

Lekcija

STRUKTURA HTML DOKUMENTA

dr Miroslava Raspopović, Andrej Stanišev, Jovana Kovač



STRUKTURA HTML DOKUMENTA

Uvod

01

02

03

Uvod

Markap jezici

HTML

Primer i rešenje

Šta su markap jezici?

Čemu služi HTML?

Verzije HTML

Struktura HTML dokumenta

Struktura HTML jezika

Zaglavlje HTML dokumenta

Telo HTML dokumenta

+ Primer i rešenje lekcije

UVOD

Cilj ove lekcije je da da pregled HTML markap jezika i da da osnovne smernice za razvoj veb sajtova

U ovoj lekciji biće obrađene sledeće teme:

1. Markap jezici
2. HTML
3. Primer i rešenje zadatka

Markap jezici

dfgdfgdf

- ❑ *Šta su markap jezici?*
- ❑ *Čemu služi HTML?*
- ❑ *Verzije HTML*

01

VEB MARKAP JEZICI

Gotovo svi markap jezici su zasnovani na internacionalnom standardu ISO 8879

Veb markap jezici (eng. markup language) su posebni jezici kojima se opisuje veb dokument. Unutar samog teksta dokumenta se upisuju posebne sekvence karaktera poznate kao tagovi ili oznake kojim se obeležavaju pojedini elementi ili grupe elemenata radi prikazivanja ili identifikacije. Zato se ovi jezici mogu nazvati i jezici obeležavanja.

Gotovo svi markap jezici su zasnovani na internacionalnom standardu ISO 8879 – Standard Generalized Markup Language (SGML). Ovaj standard je nastao 1986, dakle pre pojave veba, sa idejom da se kreira jezik kojim će se definisati format u tekstualnim dokumentima. Međutim, SGML se koristi i za definisanje drugih specijalizovanih jezika za predstavljanje dokumenata. SGML dokument koristi posebnu datoteku koja se zove Document Type Definition (DTD), koja definiše tagove koji se koriste za opis formatiranja teksta, attribute i entitete nekog markap jezika, kao i način na koji se oni zajedno koriste.

Pošto SGML opisuje svoje sopstveno formatiranje naziva se meta-jezik. SGML je vrlo složen i obiman jezik koji između ostalog uključuje i hipertekst linkove. Značajni markap jezici koji su nastali na osnovu SGML-a su HTML, XHTML i XML.

SGML DTD je u početku definisao HTML na taj način da se HTML mogao koristiti samo za jednostavnu isporuku veb sadržaja. S jedne strane ovo je bila prednost, s obzirom da je na taj način HTML bio jednostavniji za korišćenje od SGML, ali s druge strane HTML je iz istog razloga ograničen sa svojom efikasnošću i fleksibilnosti. Kada se koristi DTD to podrazumeva da su svi elementi potpuno prilagodljivi i omogućava da pravila korišćenja budu definisana u samom dokumentu.

MOGUĆNOSTI HTML I NJEGOVA INTERPRETACIJA

Za interpretaciju HTML dokumenta se koriste različiti korisnički agenti. Korisnički agent može biti vizualni veb čitač

Hipertekstualni markup jezik (HTML – Hyper Text Markup Language) je jezik koji služi za publikovanje dokumenata na World Wide Web-u. Dokumenti, takozvane veb strane, mogu da se sastoje od teksta, tabela, lista, slika i elemenata za unos podataka. Svaki dokument može da sadrži i meta informacije koje ukazuju na to šta sadrži dokument. Unutar dokumenta se uobičajeno nalaze hiperlinkovi koji pomoću URI-ja daju vezu ka drugim veb stranicama. Dokument može da sadrži i ugnježdene skriptove, aplete i/ili objekte.

Za interpretaciju HTML dokumenta se koriste različiti **korisnički agenti**. Korisnički agent može biti vizualni veb čitač koji može da radi u tekstualnom ili grafičkom režimu, nevizuelni veb čitač, koji može biti audio ili Brajov, roboti za pretraživanje, proksi ili neki drugi uređaji.

Koristeći ove mogućnosti HTML-a autori veb strana su u mogućnosti da:

- Publikuju veb dokumente koji imaju naslove, tekst, tabele, liste, slike i druge ugnježdene objekte kao što su matematički, zvučni ili video objekti,
- Omoguće čitaocima veb strana da pristupe drugim informacijama koje su referencirane hiperlinkovima,
- Uključe u veb strane forme, komandnu dugmad, izborne liste, opcione grupe i druge elemente koje služe za unos podataka i komandi.

