

## Sadržaj

HTML5 .....	3
1. Uvod .....	3
2. Okruženje i alati potrebni za izradu sajta.....	3
Instalacija WAMP paketa .....	4
3. HTML sintaksa .....	4
4. HTML Head.....	6
5. Rad sa tekstom (komentari, headings, paragraf).....	9
Break tagovi .....	10
Elementi za formatiranje teksta.....	10
HTML komentari, citati, entiteti, simboli, boje i charset .....	11
6. Linkovi .....	14
Link ka nekoj stranici na Internetu.....	15
Link ka lokalnim fajlovima .....	15
Link ka nekom elementu na strani.....	16
7. Slike .....	16
8. Liste .....	18
Uređena lista.....	18
Neuređena lista.....	18
9. Tabele.....	19
10. Forme .....	22
Text Input element.....	22
Radio buttons.....	22
Check boxes .....	23
Password Input .....	23
Text Area .....	23
Drop-down lista (menu).....	24
Dugme .....	24
Submit dugme.....	24

Atributi <form> i <input> tagova .....	25
Okvir forme i naslov .....	26
11. Blok i inline elementi(div i span) .....	27
12. Novi HTML5 elementi .....	27
<header> element.....	27
<section> element .....	28
<article> element .....	28
<nav> element .....	28
<aside> element.....	28
<footer> element .....	28
<audio> element .....	28
<video> element .....	28

# HTML5

## 1. Uvod

HTML(*Hyper Text Markup Language*) je standardizovani jezik za označavanje kojim se opisuje sadržaj *web* stranice. HTML održava i standardizuje *World Wide Web Consortium (W3C)*. W3C konzorcijum čine univerziteti, kompanije kao što su *Microsoft*, *Netscape* i *Macromedia*, i eksperti iz mnogih oblasti koje su vezane za *web*. Osnovan je od strane Tim Berners Lee-a i zadatak mu je da unapređuje i promoviše *web*. Jedna od njegovih uloga je da publikuje preporuke (eng, *recommendations*). Na te preporuke treba gledati kao na standarde. Preporuke W3C-a pokrivaju mnoge aspekte *web*-a kao na primer HTML, CSS, skript jezike, XML i još mnogo toga što je vezano za *web*.

Sadržaj jedne *web* stranice čine: tekst, slike, hiperlinkovi, forme, liste itd. Prva verzija ovog jezika objavljena je 1991. godine, a njen tvorac, ujedno izumitelj WWW-a(*World Wide Web*) i osnivač W3C-a, bio je Tim Berners Lee. 2014. godine nastaje šesta i najnovija verzija – HTML5.

HTML jezik čine tekst i tagovi (znakovi). Svrha tagova jeste da saopšte *web browser*-u na koji način da prikaže tekst napisan između dva taga. Kada se prikaže html stranica u prozoru internet pretraživača, korisnik može da vidi samo sadržaj, ne i html tagove. Dakle, zahvaljujući tagu pretraživač zna kako da protumači njegov sadržaj.

Svrha *web browser*-a (*Chrome*, *Mozilla*, *IE*, *Opera*) je da pročita html dokument i prikaže njegov sadržaj.

CSS(*Cascading Style Sheets*) tehnologija definiše izgled i položaj elemenata na *web* stranici. HTML jezik se vezuje za pitanje ŠTA, a CSS za pitanje KAKO prikazati web sadržaj. CSS počinje da se koristi od verzije HTML 4.0.

## 2. Okruženje i alati potrebni za izradu sajta

Najpopularnija arhitektura u okviru koje se kreiraju *web* sajtovi jeste *Windows*-ovo okruženje – *WAMP*. Na *Macintosh* operativnom sistemu koristi se *MAMP*.

Sam naziv “WAMP” predstavlja akronim formiran od inicijala *Windows* operativnog sistema i glavnih komponenti okruženja: *Apache*, *MySQL*, *PHP*. **Windows** je operativni sistem na kojem su instalirane tri glavne komponente paketa. **Apache** je *open source web server* i omogućava simulaciju klijent/server okruženja na lokalnoj mašini. **MySQL** je sistem za upravljanje bazama podataka. Za lak pristup *MySQL* bazi u okviru *WAMP*

paketa koristi se *PhpMyAdmin* koji je sastavni deo tog paketa. *PHP (Hypertext Pre-processor)* je *open source* jezik koji se koristi za razvoj *server-side* aplikacija, kao i dinamičkog *Web* sadržaja.

### Instalacija WAMP paketa

Sa sajta <http://www.wampserver.com> potrebno je *download*-ovati najnoviju verziju *Wamp* servera (trenutno najnovija verzija je WAMP 2.5). Klikom na instalacioni fajl pokreće se kratak proces instalacije. Instalacija je jednostavna i ne razlikuje se od uobičajenih. Po završenoj instalaciji potrebno je pokrenuti WAMP server. To činimo tako što klikom na Start/All Programs/Wamp Server pokrećemo link "start Wamp Server". Kada smo to učinili, u desnom kraju Taskbar-a nalazi se ikonica Wamp server, klikom na ikonicu otvara se meni preko kog imamo priliku da pristupimo *Localhost*-u, *phpMyAdmin*-u, *www* folderu. *Localhost* omogućava prikaz *web* sajta u Internet pretraživaču. Klikom na *localhost* link otvara se *Wamp Homepage*, u kojem pronalazimo svoj projekat. Svi sajtovi se čuvaju u C:/WAMP/WWW. Klikom na link *www directory* ulazimo u *www* folder u kojem se nalaze svi sajtovi koje smo do tada kreirali, razvrstani po folderima. Ukoliko je "primersajta" naziv foldera u kojem su smešteni svi fajlovi jednog sajta, za otvaranje sajta u Internet pretraživacu potrebno je uneti adresu <http://localhost/primersajta> .

Radni prostor za kreiranje HTML strana čine tekst editor (*Notepad ++*) i *web browser* u kome se stranice pregledaju. Takođe, neretko se koriste i napredni HTML editori kao što su: *Adobe Dreamwiever*, *Kompozer*, *CoffeeCup*...

### 3. HTML sintaksa

HTML dokument sačinjavaju HTML elementi.

