

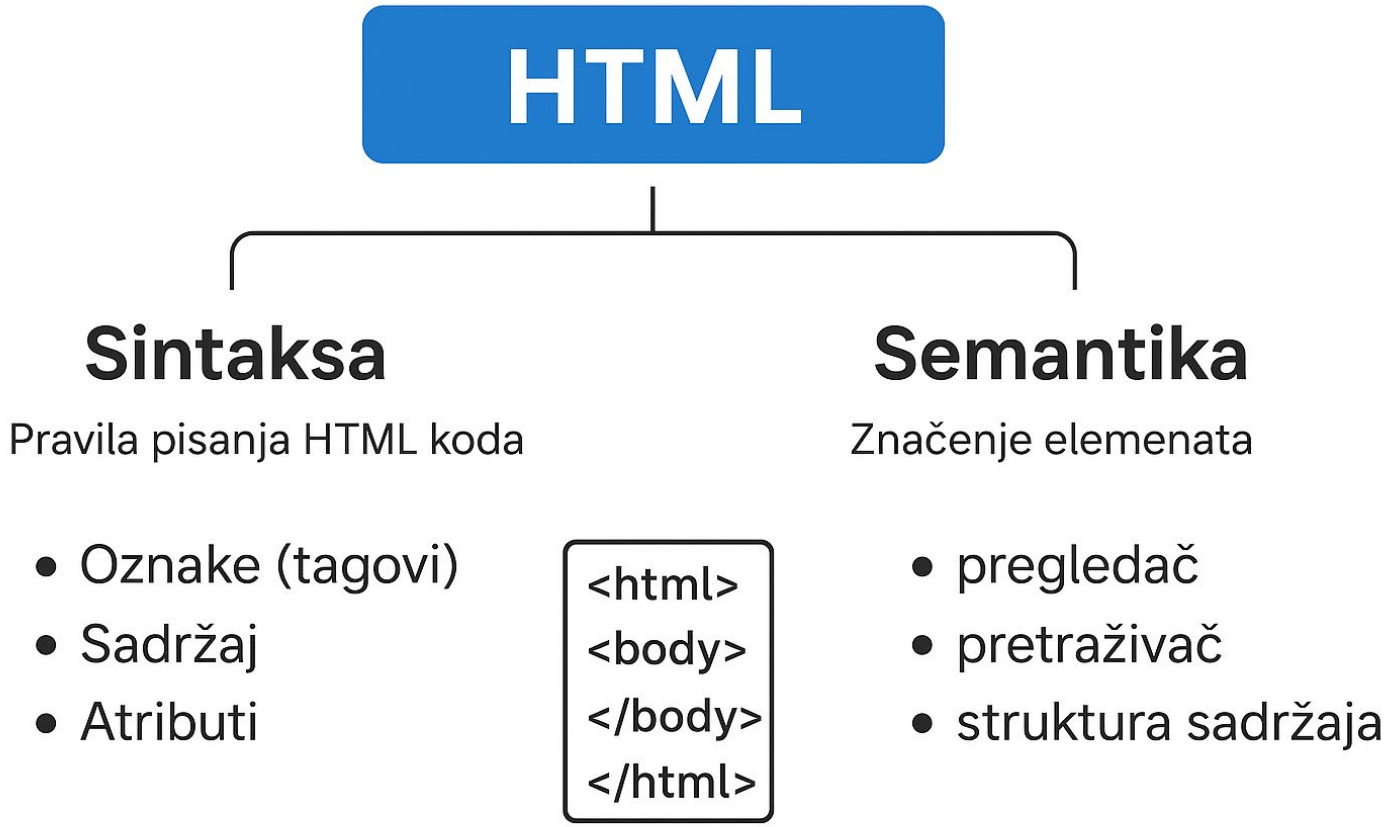
HTML

Klijentske tehnologije

ŠTA JE HTML?

- HTML je jezik za označavanje sadržaja (markup language).
- HTML definiše:
  - strukturu dokumenta
  - elemente stranice
  - hijerarhiju sadržaja
  - veze između sadržaja
  - semantičko značenje elemenata
- HTML nije programski jezik jer sam po sebi ne implementira algoritme i poslovnu logiku.

HTML	Programski jezik
Opisuje strukturu dokumenta	Implementira algoritme
Definiše elemente i njihov odnos	Implementira poslovnu logiku
`<h1>`, `<p>`, `<form>`, `<table>`...	`if`, `for`, funkcije, klase...
Ne izvršava algoritme	Izvršava algoritme
Ne obrađuje podatke kao programski jezik	Može obrađivati podatke


HTML
Sintaksa
Pravila pisanja HTML koda
Oznake (tagovi)
Sadržaj
Atributi
<html>
<body>
</body>
</html>
Semantika
Značenje elemenata
pregledač
pretraživač
struktura sadržaja



Sintaksa

Sintaksa HTML-a predstavlja skup pravila prema kojima se pišu HTML elementi i organizuje sadržaj web dokumenta.

Osnovna struktura HTML elementa je:

<oznaka> Sadržaj </oznaka>

Ovo je paragraf.

<h1>Naslov stranice</h1>

<oznaka atribut="vrednost">

Poseti stranicu

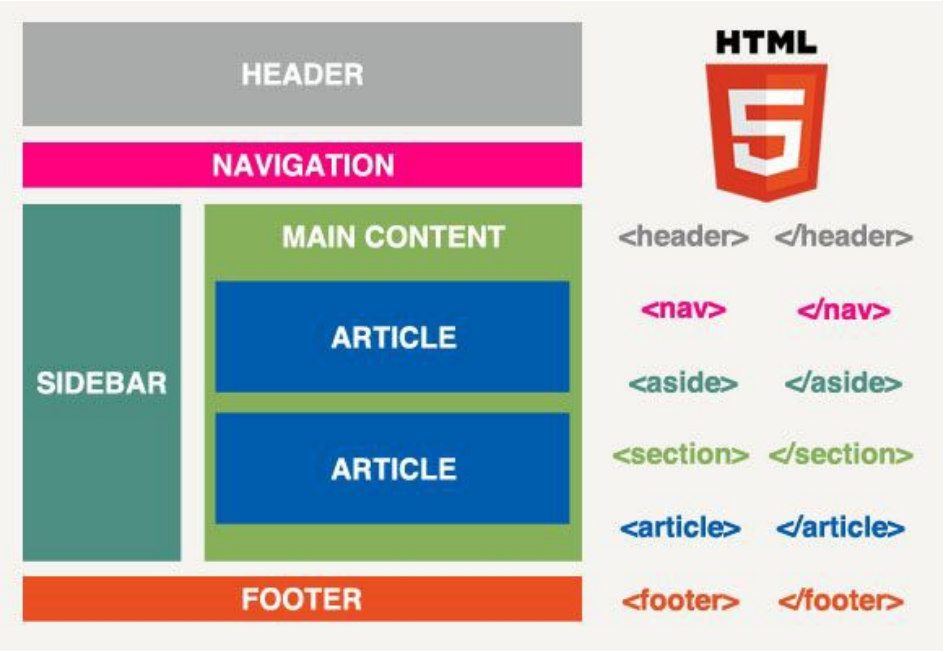
<img src="slika.jpg" alt="UAV">

SEMANTIČKI HTML

Semantički HTML koristi elemente koji opisuju značenje sadržaja.

Primeri:

- <header>
  <nav>
  <main>
  <section>
  <article>
  <aside>
  <footer>



Prednosti:
- jasnija struktura
- bolja pristupačnost
- bolji SEO
- jednostavniji razvoj
- lakše održavanje

Semantički HTML ne služi samo za prikaz sadržaja. On opisuje ulogu svakog dela stranice, čime se poboljšavaju organizacija sadržaja, pristupačnost, SEO i održavanje web aplikacije.

DOPRINOS HTML-A WEB DIZAJNU

jasnu organizaciju sadržaja

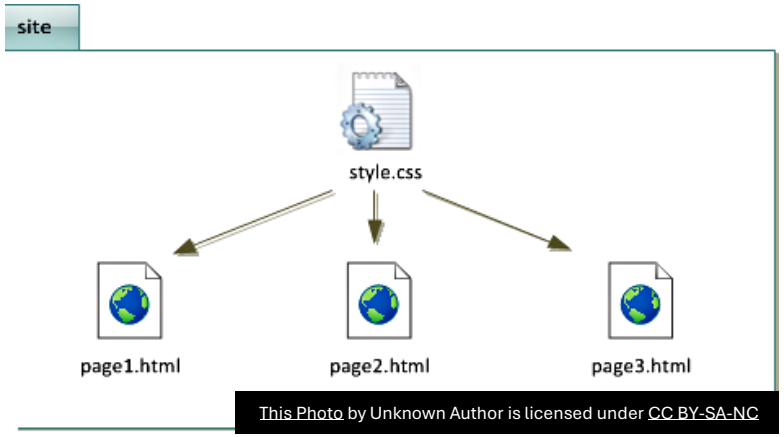
konzistentan dizajn

jednostavniju primenu CSS-a

responzivni dizajn

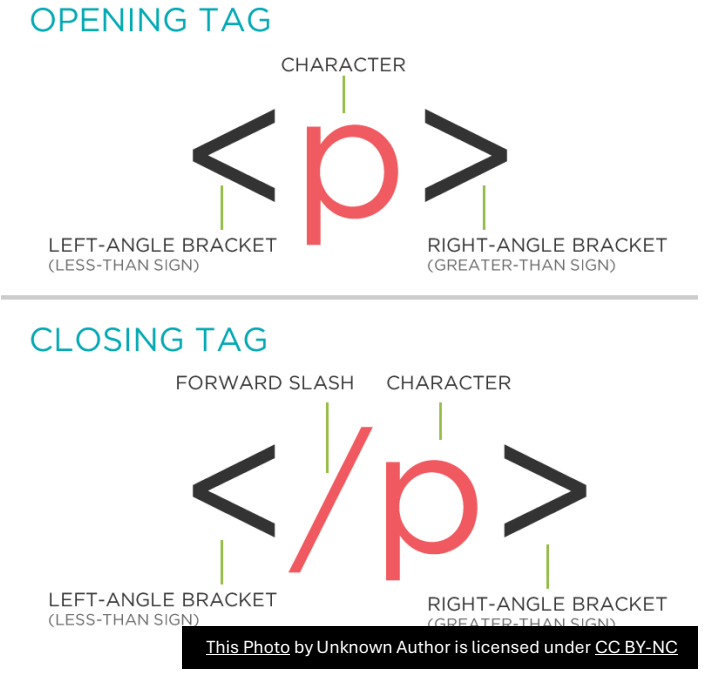
lakše održavanje

bolju pristupačnost



This Photo by Unknown Author is licensed under CC BY-SA-NC





This Photo by Unknown Author is licensed under CC BY-NC

`<header>`, `<nav>`, `<main>`, `<footer>`

Semantička struktura sajta

Organizuju zaglavlje, navigaciju, glavni sadržaj i podnožje stranice.

`<video>`, `<audio>`, `<picture>`, `<figure>`

Multimedijalni sadržaj

Omogućavaju prikaz videa, zvuka, prilagodljivih slika i ilustracija sa opisima.

`<button>`, `<details>`, `<summary>`, `<dialog>`

Interaktivni elementi

Dugmad, proširivi sadržaj, FAQ sekcije i modalni prozori.

`<button>`, `<details>`, `<summary>`, `<dialog>`

Interaktivni elementi

Dugmad, proširivi sadržaj, FAQ sekcije i modalni prozori.

`<canvas>`, `<svg>`, `<progress>`, `<meter>`

Grafika i vizuelizacija

Crtanje grafike, prikaz dijagrama, napretka i mernih vrednosti.

`<form>`, `<input>`, `<select>`, `<output>`

Napredne forme

Interaktivni upitnici, izbor opcija, unos podataka i prikaz rezultata.

