

Danijela Simić

Filip Marić

Predrag Janičić

Milan Banković

Milena Vujošević Janičić

Ve-

sna Marinković

Mladen Nikolić

Mirko Spasić

Sa-

na Stojanović Đurđević

Ivana Tanasijević

UVOD U INFORMATIKU

Beograd
2025.

Elektronska verzija (2025)

Autori:

dr Danijela Simić, docent na Matematičkom fakultetu u Beogradu

dr Filip Marić, redovni profesor na Matematičkom fakultetu u Beogradu

dr Predrag Janičić, redovni profesor na Matematičkom fakultetu u Beogradu

dr Milan Banković, docent na Matematičkom fakultetu u Beogradu

dr Milena Vujošević Janičić, vandredni profesor na Matematičkom fakultetu u Beogradu

dr Mladen Nikolić, vandredni profesor na Matematičkom fakultetu u Beogradu

dr Mirko Spasić, docent na Matematičkom fakultetu u Beogradu

dr Sana Stojanović Đurđević, docent na Matematičkom fakultetu u Beogradu

dr Ivana Tanasijević, docent na Matematičkom fakultetu u Beogradu

UVOD U INFORMATIKU

Izdavač: Matematički fakultet Univerziteta u Beogradu

Studentski trg 16, 11000 Beograd

Recenzenti:

??

??

Obrada teksta i ilustracije: *autori* (osim za slike nabrojane na kraju knjige)

Dizajn korica: *autori*

©2025. Danijela Simić, Filip Marić, Predrag Janičić, Milan Banković, Milena Vujošević Janičić, Mladen Nikolić, Mirko Spasić, Sana Stojanović Đurđević i Ivana Tanasijević

Ovo delo zaštićeno je licencom Creative Commons CC BY-NC-ND 4.0 (Attribution-NonCommercial-NoDerivatives 4.0 International License). Detalji licence mogu se videti na veb-adresi <http://creativecommons.org/licenses/by-nc-nd/4.0/>. Dozvoljeno je umnožavanje, distribucija i javno saopštavanje dela, pod uslovom da se navedu imena autora. Upotreba dela u komercijalne svrhe nije dozvoljena. Prerada, preoblikovanje i upotreba dela u sklopu nekog drugog nije dozvoljena.



Sadržaj

Sadržaj	3
1 Digitalizacija	7
1.1 Zapis teksta	7
1.2 Jezici za obeležavanje	7

Predgovor

Ovo je materijal za predmet Uvod u informatiku na prvoj godini smjera *Informatika* na Matematičkom fakultetu Univerziteta u Beogradu) Nadamo da izbor sadržaja i način prezentovanja mogu da budu zanimljivi ne samo studentima, već i svima drugima koje interesuje ova oblast računarstva.

Bićemo zahvalni čitaocima na svim ispravkama, sugestijama i komentarima koje nam pošalju.

- * Kratka istorija informatike i informaciono-komunikacionih tehnologija – Filip Marić i Predrag Janičić
- * Hardver i softver - Filip Marić, Predrag Janičić i Danijela Simić
- * Digitalizacija - Filip Marić, Predrag Janičić i Danijela Simić
- * Algoritmi i izračunljivost - Predrag Janičić
- * Sistemski softver - Milan Banković
- * Računarske mreže - Mirko Spasić
- * Programski jezici i prevodioci - Milena Vujošević Janičić
- * Proces razvoja softvera - Danijela Simić
- * Baze podataka - Ivana Tanasijević
- * Matematika i informatika - Filip Marić
- * Veštačka inteligencija - Mladen Nikolić i Predrag Janičić
- * Računarska grafika - Vesna Marinković

* Računari i društvo - Sana Stojanović Đurđević

*Danijela Simić, Filip Marić, Predrag Janičić, Milan Banković, Milena Vujošević
Jančić, Mladen Nikolić, Mirko Spasić, Sana Stojanović Đurđević i Ivana
Tanasijević*

Beograd, mart 2024.

Elektronska verzija (2025)

Digitalizacija

Današnji računari su *digitalni*. To znači da su svi podaci koji su u njima zapisani – zapisani kao nizovi brojeva. Zapisivanje tekstova, slika, zvuka i filmova u vidu brojeva zahteva pogodnu reprezentaciju koja ponekad znači i gubitak dela polaznih informacija. Kada su podaci predstavljeni u vidu brojeva, te brojeve potrebno je zapisati u računarima. Dekadni brojevni sistem koji ljudi koriste u svakodnevnom životu nije pogodan za zapis brojeva u računarima jer zahteva azbuku od 10 različitih simbola (cifara). Bilo da se radi o elektronskim, magnetnim ili optičkim komponentama, tehnologija izrade računara i medijuma za zapis podataka koristi elemente koji imaju dva diskretna stanja, što za zapis podataka daje azbuku od samo dva različita simbola. Tako, na primer, ukoliko između dve tačke postoji napon viši od određenog praga, onda se smatra da tom paru tačaka odgovara vrednost 1, a inače mu odgovara vrednost 0. Takođe, polje hard diska može biti ili namagnetisano što odgovara vrednosti 1 ili razmagnetisano što odgovara vrednosti 0. Slično, laserski zrak na površini kompakt diska „buši rupice“ kojim je određen zapis podataka pa polje koje nije izbušeno predstavlja vrednost 0, a ono koje jeste izbušeno predstavlja vrednost 1. U nastavku će biti pokazano da je azbuka od samo dva simbola dovoljna za zapisivanje svih vrsta brojeva, pa samim tim i za zapisivanje svih vrsta digitalnih podataka.