Osnovni format HTML elementa je:

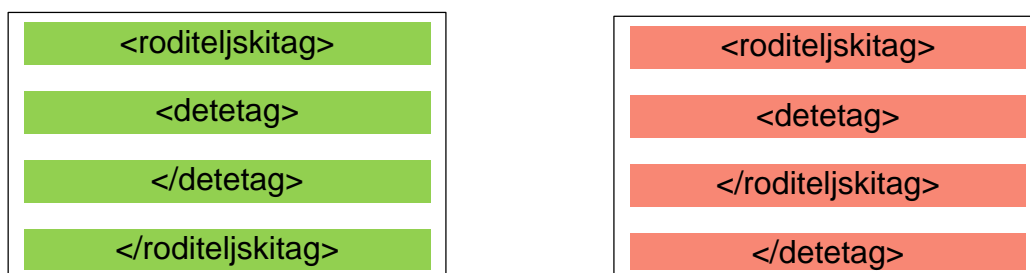
<nazivtaga>sadržaj</nazivtaga>

Prvi tag (<nazivtaga>) je početni ili otvarajući tag. Drugi tag (</nazivtaga>) se naziva poslednji ili zatvarajući. Kod zatvarajućeg taga uvek se stavlja *slash*(kosa crta „ / ") ispred samog naziva. Kada pretraživač pri iščitavanju fajla naiđe na poslednji tag jednog HTML elementa, on zna da je završen prikaz sadržaja tog taga. HTML tagovi se uvek pišu u paru (otvarajući - zatvarajući). Postoji nekoliko HTML tagova koji se nazivaju

prazni, jer ne poseduju sadržaj, pa samim tim nemaju ni zatvarajući tag. Kod ovakvih tagova predlaže se da sam otvarajući tag bude zatvoren, tako što se nakon naziva taga stavlja kosa crta (<nazivtaga/>) U nastavku će biti više reči o praznim tagovima.

Nazivi HTML tagova se uobičajeno pišu malim slovima, mada ne postoji pravilo po kojem se ne smeju upotrebljavati i velika slova. Dakle, HTML tagovi nisu *case sensitive*. Browser će isto protumačiti tagove <a> i <A>. Preporuka je da se koriste mala slova. HTML poseduje veliki broj tagova, a svi oni su predefinisani.

U HTML dokumentu podržano je ugnježdavanje elemenata. To znači da se u okviru jednog elementa(roditelj) mogu naći drugi elementi (dete). Ali, mora se obratiti pažnja prilikom zatvaranja tagova. Prvo se zatvara tag koji je poslednji otvoren (dete), a potom prvi otvoreni tag(roditelj).



HTML elementi mogu posedovati atribute. Atributi daju dodatne informacije o HTML elementu, navode se u otvarajućem tagu elementa i uvek dolaze u paru:

<nazivtaga nazivatributa="vrednostatributa">...sadržaj...</nazivtaga>

Nazivi atributa nisu *case sensitive*, ali je preporuka da se uvek pišu malim slovima. Vrednost atributa se mora navesti pod navodnicima. Koriste se ili dvostruki ili jednostruki navodnici. Preporuka je da se uvek koriste dvostruki izuzev u slučajevima kada sama vrednost atributa u sebi sadrži dvostruke navodnike, tada se koriste jednostruki. Najčešće korišćeni atributi su: *src*, *href*, *style*, *id*, *class*, *value*, *alt*, *title* itd. U daljem tekstu biće više reči o ovim atributima.

Osnovna struktura HTML dokumenta:

```
<!DOCTYPE html>

<html>

<head>

<!--Sadržaj koji nije vidljiv u browser-u, nosi informacije o dokumentu-->
```

```
</head>
<body>
<!--Sadržaj koji se prikazuje u browser-u-->
</body>
</html>
```

`<html>` je osnovni tag u HTML dokumentu. Obuhvata sve ostale tagove. Kada *browser*-u pošaljemo zahtev za html stranicom, on na osnovu ovog taga može da zna da je u pitanju html dokument. Svaki html dokument mora početi tagom `<html>` i završiti se tagom `</html>`.

`<head>` sadrži informacije o dokumentu koje nisu vidljive u *browser*-u, a koje *browser* koristi kao pomoć pri učitavanju stranice.

`<body>` obuhvata sav sadržaj koji je vidljiv posetiocu *web* stranice.

## 4. HTML Head

U okviru `<head>` taga mogu da se nađu informacije (metapodaci) kao što su: naslov dokumenta, stilovi za formatiranje sadržaja, linkovi, skripte itd.

Tagovi koji se mogu naći unutar para `<head></head>` su: `<title>`, `<link>`, `<style>`, `<script>`, `<meta>` i `<base>`.

Prva linija koda koja se piše i pre glavnog `<html>` taga je:

```
<!DOCTYPE html>
```

Ova deklaracija definiše tip HTML dokumenta i pomaže *browser*-u da ispravno prikaže *web* stranicu. Navedena deklaracija odnosi se na HTML5 verziju. Prethodne verzije html-a imale su drugačije deklaracije sa mnogo više teksta.

```
<title>
```

Obavezan tag u svim html verzijama kojim se definiše naslov dokumenta. Naslov se prikazuje u *title bar*-u prozora *browser*-a. Jedinstveni identifikator svakog html dokumenta. Primer:

```
<title>Moja prva web strana</title>
```

### <link>

Definiše vezu html dokumenta sa drugim eksternim fajlovima. Najčešće se koristi za povezivanje sa eksternim CSS fajlom. Primer:

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

O atributima ovog taga biće više reči u tekstu koji sledi.

### <style>

U okviru ovog taga ispisuju se stilovi/izgled različitih elemenata, tj. način njihovog prikaza u prozoru pretraživača. Primer:

```
<style>

p {font-size:22px; color: red;}

a:hover {color:#50786;}

</style>
```

### <script>

Služi za definisanje JavaScript koda. *JavaScript* je skripta koja se za razliku od PHP skripte, izvršava na klijentskoj strani. Povećava dinamičnost HTML stranice. Npr.

```
<script>

document.getElementById("demo").innerHTML = "Hello JavaScript!";

</script>
```

### <meta>

Meta tagovi nam omogućavaju da postavimo dodatne informacije o *web* stranici koje će omogućiti da se ona lakše pronađe na Internetu. Tačnije, *web* pretraživači rade po sledećem principu: svaki od pretraživača sadrži bazu podataka u kojoj su podaci o *web* stranicama. Na osnovu kriterijuma pretrage zadatog od strane korisnika, pretražuje se baza podataka i prikazuju rezultati. Rezultat pretrage su *web* stranice koje sadrže traženi kriterijum. Na većini ovih *web* stranica traženi kriterijum se nalazi na početku sadržaja stranice, tako da osim ako je sam početak stranice veoma deskriptivan, korisnik neće dobiti pravu sliku onoga što je na stranici. Sintaksa meta taga za dodavanje opisa stranici izgleda ovako:

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

Maksimalna dužina meta tagova iznosi 1024 karaktera, počev od prvog < i zaključno sa poslednjim > u meta tagu, što znači da uneti opisi treba da budu kratki i precizni. Još jedan koristan meta tag koji će olakšati pronalaženje *web* stranice preko pretraživača je omogućava dodavanje ključnih reči (*keywords*) koje opisuju stranicu. Sintaksa ovog taga izgleda ovako:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

Takođe u okviru meta taga možemo navesti i naziv autora dokumenta.

```
<meta name="author" content="Pavle Peric">
```

Značajna vrsta ovog taga koja se često koristi, jeste meta tag za automatski prelazak na novu HTML stranicu. Koristi se u tri slučaja:

- Kada se rade ispravke na samoj HTML stranici i korisnik se u tom sličaju preusmerava na novu stranicu.
- Kada *web* sajt poseduje uvodnu stranicu čiji prikaz traje kratak vremenski period.