OSNOVNA STRUKTURA HTML DOKUMENTA

<!DOCTYPE html>

<html>

<head>

<title>

<meta>

<body>

HTML ELEMENTI I TAGOVI

<h1> – naslov

<p> – paragraf

<a> – hiperveza

<img> – slika

<button> – dugme

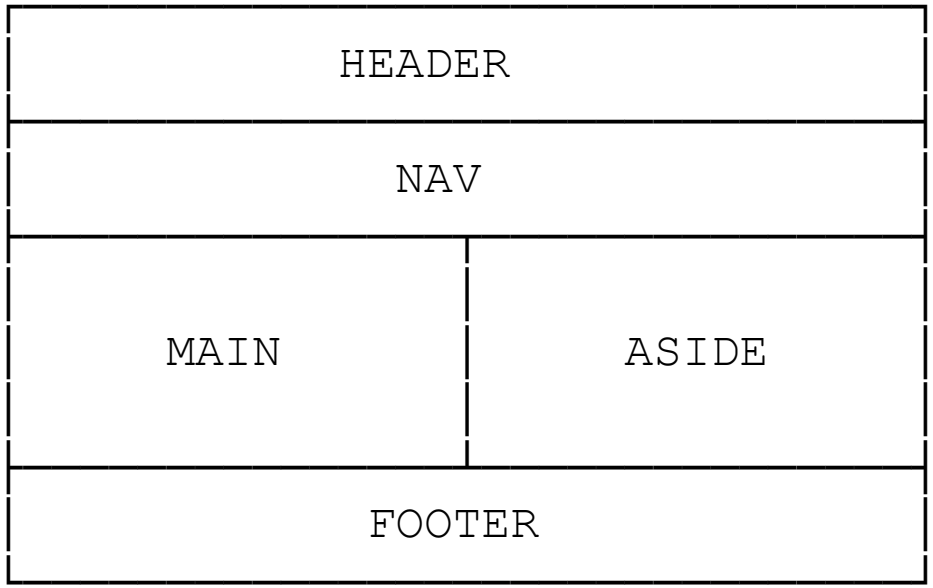
<form> – forma

HIJERARHIJA SADRŽAJA

HTML omogućava definisanje hijerarhije:

- naslov stranice
- glavne sekcije
- podsekcije
- paragrafi
- liste
- tabele
- multimedijalni sadržaj

Hijerarhija omogućava korisniku i browseru da razumeju organizaciju sadržaja.

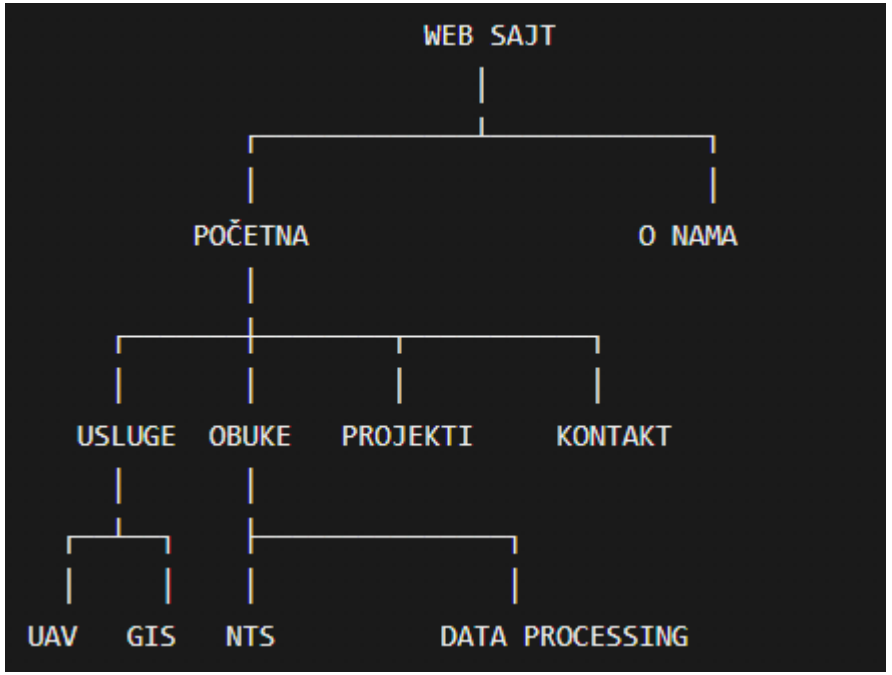



```
<header>
  <h1>UAV Training Platform</h1>
</header>

<nav>
  <a href="home.html">Početna</a>
  <a href="courses.html">Obuke</a>
  <a href="contact.html">Kontakt</a>
</nav>

<main>
  <section>
    <h2>UAV in Non-Technical Survey</h2>
    <p>Program obuke za primenu UAV tehnologija.</p>
  </section>
</main>

<footer>
  <p>2026 UAV Training</p>
</footer>
```




```
<nav>
  <a href="index.html">Početna</a>
  <a href="about.html">O nama</a>
  <a href="services.html">Usluge</a>
  <a href="training.html">Obuke</a>
  <a href="projects.html">Projekti</a>
  <a href="contact.html">Kontakt</a>
</nav>
```

Mapa web sajta

- prikazuje odnose između stranica.
- <nav> definiše navigacioni deo stranice.
- <a> omogućava povezivanje stranica.

Koji je njen značaj?
Informaciona arhitektura: pomaže u logičkom raspoređivanju sadržaja.

Navigacija: olakšava korisnicima pronalaženje informacija.

Korisničko iskustvo (UX): smanjuje broj koraka potrebnih za dolazak do željenog sadržaja.

SEO optimizacija: XML mapa sajta pomaže pretraživačima da otkriju i indeksiraju stranice.

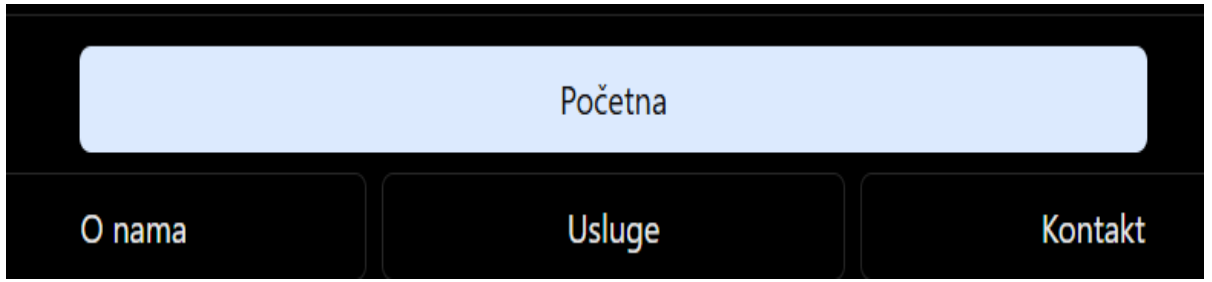
Održavanje sajta: olakšava planiranje novih stranica i izmenu postojeće strukture.

Dve vrste mape sajta

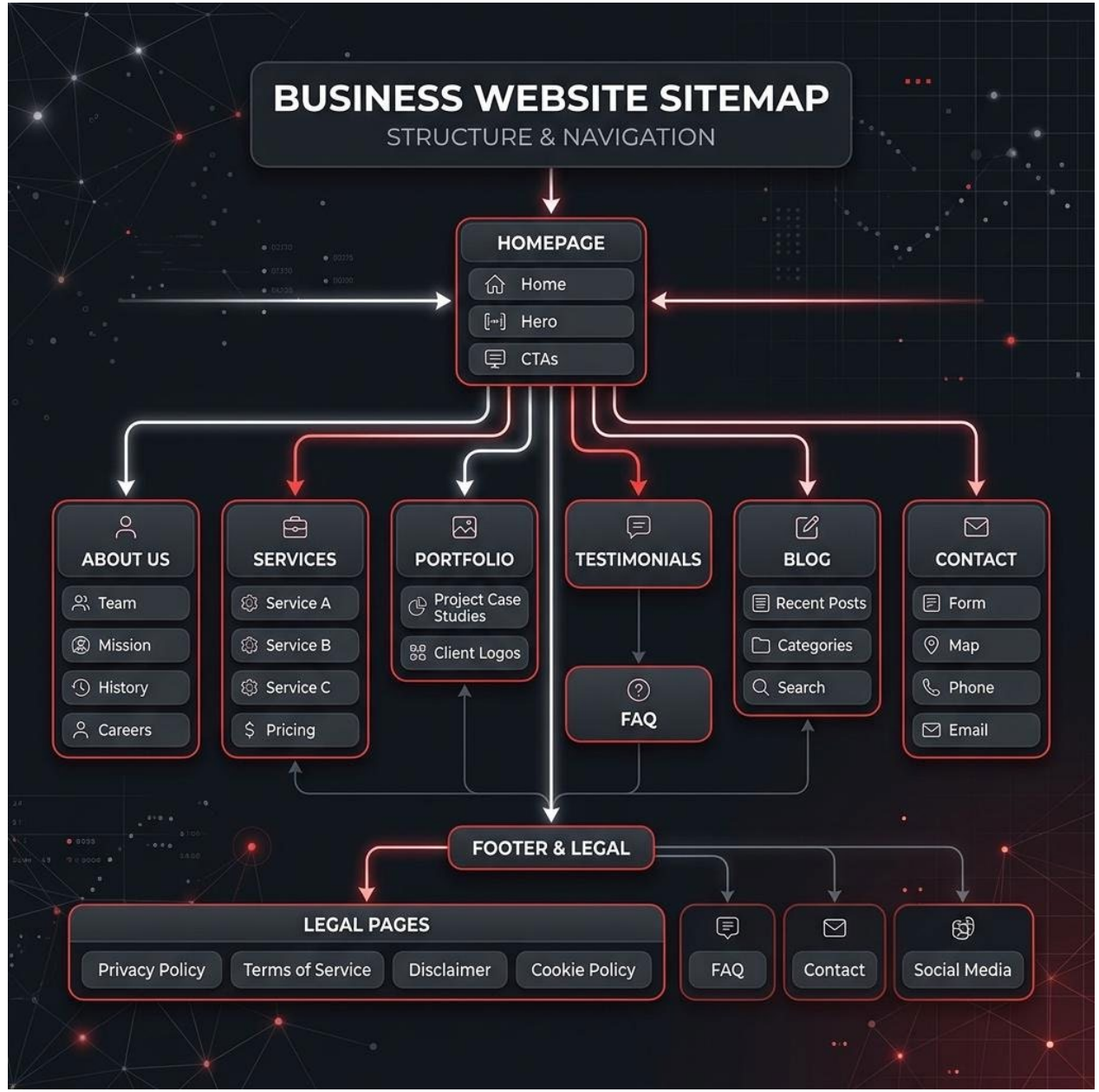
Povezuje informacionu arhitekturu sa HTML strukturom i navigacijom, čime doprinosi boljem korisničkom iskustvu i efikasnijem pronalaženju sadržaja.

