

1.Uvod u World Wide Web

Radi se o sustavu za razmjenu informacija dostupnim putem interneta. Često se Web poistovjećuje s Internetom, tako da mnogi misle da su Web i Internet jedno te isto, a nisu.

Web se temelji na konceptu **hiperteksta**, dokumenta koji sadrži izravnu vezu s nekim drugim dokumentom. Naravno, taj dokument je danas obogaćen i slikama, tablicama, zvučnim zapisima, videosekvencama, navigacijskim menijima...



World Wide Web je 1990 počeo život na mjestu gdje bi to tadanažmanje očekivali, na CERN-u, Europskom laboratoriju za fiziku čestica u Ženevi

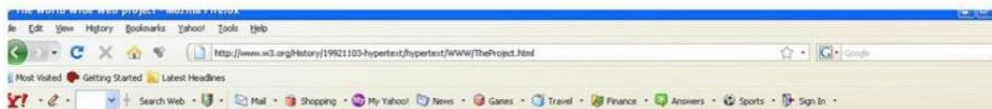


Tim Berners-Lee

Tim Berners-Lee, britanski informatičar sa CERN instituta sa nekoliko svojih kolega pokrenuo je inicijativu za **korištenje Interneta u razmjeni informacija** na način da jednostavnim klikanjem na označene dijelove teksta (**linkovi**) otvaraju novi dokumenti.

Stup cijelog projekta postaju dva elementa:

1. HTTP
2. HTML



World Wide Web

The WorldWideWeb (W3) is a wide-area [hypertext](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#), [Policy](#), November's [W3 news](#), [Frequently Asked Questions](#).

[What's out there?](#)

Pointers to the world's online information, [subjects](#), [W3 servers](#), etc.

[Help](#)

on the browser you are using

[Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#), [X11 Viola](#), [NeXTStep](#), [Servers](#), [Tools](#), [Mail robot](#), [Library](#))

[Technical](#)

Details of protocols, formats, program internals etc.

[Bibliography](#)

Paper documentation on W3 and references.

[People](#)

A list of some people involved in the project.

[History](#)

A summary of the history of the project.

[How can I help?](#)

If you would like to support the web.

[Getting code](#)

Getting the code by [anonymous FTP](#), etc.

Izgled prvih web stranica

.1. HTTP protokol

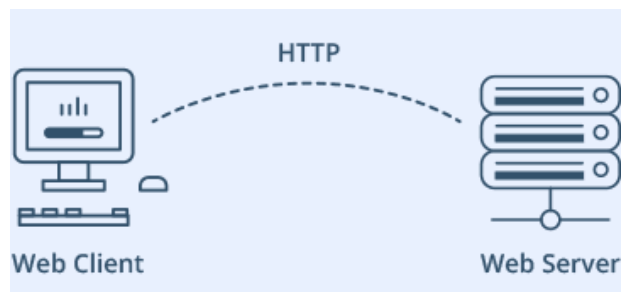
HyperText Transfer Protocol je temelj Word Wide Weba i **komunikacijski protokol** pomoću kojeg je moguće povezati se sa serverima sirom svijeta. Primarna funkcija HTTP protokola je da uspostavi vezu sa serverom, traži sve potrebne informacije za prikaz web stranice i preuzme odgovor od servera.

HTTP i HTTPS su glavna i najčešća metoda prijenosa informacija na Webu.

<http://www.nekisajt.hr>

ili

<https://www.nekisajt.hr>



Internet funkcionira otprilike slično kao i telefonski sustav u koji su također spojeni milijuni telefona, a nama je omogućeno da sa svoje linije nazovemo bilo koji drugu liniju telefona u svijetu ako znam njezin broj.

Na Internetu međusobno ne komuniciraju ljudi, već umjesto njih čine računalni programi.

Tajna je u tome što se Internet temelji na programima koje zovemo „**klijenti**“ i „**serveri**“

Serveri su **softverski programi** koji se nalaze na hostovima (računalima) te koji permanentno **očekuju upit i pruža uslugu** za koju je namijenjen.

Klijenti su **softverski programi** koji se nalaze na računalima te koji **traže upit** od servera neki podatak,. Web preglednik (npr.chtome) je jedan takav klijentski program

1.2. HTML jezik

HTML je kratica za **HyperText Markup Language**, je **prezentacijski jezik** za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. Definira sadržaj i strukturu web stranice.

HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima.

Prikaz hipertekst dokumenta omogućuje **web preglednik (Chrome, Opera, Edge,...)**.

HTML nije programski jezik niti su ljudi koji ga koriste programeri. Njime ne možemo izvršiti nikakvu zadaću, pa čak ni najjednostavniju operaciju zbrajanja ili oduzimanja dvaju cijelih brojeva.

Html datoteke su zapravo obične tekstualne datoteke, ekstenzija im je .html ili .htm. Osnovni građevni element svake stranice su **znakovi (tags)** koji opisuju kako će se nešto prikazati u web pregledniku.

1.2.1. Povijesni razvoj HTML jezika

Prvi javno dostupan opis HTML-a na internetu od strane Tim Berners-Leeja **krajem 1991**. Taj opis se sastoji od 20 elemenata (tagova). Trinaest tih elemenata još uvijek postoji u HTML-u. Tim Berners-Lee je nakon što je u listopadu 1994. napustio CERN (Europsku organizaciju za nuklearno istraživanje), osnovao organizaciju **World Wide Web Consortium** koja se danas bavi standardizacijom web-a i poznatija je kao **W3C**

Prva verzija HTML1 jezika objavljena je **1993. godine**. U to je vrijeme bio još poprilično ograničen, pa nije bilo moguće čak ni dodati slike u HTML dokumente.

U ožujku 1995. W3C objavljuje verziju **HTML 2**, koja donosi mogućnosti definicije tablica, formi,... Ovo je ostao standardni jezik za izradu web stranica do 1997.

HTML3 predstavljen je u **prosincu 1997**. pod nadzorom **W3C**. Nastavio je s prihvaćanjem tagova nametnutih od strane proizvođača različitih web preglednika, no istovremeno je pokrenuto i "čišćenje" standarda proglašavanje nekih od njih suvišnim. Preglednici poput Netscape Navigatora i Internet Explorera dominiraju tržištem.

HTML4 verzija **1999**. Najpopularnija "klasična" verzija HTML-a.. Uvodi razdvajanje strukture (HTML) i stila CSS-a. Tri varijante: *Strict*, *Transitional*, i *Frameset*.

XHTML 1.0 (Extensible HyperText Markup Language) je **2000**. nastao na preporuku W3C nova verzija jezik za izradu web stranica, a došao je kao zamjena za HTML. U osnovnoj verziji on nije puno različitiji od **HTML4**, nego se može reci da je to pročišćena verzija HTML-a.

XHTML 1.1 novi jezik za izradu web stranica pojavljuje se **2001** na preporuku W3C.

HTML 5 je nastao u suradnji World Wide Web Consortium (W3C) i Web Hypertext Application Technology Working Group (WHATWG). Do 2006. godine su ove dvije grupe radile odvojeno, WHATWG je radio sa web formama i aplikacijama, a W3C sa XHTML 2.0.

Na svu sreću odlučili su udružiti snage i kreirati novu verziju **HTML-a**. HTML5 nastaje **2014** i najveći je iskorak u povijesti razvoja HTML jezika. HTML5 donosi brojne nove mogućnosti koje HTML 4.01 i XHTML 1.x nisu imali, kao što je mogućnost reprodukcije videa i audia na stranicama bez korištenja Adobe flasha iil Microsoftovog silverlighta, plus novi semantički tagovi,...

