezni element svakog HTML dokumenta. Definiše naziv HTML dokumenta koji se koristi u nekoliko različitih situacija prilikom prikaza imena stranice u:

- prozoru browsera;
- favoritima;
- rezultatima pretrage sajtova za pretraživanje.

Jedan dokument može imati samo jedan `title` element.

Element `title` definiše se unutar `head` sekcije, a ukoliko `head` sekcija nije navedena, `title` element se definiše kao direktan potomak `html` elementa.

Ukoliko postoji `head` sekcija, dokument sa `title` elementom može izgledati ovako:

```
<!DOCTYPE html>
<html>
<head>
  <title>This is title</title>
</head>
<body>
  ...
</body>
</html>
```

S obzirom na to da `head` sekcija po pravilima HTML jezika nije obavezna, u slučaju njenog izostanka `title` element navodi se na sledeći način:

```
<!DOCTYPE html>
<html>
<title>This is title</title>
<body>
  ...
</body>
</html>
```

### **<style> element**

Element `style` koristi se za definisanje CSS stilizacije za tekući HTML dokument. Stilizacija određuje kako će HTML elementi izgledati. CSS je pojam kojem će biti posvećen drugi deo ovoga kursa.

U jednom HTML dokumentu moguće je imati veći broj `style` elemenata.



## <base> element

Element `base` definiše baznu adresu za sve relativne adrese unutar HTML dokumenta. Uloga ovog elementa prevazilazi naše trenutno znanje, te ćemo se stoga na njega vratiti u jednoj od narednih lekcija u kojoj će biti reči o linkovima.

## <link> element

Element `link` koristi se za definisanje veze sa eksternim dokumentom ili resursom. Najčešće se koristi za uključivanje eksternog fajla sa stilizacijom. `link` element može se naći samo unutar `head` sekcije, i to proizvoljan broj puta. Za ovaj element važi identično kao i za prethodni. Njime ćemo se posebno baviti u lekcijama o stilizaciji.

## <meta> element

Element `meta` koristi za definisanje metapodataka o HTML dokumentu. Metapodaci se ne prikazuju na stranici, ali se zato koriste od strane browsera, search enginea i drugih web servisa.

Element `meta` najčešće se koristi za definisanje:

- opisa stranice;
- ključnih reči;
- autora dokumenta;
- vremena kada je stranica poslednji put izmenjena;
- podataka o kompatibilnosti.

Sledeći kod ilustruje upotrebu meta elemenata:

```
<head>
  <meta charset="UTF-8">
  <meta name="description" content="HTML and CSS Fundamentals">
  <meta name="keywords" content="HTML, CSS, XML, JavaScript">
  <meta name="author" content="Vladimir Dresevic">
</head>
```

Element `Meta` uvek se piše unutar `head` sekcije, proizvoljan broj puta. U HTML5 varijanti jezika, element `meta` se formira korišćenjem samozatvarajućeg taga.

Meta-element na sebi može imati nekoliko atributa kojima se dodatno određuje njegova svrha. Atributom `charset` podešava se tip encodinga, što određuje način na koji će browser pročitati stranicu i prikazati specifične karaktere. Preporuka je da se uvek koristi `utf-8` encoding, pogotovu za HTML dokumente koji sadrže tekstualne delove srpskog jezika, ali i svih onih jezika koji poseduju neke specifične karaktere (nemački, španski, norveški, danski itd.). Korišćenjem `utf-8` encodinga osigurava se adekvatan prikaz specifičnih karaktera kao što su š, ć, ž i slično.

Ostali metapodaci elementa `meta` navode se isključivo kao parovi ključeva i vrednosti. Ključevi se definišu atributom `name`, dok se vrednosti definišu atributom `content`. Atribut `name` dozvoljava definisanje metapodataka za nekoliko predefinisanih kategorija. U prikazanom primeru su na taj način definisane informacije o opisu, ključnim rečima i autoru HTML dokumenta.