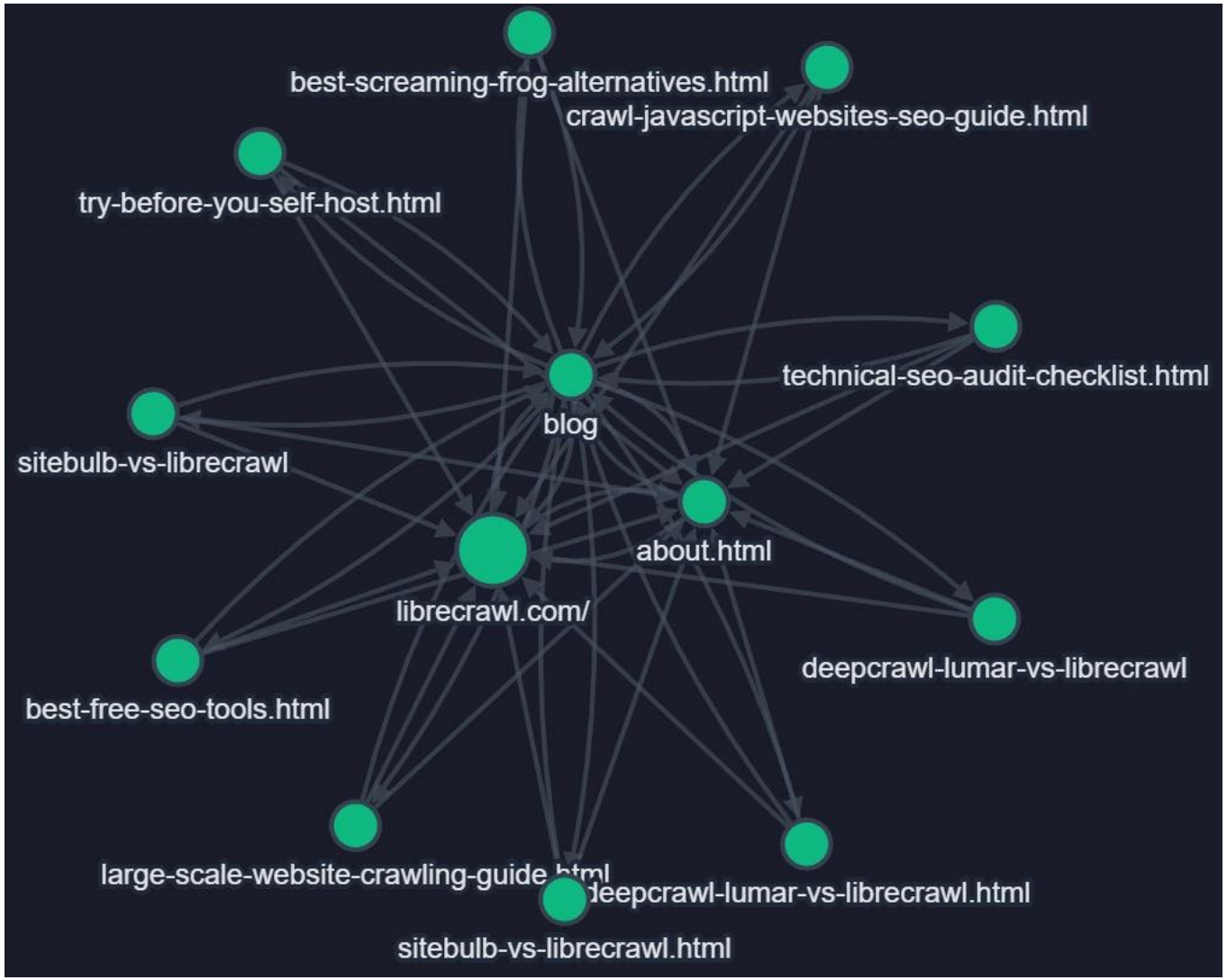
HTML mapa sajta	XML mapa sajta
Namenjena prvenstveno korisnicima.	Namenjena prvenstveno pretraživačima.
Prikazuje linkove ka stranicama.	Navodi URL adrese stranica i druge metapodatke.
Pomaže u navigaciji.	Pomaže u otkrivanju i indeksiranju stranica.

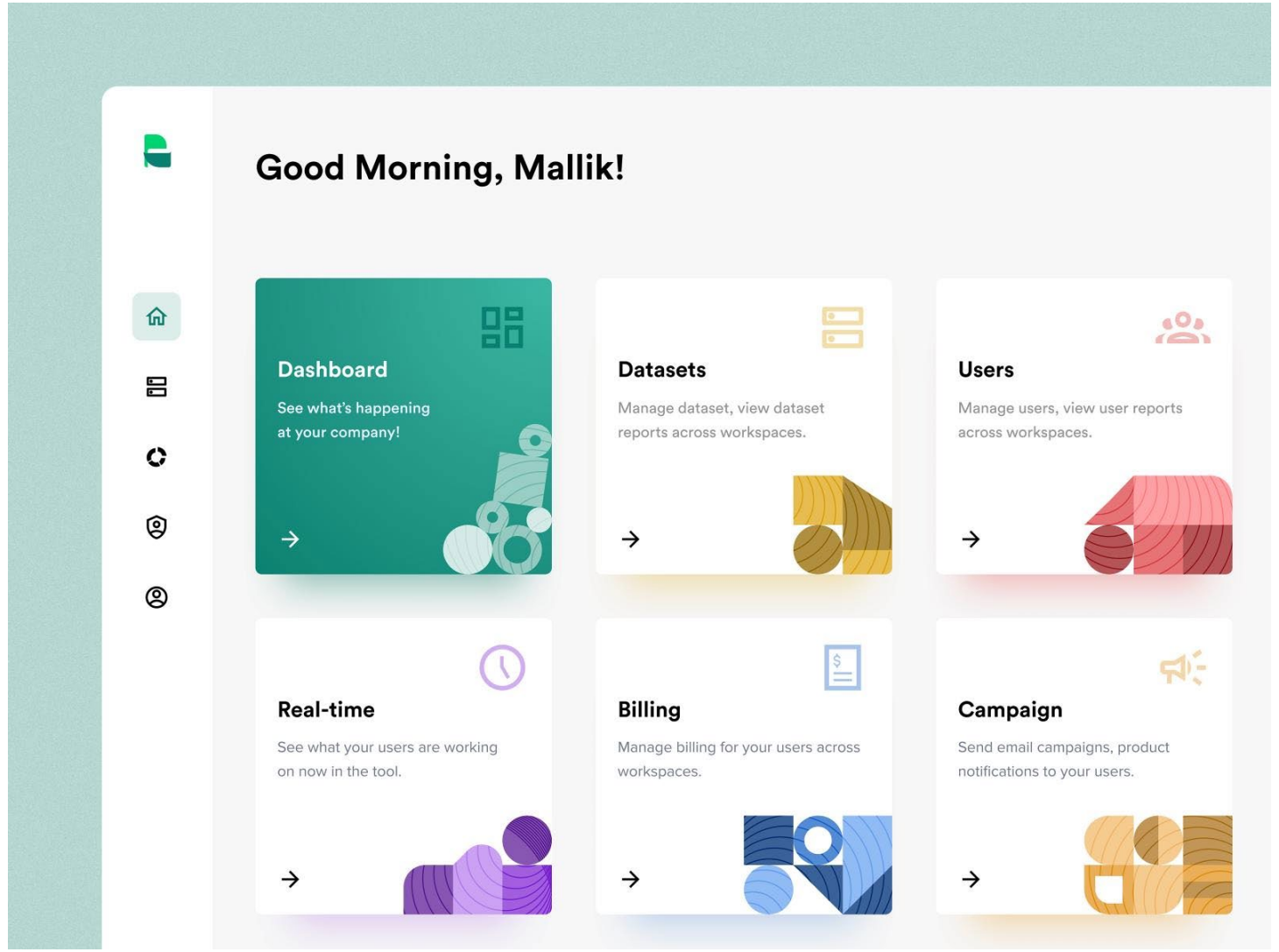





```
<nav>
 <ul>
  <li><a href="index.html">Početna</a></li>
  <li><a href="onama.html">O
nama</a></li>
  <li><a href="usluge.html">Usluge</a></li>
  <li><a
href="kontakt.html">Kontakt</a></li>
 </ul>
</nav>
```


BUSINESS WEBSITE SITEMAP
STRUCTURE & NAVIGATION
HOMEPAGE
Home
Hero
CTAs
ABOUT US
Team
Mission
History
Careers
SERVICES
Service A
Service B
Service C
Pricing
PORTFOLIO
Project Case Studies
Client Logos
TESTIMONIALS
FAQ
BLOG
Recent Posts
Categories
Search
CONTACT
Form
Map
Phone
Email
FOOTER & LEGAL
LEGAL PAGES
Privacy Policy
Terms of Service
Disclaimer
Cookie Policy
FAQ
Contact
Social Media


best-screaming-frog-alternatives.html
crawl-javascript-websites-seo-guide.html
try-before-you-self-host.html
technical-seo-audit-checklist.html
blog
sitebulb-vs-librecrawl
about.html
librecrawl.com/
deepcrawl-lumar-vs-librecrawl
best-free-seo-tools.html
large-scale-website-crawling-guide.html
deepcrawl-lumar-vs-librecrawl.html
sitebulb-vs-librecrawl.html


Good Morning, Mallik!
Dashboard
See what's happening at your company!
Datasets
Manage dataset, view dataset reports across workspaces.
Users
Manage users, view user reports across workspaces.
Real-time
See what your users are working on now in the tool.
Billing
Manage billing for your users across workspaces.
Campaign
Send email campaigns, product notifications to your users.

NASLOVI I TEKSTUALNI SADRŽAJ

Naslovi:

- <h1>
  <h2>
  <h3>
  …
  <h6>

Pravilna upotreba elemenata poboljšava:

- čitljivost
- navigaciju
- pristupačnost

Tekstulani sadržaj

Element	Namena
`<p>`	Definisanje pasusa
`<strong>`	Označavanje važnog teksta
`<em>`	Naglašavanje teksta
`<mark>`	Isticanje teksta bojom
`<blockquote>`	Izdvajanje dužeg citata
` `	Prelazak u novi red
`<hr>`	Tematsko razdvajanje sadržaja

LINKOVI I NAVIGACIJA

- Element <a> predstavlja osnovu web navigacije.

Omogućava:

- prelazak između stranica
- povezivanje sa drugim sajtovima
- navigaciju unutar stranice
- preuzimanje resursa
- povezivanje različitih delova aplikacije
- Navigacija je jedan od ključnih elemenata korisničkog iskustva.

SLIKE I MULTIMEDIJA

HTML omogućava integraciju:

- slika
- audio sadržaja
- video sadržaja
- grafike
- drugih multimedijalnih resursa

Primeri:

- <img>
  <audio>
  <video>

HTML tako omogućava da web stranica postane multimedijalna platforma.

RESPONSIVE WEB DIZAJN

Viewport je vidljivi deo web stranice koji korisnik trenutno može da vidi u prozoru browsera ili na ekranu uređaja.
Na računaru viewport zavisi od veličine prozora browsera, dok se na mobilnom uređaju prilagođava širini ekrana.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Parametar	Značenje
width=device-width	širina viewport-a jednaka je širini ekrana uređaja
initial-scale=1.0	početno uvećanje stranice je 100%
viewport	definiše kako browser treba da tretira prikazni prostor

Problemi sa reposnsive karakteristikama

- Responsive dizajn nije samo „smanjivanje stranice“
- Zahteva pravilno projektovanje strukture, sadržaja i interakcije za različite dimenzije ekrana.

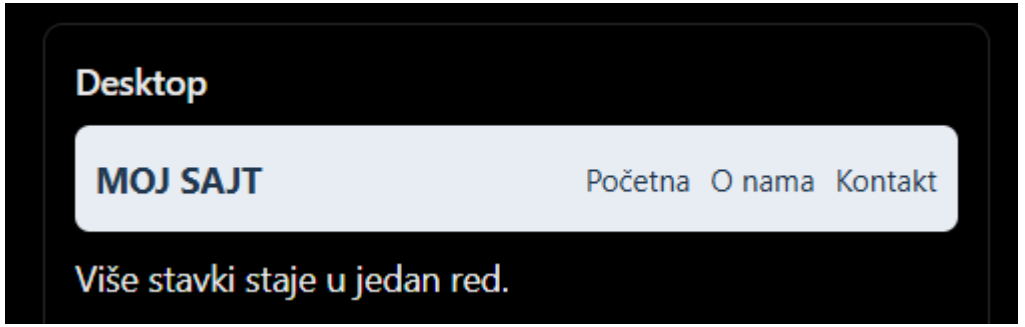
RAZLOZI

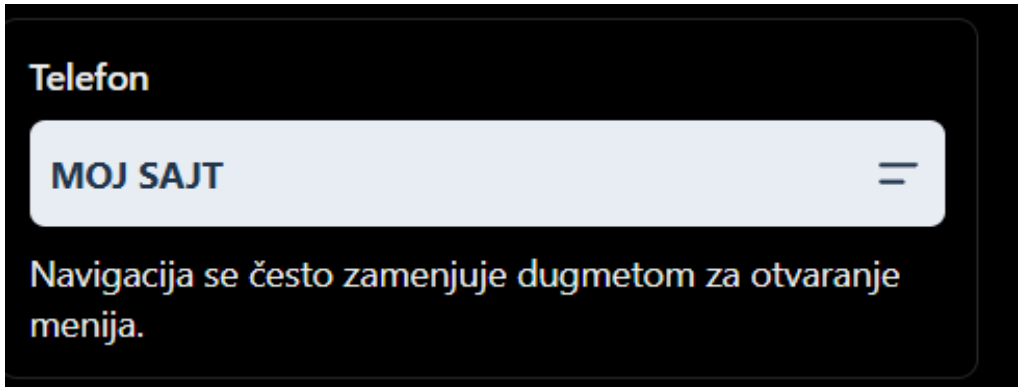
- Sajt je prvo napravljen za desktop!!!!!!
  Mobilna verzija se naknadno „prilagođava“, umesto da se dizajnira od početka za različite ekrane.
- Previše fiksnih dimenzija

width: 1200px;

Umesto fleksibilnih jedinica kao što su %, vw, rem, max-width i flex.