HTML više nije verziran klasično (npr. HTML6,...). Verzija HTML5 se stalno ažurira

1.2.2.Web preglednici - browseri



Marc
Andreessen

Svaki pojedini hipertekstualni dokument na Webu zovemo web-stranica. Na cijelom Internetu postoji ogroman broj računala koje nazivamo web-serverima. Na njima su smještene sve web-stranice i dostupne su svakom korisniku interneta, u čitavom svijetu. Da bismo pristupili web stranicama potreban je **program** kojeg zovemo **web-browser** ili u prijevodu **web-preglednik**.

Marc Andreessen je razvio prvi web-browser sa grafičkim sučeljem, browser podimenom **Mosaic**.



Firefox je besplatni preglednik otvorenog koda koji je predvodnik u inovativnosti opcija kojima se olakšava pregledavanje web stranica. U zajedništvu sa e-mail klijentom Thunderbird, čini grupaciju koju nazivamo Mozilla.

Opera je besplatan preglednik zatvorenog koda tvrtke Opera. Bio je prvi koji je prepoznao pregledavanje web stranica na mobitelima i drugim mobilnim uređajima. Zato je danas čest gost na operativnim sustavima za mobitele.



Safari je ime web preglednika kojeg je razvila američka tvrtka Apple Computer i dostupanje za operacijski sustav OSX. Paralelno sa predstavljanjem iPhone-a Apple je razvio i mobilnu inačicu ovog internet preglednika pod nazivom Mobile Safari koji se posebno prilagođen pregledu na manjim ekranima i korištenju na slabijim uređajima poput mobilnih telefona odnosno mp3 playera.

Crome je web browser dolazi iz Google-a i kombinira minimalistički dizajn i visoko sofisticiranu tehnologiju koja dozvoljava brže, sigurnije i jednostavnije pretraživanje. Chrome je besplatni preglednik otvorenog koda.



1.3. Kako se prikazuju web stranice u web pregledniku

Kada se želimo spojiti na internet, prvo moramo naći davatelja internet usluga u Hrvatskoj – **ISP (Internet Service Provider)**.

U Hrvatskoj su to npr.: CARNet, Iskon, T-com, ...

Kako je na Internetu spojeno ogroman broj računala, svakom računalu koje se spoji makar i na 5 minuta dodijeli se **jedinstvena brojčana oznaka** koju zovemo **IP adresa**.

Svako računalo na Internetu ima različitu IP adresu.

Broj IP adrese može izgleda ovako: **142. 165. 85. 43.**

1. Korisnik u web pregledniku upisuje adresu web stranice (npr. <http://www.drenik.hr>) i preglednik šalje HTTP zahtjev web serveru za dohvaćanje tražene stranice

Kako bi lakše pamtili adrese serverima se dodjeljuje **simboličke adrese** tipa **www.drenik.hr** a IP adresa te simboličke adrese je **195.252.112.4**. Naravno da je lakše zapamtiti ime nego brojeve.

drenik.hr - ovakva adresa jednostavno se naziva **domena** - to je jedinstveno ime za Internet adresu. Naziv domene je unikatna, tj. ne može biti više domena istog naziva.

Oznaka **hr** naziva se **vršna domena** i svaka država ima svoju vršnu domenu (hr = hrvatska)

au = Austrija

hu = Mađarska

it = Italija

my = Malezija

com = komercijalna adresa

org = organizacija

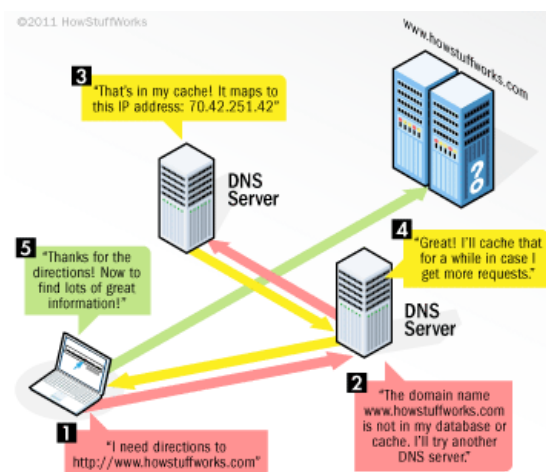
net = mrežne usluge

edu = američke visokoškolske ustanove

gov = američke vladine ustanove

mil = američke vojne ustanove

2. Pretvaranje simboličkih adresa u numeričke (IP adrese) odrađuju **DNS serveri**. DNS serveri se nalaze širom svijeta i zaduženi su za pretvaranje imena tj adresa domena u IP adrese. DNS je zasnovan na hijerarhijskom principu i jedna je od osnovnih komponenti interneta.
3. Kada je pronađena IP adresa uspostavlja se kontakt između web preglednika (klijentski program) i web servera (serverskog programa)
4. Program na serveru pruža uslugu koja je tražena njega, podaci se najprije dijele i raspoređuju u elektronske pakete. Svi ti podaci su numerirani i označeni IP adresom računala kojem se podaci šalju. Paketići ne moraju putovati istim putem do računala koje ih prima
5. Na drugom kraju veze računalo - klijent sakuplja paketiće i njihov sadržaj međusobno spaja u jednu cjelinu i prikazuje u web pregledniku.



od

Ovakav sustav ima dvije važne karakteristike:

Efikasnost – Internet je organiziran tako da prenese sve pakete na najbrži mogući način.

Pouzdanost – odredišno računalo (klijent) zna se služiti metodama po kojima provjerava svaki pristigli paket kako bi se uvjerilo da nije došlo ni do kakve greške tijekom prijenosa. Ako paket sadrži neku grešku, odredišno će računalo (klijent) poslati poruku računalu pošiljatelju (server) te zatražiti taj i taj paket pošalji ponovo.

Dva najvažnija protokola koji omogućavaju razmjenu paketa (podataka) su:

IP – koristi se za adresiranje

TCP – namijenjen je kontroli paketa i provjeravanju jesu li svi paketi pristigli do odredišta bez greške.

Ta su dva protokola toliko važna da se na internetu zove zajedničkim imenom **TCP/IP**. Milijuni računala na internetu, a na kojima su pokrenuti klijentski i serverski programi razmjenjuju pakete poštujući TCP/IP protokol.

2. Faze razvoja web stranice

Razvoj web stranice obično prolazi kroz nekoliko ključnih faza.

1. Planiranje i analiza
2. Izrada dizajna
3. Razvoj web stranice
4. Testiranje i ispravljanje pogrešaka
5. Implementacija
6. Administriranje i ažuriranje

2.1. Planiranje i analiza

Ukratko: u ovoj fazi se postavljaju **odgovori na pitanja "Zašto?", "Za koga?" i "Što točno radimo?"**. Kad je planiranje odrađeno kvalitetno, ostatak procesa ide puno jednostavnije.

Što uključuje:

- Definiranje cilja stranice (informiranje, povećati prodaju, edukacija, promocija).
- Identifikacija ciljane publike (npr. učenici, kupci, roditelji).
- Analiza konkurencije – kako izgledaju i funkcioniraju slične stranice.
- Izrada strukture sadržaja (*sitemap*) – koje će stranice i podstranice postojati i njihova hijerarhija
- Plan sadržaja (tko piše tekstove, tko dostavlja slike, rokovi).

Zašto je važno:

Ako nema jasnog plana, stranica može biti nepregledna i nefunkcionalna.

2.1.1.Organizacija sadržaja – struktura - sitemap

Strukturni prikaz stranice – pokazuje kako su stranice i podstranice međusobno povezane.