Izgled takvog taga je sledeći:

```
<meta http-equiv="refresh" content="x; url=newpage_or_URL.html">
```

gde x predstavlja vreme koliko će ova stranica biti prikazana (izraženo u sekundama), a URL je adresa na koju će se izvršiti preusmeravanje posle isteka tog vremena.

Zatim, tag se koristi kada hoćemo da saopštimo *browser*-u koliko često da osvežava dokument. Primer:

```
<meta http-equiv="refresh" content="30">
```

```
<base>
```

Ovaj tag služi za navođenje zajedničke *url* putanje do svih linkova na stranici. Sadrži dva atributa: *href* i *target*. U *href* atributu navodi se putanja do svih relativnih *url*-ova. Atribut *target* specificira da li će se linkovi otvarati u novoj kartici *browser*-a ili u postojećoj. Primer:

```
<base href="http://www.w3schools.com/images/" target="_blank">
```

Vrednost “\_blank” označava otvaranje linkova u novoj kartici.

Izgled head dela html dokumenta:

```
<!DOCTYPE html>
```



```
<html>
<head>
  <title>Moja prva web strana</title>
  <base href="http://www.w3schools.com/images/" target="_blank">
  <link rel="stylesheet" type="text/css" href="stilovi.css" />
  <meta name="description" content="Free Web tutorials on HTML and CSS">
  <meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
  <meta name="author" content="Pavle Peric">
  <meta http-equiv="refresh" content="x; url=newpage_or_URL.html">
  <meta http-equiv="refresh" content="30">
  <script>
    document.getElementById("demo").innerHTML = "Hello JavaScript!";
  </script>
  <style>
    p {font-size:22px; color: red;}
    a:hover {color:#50786;}
  </style>
</head>
</html>
```

<head> tag nije obavezan, može da se izostavi, *browser* će i tada znati šta treba da uradi sa podacima pre <body> taga.

## 5. Rad sa tekстом (komentari, headings, paragraf)

U html-u naslove (zaglavlja) ispisujemo tako što sadržaj naslova stavljamo između para tagova <hN> i </hN>.

`<hN>` Ovo je moj prvi naslov `</hN>`,

gde je N broj koji određuje veličinu naslova i kreće se od 1 do 6 (`<h1>` označava najznačajniji naslov u dokumentu, dok `<h6>` označava najmanje važan naslov). Preporučuje se upotreba `<h1>` naslova samo jednom na strani.

## Break tagovi

1. Najznačajniji *break* element za rad sa tekstom jeste par `<p></p>`. Ovaj element se koristi kada tekst hoćemo da prenesemo u novi red

`<p>`Ovo je moj prvi paragraph(pasus).`</p>`

*Web browser* automatski dodaje prazan red pre i posle paragrafa.

2. `<br/>` tag se koristi da prelomimo tekst u novi red. Ne ponaša se kao paragraf jer ne dodaje jedan prazan red pre i posle prelomljenog teksta.

Ovo je moj prvi `<br/>`paragraf(pasus).

`<br/>` tag je primer praznog taga jer ne poseduje zatvarajući tag, pa samim tim ni sadržaj.

3. `<hr/>` tag kada se umetne u tekst, iscrtava horizontalnu liniju kojom se razdvaja tekst pre i posle ovog taga.

Ovo je moj prvi `<hr/>`paragraf(pasus).

Ovo je drugi primer samozatvarajućeg, praznog taga.

## Elementi za formatiranje teksta

Kao i u *Microsoft Word*-u, HTML može preneti *web browser*-u da prikaže neke delove teksta drugačije (podebljano, iskošeno, manjim slovima, markirano, kao gornji ili donji eksponent).

Tagovi koji se koriste za formatiranje teksta u html-u su: `<b>`, `<i>`, `<small>`, `<sub>`, `<sup>`, `<ins>`, `<mark>`.

`<b>` - koristi se za podebljavanje teksta.

Element : `<p>Ovo je moj <b>prvi paragraf.</b></p>`

Rezultat: Ovo je moj **prvi paragraf.**

`<i>` - za iskošeni tekst

Element : `<p>Ovo je moj <i>prvi paragraf.</i></p>`

Rezultat: Ovo je moj *prvi paragraf.*

`<small>` - sadržaj između ovih tagova prikazuje se umanjenim slovima.

Element : `<p>Ovo je moj <small>prvi paragraf.</small></p>`

Rezultat: Ovo je moj prvi paragraf.

`<sub>` - za navođenje donjeg eksponenta (indeksa).

Element : `<p>Ovo je moj <sub>prvi paragraf.</sub></p>`

Rezultat: Ovo je moj <sub>prvi paragraf.</sub>

`<sup>` - za navođenje gornjeg eksponenta (stepena).

Element : `<p>Ovo je moj <sup>prvi paragraf.</sup></p>`

Rezultat: Ovo je moj <sup>prvi paragraf.</sup>

`<ins>` - za podvlačenje teksta (underline).

Element : `<p>Ovo je moj <ins>prvi paragraf.</ins></p>`

Rezultat: Ovo je moj prvi paragraf.

`<mark>` - za markiranje teksta.

Element : `<p>Ovo je moj <mark>prvi paragraf.</mark></p>`

Rezultat: Ovo je moj **prvi paragraf.**

## HTML komentari, citati, entiteti, simboli, boje i charset

### Komentari

Za pisanje komentara u html dokumentu koristi se izraz:

<!-- - tekst komentara - - >

Komentari se ispisuju radi lakše preglednosti html koda i mogu služiti kao mali podsetnik. Oni se ne prikazuju na učitanoj *web* stranici.

### Citati

Za navođenje citata na *web* stranici koriste se dve vrste tagova: <q> i <blockquote>.

<q> - tag se koristi za navođenje kraćih citata. A tekst unutar ovog para tagova u *browser*-u se ispisuje pod znacima navoda.