- Loše korišćenje CSS-a
  Ne koriste se pravilno:
  - Flexbox
  - CSS Grid
  - Media Queries
  - fluidne dimenzije
  - responsive typography.

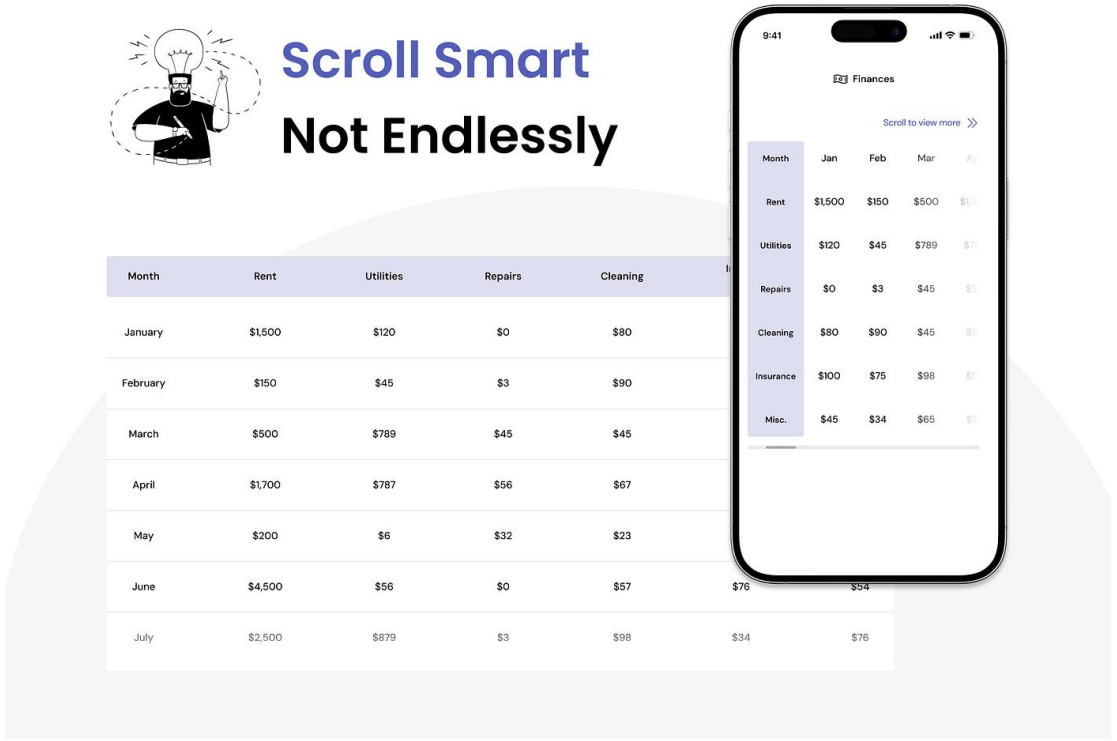




- Sadržaj nije prilagođen malom ekranu
  Tabele, meniji, slike, forme i veliki blokovi teksta mogu zahtevati potpuno drugačiji raspored na telefonu.

- Desktop-first pristup
  Često se projektuje:

  Desktop → tablet → telefon
- Razmisljati o osnovnoj strukturi koja funkcioniše na svim ekranima.
- Veliki broj uređaja i rezolucija
  Nije dovoljno testirati samo „računar i telefon“. Postoje različite širine ekrana, gustine piksela, orijentacije i browseri.



- JavaScript i kompleksne komponente
  Interaktivni meniji, popup prozori, mape, grafikoni i drugi elementi mogu dobro raditi na desktopu, a loše na telefonu.
- Performanse se zanemaruju
  Velike slike, video, fontovi i JavaScript mogu napraviti veoma sporu mobilnu stranicu čak i kada je vizuelno responsive.

HTML I CSS

HTML definiše šta postoji, a CSS definiše kako izgleda.

HTML:

- <button>Pošalji</button>
- CSS određuje:
- boju
- veličinu
- poziciju
- margine
- font
- animacije

Kvalitet HTML strukture direktno utiče na kvalitet CSS organizacije.

CSS se u HTML implementira na tri načina:

Način	Opis	Primer
Inline CSS	CSS se piše direktno u HTML elementu.	`<p style="color: red;">Tekst</p>`
Interni CSS	CSS se piše unutar elementa `<style>` u sekciji `<head>`.	`<style>p { color: red; }</style>`
Eksterni CSS	CSS se čuva u posebnom `.css` fajlu i povezuje sa HTML-om.	`<link rel="stylesheet" href="style.css">`

HTML I JAVASCRIPT

- JavaScript koristi HTML DOM strukturu za interakciju sa sadržajem.

JavaScript može:

- menjati tekst
- menjati stil
- prikazivati i skrivati elemente
- obrađivati događaje
- validirati forme
- komunicirati sa serverom

HTML predstavlja osnovu nad kojom JavaScript gradi interaktivnost.

DOM MODEL

Browser pretvara HTML dokument u DOM – Document Object Model.

- Primer:

- HTML dokument

↓

DOM stablo

↓

JavaScript pristupa elementima

↓

promena sadržaja ili ponašanja

DOM omogućava povezivanje strukture dokumenta sa programskom logikom.

HTML FORME

Forme predstavljaju osnovu komunikacije korisnika sa web aplikacijom.

Primeri:

- prijava korisnika
- registracija
- pretraga
- kontakt forma
- naručivanje
- unos podataka

Osnovni elementi:
<form>
<input>
<select>
<textarea>
<button>

TML VALIDACIJA PODATAKA

HTML omogućava osnovnu validaciju podataka.

- Primeri atributa:
- required
- min
- max
- minlength
- maxlength
- pattern
- type="email"

Prednosti:
smanjuje broj grešaka
poboljšava korisničko iskustvo
smanjuje nepotrebne zahteve serveru
povećava kvalitet aplikacije

HTML I WEB APLIKACIJE

HTML nije samo osnova statičkih stranica.

- Koristi se u aplikacijama kao što su:
- e-commerce
- informacioni sistemi
- društvene mreže
- administrativni sistemi
- obrazovne platforme
- GIS aplikacije
- poslovne aplikacije
- HTML predstavlja korisnički interfejs aplikacije.

HTML I BACKEND

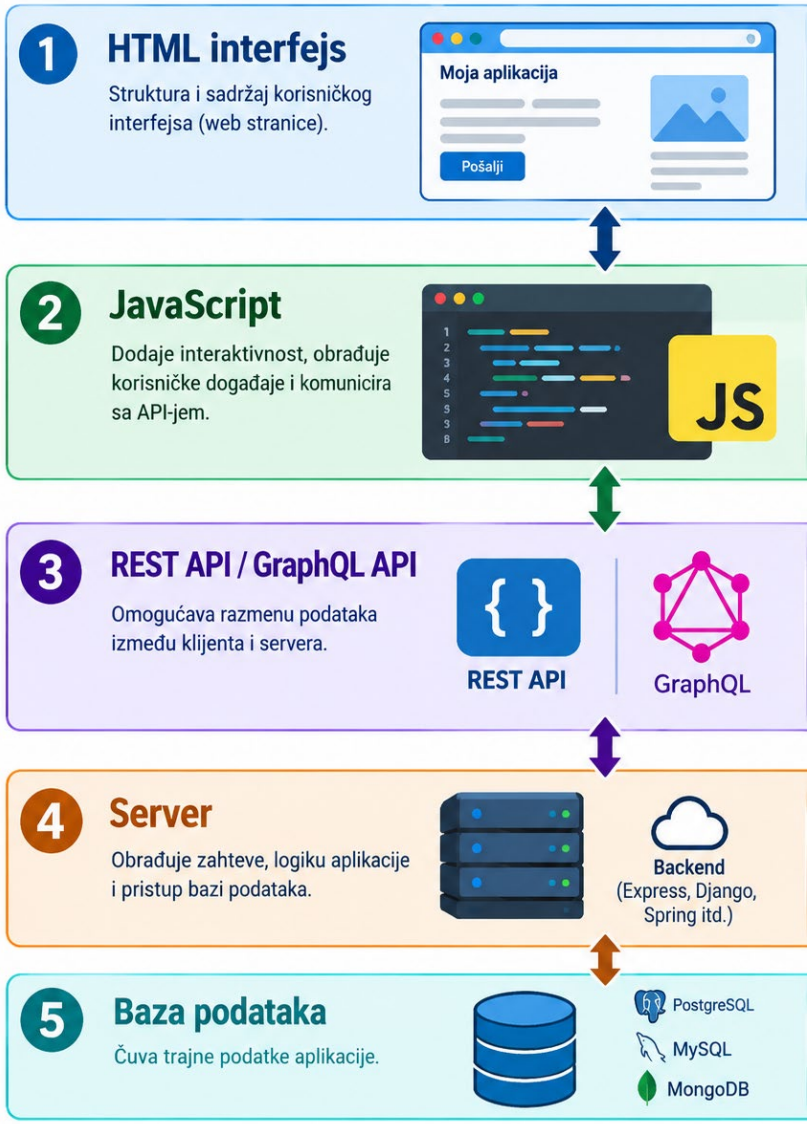
HTML forma može slati podatke serveru.

- Proces:
- Korisnik
  ↓
  HTML forma
  ↓
  HTTP zahtev
  ↓
  Backend
  ↓
  Baza podataka
  ↓
  Odgovor
  ↓
  HTML interfejs
- HTML je zato deo šire arhitekture web aplikacije.

HTML I API

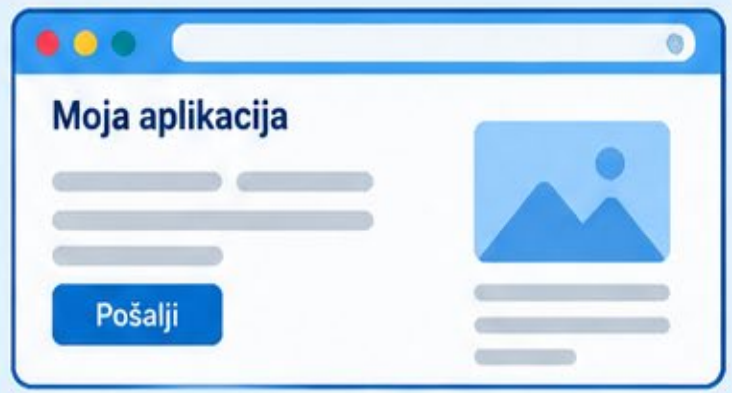
Savremene aplikacije koriste API-je za razmenu podataka.

HTML predstavlja korisnički sloj, dok JavaScript omogućava dinamičku komunikaciju sa API servisima



1 HTML interfejs

Struktura i sadržaj korisničkog interfejsa (web stranice).



- HTML definiše strukturu stranice (zaglavlje, meni, forme, tabele itd.).
- Koristi se semantički HTML (npr. <header>, <nav>, <section>, <form>, <button>).
- Čist i pristupačan HTML je osnova kvalitetnog korisničkog iskustva i SEO-a.

PRISTUPAČNOST (ACCESSIBILITY)

HTML ima ključnu ulogu u pristupačnosti web sadržaja.

- Kvalitetan HTML omogućava korišćenje sajta osobama koje koriste:
- čitače ekrana
- tastaturu
- pomoćne tehnologije
- alternativne načine navigacije

Važni elementi:

- semantički HTML
- odgovarajući naslovi
- alt tekst
- label elementi
- pravilna struktura formi

HTML I SEO

HTML utiče na način na koji pretraživači razumeju sadržaj.

Važni elementi:

- <title>
- meta opis
- naslovi
- semantički elementi
- kvalitetan tekst
- alternativni tekst slika
- pravilna struktura linkova

SEO počinje pravilnom strukturom dokumenta.

HTML I KORISNIČKO ISKUSTVO

intuitivnu navigaciju

jasnu organizaciju sadržaja

brže učitavanje

bolju pristupačnost

konzistentno ponašanje interfejsa

HTML I PERFORMANSE

Kvalitet HTML-a utiče na performanse web stranice.

- Važno je:
  - izbegavati nepotrebne elemente
  - koristiti odgovarajuće HTML elemente
  - optimizovati slike
  - pravilno učitavati skripte
  - smanjiti nepotrebnu složenost DOM-a

Cilj je brži i efikasniji web.