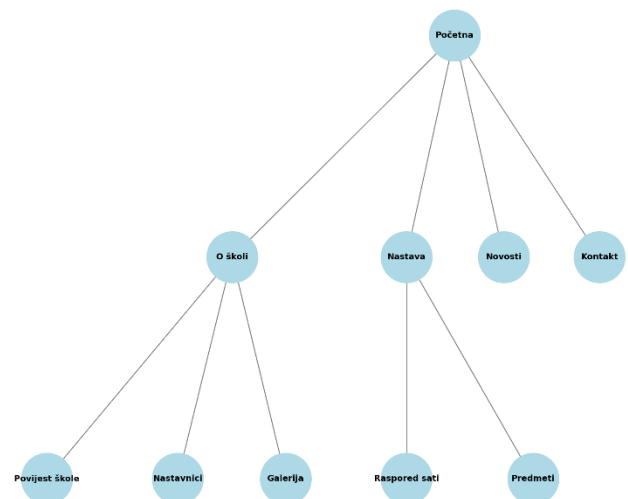
a) Hijerarhijska struktura (stablo)

Najčešća i najpreglednija. **Hijerarhijska struktura** znači da su elementi organizirani u obliku definiranih razina – od prve razine prema nižim razinama.

Glavne stranice (npr. *Početna, O školi, Nastava, Novosti, Kontakti*) nalaze se na **prvoj razini**. **Podstranice** (npr. *Povijest, Nastavnici, Galerija,...*) su **druga razina**. Ako podstranica ima svoje podstranice (npr. *Predmeti* → *Matematika*), to je **treća razina**, i tako dalje.

Primjer (školska stranica):

- Početna (home)
- O školi
 - Povijest škole
 - Nastavnici
 - Galerija
- Nastava
 - Raspored sati
 - Predmeti
- Novosti
- Kontakt



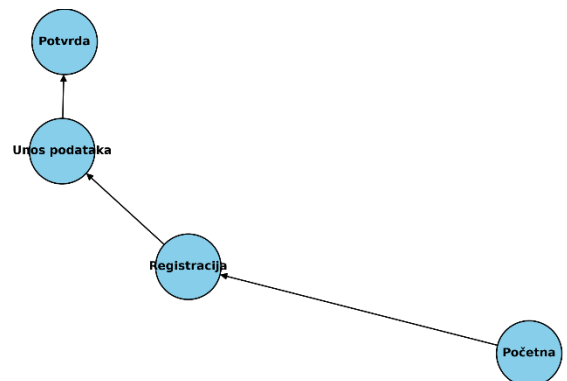
Prednost: jednostavna za navigaciju, korisnik lako pronalazi informacije.

b) Linearno povezana struktura

Stranice se otvaraju jedna za drugom, kao koraci u procesu i definiran je točan redoslijed kojim se posjećuju web stranice.

Primjer (online prijava):

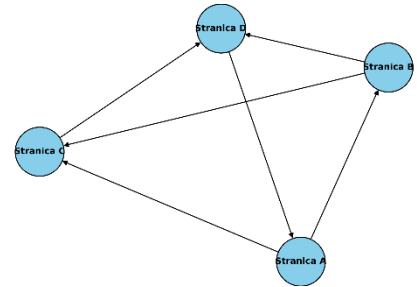
Početna → Registracija → Unos podataka → Potvrda



Prednost: korisnika vodi kroz točno određen tijek.

c) Mrežna struktura

Za razliku od hijerarhijske (stablo), gdje postoji jasan put od vrha prema dnu, mrežasta struktura je umrežena – stranice nisu „posložene po razinama“, već se mogu međusobno povezivati na više načina. Nema stroge hijerarhije. Stranice su povezane linkovima u različitim smjerovima. Korisnik može doći do sadržaja iz više polaznih točaka. Često se koristi u wiki sustavima, portalima, blogovima ili većim bazama znanja.



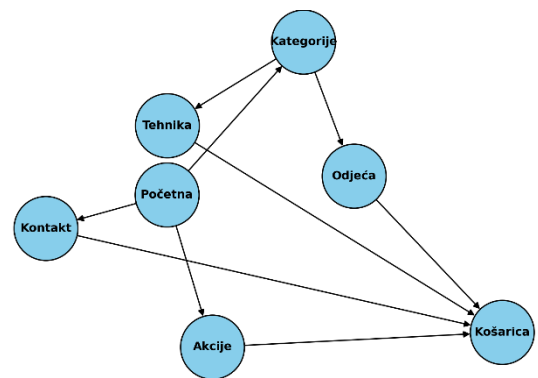
Prednost: fleksibilna i informacijski bogata.

d) Kombinirana struktura

Najčešća u praksi – osnovna hijerarhija uz dodatne veze među stranicama.

Primjer (web shop):

- Početna
- Kategorije proizvoda
 - Odjeća
 - Tehnika
- Akcije
- Kontakt
- (s bilo koje stranice se može preći na "Košaricu")



Prednost: spaja preglednost hijerarhijske strukture i fleksibilnost mrežne strukture.

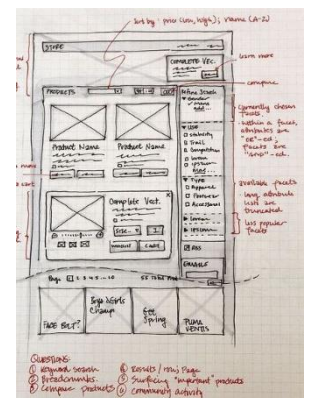
2.2. Izrada dizajna

Što uključuje:

- Izradu *wireframea* (skice izgleda stranice).
- Odabir boja, fontova, slika i vizualnog identiteta.
- Planiranje rasporeda elemenata (navigacija, izbornici, gumbi).
- Fokus na korisničko iskustvo (UX) – jednostavna, brza i intuitivna navigacija.

Zašto je važno:

Privlačan i jasan dizajn zadržava korisnika i čini stranicu funkcionalnom.



2.3. Razvoj web stranice

Nakon što je dizajn odobren, kreće tehnička realizacija – razvoj web stranice.

- Izrada kostura stranice pomoću HTML-a.
- Uređivanje vizualnog izgleda pomoću CSS-a.
- Programiranje - dodavanje interaktivnosti pomoću npr. JavaScripta.
- Povezivanje s bazama podataka ili CMS-om (npr. WordPress).

Zašto je važno:

Ovo je tehnička faza u kojoj se stranica pretvara iz ideje i dizajna u funkcionalan proizvod.

2.4. Testiranje

Što uključuje:

- Provjera radi li svaki link i funkcija (npr. obrasci, registracija, košarica).
- Testiranje na različitim uređajima (računalo, mobitel, tablet).
- Testiranje u različitim preglednicima (Chrome, Firefox, Edge).
- Brzina učitavanja i optimizacija.
- Ispravljanje bugova.

Zašto je važno:

Otkrivaju se i uklanjaju problemi prije nego korisnici dođu na stranicu.

2.5. Implementacija (objava)

Što uključuje:

- Kupnja domene (npr. moja-stranica.hr).
- Odabir i konfiguracija hostinga (mjesto gdje se stranice postavljaju).
- Prijenos datoteka na poslužitelj.
- Konačna provjera funkcioniranja stranice uživo.

Zašto je važno

Tek objavom stranica postaje dostupna korisnicima širom svijeta.

2.6. Održavanje i nadogradnja

Što uključuje:

- Redovito ažuriranje sadržaja (nove vijesti, slike, proizvodi).
- Sigurnosne nadogradnje i zaštita od napada.
- Analiza posjećenosti i ponašanja korisnika (npr. Google Analytics).
- Dodavanje novih funkcionalnosti prema potrebama korisnika.