Element :                <p>Ovo je moj prvi citirani tekst <q>Sve sto mozes da zamislis, mozes i da ostvaris</q>.</p>

Rezultat:                Ovo je moj prvi citirani tekst „Sve sto mozes da zamislis mozes i da ostvaris“.

<blockquote> - koristi se za navođenje većih citata, preuzetih sa nekog drugog izvora. Po *default*-u ovakav citat *browser* prikazuje u novom redu i uvučeno u odnosu na ostali tekst.

Element :                <p>Ovo je moj citat <blockquote cite="http://www.w3.org/html/"> In October 2014, the HTML working group published HTML5 as W3C Recommendation. This specification defines the fifth major revision of the Hypertext Markup Language (HTML), the format used to build Web pages and applications, and the cornerstone of the Open Web Platform.</blockquote></p>

Rezultat:                Ovo je moj citat

In October 2014, the HTML working group published HTML5 as W3C Recommendation. This specification defines the fifth major revision of the Hypertext Markup Language (HTML), the format used to build Web pages and applications, and the cornerstone of the Open Web Platform.

Kod ovakve vrste citata navodi se atribut *cite* čija je vrednost adresa *web* stranice sa koje je citat preuzet.

### Entiteti i simboli

Rezervisani znakovi u HTML-u se moraju zameniti entitetima. Rezervisani znakovi su obično <, >, &nbsp;, &. Da bismo, recimo, znak manje (<) napisali u tekstu, a da ga

*browser* ne protumači kao deo taga, moramo ga u kodu zameniti odgovarajućim entitetom. Postoje dva formata entiteta koje možemo koristiti:

&naziv\_entiteta ili &#broj\_entiteta

Entiteti su unapred definisani.

| Prikaz na web strani | Naziv entiteta | Broj entiteta |
|----------------------|----------------|---------------|
| razmak u tekstu      | &nbsp;         | &#160;        |
| <                    | &lt;           | &#60;         |
| >                    | &gt;           | &#62;         |
| &                    | &amp;          | &#38;         |

Entitete koristimo i kada želimo da predstavimo simbole koji ne postoje na tastaturi. To su različiti matematički i tehnički simboli, valute i drugi.

Videti listu simbola na sajtu [http://www.w3schools.com/html/html\\_symbols.asp](http://www.w3schools.com/html/html_symbols.asp)

Za umetanje razmaka između reči u tekstu koristi se izraz *&nbsp;*. Možemo koristiti onoliko znakova u nizu koliki želimo da bude razmak između reči.

Element :           <p>Ovo je moj &nbsp;&nbsp;&nbsp; prvi paragraph.</p>

Rezultat:           Ovo je moj   prvi paragraf.

## Boje

Monitori novije generacije mogu da prikažu i do 16 384 različite boje. Svaka boja u sebi sadrži vrednost crvene (*Red*), zelene (*Green*) i plave (*Blue*) – RGB model. Boje se mogu predstaviti na tri načina:

1. Naziv boje. Npr. Aqua, Coral, Chocolate, Gray,...

2. Heksadecimalno. Vrednosti za boje (rgb) se zadaju u heksadecimalnom zapisu (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Ukupno se navodi šest cifara, po dve za svaku vrednost. Ispred cifara se stavlja znak #. Npr. Heksadecimalna Indigo boja glasi „4B0082“, gde je „4B“ vrednost crvene, „00“ vrednost zelene, a „82“ vrednost plave. Crna boja se označava sa „000000“, a bela sa „FFFFFF“.

3. rgb zapis. Vrednosti za sve tri boje su zadate u rasponu od 0 do 255. Maksimalne vrednosti (R=255, G=255, B=255) daće belu, a minimalne vrednosti (R=0, G=0, B=0) prikazaće crnu boju. Npr. Indigo boja u rgb zapisu glasi rgb(75, 0, 130), gde 75

predstavlja intenzitet crvene, 0 intenzitet zelene, a 130 intenzitet plave boje u Indigo boji.

### Set karaktera (enkodiranje)

U svakom HTML dokumentu mora se navesti standard za enkodiranje. *Browser*-u je neophodna ova informacija, jer na taj način zna kako da prikaže sve karaktere i simbole koje unosimo na sajt. U HTML5 verziji koristi se *UTF-8* standard. Ovaj standard pokriva skoro sve karaktere i simbole na svetu. Sigurno ste se zapitali kako slovo „Ž“ da prikazemo na *web* stranici, jer u većini slučajeva ne prikazuje se pravilno. Za svako latinično slovo i mnoge druge simbole *UTF-8* standard nudi zamenu (u vidu dvocifrenog ili trocifrenog broja, sa znakovima `&#` ispred broja) koja se ispisuje u tekstu umesto specifičnog slova ili simbola. Tako umesto slova Ž potrebno je da unesemo „`&#381`“.

Kao što smo prethodno naveli, sve informacije koje su od značaja *browser*-u prilikom čitanja HTML dokumenta, navode se u `<head>` delu dokumenta. Standard koji stranica koristi za enkodiranje, navodi se u okviru `<meta>` taga na sledeći način.

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

Gde je „charset“ atribut sa vrednošću UTF-8.

Prethodne verzije setova karaktera za enkodiranje bile su hronološki: ASCII, ANSI, ISO-8859-1.

## 6. Linkovi

Svaka web stranica poseduje linkove ka nekim izvorima. U HTML-u linkovi se kreiraju korišćenjem `<a>` taga:

```
<a href="url">tekst linka</a>,
```

gde je *href* atribut čija je vrednost odredišna *url* adresa, a „tekst linka“ predstavlja tekst na kojem je postavljen link na web stranici, po default-u je plave boje i podvučen. Klikom na njega bivamo usmereni na odredišnu adresu. Na mestu „tekst linka“ može da se nađe slika ili bilo koji html element.

Kada su linkovi u pitanju razlikujemo tri situacije u zavisnosti na koju lokaciju nas mogu odvesti: link ka lokalnim fajlovima, link ka nekoj stranici na Internetu, link ka nekom elementu na strani.

## Link ka nekoj stranici na Internetu

Primer jednog takvog linka je sledeći:

```
<a href="http://www.w3schools.com/html/default.asp">HTML tutorijali i primeri za vezbanje</a>
```

Kada korisnik poseti tekst sa linkom (*hypertext*) "HTML tutorijali i primeri za vezbanje", on biva preusmeren na adresu <http://www.w3schools.com/html/default.asp>.