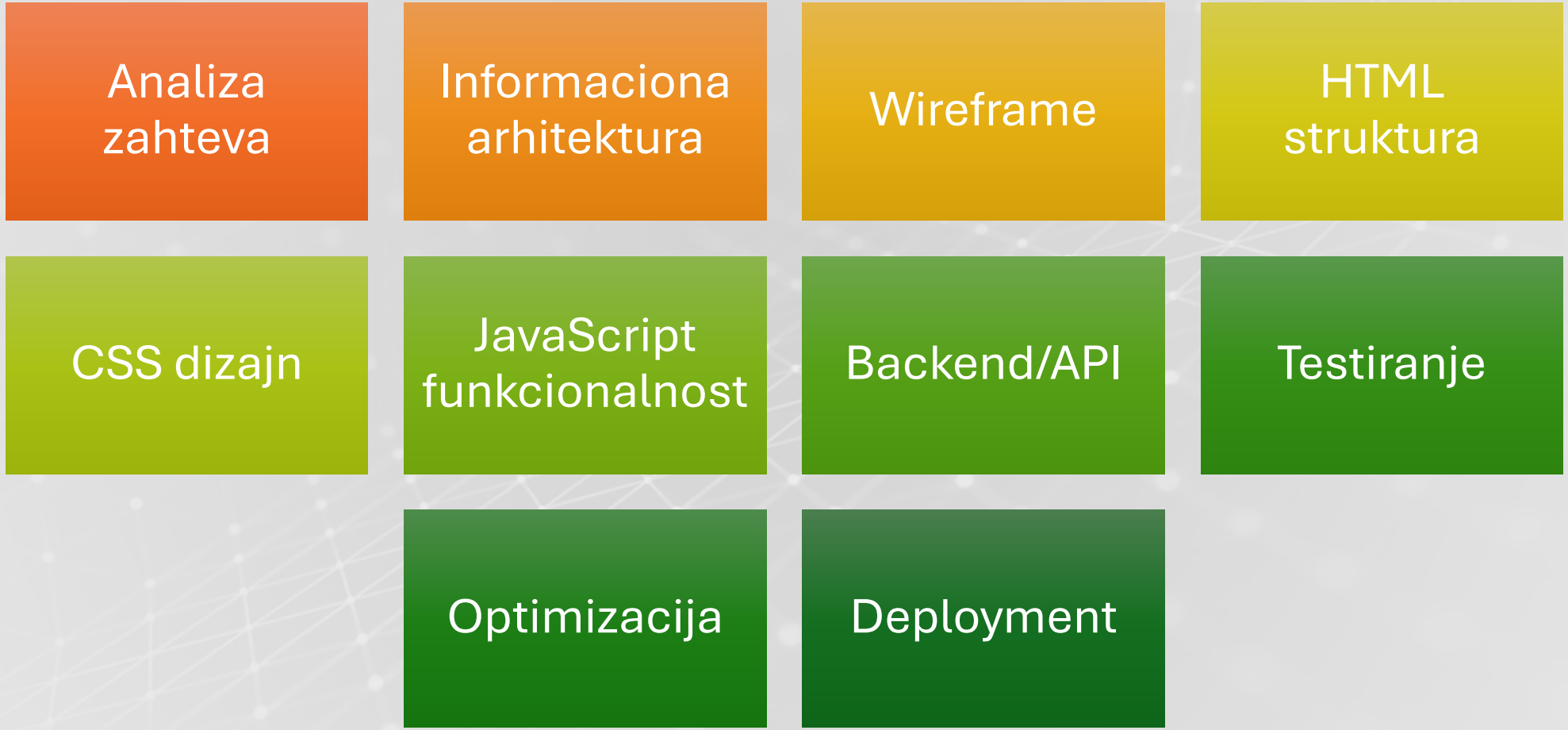
HTML5 I SAVREMENI WEB

HTML5 je doneo veliki broj mogućnosti:

- semantički elementi
- audio
- video
- canvas
- napredne forme
- lokalno skladištenje
- geolokacione mogućnosti kroz web API-je
- podršku za savremene aplikacije

HTML5 je značajno proširio mogućnosti web platforme.

HTML U RAZVOJNOM PROCESU



HTML – od strukture do kompletne web aplikacije

Jedan jezik → Mnogo mogućnosti → Bolji web.

1 HTML struktura

Osnova svakog web sajta.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Moja stranica</title>
  </head>
  <body>
    <h1>Naslov</h1>
    <p>Tekst stranice.</p>
  </body>
</html>
```

- ✓ Semantički elementi
- ✓ Logična hijerarhija
- ✓ Čista struktura

Rezultat: jasan i organizovan sadržaj.

2 Kvalitet dizajna

CSS pretvara strukturu u vizuelni doživljaj.



```
btn {
    background: #4CAF50;
    color: white;
    padding: 10px 20px;
    border-radius: 5px;
}
```

- ✓ Stil i raspored
- ✓ Responzivni dizajn
- ✓ Dosledan izgled

Rezultat: atraktivan i funkcionalan dizajn.

3 Korisničko iskustvo (UX)

Dobar dizajn + logična struktura = zadovoljan korisnik.



```
<nav>
  <ul>
    <li><a href="#">Početna</a></li>
    <li><a href="#">O nama</a></li>
    <li><a href="#">Kontakt</a></li>
  </ul>
</nav>
```

- ✓ Intuitivna navigacija
- ✓ Brza dostupnost sadržaja
- ✓ Konzistentan interfejs

Rezultat: jednostavno i prijatno korišćenje.

4 Pristupačnost

Web dostupan svima.



```
<img src="logo.png"
     alt="Opis slike">

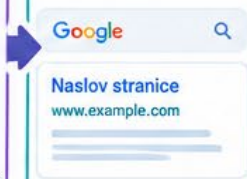
<label for="email">
  Email:
</label>
<input type="email"
     id="email">
```

- ✓ Alt tekstovi
- ✓ Label elementi
- ✓ Prava semantika

Rezultat: uključiv i pristupačan web.

5 SEO

Struktura pomaže pretraživačima.




```
<title>Naslov stranice</title>
<meta name="description"
      content="Kratak opis">
<h1>Glavni naslov</h1>
```

- ✓ Naslov i meta opis
- ✓ Semantički tagovi
- ✓ Kvalitetan sadržaj

Rezultat: bolja vidljivost na pretraživačima.

6 JavaScript interakcija

HTML postaje dinamičan i interaktivan.




```
<button id="btn">
  Klikni
</button>

<script>
  document.getElementById(
    'btn').addEventListener(
    'click', function() {
      alert('Pozdrav!');
    });
</script>
```

- ✓ Promena sadržaja
- ✓ Validacija formi
- ✓ Dinamičke funkcije

Rezultat: interaktivna i pametna aplikacija.

7 API komunikacija

Razmena podataka sa serverom.




```
fetch('https://api.example.com')
  .then(response => response.json())
  .then(data => {
    console.log(data);
  });
```

- ✓ REST API / GraphQL
- ✓ Razmena podataka
- ✓ Dinamički sadržaj

Rezultat: povezanost sa serverom i drugim servisima.

8 Kompletna web aplikacija

Sve komponente zajedno čine funkcionalnu aplikaciju.



- ✓ Struktura (HTML)
- ✓ Dizajn (CSS)
- ✓ Interakcija (JS)
- ✓ API (podaci)
- ✓ Backend (logika)
- ✓ Baza (skladištenje)

Rezultat: kompletno web rešenje spremno za korisnike.

HTML → Struktura → Dizajn → UX → Pristupačnost → SEO → JavaScript → API → Web aplikacija

Kvalitetniji HTML = bolji dizajn + bolje korisničko iskustvo + veća pristupačnost + bolji SEO + interaktivnost + povezivost = uspešna web aplikacija

DOBAR VS. LOŠ HTML

- Dobar HTML:
- semantički
- validan
- čitljiv
- pristupačan
- responzivan
- jednostavan za održavanje

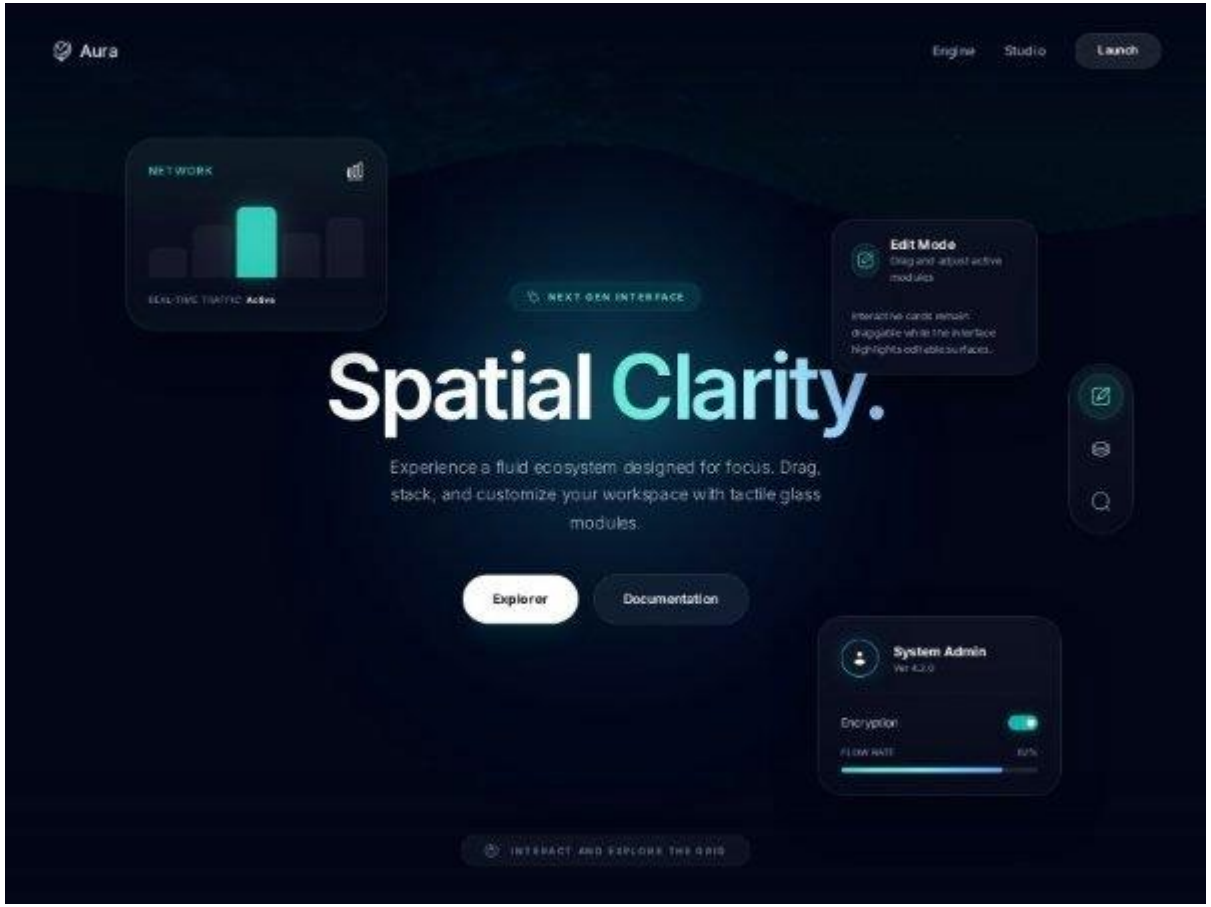
- Loš HTML:
- nejasna struktura
- prekomerna upotreba
- nedostatak semantike
- loša hijerarhija naslova
- nepristupačne forme
- nepotrebna složenost

Koncept	HTML elementi	Značaj za klijentske tehnologije
Struktura dokumenta	<html>, <head>, <body>	Osnova DOM stabla
Semantika	<main>, <article>, <nav>	Značenje i organizacija sadržaja
Korisnički unos	<form>, <input>	Prikupljanje podataka
Validacija	required, min, max, pattern	Provera podataka u pregledaču
Interaktivni sadržaj	<details>, <dialog>	Ugrađene funkcionalnosti pregledača
Prikaz podataka	<table>, <output>, <progress>	Strukturirani rezultati i statusi
Uključivanje resursa	<script>, <link>, <img>	Povezivanje HTML-a sa drugim tehnologijama

Šta je najatraktivnije u HTML-u?

Moderni interfejsi

- Glassmorphism, gradijenti, kartice, senke, mikroanimacije i prilagodljiv raspored. HTML daje strukturu, a CSS vizuelni identitet.



Interaktivne vizuelizacije

- Grafikoni, mape, vremenske linije i dinamički prikaz podataka uz JavaScript i odgovarajuće biblioteke.



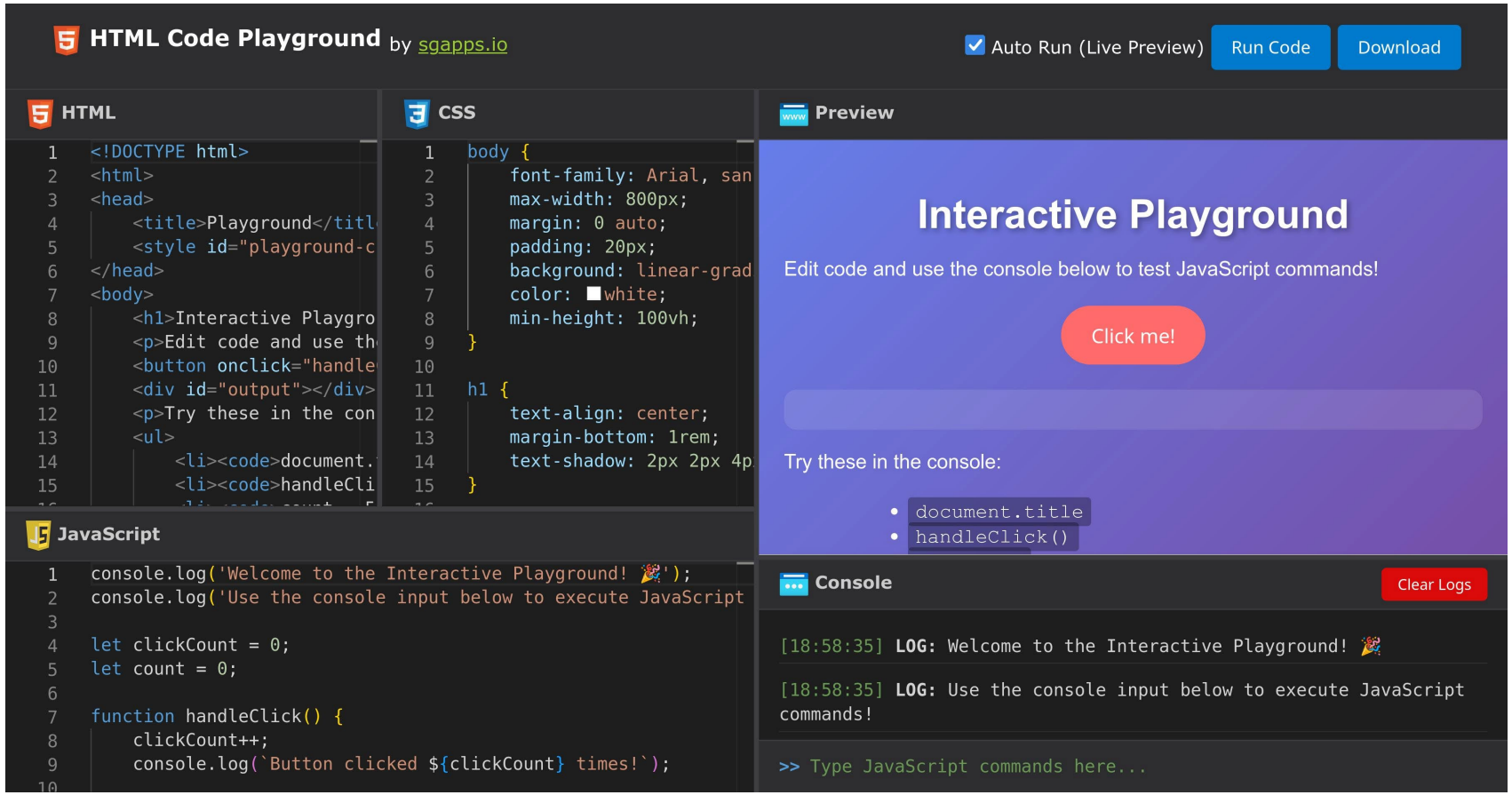
3D veb-iskustva

- Rotirajući objekti, interaktivni modeli i prostorne scene pomoću WebGL-a ili biblioteke Three.js.



Interaktivno učenje programiranja

- menja se HTML kod i odmah vidi rezultat u pregledaču. Odlično za demonstraciju strukture, naslova, pasusa i linkova.



Noviji i napredni HTML elementi

<selectedcontent>

Koristi se u okviru prilagodljivih elemenata <select> za prikaz sadržaja trenutno izabrane opcije u prilagođenom zaglavlju kontrolne liste.