1.1 Zapis teksta

1.2 Jezici za obeležavanje

Postoje dva osnovna pristupa za kreiranje multimedijalnih dokumenata: *WYSIWYG* pristup (engl. *What You See Is What You Get*) i pristup zasnovan na **jezicima za označavanje**.

WYSIWYG alati omogućavaju korisniku da odmah vidi kako će finalni dokument izgledati, koristeći grafičke kontrole za unos i uređivanje sadržaja. Ovaj pristup je intuitivan i jednostavan, pa je pogodan za korisnike koji žele trenutnu vizuelnu povratnu informaciju. Ipak, takvi alati često ograničavaju nivou preciznosti i mogućnost prilagođavanja, naročito kod kompleksnih dokumenata koji zahtevaju dosledno i detaljno formatiranje.

Jezik za označavanje

Jezik za označavanje je sistem u kome se originalni sadržaj dopunjuje *oznakama* (anotacijama) koje određuju njegovu strukturu, izgled ili semantičku ulogu u dokumentu.

Za razliku od WYSIWYG pristupa, jezici za označavanje naglašavaju preciznu organizaciju sadržaja i jasno odvajanje *strukture* od *prezentacije*. Oznake određuju raspored, hijerarhiju, stil i metapodatke dokumenta, što ih čini izuzetno pogodnim za izradu tehničkih izveštaja, naučnih radova, knjiga i web stranica.

Ključna prednost jezika za označavanje jeste visok nivo kontrole nad strukturom, formatiranjem i automatizacijom prikaza. Uz njih je lakše održavati konzistentan stil u obimnim i složenim dokumentima.

Izbor pristupa zavisi od potreba autora: WYSIWYG nudi jednostavnost i brzinu, dok jezici za označavanje pružaju fleksibilnost, preciznost i potpunu kontrolu nad dokumentom.

Jezici za označavanje imaju dugu istoriju koja potiče iz izdavačke industrije, gde su se koristile posebne notacije za označavanje načina slaganja teksta. Ti principi su kasnije preneti u digitalni svet.

SGML

SGML (Standardni Generalizovani Jezik za Označavanje) predstavlja meta-jezik koristi se za definisanje drugih jezika za označavanje. Rane verzije HTML-a bile su definisane upravo u okviru SGML-a, pa se HTML smatra jednom od najpoznatijih SGML aplikacija.

Evolucija od SGML-a dovela je do nastanka savremenih jezika kao što su:

- * **HTML** — osnovni jezik za opis strukture web stranica;
- * **LaTeX** — standard za pripremu naučnih i tehničkih dokumenata;
- * **Markdown** — minimalistički jezik za brzo formatiranje teksta;
- * **XML** — generički, strogo strukturiran jezik za razmenu podataka.

Ovi jezici danas predstavljaju osnovu digitalnog izdavaštva, web tehnologija i automatizovanog procesiranja informacija.

1.2.0.1 Opšta struktura jezika za označavanje

Jezici za označavanje jasno razdvajaju **logičku strukturu** dokumenta od njegove **vizuelne prezentacije**. Logička struktura obuhvata organizaciju sadržaja u veće i manje celine (poglavlja, sekcije, paragrafe), kao i označavanje delova teksta sa posebnom ulogom — citata, definicija, naglašavanja i sl. Vizuelna prezentacija određuje kako će takve strukture biti prikazane: tip i veličinu fonta, boje, margine, razmake među elementima i druge stilove.

Ključna ideja jezika za označavanje jeste da se *sadržaj* i *struktura* pišu ručno, dok se konačna prezentacija (izgled) generiše odvojenim alatima – pregledačem ili kompajlerom.

Primer 1.1. Primer opšte strukture

Sledeći primer prikazuje jednostavnu organizaciju kolekcije recepata u stilu XML-a:

```
<!DOCTYPE kolekcija SYSTEM "recipe-collection.dtd">
<kolekcija>
  <recept author="Marko Perić" title="Čokoladna torta">
    <sastojci>
      <sastojak>2 šolje brašna</sastojak>
      <sastojak>1 šolja šećera</sastojak>
      <sastojak>1/2 šolje kakao praha</sastojak>
    </sastojci>
    <instrukcije>
      <korak>Ugrejte rernu na 350°F (175°C).</korak>
      <korak>Pomesajte sve suve sastojke.</korak>
      <korak>Pecite 30 minuta.</korak>
    </instrukcije>
  </recept>
</kolekcija>
```

U ovom primeru vidimo osnovne gradivne elemente jezika za označavanje:

- * **elemente** (npr. <recept>, <sastojci>),
- * **atribute** (npr. author, title) koji daju dodatne informacije.