VERZIJE HTML-A

Najnovija verzija HTML-a je HTML 5.0. HTML 5.0 dopunjuje HTML sa sintaktičkim svojstvima koje uključuju elemente <video>, <audio>, <header> i <canvas>

Prva verzija HTML se zvala "HTML Tags". Međutim, prva verzija HTML koja je bila po IETF standardu je HTML 2.0 koja je objavljena 1995-te. Sledeća verzija HTML-a, HTML 3.2, je objavljen u skladu sa W3C preporukama. HTML verzija koja je bila u skladu sa standardima SGML DTD je bila verzija HTML 4.0, a koja je objavljena 1998-e godine. Njegova revidirana verzija bila je verzija HTML 4.01, koja je objavljena godinu dana kasnije. HTML verzija 4 je objavljena 1998. godine, a revidirana verzija HTML 4.01 je iz 1999. godine. Ova verzija HTML-a omogućuje kreiranje dokumenta na bilo kom jeziku i korišćenje različitih pisama. Time se postiže efektivno indeksiranje dokumenata za mašine za pretraživanje, kvalitetnija tipografija i bolja konverzija iz teksta u govor. Posebna pažnja pri razvoju HTML-a je posvećena poboljšanju dostupnosti sadržaja osobama sa fizičkim ograničenjima.

Radi pojednostavljenja opisa veb strana, HTML 4.0 uvodi veću razliku između strukture dokumenta i prezentacije, ohrabrujući tako upotrebu stilova (eng. *style sheets*) umesto HTML prezentacionih elemenata i atributa. Informacije o stilu mogu biti specificirane unutar HTML dokumenta ili u eksternim datotekama i mogu se odnositi na individualne elemente ili na grupe elemenata.

Mehanizam za definisanje stilova ne zavisi od jezika kojim se opisuje stil. U nameri da se nametne korišćenje stilova WWW konzorcijum planira da u narednim verzijama ukine neke prezentacione elemente i attribute. Takvi elementi (tagovi) su u verziji 4.01 označeni za izbegavanje (eng. deprecated).

Najnovija verzija HTML-a je HTML 5.0. HTML 5.0 dopunjuje HTML sa sintaktičkim svojstvima koje uključuju elemente <video>, <audio>, <header> i <canvas>. Ove osobine su dodate kako bi omogućile lakše upravljanje multimedijalnim i grafičkim sadržajima bez dodatne potrebe za plugin-ovima i API. HTML5 takođe omogućava obogaćenje i semantičkog zadržaja dokumenta sa atributima kao što su <section>, <article>, <header> i <nav>. Iz istog razloga su dodati i neki novi atributi, dok su određeni elementi i atributi odbačeni u odnosu na prethodnu verziju.

HTML

-
- ❑ *Struktura HTML dokumenta*
 - ❑ *Struktura HTML jezika*
 - ❑ *Zaglavlje HTML dokumenta*
 - ❑ *Telo HTML dokumenta*

02

PRIMER HTML DOKUMENTA

HTML dokument se sastoji iz više elemenata grupisanih u verziju, zagavlje i telo dokumenta

HTML 4.01 se sastoji iz više različitih elemenata od kojih se većina može kategorizati u tri grupe:

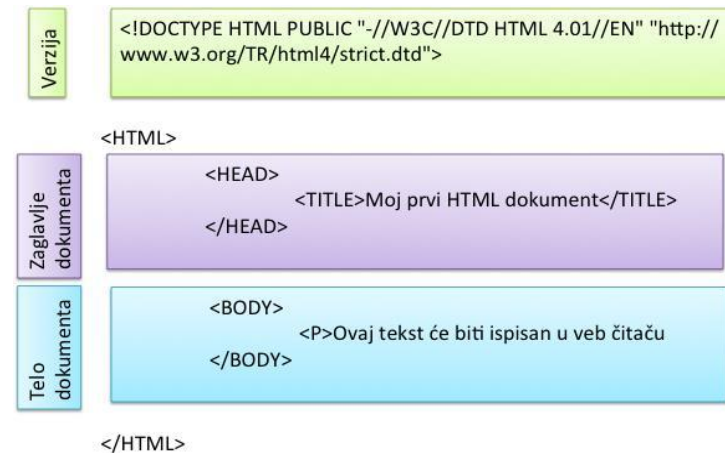
- *Verzija dokumenta*
- *Deklarativno zaglavlje dokumenta (eng. header)*
- *Telo dokumenta (eng. body)*

Primer jednostavnog HTML dokumenta dat je na Slici 1.