## Link ka lokalnim fajlovima

Jedan sajt može imati veći broj stranica, slika i drugim multimedijalnih sadržaja. Lokalni fajlovi su fajlovi koji se nalaze na istom serveru i pripadaju jednom projektu tj. nalaze se u okviru projektnog foldera. Jedan takav folder mora biti pregledan i organizovan na što je moguće bolji način radi lakšeg snalaženja u njemu, to uključuje postojanje foldera u kojem se nalaze samo slike, foldera za css fajlove, foldera u kojem su smešteni JavaScript fajlovi i nekoliko drugih. Fajlovi koji predstavljaju stranice web sajta se obično nalaze u korenom(root, glavnom) tj. projektnom folderu. Kod kreiranja linkova ka lokalnim fajlovima razlikujemo tri slučaja: odredišni fajl se nalazi u istom folderu (isti nivo hijerarhije), odredišni fajl se nalazi u podfolderu (nizi nivo hijerarhije), odredišni fajl se nalazi u nadfolderu (viši nivo hijerarhije).

1. Odredišni fajl se nalazi u istom folderu.

Pristupanje fajlu koji se nalazi na istom nivou hijerarhije kao i fajl od kojeg vodi link, se izvodi na sledeći način. Pretpostavimo da je prva.html stranica na kojoj se nalazi tekst sa linkom ka stranici druga.html. Pri čemu se stranice nalaze u istom folderu. HTML element izgleda ovako:

```
<a href="druga.html">Ovde možete pogledati našu drugu stranicu</a>
```

Kada korisnik poseti link [Ovde možete pogledati našu drugu stranicu](#), biva preusmeren na stranicu druga.html.

2. Odredišni fajl se nalazi u folderu nižeg nivoa hijerarhije.

Pretpostavimo da smo projektnom folderu dali naziv "mojprojekat". U okviru tog foldera nalazi se fajl prva.html i folder "slike", a unutar foldera "slike" jedna slika sa nazivom "primer.jpg". Želimo da sa stranice prva.html vodi link ka slici primer.jpg. To činimo sledećim html elementom:

```
<a href="slike/primer.jpg">Ovde možete pogledati primer slike</a>
```

Da bismo pristupili slici na nižem nivou hijerarhije najpre moramo da navedemo folder (ili foldere) a potom sliku koja se tu nalazi. Kada navodimo putanju uvek polazimo od trenutnog fajla, tj. mesta gde se on nalazi u hijerarhiji.

3. Odredišni fajl se nalazi u folderu višeg nivoa hijerarhije.

Pretpostavimo da se fajl prva.html nalazi na sledećem nivou hijerarhije: mojprojekat/fajlovi/prva.html. Takođe, odredišni fajl kojem želimo da pristupimo putem linka sa prva.html, nalazi se na putanji: mojprojekat/druga.html. Element za ubacivanje linka na stranicu prva.html izgleda ovako:

```
<a href= "../druga.html">Ovde možete pogledati primer slike</a>
```

Da bismo došli do odredišne stranice koja je na višem nivou (druga.html), moramo da izađemo iz trenutnog foldera (fajlovi), to činimo pomoću izraza ("..") – jedan folder unazad). Tako smo ušli u sadržaj foldera "mojprojekat", a u njemu se nalazi tražena stranica. U sličaju da je potrebno da izađemo iz više foldera, navodimo više izraza ("..").

Pristup lokalnim fajlovima se najčešće događa u navigaciji.

### Link ka nekom elementu na strani

Kada su u pitanju duge web stranice ili jednostrani sajtovi sa više sekcija, da bi se ubrzao pristup pojedinačnim sekcijama, kreiraju se linkovi koji vode ka pojedinačnim sekcijama. Primer jednog takvog linka je sledeći:

```
<a href= "#petasekcija">Link ka petoj sekciji</a>
```

Element sekcije mora da poseduje jedinstveni identifikator (id) kao atribut koji ima svoju vrednost. U našem sličaju vrednost atributa id je "petasekcija". Kao putanja do sekcije navodi se vrednost njenog id atributa sa znakom # ispred vrednosti. O id atributu biće više reči u narednom tekstu.

Kada kliknemo na bilo koji link, browser će novu stranicu otvoriti u istom tab-u. Ukoliko to želimo da sprečimo i da kažemo browser da svaki link otvara u novoj kartici, <a> tagu moramo da dodamo jedan atribut "target". Vrednost atributa postavljamo na "\_blank" kao u primeru:

```
<a href= "druga.html" target= "_blank">Ovde možete pogledati našu drugu stranicu</a>
```

## 7. Slike



Kada sliku ubacimo odmah nakon teksta ili u tekst ona će se pojaviti u redu sa tekstom bez prelaska u novi red. Takva slika se naziva „inline“ slika. Za prikaz slika na HTML stranici koristi se <img> tag. To je prazan, samozatvarajući tag.

HTML format za inline image tag izgleda ovako:

```
,
```

gde je slika.jpg JPG slika koja je smeštena u istom folderu gde je smešten i HTML dokument. Atribut „src“ je obavezan jer specificira putanju do slike.

Kao i u prethodnom poglavlju sa linkovima, i kod prikaza slika se navode putanje, samo što se u ovom slučaju koristi *src* atribut umesto *href* atributa. Slika može da bude lokalna, tj. nalazi se u folderu projekta (relativna putanja) ili da se nalazi na nekoj adresi na Internetu (apsolutna putanja).

Ako želimo da slika ne bude „inline“ već „block“ (tj. da se prikaže u posebnom redu) potrebno je da <img> tag okružimo blok elementom (<p>, <div>). Npr.

```
<p></p>
```

Uvek je potrebno naznačiti dimenzije( širina i visina) slike kako bi browser brže prikazao sliku, pa i samu stranicu. Za definisanje dimenzija koriste se dva atributa u <img> tagu, to su : „width“ i „height“. Dimenzije se izražavaju u pikselima, po defaultu.

```

```

Dimenzije mogu da se izraze i procentualno ili u drugim jedinicama mere.

Sledeći neizostavan atribut je „alt“. Vrednost ovog atributa browser koristi kada iz nekog razloga ne može da prikaže sliku, već umesto nje prikazuje sadržaj atributa. Primer.

```

```

U prethodnom odeljku o linkovima smo rekli da se *hyperlink* može postaviti i na druge elemente, ne samo na tekst. Posle teksta drugi najčešće korišćeni element u ovom slučaju, jesu slike. Link se postavlja na sliku na sledeći način:

```
<a href="druga.html">  
      
</a>
```

## 8. Liste

Lista predstavlja skup nabiranja (stavki). U HTML-u liste se klasifikuju na dva tipa: uređene i neuređene liste. Razlika je u znaku koji se nalazi ispred nabiranja. Kod uređenih bitan je redosled stavki, a znakovi su brojevi, slova, rimski brojevi. Kod neuređenih listi, gde nije bitan redosled navođenja stavki, obično se koriste različiti geometrijski oblici (kružići, kvadrati, zvezde,...) ili slike.