Elementi predstavljaju logičke celine dokumenta, dok atributi obezbeđuju metapodatke koji opisuju osobine tih celina.

Jedan od važnih elemenata jezika zasnovanih na SGML tradiciji jeste **definicija tipa dokumenta** (engl. *Document Type Definition* — DTD). Ona propisuje koje oznake su dozvoljene, kojim redosledom mogu biti ugnježdene i koje attribute mogu sadržati.

DOCTYPE deklaracija povezuje dokument sa odgovarajućim DTD-om:

```
<!DOCTYPE kolekcija SYSTEM "recipe-collection.dtd">
```

Ovom deklaracijom navodi se da dokument koristi tip kolekcija, čija je struktura definisana u datoteci `recipe-collection.dtd`. DTD određuje pravila, ograničenja i hijerarhiju elemenata.

Primer 1.2. Primer jednostavnog DTD fajla

```
<!ELEMENT kolekcija - (recept+)>
<!ELEMENT recept - (sastojci, instrukcije)>
<!ELEMENT sastojci - (sastojak+)>
<!ELEMENT instrukcije - (korak+)>
<!ELEMENT sastojak - (#PCDATA)>
<!ELEMENT korak - (#PCDATA)>
```

U ovim pravilima:

- * simbol + označava da se element mora pojaviti bar jednom;
- * #PCDATA znači da element sadrži tekst (parsiran sadržaj);
- * struktura definiše tačan redosled i ugnježđivanje elemenata.

Ovakva formalizacija osigurava konzistentnost i validnost dokumenata, što je posebno važno pri automatskoj obradi i prenosu podataka.

1.2.0.2 HTML (HyperText Markup Language)

HTML (HyperText Markup Language) predstavlja osnovni jezik za opis strukture web stranica i digitalnih dokumenata namenjenih prikazu u veb pregledačima. HTML ne definiše vizuelni stil niti ponašanje elemenata, već primarno opisuje *koji* se sadržaj pojavljuje na stranici i *kako je organizovan*. U tom smislu, HTML čini temelj modernog veba, dok se izgled i interaktivnost tipično postižu korišćenjem CSS-a i JavaScript-a.

Kratka istorija. HTML je razvijen početkom devedesetih godina od strane Tim Bernersa-Lija kao deo prve verzije World Wide Web sistema. Rane HTML specifikacije nisu bile strogo formalizovane, već su evoluirale kroz praksu i potrebe zajednice, što je dovelo do brojnih nedoslednosti među pregledačima. Od verzije 3.2 razvoj standarda preuzima *World Wide Web Consortium* (W3C), uvodeći jasnije specifikacije i stabilniji model razvoja. Savremena specifikacija, poznata pod imenom **HTML5**, standardizuje širok skup elemenata, uvodi semantičke oznake i pruža snažnu osnovu za razvoj interaktivnih veb aplikacija.

Osnovna struktura dokumenta. HTML dokument počinje deklaracijom tipa dokumenta:

```
<!DOCTYPE html>
```

Ova deklaracija informiše pregledač da se radi o dokumentu u skladu sa HTML5 standardom. Sam dokument predstavlja jedan `<html>` element, unutar koga se nalaze dve osnovne celine:

- * `<head>` – deo koji sadrži metapodatke o dokumentu: naslov, kodiranje karaktera, linkove ka spoljnim resursima i druge informacije važne za interpretaciju dokumenta;
- * `<body>` – deo vidljiv korisnicima, koji obuhvata sve elemente prikazane na stranici: tekst, slike, tabele, multimediju i dinamički sadržaj.

Primer 1.3. Osnovna struktura HTML dokumenta

```
<!DOCTYPE html>
<html>
  <head>
    <title>Moja prva stranica</title>
  </head>
  <body>
    <h1>Dobrodošli!</h1>
    <p>Ovo je primer HTML označavanja.</p>
  </body>
</html>
```

Elementi i atributi. HTML dokument se sastoji od **elemenata**, koji predstavljaju logičke komponente dokumenta. Svaki element je obeležen parom tagova: početni tag (`<ime-elementa>`) i završni tag (`</ime-elementa>`).

Između ova dva taga nalazi se sadržaj elementa. Neki elementi nemaju sadržaj (npr. `<br>` ili `<img>`) i u tom slučaju se koriste kao samostalni tagovi. Elementi mogu imati i **atribute**, koji dopunjuju informacije o elementu. Atributi određuju, na primer, URL slike koju treba učitati, opis elementa, identifikator elementa, ili druge metapodatke važne za prikaz.

Sekcija `<head>`. Element `<head>` sadrži informacije koje nisu direktno prikazane na samoj stranici, ali utiču na njeno ponašanje i prikaz. Najznačajniji elementi su:

- * `<title>` – naslov stranice prikazan u kartici pregledača;
- * `<meta charset="UTF-8">` – navođenje kodiranja karaktera;
- * `<meta name="viewport" content="width=device-width, initial-scale=1.0">` – prilagođavanje prikaza mobilnim uređajima;

- * `<link>` – povezivanje eksternih resursa, najčešće CSS fajlova i veb-fontova ili JavaScript dokumenata.