U ovom primeru se može videti da se radi o HTML dokumentu verzije 4.01

i da je specificirano da se korisni striktni DTD (strict.dtd), što znači da se atributi ne mogu izbegavati.

Telo dokumenta predstavlja sadržaj koji će biti prikazan preko veb čitača.



Slika-1 Primer HTML dokumenta

HTML KONSTRUKCIJE

HTML jezik koristi sledeće konstrukcije: elemente, attribute, karakter reference i komentare

HTML datoteka, koja opisuje jednu stranicu, sastavljena je od teksta i tagova. Tag je osnova svake HTML strane i kazuje veb čitaču kako da prikaže Veb stranu. HTML tag predstavlja jednu reč ili skraćenicu koja se nalazi u trouglastim zagradama. Na primer, tag može biti napisan kao

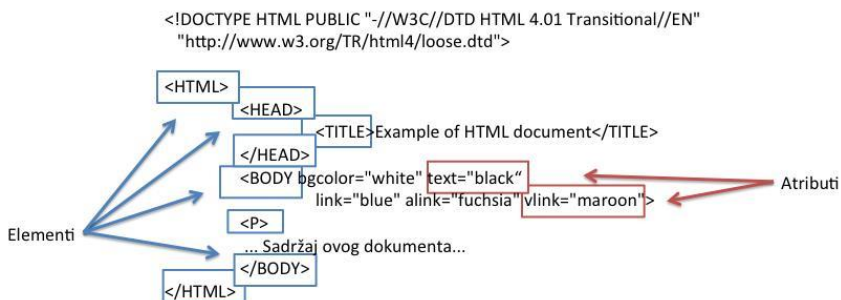
`<html>`

HTML datoteka mora da ima ekstenziju tipa `.html` , `.htm`, ali se može kreirati u bilo kom tekst editoru ili pomoću nekog alata za kreiranje HTML strana.

HTML jezik koristi sledeće konstrukcije:

- *elemente*
- *attribute*
- *karakter reference*
- *komentare.*

Primer elemenata i atributa je ilustrovan na Slici 1.



Slika-1 Primer HTML dokumenta sa ilustrovanim primerima za elemente i attribute

HTML ELEMENTI

Svaka deklaracija nekog tipa elementa opisuje tri dela: početni tag, sadržaj, završni tag.

HTML dokumenti su definisani pomoću njihovih elemenata. Tako HTML jezik ima tipove elemenata kojim se predstavljaju naslovi, pasusi, hipertekstualni linkovi, liste, tabele, slike itd. Svaka deklaracija nekog tipa elementa opisuje tri dela:

- *početni tag*
- *sadržaj*
- *završni tag*

Ime elementa se pojavljuje i u startnom i u završnom tagu, s tim što se u završnom tagu ispred imena elementa upisuje znak „/“ kako bi se znalo da je u pitanju završni tag. Imena tagova se nalaze između zagrada „<“ i „>“.

U sledećem primeru je prikazano korišćenje tipa elementa TITLE. Između startnog taga i završnog taga nalazi se sadržaj „Naziv HTML dokumenta“.

Neki HTML tipovi elemenata dozvoljavaju autorima da se startni ili završni tag izostavi, mada se u tom slučaju gubi kompatibilnost sa XHTML-om.

Tako na primer tagovi <P> i ne moraju da imaju završne tagove, a <BODY> i <HEAD> ne moraju da imaju startne tagove.

Neki tagovi nemaju sadržaj, pa nema potrebe da imaju i završni tag.

Primer ovakvog taga je
, kojim se označava završetak linije teksta. DTD definiše kod kojih tipova elemenata je dozvoljeno

izostavljanje početnog ili završnog taga i sadržaja.