### Uređena lista

Za uređene liste (ordered list) koriste se `<ol>` `</ol>` tagovi. Kao sadržaj ovog elementa navode se stavke liste, a označavaju se tagovima `<li>` `</li>`. Primer:

`<ol>`

`<li>Prva stavka</li>`

1. Prva stavka

`<li>Druga stavka</li>`

rezultat →

2. Druga stavka

`<li>Treća stavka</li>`

3. Treća stavka

`</ol>`

`<ol>` tag označava početak i kraj liste, i `<li>` označava svaku stavku liste. Browser će po defaultu koristiti oznake počev od „1”. Da bismo promenili način prikaza stavki liste, koristimo atribut „type”. Vrednosti ovog atributa mogu da budu: 1 (default), A, a, I (rimski broj), i (rimski mali broj).Primer:

`<ol type="I">`

`<li>Prva stavka</li>`

I. Prva stavka

`<li>Druga stavka</li>`

rezultat →

II. Druga stavka

`<li>Treća stavka</li>`

III. Treća stavka

`</ol>`

### Neuređena lista

U sintaksi neuređena lista (unordered list) se razlikuje od uređene po tagovima koje koristi za početak i kraj liste, to su `<ul>` `</ul>`. Tagovi za stavke su isti `<li>` `</li>`. Primer:

`<ul>`

```
<li>Prva stavka</li>
```

```
<li>Druga stavka</li>
```

```
<li>Treća stavka</li>
```

```
</ul>
```

rezultat

- Prva stavka
- Druga stavka
- Treća stavka

HTML5 verzija ne podržava korišćenje „type” atributa u neuređenim listama. Za promenu znakova neuređene liste koriste se stilovi (upotreba CSS-a) koji će kasnije biti objašnjeni.

Liste mogu biti ugnježdene jedna u drugu. Tačnije jedna stavka može da sadrži sopstvene stavke itd. Neuređena lista u sebi može da sadrži uređenu i obrnuto. Primer:

```
<ul>
```

```
<li>Prva stavka</li>
```

```
<li>Druga stavka
```

```
<ol>
```

```
<li>Prva stavka druge stavke</li>
```

```
<li>Druga stavka druge stavke</li>
```

```
<li>Treća stavka druge stavke</li>
```

```
</ol>
```

```
</li>
```

```
<li>Prva stavka</li>
```

```
</ul>
```

rezultat

- Prva stavka
- Druga stavka
  1. Stavka prva
  2. Stavka druga
  3. Stavka treća
- Treća stavka

## 9. Tabele

Za prikaz tabela u HTML-u se koriste tagovi `<table></table>`. Redovi tabele se označavaju tagovima `<tr></tr>`. Svaki red se sastoji od ćelija (podataka). Svaka pojedinačna ćelija se označava tagovima `<td></td>`. Broj ćelija u redu, određuje broj kolona tabele. Primer:

|        |    |            |
|--------|----|------------|
| Milica | 30 | racunovođa |
| Petar  | 25 | arhitekta  |
| Milan  | 42 | masinovođa |

rezultat

```

<table>
<tr>
<td>Milica</td>
<td>30</td>
<td>racunovođa</td>
</tr>
<tr>
<td>Petar</td>
<td>25</td>
<td>arhitekta</td>
</tr>
<tr>
<td>Milan</td>
<td>42</td>
<td>masinovođa</td>
</tr>
</table>

```

Na prethodnom primeru predstavljena je osnovna struktura tabele. Sama tabela nije formatirana, već su samo podaci razvrstani po redovima i kolonama. Prethodna tabela ne poseduje okvir. Za dodavanje okvira i drugih stilova koristi se CSS, koji će biti opisan u narednim poglavljima. U ranijim verzijama HTML-a atribut „border“ se dodavao <table> tagu, sada se taj način ređe praktikuje. Tag izgleda kao na primeru:

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

gde je „border“ atribut sa vrednošću 1, tj. debljina okvira je 1px. Ovim se okvir dodaje oko cele tabele i oko svake ćelije u tabeli.

Navikli smo da kolone tabele imaju svoje nazive. U html-u se naslovi kolona zadaju preko tagova <th></th>.

| Ime    | Broj godina | Zanimanje  |
|--------|-------------|------------|
| Milica | 30          | racunovođa |
| Petar  | 25          | arhitekta  |

```

<table border="1">
<tr>
<th>Ime</th>
<th>Broj godina</th>
<th>Zanimanje</th>
</tr>
<tr>
<td>Milica</td>
<td>30</td>
<td>racunovođa</td>
</tr>
<tr>
<td>Petar</td>
<td>25</td>

```

rezultat

```
<td>arhitekta</td>
</tr>
</table>
```

Browser po defaultu sadržaj unutar <th></th> tagova centrira po sredini ćelije i podeblja.

Sledeće što možemo da uradimo sa tabelom je da koristimo attribute koji govore na koliko redova i kolona da se prostire jedna ćelija tabele. To su „colspan“ i „rowspan“ atributi koje možemo dodavati tagovima <td> i/ili <th>.

„colspan“ atribut kada se doda <td> tagu jedne ćelije proširuje ćeliju na određeni broj kolona (default je 1). Primer:

```
<table border="1">
<tr>
<th>Ime</th>
<th>Zanimanje</th>
</tr>
<tr>
<td>Milica</td>
<td>racunovođa</td>
<td>programer</td>
</tr>
</table>
```

rezultat →

| Ime    | Zanimanje  |           |
|--------|------------|-----------|
| Milica | racunovođa | programer |

„rowspan“ atribut proširuje ćeliju na dva ili više redova.

```
<table border="1">
<tr>
<th>Ime</th>
<td>Milica</td>
</tr>
<tr>
<th rowspan="2">Zanimanje</th>
<td>racunovođa</td>
</tr>
<tr>
<td>programer</td>
</tr>
</table>
```

rezultat →

| Ime       | Milica     |
|-----------|------------|
| Zanimanje | racunovođa |
|           | programer  |

Tabela može imati naslov, a on se zadaje tagovima <caption></caption>. Ovaj tagovi sa tekstom naslova kao sadržajem moraju biti smešteni odmah ispod <table> taga.

## 10. Forme

HTML forme se koriste za prikupljanje korisničkih podataka. HTML tagovi za formu su `<form></form>`. Forme se sastoje iz elemenata, a to mogu da budu: jednolinijska polja za unos teksta (text fields), radio buttons, check boxes, padajuća lista (drop-down list), polje za unos većeg teksta (text area), submit dugme, password polje.