Primer 1.4. Tipična head sekcija

```
<head>
  <title>Moja stranica</title>
  <meta charset="UTF-8">
  <link rel="stylesheet" href="styles.css">
  <link href="https://fonts.googleapis.com/css2?family=Roboto"
    rel="stylesheet">
</head>
```

Sekcija `<body>`. U okviru `<body>` elementa nalaze se sve komponente koje korisnik vidi i sa kojima može da interaguje: naslovi, paragrafi, liste, linkovi, slike, forme, skriptovi i složenije strukture. Ovaj deo HTML-a čini temelj vizuelnog i semantičkog prikaza stranice, a u nastavku će biti predstavljeni najčešće korišćeni elementi iz ove sekcije.

Naslovi. HTML obezbeđuje hijerarhijski sistem naslova pomoću elemenata `<h1>`–`<h6>`, koji određuju relativnu važnost pojedinih sekcija dokumenta. Element `<h1>` obično predstavlja glavni naslov stranice, dok se niži nivoi (`<h2>`–`<h6>`) koriste za organizaciju podsekcija i potsekcija. Pravilna upotreba naslova omogućava preglednu strukturu dokumenta i olakšava njegovu interpretaciju kako korisnicima, tako i pretraživačima i alatima za pristupačnost.

Paragrafi. Element `<p>` predstavlja osnovnu jedinicu tekućeg teksta. Paragrafi služe za organizovanje sadržaja u celovite logičke blokove, što poboljšava čitljivost i olakšava vizuelnu segmentaciju teksta. Pregledač, prema podrazumevanim pravilima, oko svakog paragrafa dodaje odgovarajući razmak.

Liste. Za strukturiranje nabiranja i uređenih sekvenci HTML nudi tri osnovna elementa: `<ul>` za neuređene liste, `<ol>` za uređene liste i `<dl>` za liste opisa, namenjene predstavljanju parova termin - definicija. Ovi elementi omogućavaju prirodno predstavljanje različitih tipova nabiranja, poput postupaka, karakteristika, pojmova ili objašnjenja.

Pored osnovne uloge u strukturiranju teksta, liste imaju važnu primenu i u komponovanju delova interfejsa: neuređene liste se često koriste kao osnova za izgradnju navigacionih menija, traka sa alatima ili struktura na koje se primenjuju stilovi i skriptovi. Time liste prevazilaze čisto tekstualnu funkciju i postaju ključni element u organizaciji i pristupu različitim delovima veb stranice.

Tekstualni elementi. HTML obuhvata i niz elemenata koji služe za obeležavanje naglašavanja i specifičnih stilskih ili semantičkih uloga u tekstu. Elementi `<em>` i `<strong>` koriste se za označavanje naglašenog i snažno naglašenog teksta i imaju semantičku dimenziju, dok elementi `<i>` i `<b>` pretežno određuju vizuelni stil (kurziv odnosno podebljavanje). Dodatno, `<small>` se koristi za sitniji tekst, a `<br>` za umetanje preloma reda bez započinjanja novog paragrafa. HTML takođe podržava komentare u obliku `<!-- komentar -->`, koji služe za beleške i organizaciju koda.

Primer 1.5. Kombinovana upotreba osnovnih elemenata

```
<h1>Uvod u HTML</h1>

<p>HTML je osnovni jezik za strukturiranje web stranica. U okviru
→ dokumenta
koriste se naslovi različitih nivoa, paragrafi i drugi elementi
za organizaciju sadržaja.</p>

<h2>Osnovne strukture</h2>

<p>Elementi u HTML-u mogu označavati <em>naglašeni tekst</em>,
<strong>važne informacije</strong> ili stilske efekte poput
<i>kurziva</i> i <b>podebljavanja</b>.</p>

<ul>
  <li>Neuređena lista</li>
  <li>Sa više stavki</li>
</ul>

<ol>
  <li>Prvi korak</li>
  <li>Drugi korak</li>
</ol>

<p>Ovo je linija teksta.<br>Ovo je nova linija.</p>

<!-- Ovaj komentar nije vidljiv u pregledaču -->
```

Hipertekst. Koncept *hiperteksta* predstavlja jednu od ključnih ideja na kojima počiva savremeni web. Za razliku od linearnog teksta, hipertekst omogućava nelinearnu organizaciju sadržaja, u kojoj se čitanje može odvijati putem prelaska sa jedne informacije na drugu korišćenjem tzv. *hiperveza* (engl. *hyperlinks*).

Hipertekst

Hipertekstualni dokument predstavlja mrežu logički povezanih celina. Kretanje kroz takav dokument odvija se interaktivnim izborom veza, što korisniku omogućava da sam određuje tok i dubinu pregleda informacija.

Ideja hiperteksta značajno prethodi savremenom webu. Još četrdesetih godina, Vannevar Bush je u svom eseju *As We May Think* opisao koncept „Memeksa“, uređaja koji omogućava nelinearno povezivanje informacija. Kasnije, šezdesetih godina, Ted Nelson uvodi sam termin *hypertext* i razvija viziju sistema za trajno povezivanje dokumenata. Tim Berners-Lee je početkom devedesetih iskoristio ove ideje kroz prvu verziju World Wide Web-a, koristeći HTML za strukturu, HTTP za komunikaciju i URL za adresiranje resursa. Time su koncepti hiperteksta postali tehnološki dostupni na globalnom nivou.