Slika-2 Primer korišćenja TITLE elementa

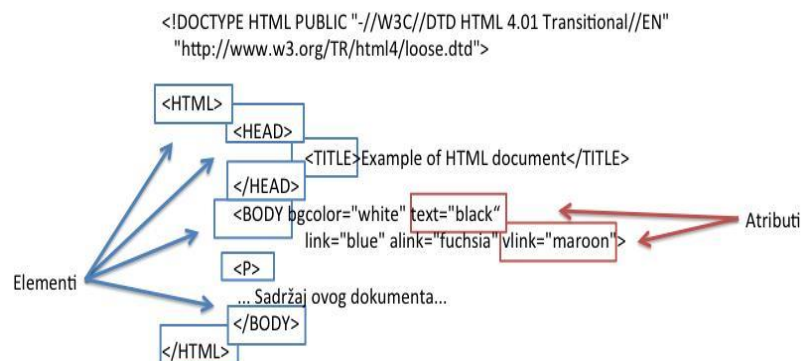
PRIMER HTML ELEMENATA

HTML, HEAD, TITLE, BODY i P

Neki osnovni tagovi su prikazani na Slici 1, a to su HTML, HEAD, TITLE, BODY i P. Element HTML definiše čitav HTML dokument i njegov početni tag je `<HTML>`, a završni tag je `</HTML>`. Njegov sadržaj predstavljaju drugi elementi, HEAD i BODY. Oni predstavljaju ugnježdene elemente.

Element BODY definiše telo HTML dokumenta i njegov početni tag je `<BODY>`, a završni tag je `</BODY>`. Sadržaj ovog elementa je drugi element, P. Element P ima početni tag `<P>` i definiše novi paragraf. Ovaj element ne zahteva završni tag, mada se može kao završni tag koristiti `</P>`. Sadržaj ovog elementa je tekst "...Sadržaj ovog dokumenta..."

Treba naglasiti da HTML tagovi ne moraju biti isključivo napisani velikim slovima. Dozvoljeno je da se koriste i mala i velika slova u imenu tagova, odnosno `<HTML>` i `<html>` se može koristiti.



Slika-1 Primer HTML dokumenta sa ilustrovanim primerima za elemente i atribute

HTML ATRIBUTI

Atributi i njihove vrednosti se zadaju unutar početnog taga

Elementi imaju pridružene atribute koji mogu uzimati podrazumevajuće vrednosti, vrednosti zadate od strane autora ili vrednosti definisane nekom skriptom. Atributi i njihove vrednosti se zadaju unutar početnog taga. Vrednost atributa se upisuje između znakova navoda. U sledećem primeru atribut *align*, koji ima vrednost „left“, upisan je unutar početnog taga.

```
<h1 align="left"> Tekst u Heading 1 formatu koji će biti postavljen uz levu marginu </ h1>
```

Kada se poziva hiperlink u hipertekstu, atributi se takođe koriste. Na primer, atribut *href* se može koristiti na sledeći način:

```
<a href="http://www.metropolitan.edu.rs">Ovo je link</a>
```

HTML KOMENTARI

Komentari se pišu između karaktera "<!--" i "-->"

HTML kod treba da bude iskomentarisano, kao i bilo koji drugi kod. Cilj komentarisanja koda je lakše snalaženje u kodu, kako bi se u njemu lakše obavile izmene u budućnosti. Komentari se pišu između karaktera "<!--" i "-->". Primer jednog komentara je:

```
<!--Ovo je komentar  
koji može da bude napisan  
u više redova -->
```

Informacije u komentaru ne interpretira čitač veb strana, što znači da korisnik neće videti sadržaj komentara među ostalim sadržajem veb stranice.

ZAGLAVLJE HTML DOKUMENTA - <HEAD>

Element zaglavlja dokumenta HEAD sadrži informacije o dokumentu, kao što su naslov, ključne reči koje će biti korišćene od strane mašina za pretraživanje i druge podatke

Element zaglavlja dokumenta HEAD sadrži informacije o dokumentu, kao što su naslov, ključne reči koje će biti korišćene od strane mašina za pretraživanje i druge podatke koji se ne smatraju sadržajem dokumenta. Korisnički agent ne iscrta elemente koji se pojavljuju u zaglavlju kao sadržaj, već koristi neke druge mehanizme da bi ih učinio dostupnim korisniku.

Unutar HEAD elementa se mora naći TITLE element koji služi za identifikovanje sadržaja dokumenta. Unutar HEAD elementa se mogu naći i META elementi. META element služi za specificiranje metapodataka, odnosno informacija o dokumentu, kao što su autor, lista ključnih reči, datum publikovanja, jezik sadržaja dokumenta. Svi ovi podaci mogu biti jako korisni mašinama za pretraživanje.