Neki elementi forme koriste isključivo `<input>` tag sa „type“ atributom koji govori o kom elementu je reč. Drugi elementi koriste sopstvene tagove (textarea, drop-down list, button).

### Text Input element

Jednolinijska polja za unost teksta (Ime. Prezime, email,...). Default širina je 20 karaktera, a mogu se kreirati i polja sa drugim veličinama unošenjem *size* vrednosti. Može se ograničiti i broj unosa karaktera podešavanjem vrednosti u MAXLENGTH atributu. Text input polja će biti prazna kada se stranica učitava, osim ako korisnik ne zada početni tekst kroz VALUE opciju. Primer:

```
<form>
```

```
  <input type="text" size="20" maxlength="15" name="ime" value="Ime"/><br/>
```

```
  <input type="text" size="30" maxlength="25" name="prezime" value="Prezime"/>
```

```
</form>
```

Prvo input polje za unos imena široko je 20 karaktera, maksimalan broj karaktera koje možemo uneti u ovo polje je 15, naziv polja je „ime“, podrazumevana vrednost u polju koja će biti ispisana kada se stranica učitava je „Ime“.

Atribut „name“ definiše naziv Input elementa. Vrednost ovog atributa se koristi za referenciranje podatka unetog od strane korisnika, nakon što se submit-uje forma.

### Radio buttons

Korisnik može izabrati samo jednu opciju od ponuđenih. Kada izabere željenu opciju ostale postaju deselektovane (nedostupne). Jedan set radio dugmića može imati više ponuđenih opcija (radio dugmića). Svi dugmići u okviru istog seta imaju istu „name“ vrednost, a da bi se međusobno razlikovali koristi se „value“ atribut, čija vrednost zajedno sa vrednošću „name“ atributa se koristi za referenciranje podatka.

Dodavanjem opcije CHECKED jednom od dugmića u setu, on će biti selektovan po default-u, kada se stranica učitava.

`<form>`

```
<input type="radio" name="brend" value="nike">Nike</input><br/>
```

```
<input type="radio" name="brend" value="adidas" checked>Adidas</input><br/>
```

```
<input type="radio" name="brend" value="reebok">Reebok</input>
```

`</form>`

U okviru jedne forme možemo napraviti više setova radio dugmića, iz tog razloga se koriste dva atributa „name“ (identifikuje kom setu podatak pripada) i „value“ govori kom dugmiću pripada taj podatak.

### Check boxes

Ovaj element je sličan radio buttons elementu, razlika je u tome što se kod ovog elementa može selektovati nijedna ili više opcija. Vrednost poslata u web formi je vrednost checkbox-a koji je selektovan, u suprotnom ta vrednost ne postoji. Dodavanjem opcije CHECKED u checkbox posebno će se istaći taj checkbox dok se strana učitava. Primer:

```
<input type="checkbox" name="brend" value="nike">Nike</input><br/>
```

```
<input type="checkbox" name="brend" value="adidas">Adidas</input><br/>
```

### Password Input

Ovaj element je identičan text input elementu, osim kada korisnik unosi podatke u njih. Videće "bullet" karaktere, tj. tačkice, a ne slova koja unosi kao šifru (password). Password tekst je šifrovan kroz prenos i potom na kraju dešifrovan kada je podatak polja učitao na server. Primer:

```
<input type="password" size="25" maxlength="15" name="sifra"/>
```

### Text Area

Ovaj element je tekst polje koje sadrži više od jedne linije za unos podataka i može se scroll-ovati tako da korisnik može da unese više teksta. Ovom polju možemo zadati veličinu, unošenjem broja redova i kolona kroz „rows” i „cols” attribute. Primer:

```
<textarea name="opis" rows="8" cols="14">Ovde se unosi opis (Obrisi me i pocni da opisujes!)</textarea>
```

### Drop-down lista (menu)

Tag <select> se koristi za definisanje padajućeg menija koji korisniku omogućava da izabere jednu od ponuđenih stavki u listi. <option> tag definiše tekst koji se pojavljuje za stavku menija. Dodavanjem SELECTED opcije jednoj stavci, ta stavka je selektovana kada se stranica učitava, u suprotnom, prva stavka je selektovana kao default. Primer:

```
<select name="prvaLista">
    <option value="nike">Nike</option>
    <option value="adidas" selected>Adidas</option>
</select>
```

### Dugme

Kada želimo da dugme dodamo na formu koristimo input element koji je tipa „button”. Elementu se dodaje atribut „value” čija vrednost se ispisuje na dugmetu. Primer:

```
<input type="button" value="Saznaj više"/>
```

### Submit dugme

Submit dugme govori web browser-u da sakupi sve selekcije, vrednosti, i uneti tekst iz polja i smesti ga na mesto definisano u „action” atributu <form> tag-a (O atributu pročitajte u daljem tekstu). Primer:

```
<input type="submit" value="Potvrdi"/>
```



Navedeni su najčešće korišćeni elementi forme koji su podržani od strane svih browsera. Navešćemo još neke elemente koji mogu da se nađu na formi: izbor boje, selektovanje datuma (kalendar), upload fajla, email polje, podešavanje opsega (scroll-er), url polje, search polje,...

## Atributi `<form>` i `<input>` tagova

### `<form>` tag

`<form>` tag mora da poseduje dva atributa. To su: „action” i „method” atributi.

#### 1. „action” atribut.

Ovaj atribut služi za definisanje stranice kojoj se podaci sa forme prosleđuju na obradu (kada se potvrdi „submit” dugme). Primer:

```
<form action="obrada.php">...</form>
```

U ovom primeru podaci sa forme se prosleđuju „obrada.php” stranici koja se nalazi u istom folderu kao i trenutna stranica.

#### 2. „method” atribut

Ovaj atribut može da ima dve vrednosti: GET i POST. Oba HTTP metoda opisuju način na koji se podaci prenose do fajla navedenog u „action” atributu. Primer:

```
<form action="obrada.php" method="GET">...</form>
```

ili

```
<form action="obrada.php" method="POST">...</form>
```

Koji ćemo metod koristiti zavisi od osetljivosti podataka koji se prenose.

GET metod se koristi ako prenosimo manje količine podataka. Podaci koji se prenose do servera su vidljivi u adresnoj liniji browsera. Zbog toga ovaj metod se ne preporučuje za osetljive podatke (broj kreditne kartice, password,...)

Podaci koji se šalju POST metodom nisu vidljivi u url-u i ne postoje ograničenja u pogledu količine podataka.