U HTML-u se hiperveze realizuju pomoću elementa `<a>` (engl. *anchor*), koji povezuje dokumenti ili njihove delove sa drugim resursima. Referenca ka cilju veze navodi se atributom `href`, koji može upućivati na drugu web stranicu, fajl ili poziciju unutar istog dokumenta

U HTML-u se ciljevi hiperveza mogu navoditi kao *apsolutne* ili *relativne* putanje. Apsolutna adresa uvek navodi potpunu lokaciju resursa, uključujući protokol, domen i eventualnu putanju, na primer:

```
<a href="https://www.example.com/dokumenti/uputstvo.pdf">
  Otvori uputstvo
</a>
```

Relativne adrese izražavaju lokaciju resursa u odnosu na trenutni dokument. Ovo omogućava organizaciju veb-sajtova kao hijerarhije direktorijuma, bez potrebe da se za svaki resurs navodi puna URL putanja. Relativne putanje olakšavaju održavanje i prenosivost veb-projekata, dok apsolutne putanje obezbeđuju jednoznačno adresiranje nezavisno od konteksta.

Primer 1.6. Primer apsolutnih i relativnih veza

```
<!-- Apsolutna veza -->
<a href="https://www.example.org">Otvorite Example</a>

<!-- Relativna veza ka poddirektorijumu -->
<a href="materijali/predavanje1.pdf">Preuzmi prezentaciju</a>

<!-- Veza na deo istog dokumenta -->
<a href="#poglavlje2">Idi na Poglavlje 2</a>

<h2 id="poglavlje2">Poglavlje 2</h2>
<p>Ovo je cilj interne hiperveze.</p>
```

Tabele. Tabele u HTML-u koriste se za prikaz podataka u strogo strukturiranom, tabelarnom obliku. Osnovu tabele čine redovi i kolone, pri čemu se redovi definišu elementom `<tr>`, ćelije podataka elementom `<td>`, a ćelije zaglavlja elementom `<th>`.

HTML omogućava i spajanje ćelija pomoću atributa `colspan` i `rowspan`, kao i navođenje naslova tabele putem elementa `<caption>`.

Iako su tabele istorijski korišćene i za raspoređivanje elemenata na stranici, u savremenom webu njihova upotreba uglavnom je ograničena na situacije gde se prirodno javlja tabelarni prikaz podataka, dok se za vizuelno oblikovanje stranica koriste CSS mehanizmi poput fleksibilnih i mrežnih rasporeda.

Slike i multimedija. HTML omogućava uključivanje različitih oblika multimedijalnog sadržaja, među kojima su najzastupljenije slike, zvučni zapisi i video materijali. Slike se ugrađuju pomoću elementa `<img>`, koji je prazan element i ne sadrži završni tag. Osnovni atribut `src` određuje putanju do slike, dok atribut `alt` obezbeđuje kratki opis njenog sadržaja i važan je za pristupačnost, jer ga koriste čitači ekrana i pregledači koji ne mogu da učitaju sliku.

Pored slika, savremeni web koristi i mogućnost ugrađivanja multimedije iz spoljnih izvora (npr. video platformi). Za takve sadržaje često se koristi element `<iframe>`, koji omogućava umetanje spoljnog resursa u okviru stranice – poput video zapisa ili interaktivnih mapa.

Primer 1.7. Osnovni multimedijalni sadržaji u HTML-u

```
<!-- Slika sa alternativnim tekstom -->


<!-- Ugrađeni video zapis iz spoljnog izvora -->
<iframe src="https://www.example.com/video/embed/12345"
  title="Video zapis"></iframe>
```

Forme. Forme u HTML-u predstavljaju osnovni mehanizam za prikupljanje podataka od korisnika. One omogućavaju unos tekstualnih podataka, biranje opcija, slanje fajlova i druge interakcije koje su ključne za rad web aplikacija. Forme se definišu elementom `<form>`, dok se pojedinačni elementi za unos podataka obeležavaju elementima poput `<input>`, `<textarea>`, `<select>` i `<button>`.

Element `<form>` može sadržati attribute koji određuju način obrade podataka. Najčešće se koriste atribut `action`, koji navodi adresu servera na koji se podaci šalju, i atribut `method`, kojim se određuje HTTP metoda slanja (najčešće GET ili POST). Iako savremene web aplikacije često koriste JavaScript za obradu interakcija, struktura forme u HTML-u i dalje čini osnovu za organizovanje elemenata za unos podataka.

Primer 1.8. Osnovna HTML forma

```
<form action="/prijava" method="post">
  <label for="ime">Ime:</label>
  <input type="text" id="ime" name="ime">

  <label for="lozinka">Lozinka:</label>
  <input type="password" id="lozinka" name="lozinka">

  <button type="submit">Prijava</button>
</form>
```

U ovom primeru labela opisuje pripadajuće polje za unos, dok element dugmeta pokreće slanje forme. Stilizovanje izgleda, rasporeda i ponašanja forme obično se ostvaruje korišćenjem CSS-a i JavaScript-a, dok HTML obezbeđuje semantičku strukturu i osnovne interakcione mogućnosti.