### Pitanje

Kojim se elementom definiše naziv HTML dokumenta?

- a) link
- b) meta
- c) title**
- d) html

### Objašnjenje:

*Element title je obavezni element svakog HTML dokumenta. Obično se definiše unutar head sekcije, a ukoliko head sekcija nije navedena, title element se definiše kao direktan potomak html elementa. Element title definiše naziv HTML dokumenta.*

### <script> element

script element se koristi za integraciju JavaScript koda u HTML dokument. script Element može da sadrži direktno kod ili da upućuje na eksterni fajl sa klijentskom skriptom.

Veliki deo ovog programa, u narednim kursevima, biće posvećen jeziku za kreiranje klijentskih skripti – JavaScriptu, kada će korišćenje ovog elementa zaglavlja biti detaljno objašnjeno.

### <noscript> element

U uvodnom kursu ovoga programa rečeno je da je jezik JavaScript jedna od tri osnovne komponente današnjeg weba. JavaScript je programski jezik koji se izvršava unutar browsera, a ono što možda niste znali jeste to da gotovo svi browseri poseduju opciju za deaktivaciju, odnosno blokiranje JavaScript jezika. U takvim situacijama bilo koji JavaScript kod koji HTML dokument poseduje ne može da se izvrši. Za ovakve situacije posebno je koristan noscript element.

Element noscript koristi se za definisanje alternativnog sadržaja koji će biti prikazan korisnicima u čijim je browserima onemogućeno ili nije podržano izvršavanje skripti. Ovaj element može se definisati unutar head ili body sekcije. Primer upotrebe noscript elementa:

```
<noscript>Your browser does not support JavaScript!</noscript>
```

### Primer – transformacija uvodnog primera

Za kraj, primer iz prethodne lekcije biće modifikovan i unapređen tako da ispoštuje osnovnu HTML strukturu i da inkorporira neke od elemenata zaglavlja koje smo videli u ovoj lekciji:

### Radno okruženje

```
<!DOCTYPE html>
<html>
<head>
  <title>First HTML document</title>
```

```
<meta charset="UTF-8">
<meta name="description" content="First example on HTML & CSS
Fundamentals course">
<meta name="keywords" content="HTML,CSS">
<meta name="author" content="Vladimir Dresevic">
</head>
<body>
  <h1>This is a heading</h1>
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec
luctus sapien magna, eu sollicitudin augue porttitor nec. Nulla
pharetra, mi eget maximus posuere, quam quam malesuada metus, a
volutpat mauris arcu id lorem.</p>
  <p>Vivamus neque felis, pharetra sed odio a, mattis lacinia neque.
Donec id urna quam. Fusce ac risus vel diam vestibulum maximus vel sit
amet erat.</p>
</body>
</html>
```

U navedenom primeru, unutar radnog okruženja možete napraviti izmene, tako što ćete dodati više naslova ili paragrafa.

## Rezime

- HTML jezik definiše osnovnu strukturu koju mora ispoštovati svaki HTML dokument.
- HTML dokumenti započinju DOCTYPE deklaracijom koja govori browseru koja se verzija HTML-a koristi u dokumentu.
- HTML elementi unutar dokumenta grade strukturu stabla tako što jedan element može sadržati druge HTML elemente.
- Koreni element svakog HTML dokumenta je `html` element.
- `html` element može da sadrži elemente `head` i `body`.
- Element `body` predstavlja telo HTML dokumenta.
- Element `head` je koreni element za sve podatke zaglavlja jednog HTML dokumenta.
- `head` element nije obavezno definisati kako bi stranica bila validna.
- Unutar `head` elementa mogu se naći različiti elementi od koji je jedini obavezni element `title`.
- Element `title` je obavezni element svakog HTML dokumenta koji definiše naziv HTML dokumenta; ukoliko dokument ne poseduje zaglavlje, `title` se definiše kao direktni potomak `html` elementa.