```
<select>
  <button>

<selectedcontent></selectedcontent>
  </button>
  <option>HTML</option>
  <option>JavaScript</option>
</select>
```

<dialog>

Prikazuje dijalog ili modalni prozor, bez potrebe za izradom celog meha-nizma od nule.

```
<dialog open>
  <h2>Obaveštenje</h2>
  <p>Podaci su uspešno obrađeni.</p>
  <form method="dialog">
    <button>Zatvori</button>
  </form>
</dialog>
```

<search>

Semantički označava deo stranice namenjen pretrazi ili filtriranju sadržaja.

```
<search>
 <form action="/search">
  <input name="q"
type="search"
       placeholder="Pretraži
sadržaj">
  <button
type="submit">Pretraži</button
>
 </form>
</search>
```

<template>

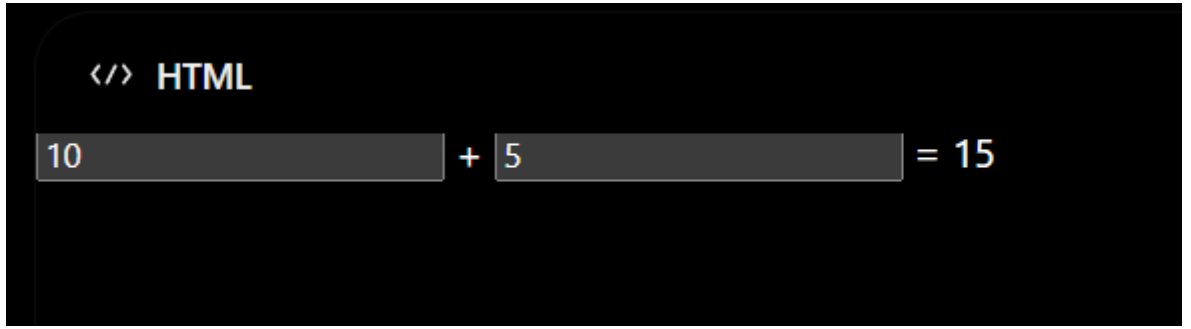
Čuva HTML strukturu koja se ne prikazuje odmah, već se može upotrebiti kasnije, najčešće uz JavaScript.

```
<template id="user-card">
  <article>
    <h2>Korisnik</h2>
    <p>Podaci o korisniku</p>
  </article>
</template>
```

<output>

Predstavlja rezultat proračuna ili korisničke akcije.

```
<form oninput="result.value =
 Number(a.value) + Number(b.value)">
 <input id="a" type="number" value="10">
 +
 <input id="b" type="number" value="5">
 =
 <output name="result" for="a b">15</output>
</form>
```



Novine u HTML-u u radu sa fotografijama

- U savremenom HTML-u, rad sa fotografijama nije ograničen samo na element <img>.

Razvijene su mogućnosti za

- prilagođavanje slike uređaju,
- izbor odgovarajuće rezolucije,
- optimizaciju učitavanja i
- prikaz alternativnog sadržaja.

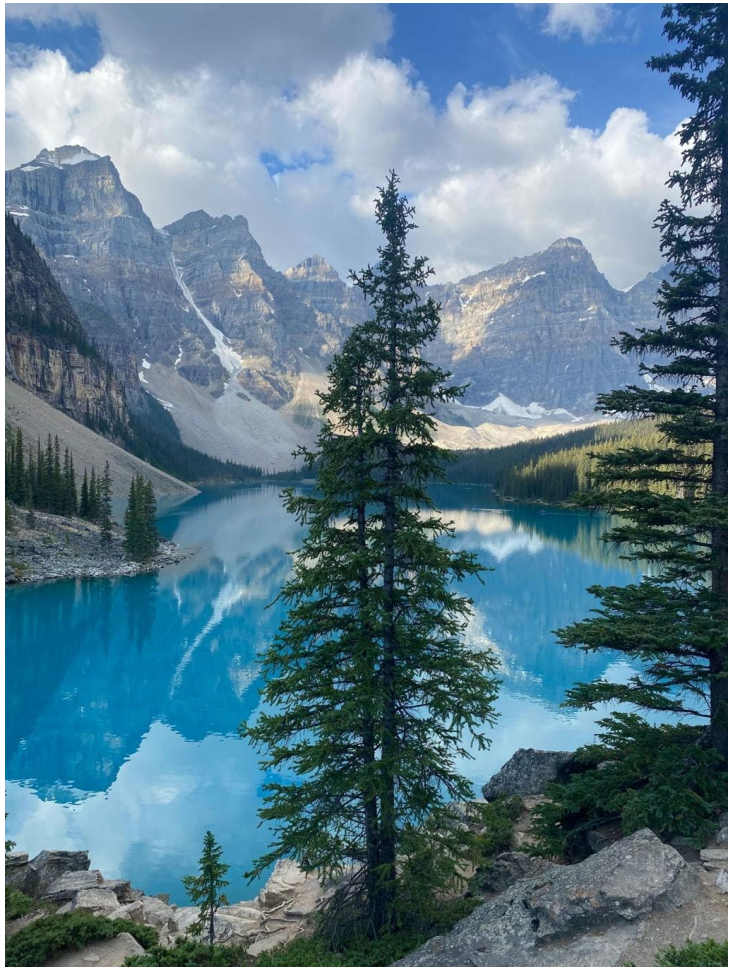
- <picture> — izbor između različitih verzija fotografije.
- srcset i sizes — izbor rezolucije u skladu sa potrebama prikaza.
- loading="lazy" — odloženo učitavanje fotografija.
- decoding="async" — asinhrono dekodiranje slike.
- <img> atributi width, height i alt — dimenzije, stabilnost prikaza i pristupačnost.

- <picture> — različite fotografije za različite uređaje
- Primer: desktop prikazuje panoramsku fotografiju, a telefon vertikalno kadriranu fotografiju istog motiva.



```
<picture>
 <source
  media="(max-width: 600px)"
  srcset="mountain-mobile.jpg">

 <img
  src="mountain-desktop.jpg"
  alt="Planinski pejzaž"
  width="1200"
  height="600">
</picture>
```



srcset i sizes — izbor optimalne rezolucije

Ako je motiv isti, nije potrebno da telefon preuzima fotografiju od nekoliko hiljada piksela kada je prikazuje u malom prostoru. Pomoću srcset i sizes pregledač može da odabere odgovarajući izvor.