Opšti elementi. HTML obezbeđuje dva neutralna, nestrukturirana kontejnerska elementa — `<div>` i `<span>` — koji služe za grupisanje sadržaja kada nije potreban poseban semantički element. Razlika među njima odnosi se pre svega na način prikazivanja: `<div>` je blokovski element i koristi se za organizovanje većih celina, dok je `<span>` jednolinijski element namenjen označavanju manjih segmenata teksta unutar postojeće linije.

`<div>` se najčešće koristi za organizovanje strukture stranice, npr. za odeljke, navigacione oblasti ili grupisanje elemenata koji čine vizuelnu celinu. U kombinaciji sa atributima `id` i `class`, ovi kontejneri predstavljaju osnovu za stilizovanje i raspoređivanje elemenata pomoću CSS-a.

Element `<span>` se koristi za obeležavanje dela teksta kojem je potrebno primeniti poseban stil ili ponašanje, bez narušavanja toka rečenice.

Primer 1.9. Korišćenje `div`, `span` i jednostavnog CSS stilizovanja

```
<!-- HTML dokument -->
<div class="kontejner">
  <h2>Informacije o računaru</h2>
  <p>Glavne komponente su
    <span class="istaknuto">procesor</span>,
    <span class="istaknuto">memorija</span>
    i <span class="istaknuto">I/O uređaji</span>.
  </p>
</div>
```

```
/* Spoljni CSS fajl (stil.css) */  
.kontejner {  
    padding: 1em;  
    background-color: #f2f2f2;  
}  
  
.istaknuto {  
    font-weight: bold;  
    color: #2a4d8f;  
}
```

U ovom primeru `<div>` omogućava da se celina stilizuje kao blok, dok `<span>` omogućava isticanje pojedinačnih termina. Takva podela uloga odslikava tipičan odnos između HTML-a, koji definiše strukturu, i CSS-a, koji oblikuje vizuelnu prezentaciju.

HTML je ključan za kreiranje strukture web stranica. U kombinaciji sa CSS-om za stilizovanje i JavaScript-om za interaktivnost, HTML služi kao osnova za sav web razvoj.

1.2.0.3 LaTeX

LaTeX je sistem za pripremu dokumenata koji omogućava izradu visokokvalitetnih tekstova, posebno onih koji sadrže matematičke formule, naučnu notaciju i složene strukture kao što su poglavlja, reference, bibliografije i indeksi. Za razliku od WYSIWYG alata, LaTeX naglašava preciznu kontrolu nad strukturom dokumenta i doslednost formatiranja, što ga čini standardom u akademskom izdavaštvu, naročito u matematici, računarstvu, fizici i inženjerstvu.

LaTeX potiče iz ranih 1980-ih godina i predstavlja proširenje sistema TeX koji je razvio Donald Knut sa ciljem da obezbedi precizno i pouzdano digitalno slaganje teksta. Leslie Lamport je, nadgrađujući TeX, definisao LaTeX kao strukturalni i semantički sloj koji autorima omogućava da opišu logičku organizaciju dokumenta, dok sistem automatski preuzima brigu o njegovoj tipografskoj realizaciji. Ovakav model rada doprineo je tome da LaTeX postane osnovno sredstvo u naučnim i tehničkim disciplinama, gde se zahteva doslednost, kontrola i visoka tipografska tačnost.

LaTeX koristi tekstualnu datoteku sa ekstenzijom `.tex` u kojoj se pomoću naredbi i okruženja opisuje struktura dokumenta. Da bi se takva datoteka pretvorila u rezultat (najčešće PDF), koristi se LaTeX kompajler, poput `pdflatex`, `xelatex` ili `lualatex`. Kompajliranje vrši automatsko generisanje naslova, brojeva strana, bibliografije, referenci i drugih elemenata dokumenta.

LaTeX dokument se sastoji iz dva osnovna dela: preambule i glavnog dela dokumenta. Preambula se nalazi između početne deklaracije klase dokumenta (`\documentclass`) i komande `\begin{document}`, i služi za definisanje tipa dokumenta, uključivanje paketa i podešavanje globalnih parametara (npr. fontova, margina, formata stranice).

Komanda `\documentclass` određuje osnovni izgled i namenu dokumenta. Najčešće korišćene klase su `article`, `report` i `book`, dok se dodatne opcije (veličina fonta, format papira, broj kolona) navode u uglaste zagrade:

```
\documentclass[12pt, a4paper]{article}
```

Funkcionalnost LaTeX-a proširuje se pomoću paketa, koji se uključuju komandom `\usepackage`. Paketi omogućavaju rad sa grafikama, naprednim matematičkim zapisima, bibliografijama i drugim specijalizovanim elementima dokumenta. Tipični primeri su `graphicx` za rad sa slikama i `amsmath` za matematička okruženja.