Zašto je važno:

Stranica koja se ne održava brzo zastari, postaje nesigurna ili neprivlačna korisnicima.

3. Tehnologije i jezici za izradu web stranice

Neki bitni pojmovi:

web stranica (engl. **web page**) je jedan HTML dokument koji se prikazuje u web pregledniku (Chrome, Firefox, Safari, itd.)

web mjesto (engl. **website**) je skup povezanih web stranica koje dijele zajedničku domenu i obično su postavljene na istom web serveru.

web poslužitelj (engl. **web server**) je posebno računalo koje je umreženo na internetu i njegov softver "poslužuje" web stranice korisnicima na zahtjev u web pregledniku.

web preglednik (engl. **web browser**) je softverska aplikacija koja vam omogućuje pristup i prikaz web stranica na internetu. To je vaš "prozor u svijet weba" - alat koji prevodi HTML kod u vizualno prikazane stranice.

Za izradu web-stranica koristi se kombinacija tehnologija i programskih jezika, od kojih svaka ima svoju ulogu u procesu razvoja.

3.1. Osnovni jezici

HTML (HyperText Markup Language) – jezik za označavanje sadržaja (struktura stranice: naslovi, tekst, slike, linkovi).

CSS (Cascading Style Sheets) – služi za stiliziranje i oblikovanje elemenata (boje, fontovi, raspored).

JavaScript – programski jezik za dodavanje interaktivnosti (npr. animacije, forme, dinamičko učitavanje podataka).

3.2. Baze podataka

- **Relacijske:** MySQL, PostgreSQL, MariaDB, MS SQL
- **NoSQL:** MongoDB, Firebase, Redis

3.3. Programske tehnologije i okviri

Frontend (kljentska strana)

- **React, Angular, Vue.js** → popularni JavaScript okviri za izradu modernih interaktivnih aplikacija.
- **Bootstrap, Tailwind CSS** → biblioteke za brže oblikovanje dizajna.

Frontend je dio web-stranice ili web-aplikacije koji **korisnik vidi i s kojim direktno komunicira** u svom pregledniku. Možeš ga zamisliti kao "**lice stranice**" – sve što ti se prikaže na ekranu.

Što sve spada u frontend?

1. **Struktura stranice (HTML)**

naslovi, tekstovi, slike, linkovi, forme...

2. **Izgled stranice (CSS)**

boje, fontovi, raspored elemenata, dizajn za mobitel i računalo (responsivni dizajn).

3. **Interaktivnost (JavaScript)**

padajući izbornici, galerije slika, validacija formi, animacije, dinamičko učitavanje podataka

Alati i tehnologije za frontend

1. **Osnovno:** HTML, CSS, JavaScript
2. **Biblioteke i frameworkovi:** React, Angular, Vue.js
3. **CSS alati:** Bootstrap, Tailwind CSS
4. **Dizajn alati:** Figma, Adobe XD (za planiranje izgleda stranice).

Backend (serverska strana)

- **Node.js** (JavaScript na serveru)
- **PHP** (često korišten za WordPress, Joomla, Drupal)
- **Python** (Django, Flask)
- **Ruby on Rails**
- **Java (Spring Boot)**
- **C# (.NET)**

Backend je dio web-stranice ili aplikacije koji radi na serveru i brine se za:

- obradu podataka,
- komunikaciju s bazom podataka,
- provjeru korisnika (login, registracija),
- logiku aplikacije (računanje, filtriranje, slanje e-mailova, obrada plaćanja...).

Korisnik **ne vidi direktno backend**, ali on omogućuje da sve "**iza scene**" radi kako treba.

3.4. Statičke i dinamičke web stranice

Statičke web stranice

Sadržaj je fiksiran, zapisan u HTML-u. Svaki korisnik vidi **isti sadržaj**, osim ako se stranica ručno ne promijeni.

Tehnologije: HTML, CSS, malo JavaScripta (za interaktivnost).

Primjeri korištenja:

- Osobni portfolio
- Jednostavne prezentacijske stranice
- Informacijske stranice (npr. "O nama", "Kontakt")

Prednosti:

- Brze za učitavanje
- Jednostavne za izradu i hosting

Nedostaci:

- Teško ažuriranje – svaka promjena mora se ručno pisati u kodu
- Nema personalizacije sadržaja

Dinamičke web-stranice

Sadržaj se **generira u realnom vremenu**, često ovisi o korisniku ili podacima iz baze. Stranica može izgledati drugačije za različite ljude.

Tehnologije: HTML, CSS, JavaScript + **backend jezik** (PHP, Python, Node.js, Java...) i **baza podataka** (MySQL, MongoDB...).

Primjeri korištenja:

Web shopovi (Amazon, eBay)

Društvene mreže (Facebook, Instagram)

Vijesti i blogovi (WordPress, portali)

Prednosti:

- Lakše ažuriranje (sadržaj se povlači iz baze)
- Personalizacija sadržaja za korisnika
- Puno više funkcionalnosti (registracija, komentari, pretraga...)

Nedostaci:

- Složenija izrada i održavanje
- Zahtijeva server i bazu podataka
- Potencijalno sporija (ako nije dobro napravljena)

4. Pisanje koda u HTML jeziku

4.1. Tagovi ili oznake

Kako se HTML stranice sastoje od običnog tekstualnog koda, mogu se pisati i u najjednostavnijim tekstualnim uređivačima kao što je npr. Notepad.

U HTML-u, koristimo **tag-ove ili oznake**, a to je zapravo slovo ili riječ opasani uglatim zagradama.

Većina tagovi dolaze u parovima. Prvi dio taga aktivira naredbu, a drugi dio taga je deaktivira. Razlikuje ih kosa crta na početku drugog taga. Sadržaj koji se nalazi između prvog i drugog dijela taga aktiviran je njihovom funkcijom (namjenom).

Tagovi koji dolaze samostalno mogu se pisati s kosom crtom ili bez nje.

kosa crta

Primjer 1

<p>	Sadržaj	</p>
↑		↑
prvi dio taga		drugi dio taga

Primjer 2

<h1>	Sadržaj	</h1>
↑		↑
prvi dio taga		drugi dio taga

Primjer 3

 ili
↑ ↑
tag koji nema para

Postoje osnovna pravila pisanja tagova

Tagovi ili oznake se pišu malim slovima

Krivo

`<BODY>`

`<P>This is a
paragraph</P>`

`</BODY>`

Ispravno

`<body>`

`<p>This is a paragraph</p>`

`</body>`

Tagovi ili oznake se moraju zatvoriti

Krivo

`<p>This is a paragraph`

`<p>This is another paragraph`

Ispravno

`<p>This is a paragraph</p>`

`<p>This is another paragraph</p>`

Tagovi koji nemaju para:

This is a break`<br />`

This is a break`<br>`

Here comes a horizontal rule:`<hr />`

Here comes a horizontal rule:`<hr >`

Here's an image `<img />`

Here's an image `<img>`

Tagovi ili oznake moraju biti pravilno ugniježđeni

Krivo

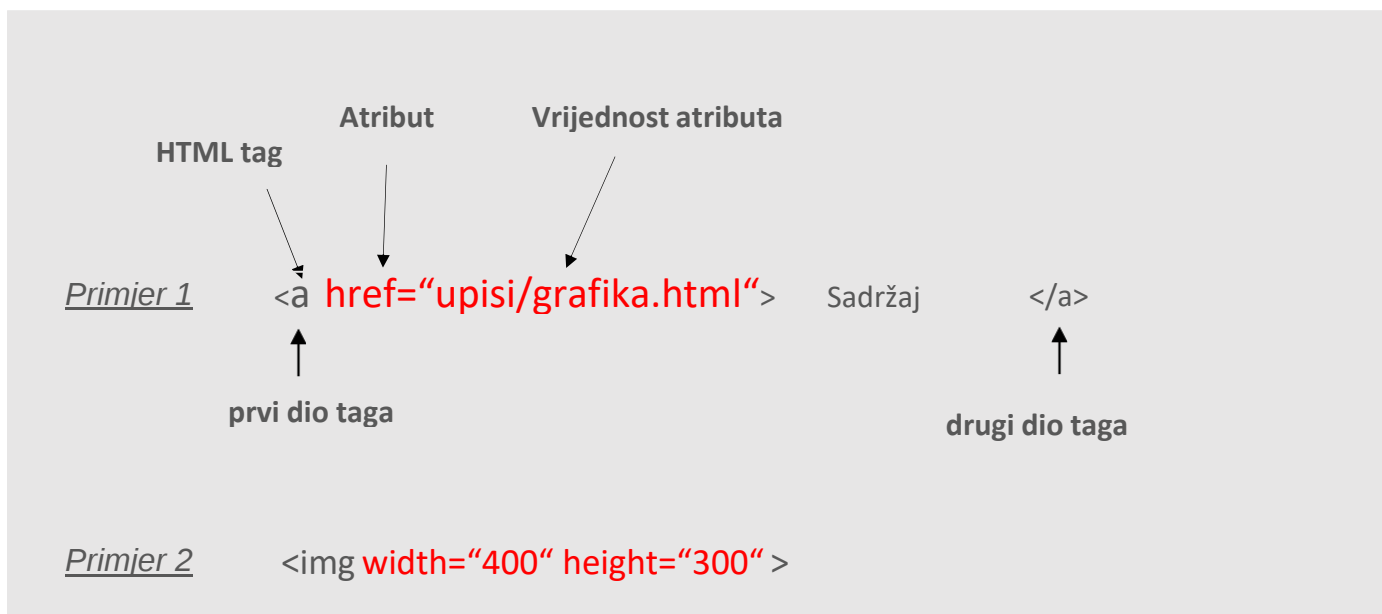
```
<p><b>This text is bold and italic</p></b>
```

Ispravno

```
<p><b>This text is bold and italic</b></p>
```

4.2. Atributi i vrijednosti tagova

Karakteristika HTML je da tagovi (oznake) imaju svoje atribute. Atributi određenog taga pišu se unutar uglatih zagrada prvog dijela taga. Atributi imaju svoje vrijednosti koje se stavljaju pod navodnike. Između vrijednosti i atributa nalazi se znak jednakosti. Tagovi mogu imati više atributa. Atributima se definira određene karakteristike sadržaja koji se nalazi unutar naredbi.



Atributi također moraju se pisati malim slovima

Krivo

```
<table WIDTH="100%">
```

Ispravno

```
<table width="100%">
```

Vrijednosti atributa moraju se stavljati pod navodnicima

Krivo

```
<table width=100%>
```

Ispravno

```
<table width="100%">
```

5.Osnovna struktura HTML jezika

5.1.Doctype najava

Prilikom stvaranja HTML dokumenta, prva linija u dokumentu trebao bi biti ovaj tag:

<!DOCTYPE ... >.

Doctype oznaka uvijek počinje s uskličnikom i uvijek je stavljen na početak dokumenta, prije bilo koje druge oznake. Doctype deklaracija **ne mora se** nužno pisati velikim slovima, ali zvanična W3C preporuka je korištenje velikih slova doctype.

Korištenje DOCTYPE oznaku je poput potpisivanja ugovora. To je tag, koji kaže koja je **tekuću verziju HTML jezika** i da će **naše kodiranje** biti u skladu sa tom verzijom HTML jezika.

Ako koristimo **HTML 4.01 jezik transitional** doctype najava izgleda

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Ako koristim **XHTML 1.0 jezik transitional** doctype najava izgleda

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Ako koristim **HTML 5 jezik** doctype najava izgleda

```
<!DOCTYPE html>
```

5.2. Osnovni tagovi u strukturi HTML dokumenta

Tri osnovna taga sačinjavaju osnovnu strukturu HTML stranice. To su:

HTML je glavni tag unutar kojeg je nalaze dva taga HEAD i BODY. On nam definira početak i kraj HTML dokumenta.

HEAD je tag unutar kojeg se stavljaju **tagovi** koji se **ne prikazuju** direktno na stranici, ali pružaju određene informacije o stranici.

BODY je tag unutar kojeg se stavljaju tagovi koji će nam **prikazati sadržaj stranice**. Svi tagovi koje stavljamo unutar BODY taga bit će prikazani u web pregledniku i utjecati će na izgled stranice u pregledniku.

Primjer - tagovi osnovne strukture web stranice HTML5

<!DOCTYPE html>

<html lang="en">

<head>

Između dolaze tagovi koji daju određene informacije o web stranici

</head>

<body>

Između dolaze tagovi koji će prikazivati sadržaj web stranice u pregledniku

</body>

</html>

Unutar taga <html> dolazi atribut **lang** npr. <html lang="en">. Atribut lang određuje jezik sadržaja elementa. Uobičajeni primjeri su "en" za engleski, "es" za španjolski, "fr" za francuski i tako dalje.

Atribut **lang** je globalni atribut i može se koristiti na bilo kojem HTML elementu.

<p lang="fr">Ceci est un paragraphe.**</p>**

5.3. Osnovni tagovi unutar <head> taga

5.3.1. Naslov stranice - <title> tag

Jedan od obaveznih tagova koji dolazi unutar <head> taga je tag <title>. Naslov stranice vidjeti ćemo na naslovnoj liniji preglednika i on se **ne izostavlja** kod izrade web stranice. Naslovom stranice daje se ime i s nekoliko riječi opisuje namjena te stranice.

Tagom title:

- definira naslov na naslovnoj liniji preglednika,
- naslov stranice se dodaje kada želimo stranicu označiti u favoritima,
- naslov stranice bitan kod tražilica (google)

```
<!DOCTYPE html>
<html>
<head>
<title> Zrno Soli Restaurant Mediterranean Cuisine by UNESCO </title>
</head>
<body>
</body>
</html>
```

5.3.2. Meta tagovi

Meta tagovi su također smješteni u head prostoru naših web stranica. To su HTML tagovi koji pružaju informacije nekom web pregledniku o sadržaju web stranice i korisnici uglavnom nisu svjesni njihove prisutnosti. Meta elementi se obično koriste za određivanje opisa stranice, ključnih riječi, autora dokumenta, zadnje izmjene i drugih metapodataka.

```
<head>
<meta charset="UTF-8">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
<meta name="author" content="John Doe">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Meta tag character-encoding

Jedan od obaveznih meta tagova koji dolazi unutar <head> taga. Govori pregledniku koji skup znakova treba koristiti za prikazivanje tekstualnih dijelova web stranice. Za hrvatske znakove u izradi web stranica koristile su se tri kodne stranice: windows-1250, ISO-8859-2 i Unicode UTF-8. Danas je najrašireniji UTF-8 raspored znakova. Zadani raspored znakova za HTML5 je UTF-8. i samim tim je riješeno i pitanje hrvatskih dijakritičkih znakova.

<meta charset="utf-8">

6.Tagovi za prikazivanje teksta i naslova

6.1. Naslovi

Naslovi na web stranicama funkcioniraju na isti način kao naslovi u tiskanim dokumentima. Oni odvajaju tekst u logičke cjeline, tj. odjeljke. Postoji šest tagova za naslove (heading), i svi su forme: `<hn>`, gdje je **n** broj od 1 do 6. Naslovi su **podebljanog pismovnog reza**, a broj 1 označava veličinom najveći naslov. Ovi tagovi dolaze u paru.

<h1> veličinom najveći naslov i u strukturi web stranice ima najveću vrijednost `</h1>`

<h6> veličinom najmanji naslov i u strukturi web stranice ima najmanju vrijednost `</h6>`