```
<img
 src="pejzaz-800.jpg"
 srcset="
  pejzaz-400.jpg 400w,
  pejzaz-800.jpg 800w,
  pejzaz-1600.jpg 1600w"
 sizes="(max-width: 600px) 100vw,
      800px"
 width="800"
 height="500"
 alt="Planinski pejzaž">
```

srcset navodi dostupne verzije fotografije i njihove izvorne širine.
sizes opisuje očekivanu širinu prostora za sliku.
Pregledač bira odgovarajući izvor prema veličini prikaza i karakteristikama uređaja.

- loading="lazy" — odloženo učitavanje
- Fotografije koje su daleko ispod vidljivog dela stranice mogu se učitati tek kada se korisnik približi tom sadržaju. To može smanjiti početni prenos podataka i ubrzati prikaz stranice.
- za glavnu fotografiju vidljivu odmah pri otvaranju stranice obično ne treba koristiti loading="lazy"

```
<img
  src="vodopad.jpg"
  loading="lazy"
  width="800"
  height="600"
  alt="Vodopad u šumi">
```



- decoding i fetchpriority — kontrola učitavanja
- decoding="async" daje pregledaču smernicu da dekodira sliku asinhrono.
- fetchpriority="high" daje smernicu da se fotografija preuzme sa višim prioritetom.
- width i height omogućavaju pregledaču da unapred rezerviše prostor i smanji pomeranje sadržaja tokom učitavanja

```
<!-- Fotografija koja nije prioritetna -->
<img
 src="galerija-1.jpg"
 loading="lazy"
 decoding="async"
 width="600"
 height="400"
 alt="Fotografija iz galerije">

<!-- Glavna fotografija stranice -->
<img
 src="naslovna.jpg"
 fetchpriority="high"
 width="1200"
 height="700"
 alt="Glavni pejzaž">
```

HTML mogućnost	Glavna svrha
<picture>	Različiti kadrovi ili formati slike
srcset	Izbor između više rezolucija
sizes	Opis očekivane širine prikaza
loading="lazy"	Odloženo učitavanje
decoding="async"	Smernica za asinhrono dekodiranje
fetchpriority	Određivanje relativnog prioriteta preuzimanja
width, height	Rezervisanje prostora za sliku
alt	Tekstualni opis fotografije za pristupačnost

Sta podrazumeva ` ` napredni html` `

<canvas> je HTML el površinu dimenzija 400 × 200 piksela, ali je ona još prazna. ement koji definiše površinu za crtanje unutar veb stranice.

Sam <canvas> ne crta automatski ništa. On predstavlja prostor u kome JavaScript može da crta:

- linije i geometrijske oblike,
- grafikone,
- slike,
- animacije,
- jednostavne igre,
- različite vizuelne efekte.

```
<canvas id="mojCanvas" width="400" height="200">
   Vaš pregledač ne podržava canvas.
</canvas>
```



površina dimenzija 400 × 200 piksela, ali je ona još prazna.

```
const canvas = document.getElementById("mojCanvas");
const ctx = canvas.getContext("2d");

ctx.fillStyle = "blue";
ctx.fillRect(50, 50, 150, 80);
```



Pravougaonik je plave boje, dimenzija 150 × 80 piksela.
Boja: plava ("blue").
Širina: 150 piksela.
Visina: 80 piksela.
Početna pozicija: 50 piksela od leve ivice i 50 piksela od gornje ivice platna.

SVG – Scalable Vector Graphics

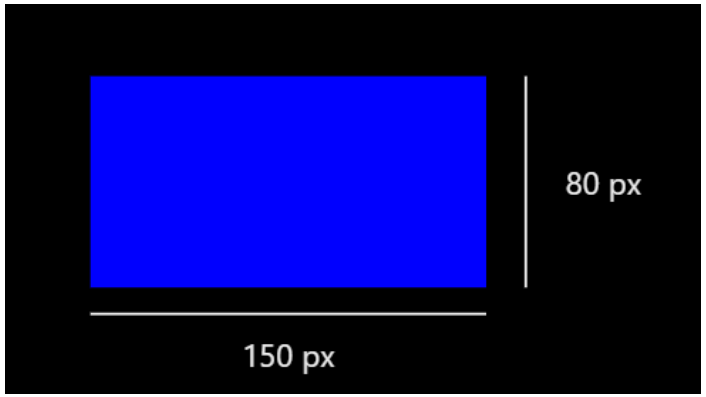
SVG je format za opisivanje vektorske grafike pomoću XML oznaka.

Koristi za prikazivanje geometrijskih oblika, ikona, dijagrama, ilustracija i grafike koja može da se skalira bez gubitka kvaliteta.

Svaki oblik može da bude zaseban element dokumenta, sa sopstvenim atributima i mogućnošću stilizovanja.

```
<svg width="400" height="160"
 xmlns="http://www.w3.org/2000/svg">
  <rect x="50" y="30"
      width="150" height="80"
      fill="blue" />
</svg>
```





Atribut	Značenje
width="400"	Širina SVG površine
height="160"	Visina SVG površine
x="50"	Horizontalna pozicija pravougaonika
y="30"	Vertikalna pozicija pravougaonika
width="150"	Širina pravougaonika
height="80"	Visina pravougaonika
fill="blue"	Boja ispune

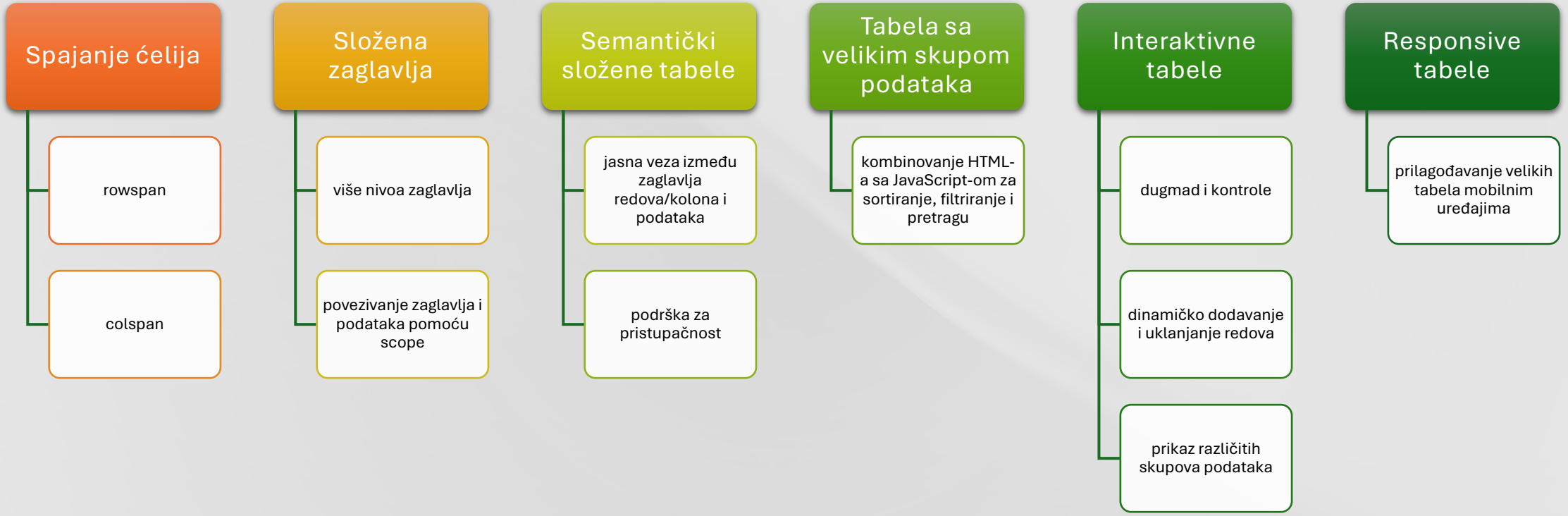
SVG i Canvas – koja je razlika?

SVG	Canvas
Grafika se definiše pomoću elemenata, kao što su <rect>, <circle> i <line>.	Grafika se najčešće iscrtava pomoću JavaScript Canvas API-ja.
Oblici ostaju zasebni elementi kojima se može pristupiti i manipulisati.	Nacrtani oblici uglavnom nisu zasebni DOM elementi.
Pogodan za ikone, logotipe, dijagrame i skalabilne ilustracije.	Pogodan za igre, simulacije i grafiku sa čestim iscrtavanjem.
Skalira se bez gubitka kvaliteta.	Rezolucija zavisi od dimenzija platna i načina prikazivanja.

Strukturiranje i složene HTML tabele“

- <caption> – definiše naslov/opis tabele.
- <thead> – sadrži zaglavlje tabele.
- <tbody> – sadrži glavni deo podataka.
- <tfoot> – sadrži završni deo, na primer zbir ili ukupnu vrednost.
- <th> – definiše zaglavlje reda ili kolone.
- <td> – definiše pojedinačnu ćeliju sa podacima.

Novine su način na koji se tabela koristi i kombinuje sa drugim mogućnostima HTML-a.


Spajanje ćelija
rowspan
colspan
Složena zaglavlja
više nivoa zaglavlja
povezivanje zaglavlja i podataka pomoću scope
Semantički složene tabele
jasna veza između zaglavlja redova/kolona i podataka
podrška za pristupačnost
Tabela sa velikim skupom podataka
kombinovanje HTML-a sa JavaScript-om za sortiranje, filtriranje i pretragu
Interaktivne tabele
dugmad i kontrole
dinamičko dodavanje i uklanjanje redova
prikaz različitih skupova podataka
Responsive tabele
prilagođavanje velikih tabela mobilnim uređajima

```
<table>
  <caption>Plan ispita</caption>
  <thead>
    <tr>
      <th rowspan="2">Predmet</th>
      <th colspan="2">Ispit</th>
      <th rowspan="2">ECTS</th>
    </tr>
    <tr>
      <th>Datum</th>
      <th>Vreme</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>HTML i web dizajn</td>
      <td>15.06.</td>
      <td>10:00</td>
      <td>6</td>
    </tr>
  </tbody>
</table>
```



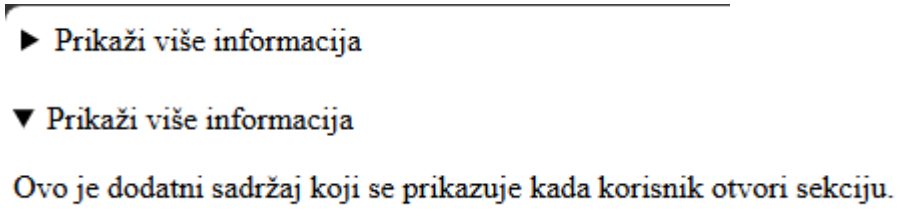
Zaglavlje sa više nivoa, rowspan i colspan, odnosno povezivanje više ćelija u jednu logičku strukturu.

Osnovno: <table>, <tr>, <th>, <td>
Strukturirano: <caption>, <thead>, <tbody>, <tfoot>
Naprednije: rowspan, colspan, scope, složena zaglavlja, pristupačnost i kombinovanje sa JavaScript-om.

Interaktivni HTML elementi

određeni oblici interakcije direktno u HTML-u.

```
<details>
  <summary>Prikaži više informacija</summary>
  <p>
    Ovo je dodatni sadržaj koji se prikazuje
    kada korisnik otvori sekciju.
  </p>
</details>
```



Pregledač automatski omogućava otvaranje i zatvaranje sadržaja.
HTML sam obezbeđuje osnovnu interakciju.

<dialog>

<dialog> predstavlja dijaloški prozor koji može sadržati tekst, obrasce, dugmad i druge HTML elemente.