Primer 1.10. Osnovna struktura LaTeX dokumenta

```
\documentclass[12pt, a4paper]{article}

\usepackage{graphicx}
\usepackage{amsmath}

\title{Primer LaTeX dokumenta}
\author{Danijela Simić}
\date{\today}

\begin{document}

\maketitle

\section{Uvod}
LaTeX omogućava izradu tehnički preciznih dokumenata uz minimalna
ručna tipografska podešavanja.

\subsection{Primer slike}
\begin{figure}[h]
  \centering
  \includegraphics[width=0.4\textwidth]{slika.png}
  \caption{Ilustrativna slika}
\end{figure}

\end{document}
```

Unutar okruženja `\begin{document} ... \end{document}` nalazi se stvarni sadržaj dokumenta. LaTeX podrazumeva hijerarhijsku organizaciju teksta kroz naslove različitog nivoa, kao što su `\section`, `\subsection` i `\subsubsection`, čime se formira logička struktura dokumenta. Ovakva organizacija obezbeđuje konzistentno numerisanje i automatsku izradu sadržaja.

Pored strukturalnih jedinica, LaTeX nudi i okruženja za predstavljanje različitih tipova sadržaja. Liste se definišu okruženjima `itemize` (neuređene liste) i

enumerate (uređene liste). Tabela prikaz podataka ostvaruje se okruženjem tabular ili strukturiranim okruženjima unutar table okvira, dok se slike uključuju pomoću paketa graphicx i komande \includegraphics.

LaTeX pruža preciznu kontrolu nad tipografijom i formatiranjem teksta. Podebljani, kurzivni i podvučeni tekst ostvaruju se naredbama \textbf{ }, \textit{ } i \underline{ }. Komentari se označavaju znakom procenta (%), pri čemu se deo linije nakon ovog znaka ignoriše tokom kompajliranja.

Jedna od najistaknutijih prednosti LaTeX-a jeste njegova sposobnost da precizno i tipografski ispravno predstavi matematičke izraze. Matematički sadržaj se može unositi kao deo tekućeg teksta, korišćenjem uokviravanja izrazom $\$ \dots \$$, ili kao izdvojena jednačina koja se postavlja u centar reda, najčešće pomoću okruženja $\$ \$ \dots \$ \$$ ili okruženja equation kada je potrebno automatsko numerisanje.

Primer 1.11. Primer matematičkih izraza

Formula u jednoj liniji: $a^2 + b^2 = c^2$

Zaseban izraz:

$\$ \$$

$\int_a^b x^2 \, dx = \frac{b^3 - a^3}{3}$.

$\$ \$$

LaTeX nudi široku paletu matematičkih okruženja (matrice, sistemi jednačina, operatori, limiti, sume), kao i bogat skup simbola. Napredne funkcionalnosti omogućene su paketima iz AMS kolekcije, naročito amsmath, amssymb i amsthm, koji predstavljaju standard u matematičkom i tehničkom izdavaštvu.

LaTeX automatski generiše sadržaj dokumenta na osnovu strukturnih jedinica (\section, \subsection, \subsubsection). Komanda \tableofcontents postavlja sadržaj na željeno mesto u dokumentu, tipično odmah nakon naslovne strane.

Za ispravno generisanje sadržaja neophodno je dokument kompajlirati najmanje dva puta: prvi prolaz kreira pomoćne podatke o numeraciji i rasporedu, dok drugi prolaz omogućava njihovo korektno prikazivanje u rezultatnom dokumentu.

Za upotrebu srpske latinice i ćirilice u LaTeX dokumentima preporučuju se kompajleri xelatex i lualatex, jer oni omogućavaju direktno korišćenje Unicode karaktera.

Za srpsku latinicu i ćirilicu najčešće se koristi paket:

- * polyglossia – savremeni paket za rad sa više jezika; za srpski se koristi komanda:

```
\usepackage{polyglossia}
\setmainlanguage{serbian}
```

Paket automatski podržava latinicu i ćirilicu, kao i prelamanje po pravilima srpskog jezika.

- * alternativno, u tradicionalnom LaTeX-u (sa pdf_latex-om) koristi se:

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[serbian]{babel}
```

Ova kombinacija omogućava rad sa srpskim jezikom u UTF-8 kodiranju.

LaTeX je postao centralni alat u akademskom izdavaštvu upravo zbog sposobnosti da kombinuje semantički opis strukture sa preciznom tipografskom kontrolom. Uparivanje sa sistemima za upravljanje bibliografijama (najčešće BibTeX ili BibLaTeX), kao i bogat ekosistem paketa, omogućava izradu složenih tekstova – od seminarskih radova do doktorskih disertacija – na pouzdan i dosledan način.

1.2.0.4 Markdown

Markdown je razvijen 2004. godine od strane Džona Grubera kao lagani jezik za označavanje čiji je cilj bio da omogući pisanje tekstova koji su istovremeno čitljivi u svom izvornom obliku i mogu se lako pretvoriti u HTML. Zbog jednostavne sintakse i minimalizma, Markdown je postao široko prihvaćen na savremenim platformama za razmenu i dokumentovanje, uključujući GitHub, StackOverflow i brojna blog okruženja.

Markdown je osmišljen tako da koristi intuitivne tipografske simbole za označavanje strukture teksta: zarezi, zvezdice, crte i drugi osnovni znaci koriste se za formiranje naslova, listi, naglašenog teksta, linkova i blokova koda. Za razliku od HTML-a, Markdown teži minimalizmu i čitljivosti, pa dokument u svom izvornom obliku ostaje razumljiv bez potrebe za dodatnim alatima.

Markdown podržava mali skup osnovnih elemenata koji pokrivaju najvažnije potrebe u svakodnevnom pisanju:

- * **naslovi** se označavaju znakom # (jedan do šest znakova određuju nivo naslova);
- * **naglašavanje teksta** postiže se zvezdicama ili donjim crtama (italic i bold);
- * **liste** mogu biti uređene (numerisane) ili neuređene (označene zvezdicama ili crticama);
- * **linkovi i slike** koriste sintaksu uglastih i obličnih zagrada;
- * **blokovi koda** i isticanje sintakse koriste obeležavanje pomoću oznaka.

Primer 1.12. Reprezentativni Markdown primer