Sledeći primer pokazuje kako je pomoću META elementa specificiran autor veb stranice:

```
<META Name="Author" content="Stefan Milić">
```

Kada je potrebno specificirati jezik kojim je zadata vrednost atributa content u META elementu, koristi se atribut `lang`. Navođenje jezika može da bude korisno kako mašinama za pretraživanje tako i audio čitačima veb strana koji pomoću ovog atributa određuju koja će se pravila za izgovaranje primeniti za sadržaj.

U sledećem primeru `lang` atribut je iskorišćen da bi se opisalo na kom jeziku su date ključne reči iz sadržaja dokumenta. Primer se odnosi na US engleski, britanski engleski i francuski.

```
<!-- For speakers of U.S. English -->  
<META name="keywords" lang="en-us"  
      content="vacation, Greece, sunshine">  
<!-- For speakers of British English -->  
<META name = "keywords" lang = "en"  
      content = "holiday, Greece, sunshine">  
<!-- For speakers of French -->  
<META name = "keywords" lang = "en"  
      content = "vacances, Grèce, soleil ">
```

TELO HTML DOKUMENTA - <BODY>

Telo dokumenta se sastoji od opisa sadržaja dokumenta

Telo dokumenta se sastoji od opisa sadržaja dokumenta. Autori HTML dokumenta treba da vode računa o tome da je od verzije 4 preporučeni način za specifikaciju prezentacije dokumenata pomoću stilova (eng. *style sheets*) i da su svi prezentacioni atributi elementa **BODY** označeni za izbegavanje (eng. *deprecated*). Naredni primer ima dva dela.

Primer 1 predstavlja HTML dokument koji koristi **BODY** atribute koji su označeni za izbegavanje. Primer 2 predstavlja document koji koristi stilove kako bi se postigao isti efekat kao u Primeru 1. U prvom slučaju se atributi elementa **BODY** koriste da se specificira boja pozadine (bela), teksta (crna) i linkova ("fuchsia" kada je aktivan i "maroon" kada je link već posećen).

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">  
<HTML>  
<HEAD>  
<TITLE>Example of HTML document</TITLE>  
</HEAD>  
<BODY bgcolor="white" text="black"  
link="blue" alink="fuchsia" vlink="maroon">  
... Sadržaj dokumenta...  
</BODY>  
</HTML>
```

Primer 1. Upotreba atributa za izbegavanje

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">  
<HTML>  
<HEAD>  
<TITLE>Example of HTML document</TITLE>  
<STYLE type="text/css">  
BODY {background: white; color: black}  
A:link {color: blue }  
A:visited {color: maroon }  
A:active {color: fuchsia }  
</STYLE>  
</HEAD>  
<BODY>  
... Sadržaj dokumenta...  
</BODY>  
</HTML>
```

Primer 2. Upotreba stilova (Cascading Style Sheet)

STIL STRANICE U EKSTERNOJ DATOTECI

Stil stranice se može referencirati u eksternoj datoteci

Željeni stil ne mora da bude opisan u samom dokumentu, kao što je to urađeno u prethodnom primeru, već se može definisati u nekoj eksternoj datoteci, kao što je prikazano Primerom 3. U Primeru 3 se više strana mogu referencirati na isti stil. Stil stranice je definisan u eksternoj datoteci *style.css*.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<HTML>
<HEAD>
  <TITLE>Example of HTML document</TITLE>
  <LINK rel="stylesheet" type="text/css" href="style.css">
</HEAD>
<BODY>
  ... Content of the document...
</BODY>
</HTML>
```

Primer 3 – Upotreba eksterne datoteke za stilizaciju

Primer i rešenje zadatka

03

PRIMER I REŠENJE DANAŠNJE LEKCIJE

Napraviti HTML stranicu (Notepad ++, Sublime Text ili u nekom Txt Editoru po izboru) koja ce biti dobro struktuirana I koja ce sadrzati sve pomenute attribute iz ove lekcije. Takodje napraviti paragraph sa imenom i prezimenom.

Rešenje:

```
<!DOCTYPE html>
<html>
<head>
<title>Ovo je naslov koji stoji u head tagu</title>
</head>
<body>
<p> Ovo je paragraf koji uvek stoji u body tagu </p>

<h1> Ime prezime ,.. </h1>

<!--Ovo je komentar koji nece biti prikazan u browseru niti ce biti
izvršen, ali služi da komentarisete kod kako bi vam bilo lakše da
se snadjete-->
</body>
</html>
```

Ovo je paragraf koji uvek stoji u body tagu

Ime prezime ,..

Slika 1. predstavlja izvršen kod