```
<dialog open>
  <h2>Obaveštenje</h2>
  <p>Podaci su uspešno sačuvani.</p>
  <button>Zatvori</button>
</dialog>
```



Atribut open omogućava da se dijalog prikaže.

JavaScript nije neophodan za postojanje i osnovno korišćenje <dialog> elementa ali je poželjan

HTML = struktura + semantika + određeno ugrađeno ponašanje

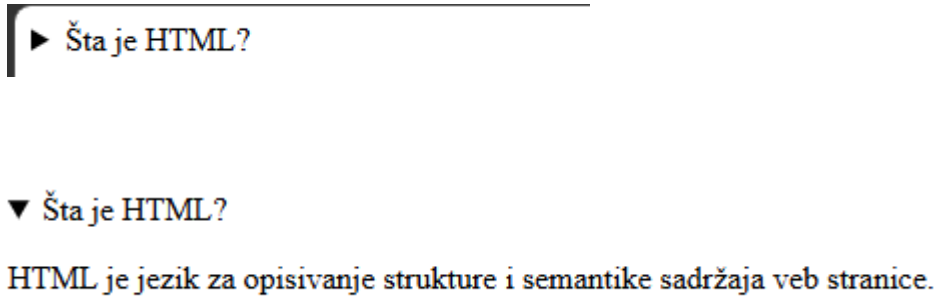
Element	Funkcija	JavaScript za osnovnu funkcionalnost
<details>	Otvaranje/zatvaranje sadržaja	Nije potreban
<summary>	Kontrola <details> elementa	Nije potreban
<dialog>	Dijaloški prozor	Nije uvek potreban
<button>	Interaktivno dugme	Nije potreban za samo dugme

<summary> je HTML element koji definiše naslov odnosno vidljivi deo <details> elementa.

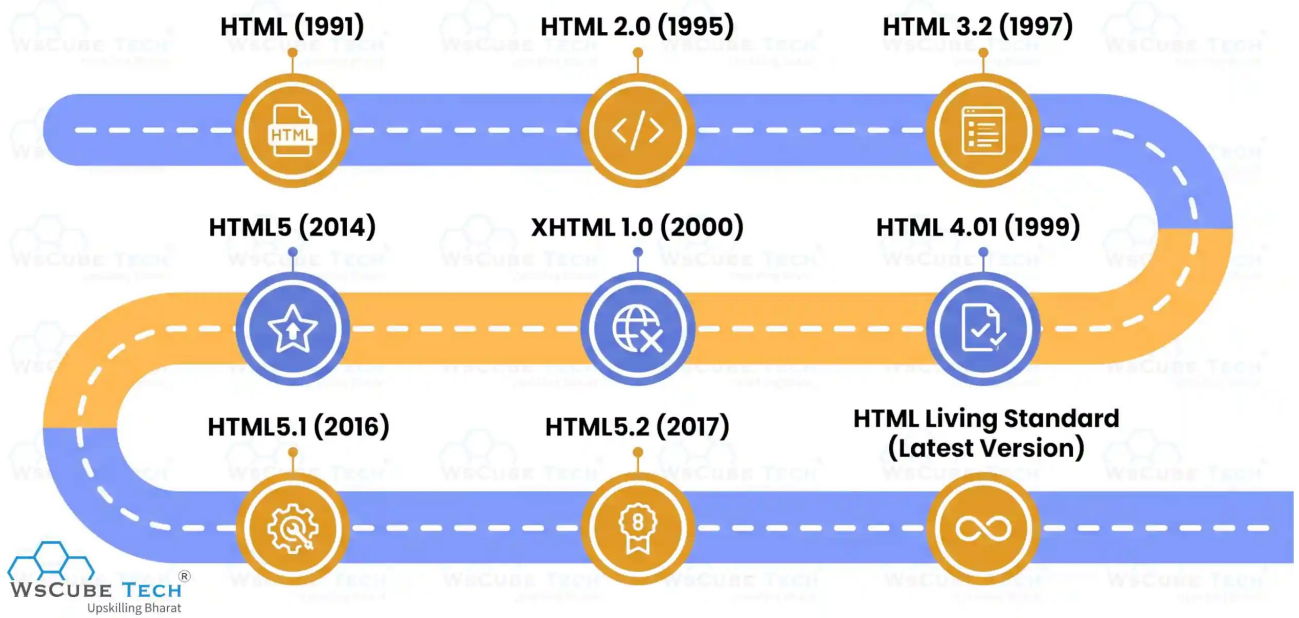
Korisnik klikom na <summary> otvara ili zatvara sadržaj koji se nalazi unutar <details>.

<details> = kontejner za sadržaj koji se može otvoriti/zatvoriti
<summary> = naslov/kontrola kojom korisnik otvara i zatvara taj sadržaj

```
<details>
  <summary>Šta je HTML?</summary>
  <p>
    HTML je jezik za opisivanje strukture i semantike sadržaja veb stranice.
  </p>
</details>
```



<summary> ne radi samostalno. Njegova posebna funkcija proizlazi iz toga što se nalazi unutar <details> elementa.


HTML (1991)
HTML 2.0 (1995)
HTML 3.2 (1997)
HTML5 (2014)
XHTML 1.0 (2000)
HTML 4.01 (1999)
HTML5.1 (2016)
HTML5.2 (2017)
HTML Living Standard
(Latest Version)
WsCube Tech
Upskilling Bharat

01

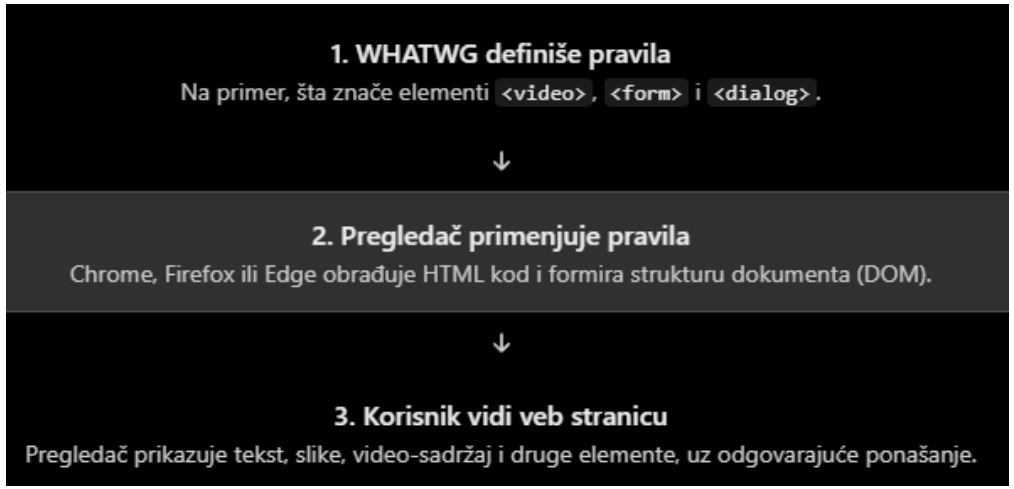
HTML Living Standard predstavlja aktuelnu specifikaciju HTML jezika koju održava organizacija WHATWG.

02

Living Standard se kontinuirano razvija i ažurira.

03

HTML6 nije zvanično objavljena verzija standarda — aktuelni standard se razvija kroz kontinuirane izmene HTML Living Standard specifikacije.



- HTML Living Standard – savremeni standard HTML-a

- HTML Living Standard – definiše pravila i značenje HTML elemenata.
- Veb pregledač (Chrome, Firefox, Edge) –primenjuje ta pravila.
- HTML dokument – konkretna veb stranica koju je developer napisao.

Razvoj i značaj HTML Living Standard-a

HTML 5.2 (2017) – jedna od poslednjih zasebno numerisanih W3C preporuka za HTML.

→

2019. godina – W3C i WHATWG usaglasili su saradnju na jedinstvenoj verziji HTML i DOM standarda.

→

Kontinuirani razvoj – WHATWG redovno ažurira HTML Living Standard, bez potrebe za uvođenjem novih numerisanih verzija.

↓

Kompatibilnost – promene se razvijaju uz nastojanje da se očuva podrška za postojeće veb stranice i pregledače.

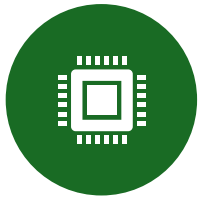
→

Praktični značaj – programeri treba da prate aktuelnu specifikaciju i podršku pregledača, umesto da se oslanjaju isključivo na oznaku verzije HTML-a.

• Šta je WHATWG?



WHATWG (Web Hypertext Application Technology Working Group)



Razvoj i unapređivanjem standarda za veb tehnologije, pre svega HTML-a i DOM-a.



Osnovana je 2004. godine

• Glavne aktivnosti WHATWG-a

- Razvoj HTML-a – održavanje i unapređivanje HTML Living Standard specifikacije.
- Razvoj DOM-a – definisanje načina na koji pregledači predstavljaju HTML dokument i omogućavaju pristup njegovim elementima.
- Kompatibilnost – preciziranje pravila ponašanja veb pregledača radi ujednačenog prikazivanja i obrade sadržaja.
- Kontinuirano unapređivanje – objavljivanje izmena specifikacija bez potrebe za uvođenjem nove numerisane verzije HTML-a.

Pojam	Značenje
WHATWG	Radna grupa koja održava HTML Living Standard.
W3C	Međunarodni konzorcijum koji razvija i koordinira brojne veb standarde.
HTML	Jezik kojim se definišu struktura i značenje sadržaja veb stranica.
HTML Living Standard	Aktuelna, kontinuirano ažurirana specifikacija HTML-a koju održava WHATWG.

```
!DOCTYPE html>
<html lang="sr">
<head>
  <meta charset="UTF-8">
  <title>Moja stranica</title>
</head>
<body>
  <h1>Dobro došli!</h1>
  <p>Ovo je moja prva veb stranica.</p>
  <button>Potvrdi</button>
</body>
</html>
```

Kada datoteku otvorimo u pregledaču:

- Pregledač prepoznaje HTML oznake.
- Na osnovu njih formira DOM stablo, odnosno unutrašnju strukturu dokumenta.
- Prikazuje naslov, tekst i dugme.
- Dugme je vidljivo, ali samo po sebi ne izvršava posebnu akciju jer mu nismo definisali šta treba da radi.

Ovakav način obrade HTML-a definisan je pravilima HTML Living Standard-a

Novine u radu sa video sadržajima

- <video> — video postaje nativni HTML sadržaj
- Najveća promena u HTML5 bila je mogućnost da pregledač direktno reprodukuje video!!!

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

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

- <track> — titlovi i sinhronizovani tekst

Ovo je jedna od najvažnijih mogućnosti za nastavu

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

  <track
    src="lecture-sr.vtt"
    kind="subtitles"
    srclang="sr"
    label="Srpski">
</video>
```

Omogućava:
•titlove,
•prevode,
•opise za osobe sa oštećenjem sluha,
•druge vremenski sinhronizovane tekstualne podatke.

poster — fotografija pre početka videa

```
<video
   controls
   poster="preview.jpg">
   <source src="lecture.mp4"
type="video/mp4">
</video>
```

Umesto praznog video okvira, korisnik prvo vidi fotografiju.
Slika → Play → Video

- Autoplay, muted, loop, playsinline

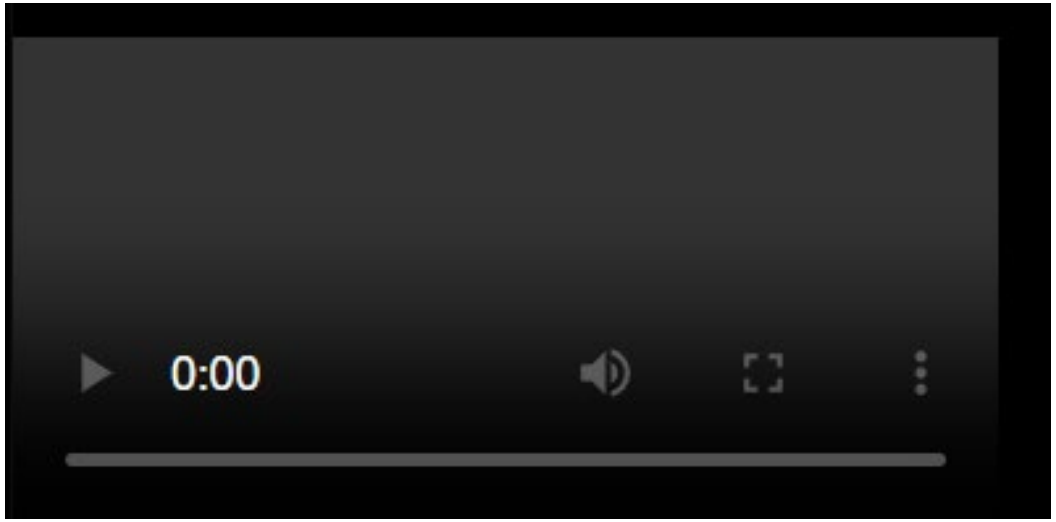
```
<video
   autoplay
   muted
   loop
   playsinline>
   <source src="background.mp4"
         type="video/mp4">
</video>
```

omogućava preciznije definisanje ponašanja videa

autoplay + muted + loop + playsinline
koja se koristi za kratke pozadinske video sekvence.

- preload — kontrola učitavanja

```
<video
   controls
   preload="metadata">
   <source src="video.mp4" type="video/mp4">
</video>
```



none
metadata
auto

- HTML video + JavaScript

```
<video id="myVideo" controls>
  <source src="lecture.mp4" type="video/mp4">
</video>
```



HTML definiše video

JavaScript može da upravlja njime



```
const video =
document.querySelector("#myVideo");
video.play();
video.pJavaScript može da upravlja njime:ause();
video.currentTime = 30;
```

HTML mogućnost	Šta donosi
<video>	Nativna reprodukcija videa
<source>	Više video formata
<track>	Titlovi i sinhronizovani tekst
poster	Početna fotografija
autoplay	Automatsko pokretanje
muted	Kontrola zvuka pri automatskoj reprodukciji
loop	Automatsko ponavljanje
playsinline	Reprodukcija unutar stranice, naročito značajna na mobilnim uređajima
preload	Kontrola predučitavanja
JavaScript API	Programska kontrola videa