```
# Naslov dokumenta

Markdown omogućava podebljan, italic i kombinovani tekst.

## Lista zadataka
- prva stavka
- druga stavka
- podstavka:
  - detalj

## Link i slika
[Zvanični sajt W3C](https://www.w3.org)

![Primer slike](slika.png)

## Primer koda
``python
def saberi(a, b):
    return a + b
...

```

Zahvaljujući jednostavnoj sintaksi, malom broju pravila i velikoj čitljivosti izvornog koda, Markdown predstavlja jedan od najrasprostranjenijih jezika za obeležavanje u savremenom digitalnom izdavaštvu. Pogodan je za tehničku dokumentaciju, programerske projekte i kraće akademske materijale, a njegova mogućnost transformacije u HTML i druge formate čini ga izuzetno fleksibilnim i prilagodljivim različitim okruženjima.

1.2.0.5 XML (eXtensible Markup Language)

XML je proširivi jezik za označavanje čija je primarna namena opisivanje, struktuiranje i razmena podataka. Za razliku od HTML-a, koji se fokusira na prezentaciju sadržaja, XML je dizajniran tako da razdvoji *semantiku* podataka od načina njihovog prikazivanja. Ključna karakteristika XML-a jeste mogućnost definisanja sopstvenih oznaka (tagova), čime se omogućava opisivanje različitih tipova podataka na standardizovan, hijerarhijski i lako čitljiv način.

XML je tokom dužeg perioda predstavljao osnovni format razmene podataka na internetu i u distribuiranim sistemima. Koristi se u web servisima konfiguracionim datotekama, kao i u mnogim industrijskim standardima. Brojni sistemi za pristup podacima – uključujući i ranije verzije servisa za pristup YouTube podacima – bili su zasnovani na XML-u. Zahvaljujući jasnoj strukturi, mogućnosti ugnježđivanja elemenata i formalnim mehanizmima validacije, XML je postao jedan od ključnih formata za prenos podataka u velikim softverskim okruženjima.

Primer 1.13. Primer XML zapisa

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book category="fiction">
    <title lang="en">The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
    <year>1925</year>
    <price>10.99</price>
  </book>
  <book category="non-fiction">
    <title lang="en">Sapiens</title>
    <author>Yuval Noah Harari</author>
    <year>2011</year>
    <price>14.99</price>
  </book>
</bookstore>
```

U ovom primeru prikazana je hijerarhijska struktura podataka o knjigama: svaki `<book>` element sadrži ugnježdene oznake koje opisuju njegove atribute (naslov, autora, godinu, cenu), dok se dodatne informacije mogu smestiti u attribute elemenata. Ovakva kombinacija semantike i strukture omogućava da XML bude lako obradiv i za ljude i za programe.

Upotreba XML-a. XML se tradicionalno koristi u širokom spektru scenarija:

- * **Razmena podataka i web servisi.** XML je bio osnovni format u servisima i brojnim ranim aplikacijama.
- * **Konfiguracioni fajlovi.** Mnoge aplikacije i dalje koriste XML kao čitljiv i strukturiran format konfiguracije (npr. Maven, Android manifest fajlovi, razni serveri i IDE okruženja).
- * **Standardizovani formati podataka.** Savremeni dokument formati kao što su Office Open XML, SVG (Scalable Vector Graphics) i MathML zasnivaju se na XML sintaksi.
- * **Interoperabilnost.** Zahvaljujući formalnim mehanizmima validacije (DTD, XML Schema), XML omogućava proveru ispravnosti strukture podataka, što je važno u velikim i distribuiranim sistemima.

Iako XML i dalje ima važnu ulogu u sistemima gde su potrebni stroga validacija i definisanje složenih struktura podataka, u brojnim savremenim internet servisima zamijenjen je jednostavnijim formatima za razmenu podataka, pre svega JSON-om (JavaScript Object Notation). JSON je minimalistički format namenjen predstavljanju struktura podataka. XML i dalje ostaje važan u kontekstima gde su potrebni

striktna struktura, formalna validacija, stabilnost standarda i duga životna vernost podataka.

Elektronska verzija (2025)