```
<body>
<h1>naslov h1</h1>
<h2>naslov h2</h2>
<h3>naslov h3</h3>
<h4>naslov h4</h4>
<h5>naslov h5</h5>
<h6>naslov h6</h6>
</body>
```

naslov	Veličine u piksela
<code><h1></code>	32
<code><h2></code>	24
<code><h3></code>	18
<code><h4><p></code>	16
<code><h5></code>	13
<code><h6></code>	9

Naslovi `<h5>` i `<h6>` zapravo manji od teksta `<p>`, ali to su samo zadane postavke i te naslove možete redefinirati tj., da se veličinom pojavljuju na način na koji mi to želimo.

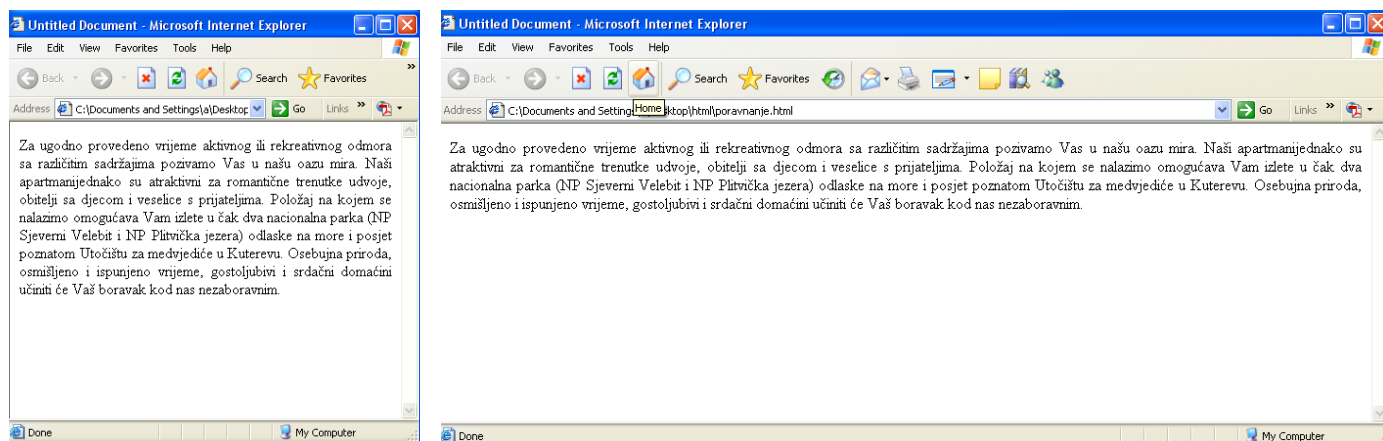
6.2. Odlomci ili paragrafi

U HTML-u je praksa da se odlomci označe s tagom `<p>` i `</p>`. Ovi tagovi dolaze u paru i web preglednik će automatski dodati razmak (bjelina od jednog retka) između dva paragrafa u tekstu.

```
<body>
<p>Uz potpunu i suvremenu opremljenost, sve naše jedinice su klimatizirane te je svaka obogaćena pripadajućom, opremljenom terasom, s kojih se pruža predivan pogled na kvarnerske otoke Cres, Plavnik i Rab te na najljepši mediteranski ortdanski prolaz - između otoka Cresa i Plavnika.</p>
<p>Ljubazni sumještani ponuditi će Vam na prodaju oveći broj lokalni gastro-specijaliteta: pršut, ovčje meso, mlijeko i sir, više vrsta riba, specifično povrće, šparoge i brokule te najraznovrsnije voće. Uz to će Vam mnogi domaćini ponuditi raznovrsno domaće, crno i bijelo vino.</p>
</body>
</html>
```

HTML ignorira prelaske u novi red (entere) realizirane u nekom od HTML editora. Ako smo sav tekstenapisali unutar jednog taga `<p>` takav tekst je „u biti“ napisan u jednom retku. U koliko će se redaka takav tekst prikazati u pregledniku ovisi o količini teksta i širini prikaza web preglednika.

Manja širina preglednika prikaz većeg broja redaka i obratno (prikaz donjih slika).



6.3. Prelazak u drugi red

Line break omogućava premještanje teksta u drugi red odmah ispod bez dodanog razmaka (praznog retka). Za prelazak u novi red koristimo `<br>` tag koji ne dolazi u paru.

U HTML5 jeziku tagovi koji dolaze samostalno mogu se pisati na dva načina

`<br>` - češće, ali i `<br />`

7. Tagovi za oblikovanje teksta

HTML nudi dosta tagova za oblikovanje teksta, a svaki od njih ima određeni vizualni ishod (kako se tekst prikazuje) i semantičko značenje (kako ga čitaju npr. pretraživači ili čitači ekrana)

7.1. Osnovno oblikovanje

Tagovi služe za naglašavanje i izgled dijela teksta unutar rečenice.

** - podebljani tekst**

`<p>`The two most popular science courses offered by the school are `<b>`chemistry `</b>` (the study of chemicals and the composition of substances) and `<b>`physics`</b></p>`.

** - podebljan ali i važan tekst**

`<p>` the most important rule, the rule you can never forget, no matter how much he cries, no matter how much he begs: `<strong>`never feed him after midnight`</strong>`. `</p>`

<i> - kurziv ili italik tekst

`<p>`The two most popular science courses offered by the school are `<i>`chemistry `</i>` (the study of chemicals and the composition of substances) and `<i>`physics`</i></p>`.

** - kurziv ili italik ali i važan tekst**

`<p>` the most important rule, the rule you can never forget, no matter how much he cries, no matter how much he begs: `<em>`never feed him after midnight`</em>`. `</p>`

<mark> - označen (highlight) obično žuta pozadina.

`<p>`Most `<mark>`salamander`</mark>`s are nocturnal, and hunt for insects, worms, and other small creatures.`</p>`

7.2.Specifična naglašavanja teksta

Tagovi služe za *posebne tipove sadržaja* (citati, kod, skraćenice...).

<small> - mali tekst

```
<p><small>The content is licensed under a Creative Commons Attribution-ShareAlike 2.5  
Generic License.</small></p>
```

 - izbrisani tekst i (prikazuje precrtano)

<ins> - umetnuti tekst (prikazuje podvučen)

```
<p>  
Cijena proizvoda je bila  
<del>50 €</del>,  
a sada je  
<ins>40 €</ins>.  
</p>
```

<sub> - tekst umanjen i ispod pismovne linije

```
<p>Almost every developer's favorite molecule is  
C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>, also known as "caffeine."  
</p>
```

<sup> - tekst umanjen i iznad pismovne linije

```
<p>The Pythagorean theorem is often expressed as the following equation:</p>  
<p>a<sup>2</sup> + b<sup>2</sup> = c<sup>2</sup></p>
```

<abbr> - kada mišom pređemo preko RH pojavi se tooltip (Republika Hrvatska)