#### 3. autocomplete="on" ili autocomplete="off"

Ako vrednost atributa postavimo na „on”, kada prvi put submitujemo formu pa je refrešujemo i počnemo ponovo da unosimo podatke, pri kucanju teksta daje nam se

predlog za unos (ranije uneti podaci u to polje). Podrazumevana vrednost ovog atributa je „off”. Ovaj atribut može da se koristi i za neke input elemente (text, search, url, password, range, color, email).

### **<input> tag**

Pored već pomenutih atributa unutar <input> taga, postoji i veliki broj drugih. Neki od njih su:

#### 1. placeholder="Ovde ide tekst"

Koristi se da bi se ispisao tekst u text polju ili text area polju, prilikom učitavanja forme. Kada kliknemo na polje i počnemo da unosimo tekst, podrazumevani tekst iz „placeholder”-a se gubi. Predstavlja zamenu za „value” atribut u text polju i tekst između <textarea></textarea> tagova.

#### 2. readonly

Ovaj atribut nema vrednost, a dodaje se polju kako bi se naznačilo da nije moguć unos teksta.

#### 3. autofocus

Kada se doda <input> tagu, u tom polju će biti kursor pri učitavanju strane.

#### 4. multiple

Koristi se za input elemente koji su tipa „file”. Nudi opciju da možemo da uploadujemo više fajlova istovremeno. Po defaultu moguće je selektovati samo jedan fajl.

#### 5. required

Koristimo kada želimo da obavestimo korisnika da je popunjavanje polja obavezno. Tačnije, ako korisnik ne unese podatak u polje, dobija obaveštenje da to mora učiniti. Koristi se za input polja koja su tipa: text, checkbox, radio, search, file, password, url,...

### **Okvir forme i naslov**

Za postavljanje okvira forme koristi se tag <fieldset>. Da bismo postavili i naslov forme odmah ispod <fieldset> taga navodimo tag <legend>. Primer:

```
<form action="obrada.html" method="POST">
```

```
  <fieldset>
```

```
    <legend>Ovo je naslov forme</legend>
```

```
<input type="checkbox" name="brend" value="nike">Nike</input><br/>
<input type="checkbox" name="brend" value="adidas">Adidas</input><br/>
</fieldset>

</form>
```

## 11. Blok i inline elementi (div i span)

Sve prethodno pomenute HTML elemente možemo svrstati u dve grupe: blok i inline elementi. Blok elementi su oni koji se prikazuju sami u novom redu. U ovu grupu spadaju: <p>, <table>, <h1>, <ul>, ... Inline elementi se prikazuju u istom redu sa drugim elementima, bez prelaska u novi red. To su elementi: <img>, <a>, <input>, <td>, ...

Spomenućemo još dva elementa koja se dosta koriste pri kreiranju web stranica. To su: <div> i <span>.

<div> je dosta korišćeni blok element i upotrebljava se kao kontejner za druge elemente. Najčešće se koristi za definisanje sekcija unutar stranice.

<span> je inline element. Koristi se najčešće u tekstu kada je potrebno jedan deo naglasiti ili drugačije stilski definisati.

## 12. Novi HTML5 elementi

Nova HTML5 verzija donosi i neke nove elemente. Pored već pomenutih elemenata forme, javljaju se semantički elementi, elementi za dodavanje audio i video sadržaja, grafički elementi, ...

Semantički naziv dolazi otuda što sam naziv taga dovoljno govori programeru koji sadržaj je u pitanju. Semantički elementi se koriste za organizovanje sadržaja na sajtu po blokovima. Svi semantički elementi su blokovski. Imaju istu funkciju kao i <div> element. To su : <header>, <footer>, <article>, <section>, <nav>, <aside>.

### <header> element

Ovaj element se koristi za definisanje header dela (zaglavlja) stranice ili sekcije. Svi elementi koji čine header jedne stranice (logo, menu, dodatne informacije) smeštaju se unutra tagova <header></header>. Ukoliko se sajt sastoji od više stranica, obično sve one imaju isti header deo, pa se u kodu ovaj deo samo prekopira na ostale stranice.

### **<section> element**

Koristi se za definisanje sekcije na stranici. Jednu sekciju čini skup povezanih elemenata.

### **<article> element**

Ovaj element ima skoro isto značenje kao i <section>, jedina razlika je ta što se sadržaj <article> elementa pojavljuje na stranici kao potpuno nezavisan blok povezanih elemenata. Obično se koristi za prikaz pojedinačnih postova na blogu, članaka u novinama, postova na forumima.

Artikli se mogu naći u okviru sekcije, ali takođe i sekcije u okviru artikla.

### **<nav> element**

Koristi se kao kontejner za navigaciju na sajtu. Sastoji se od stavki menija predstavljenih linkovima.

### **<aside> element**

Ovaj element definiše sadržaj koji se nalazi sa leve ili desne strane u odnosi na glavni sadržaj (levi ili desni sidebar).

### **<footer> element**

Ovim elementom definiše se podnožje stranice (poslednji blok). Sadržaj ovog bloka su obično dodatne informacije: kontakt, copyright, kratka navigacija, linkovi itd.

### **<audio> element**

Element se koristi za dodavanje muzičkog sadržaja. Primer:

#### **<audio controls>**

```
<source src="animal.mp3" type="audio/mpeg"/>
```

Browser ne podržava audio element.

#### **</audio>**

Element <source> je neizostavan jer se u okviru njega navodi putanja do audio fajla i media tip (u ovom slučaju „audio/mpeg“ odgovara .mp3 formatu fajla). Standardni format audio fajla je MP3. Koriste se još i Ogg i Wav. Tekst se piše nakon <source> taga za slučaj da browser ne može da učitati fajl. Atribut „controls“ se koristi za prikaz kontrola kao što su: pause, play, volume.

### **<video> element**

Ovaj element se koristi za prikaz video sadržaja. Primer:

```
<video width="300" height="350" autoplay>
```

```
<source src="animal.mp4" type="video/mp4"/>
```

Browser ne podržava video element.

```
</video>
```

Element dosta podseća na <audio>. Razlika je u formatu fajla, obično se za video sadržaje koristi MP4 format fajla. U zavisnosti od formata fajla, navodi se i media tip (video/mp4). Ostali formati su: Ogg i WebM. Preporučuje se navođenje „width“ i „height“ atributa kojima se specificira veličina videa na stranici. Atribut autoplay pokreće video čim se stranica učitava u browser-u. Može da se koristi i „controls“ atribut kao u <audio> tagu.

Da bismo na stranici postavili Youtube sadržaj koristimo <iframe> element. Primer:

```
<iframe width="400" height="290" src="
https://www.youtube.com/watch?v=E2VCwBzGdPM">
</iframe>
```