```
<p>  
<abbr title="Republika Hrvatska">RH</abbr> je članica Europske unije od 2013. godine.  
</p>
```

<blockquote> - semantičke elemente za citiranje dužeg teksta u bloku.

<p>Albert Einstein je rekao:</p>

<blockquote>

Mašta je važnija od znanja. Znanje je ograničeno, dok mašta obuhvaća cijeli svijet, potiče napredak i rađa evoluciju.

</blockquote>

<q> - semantičke elemente za kratka citiranje u liniji.

<p>

Albert Einstein je rekao: <q>Mašta je važnija od znanja.</q>

</p>

<pre> - formatiran tekst čuva razmake i prijelome redaka i koristi monospace font

<pre>

Ovo je tekst unutar pre taga.

Čuva razmake i prijelome redova.

Koristi monospace font.

</pre>

<hr> - nema para i umeće vodoravnu liniju (razdjelnik)

<p>Ovo je običan odlomak. </p>

<hr>

<code> - Označava dijelove programskog koda.

<p>Kod: <code>console.log("Pozdrav svijete!"); </code> </p>

8.Prikaz specijalnih simbola u HTML-u

Specijalne simbole možemo ubaciti u svoje HTML na nekoliko načina:

1.) Pomoću slovnih znakova (Character entity reference)

Prije slovne oznake određenog simbola potrebno je upisati znak & zatim tekstualnu oznaku i na kraju ;

Na primjer **©** u HTML-u dati će nam u web pregledniku simbol ©. Često korišteni HTML entitet je non-breaking space: ** **; razmak koji se neće prelomiti u novi red.

2.) Korištenjem brojčane oznake (Numeric character reference)

Prije broja potrebno je upisati znakove **&#** zatim broj, a na kraju ;

Na primjer **©** dati će nam © odnosno simbol za copyright

Prikaz numeričkih i slovnih oznaka na stranici https://www.html.am/reference/html-special-characters.cfm?utm_source=chatgpt.com

Ako znak ima posebno značenje u HTML-u (<, >, &) → **uvijek koristiti character entity**.

Ako znak nije standardni na tipkovnici (€, ♥, ★) → **može jedno i drugo**.

9. Liste

Liste su veoma korisne jer pregledno prikazuju informacije .Razlikujemo

- Uređene ili numerirane liste
- Neuređene ili nenumerirane liste
- Liste definicija

9.1. Numerirane ili uređene liste

Numerirane liste koriste se kad želite prikazati sadržaj koji ima abecedni ili numerički redoslijed. Tag za uređene liste je **** koji dolazi u paru ****. Tekst između taga **...** smatra se djelom liste, alementi liste nalaze se unutar ** ... ** tagova

```
<ol>
  <li>Ime datoteke (Vanishing Point)</li>
  <li>Format datoteke (psd)</li>
  <li>Trenutna veličina prikaza (66,7%) </li>
  <li>Na kojem smo nivou(leyeru) slike (Background) </li>
  <li>Modeli boja (RGB) </li>
</ol>
```

Tag **** ima dva atributa – **type** i **start**.

Atribut	Vrijednost	Opis
TYPE	1, I, i, A ili a.	To predstavlja tip uređivanja koji želite: brojevi, rimski brojevi (veliki ili mali) i slova. TYPE je po defaultu namješten na brojeve.
START	BROJ	U sklopu ovog atributa potrebno je navesti broj kojim želimo započeti listu. Ako želimo listu započeti sa 6 ili sa f, potrebno je namjestiti start="6".

Tag **** ima atribut **value**.

Atribut	Vrijednost	Opis
VALUE	BROJ	VALUE definira vrijednost određene stavke. Ako želimo bilo koju stavku započeti sa npr. 10 ili sa slovom j, potrebno je namjestiti value="10". Sljedeća stavka ima vrijednost 11 ili k.

```
<ol type="I" start="5" >
  <li>Ime datoteke (Vanishing Point) </li>
  <li>Format datoteke (psd) </li>
  <li value="12">Trenutna veličina prikaza (66,7%) </li>
  <li>Na kojem smo nivou(leyeru) slike (Background) </li>
  <li>Modeli boja (RGB) </li>
</ol>
```

9.2. Nenumerirane liste ili neuređene liste

nenumerirane liste koriste se kad želimo prikazati sadržaj koji nema bitan abecedni ili numerički redoslijed, već grafičke simbole. Ako želite kreirati neuređenu listu koristimo tag **...**. Stavke (sadržaj) u listi također dodavaju se upotrebom **** taga.

```
<ul>
  <li> Ime datoteke (Vanishing Point) </li>
  <li> Format datoteke (psd) </li>
  <li> Trenutna veličina prikaza (66,7%) </li>
  <li> Na kojem smo nivou(leyeru) slike (Background) </li>
  <li> Modeli boja (RGB) </li>
</ul>
```

Tag **** imao je atribut **type** kojim smo mogli mijenjati izgled simbola koji označavaju listu, ali se ovaj atribut **ne može koristiti** u HTML5 verziji jezika.

9.3. Ugniježdene liste

HTML dozvoljava da se stavi lista u listu. Takva lista se zove ugniježdена. Unutar elementi liste koji su okruženi `<li> ... </li>` tagovima stavljamo novu listu.

Postoje četiri kombinacije:

- Numerirana lista nalazi se unutar numerirane liste
- Numerirana lista nalazi se unutar nenumerirane liste
- Nenumerirana lista nalazi se unutar numerirane liste
- Nenumerirana lista nalazi se unutar nenumerirane liste

```
<ol>
  <li>Brojevi</li>
  <li>Uvod u programiranje
    <ul>
      <li> C </li>
      <li> C++ </li>
      <li> Java </li>
      <li> VB </li>
      <li> Perl </li>
    </ul>
  </li>
  <li> Diferencijalne jednađbe </li>
  <li> Analiticka geometrija </li>
  <li> Statistika </li>
</ol>
```

9.4. Lista definicija

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink.....</dd>
  <dt>Milk</dt>
  <dd>White cold drink.....</dd>
</dl>
```

Tag **<dl>...</dl>** definira listu definicija. Tagovi **<dt>...</dt>** i **<dd>...</d.d.>** nalaze se unutar taga **<dl>**.

Tag **<dt>** **definira** određeni pojam / ime.

Tag **<dd>** **opisuje** određeni pojam / ime.

10. Slike

Web stranice bi izgledale prilično dosadne da nema slika na njima. Slike su te koje Webu daju vizualni izgled. Slike za Web trebaju biti male, a pod pojmom male ne misli se na dimenzije, već na veličinu datoteke koja zauzima prostor na disku i koja se učitava prilikom prijenosa podataka.

Kada spremamo sliku za Web pred nama se postavljaju dva cilja i to:

- što manja veličina datoteke (što manje kb)
- što kvalitetnija fotografija

Veličina datoteke je bitna zbog brzine prijenosa podataka. Što je manja datoteka, brže će nam se podaci učitati, i obratno. Veličinu datoteke najčešće mjerimo u kb (kilobajtima) i Mb (megabajtima)

10.1.Formati slika za Web

JPEG (Graphics Interchange Format)- format koji podržava 16 milijuna boja, ne podržava prozirnost, komprimira veličinu datoteke (smanjuje je), ali žrtvuje kvalitetu (podatke) sa slike. Koristi se za spremanje kolor slika zbog toga što podržava 16 milijuna boja.

- Ekstenzija: .jpg ili .jpeg
- Najčešći format za **fotografije i realistične slike**.
- Podržava milijune boja.
- Koristi kompresiju s gubicima → **manja veličina datoteke, ali slabija kvaliteta**.

PNG koji podržava 16 milijuna boja. PNG formati podržavaju prozirnost, komprimiraju veličinu datoteke (smanjuje je), ali ne žrtvuje kvalitetu (podatke) sa slike. Nekad se koristio za slike vrlo malih dimenzija jer fine detalje unakazi JPEG kompresija s gubitkom. Međutim, povećanjem brzine prijenosa podataka ovaj format postaje sve prihvatljiviji jer želimo što kvalitetnu fotografiju.

- Ekstenzija: .png
- Koristi se za manje fotografije, ikone, logotipe.
- Podržava transparentnu pozadinu.
- Podržava milijune boja
- Bez gubitaka u **kvaliteti slike (ali veća datoteka nego JPG)**.

GIF– format koji podržava 256 boja, podržava prozirnost, komprimira veličinu datoteke (smanjuje je), ali ne žrtvuje kvalitetu (podatke) sa slike. Koristi se za slike kod koje nisu kolor fotografije, logotipe (zaštitni znak firme)

- Ekstenzija: .gif
- Ograničen na 256 boja.
- Podržava animacije i transparentnost.
- Koristi se za jednostavne animacije ili male ikone.

WEBP je format datoteke koji je razvio Google, a namijenjen je kao zamjena za JPEG, PNG i GIF formate datoteka.

- Ekstenzija: .webp
- Kompresija: može biti s gubicima (kao JPG) ili bez gubitaka (kao PNG).
- Veličina: datoteke su **manje** od JPG/PNG → brže učitavanje stranica.
- Kvaliteta: **bolja od** JPG-a pri istoj veličini.
- Podrška: transparentnost (kao PNG) i animacije (kao GIF).

SVG– grafički format napravljen za web, koji se koristi za dodavanje vektorskih slika stranicama. W3C je uveo SVG format kao standard 2001. godine. Glavna prednost je što se može beskrajno uvećavati, a da se ne izgubi kvaliteti. To čini ovaj format veoma pogodnim za responsive web dizajn.

- Ekstenzija: .svg
- Vektorski format – slika se ne gubi u kvaliteti pri povećavanju.
- Odličan za **logotipe, ikone i ilustracije**.

10.2.Tag i atributi za prikazivanje slika

Tag koja se koristi za prikazivanje slika na web stranicu je ****. Tag ne dolazi u paru. I možemo ga pisati i ****

10.2.1.Atribut src

Kratika od source (izvor). Koristi se da odredi putanju (URL adresa) do vanjske datoteke koju HTML dokument treba ugraditi-učitati (slika)

```
<img src= "umjetnost/slika.jpg" />
```

↗ - URL adresa - putanja do datoteke

. Jednostavno rečeno: **URL** (Uniform Resource Locator) **adresa** resursa na internetu (web-stranice, **slike**, videa, datoteke), s danom putanjom koja opisuje kako doći do tog dokumenta ili datoteke.

Vrste putanja

Relativni URL – u kojem su međusobnom položaju na web site-u web stranica i slika koja se poziva u web stranicu.

```
<img src= "slika.jpg">
```

Prednosti relativnih putanja

- Funkcioniraju bez obzira na to gdje se projekt prebaci (npr. lokalno  server).
- Kraće i preglednije od apsolutnih URL-ova.

Ukratko: **relativne putanje su „lokalne adrese“ datoteka u odnosu na HTML dokument.**

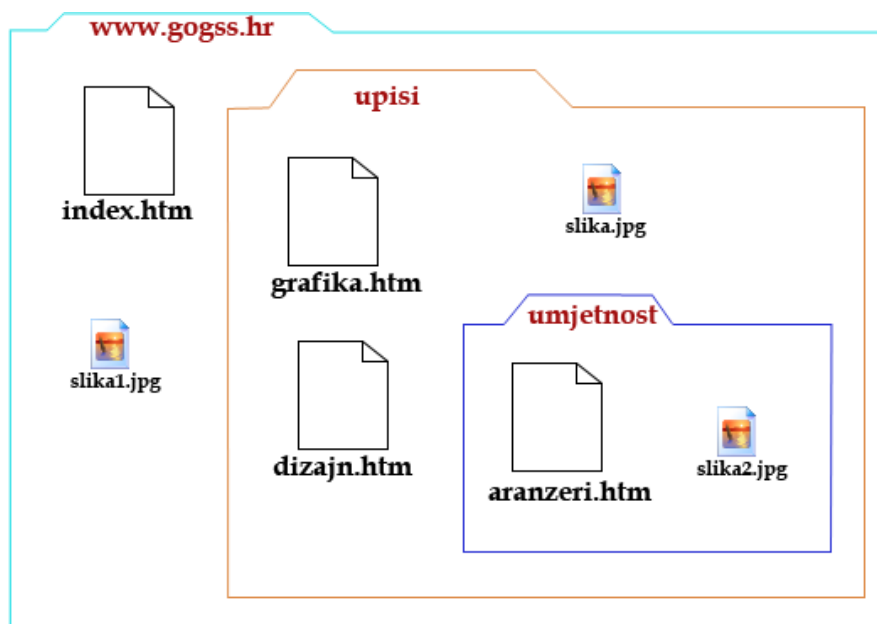
Apsolutni URL – je puna adresa do resursa na internetu. Sadrži sve dijelove URL-a: protokol, domen i putanju do datoteke. Koristi se za kada želimo točno odrediti adresu datoteke na internetu (npr. vanjski resursi - slike koje su na internetu izvan našeg web sita-a.)

```

```

Prednosti apsolutnih putanja

- adresa uvijek vodi na istu sliku, bez obzira na to gdje se HTML dokument nalazi
- ne ovisi o strukturi direktorija.
- potrebno kada se ugrađuju slike s drugih web stranice.



Primjeri: 1

Ako u web stranicu **index.html** želimo ugraditi sliku pod imenom **slika1.jpg** koja se nalazi u istoj mapi

apsolutni URL

relativni URL

Primjeri: 2

Ako u web stranicu **grafika.html** želimo ugraditi sliku pod imenom **slika2.jpg** koja se nalazi u mapi umjetnost

apsolutni URL

relativni URL

Primjeri: 3

Ako u web stranicu **index.html** želimo ugraditi sliku pod imenom **slika2.jpg** koja se nalazi u mapi umjetnost

apsolutni URL

relativni URL

Primjeri: 4

Ako u web stranicu **aranzeri.html** želimo ugraditi sliku pod imenom **slika.jpg** koja se nalazi u mapi upisi

apsolutni URL

relativni URL

Primjeri: 5

Ako u web stranicu **aranzeri.html** želimo ugraditi sliku pod imenom **slika1.jpg** koja se nalazi u root-u

apsolutni URL

relativni URL

10.2.2. Atribut alt

Alternativni tekst. Dodaje se u tag i opisuje sadržaj slike.

- tekst u alt atributu prikazuje se umjesto slike ako se slika ne može učitati.
- također ga koriste čitači ekrana pomaže osobama s oštećenim vidom (pristupačnost).
- za optimizaciju– Google koristi alt tekst kako bi razumio o čemu je slika.

Primjeri lošeg alt atributa

, ,

Primjeri dobrog alt atributa

10.2.3. Atribut title

Tekst koji se prikaže kada korisnik prijede mišem preko slike.

10.2.3. Atribut loading

Način učitavanja slike (HTML5).

- lazy → učitava se tek kad je korisnik vidi (štedi vrijeme učitavanja web stranice).
- eager → učitava se odmah.
- Auto - preglednik sam odlučuje kada će sliku učitati (zadano)

10.2.3. Atribut decoding

Kontrola dekodiranja slike, koji pomaže pregledniku da odluči **kako i kada dekodirati sliku** radi bržeg prikaza stranice. Kad preglednik učitava sliku, mora je **dekodirati** (pretvoriti iz formata poput JPG, PNG, WebP u prikazivu bitmapu).

sync (sinkrono)

- slika se dekodira odmah.
- može blokirati ostatak sadržaja dok se ne prikaže slika.
- koristi se kada je slika kritična za izgled stranice (npr. logo u zaglavlju).

async (asinkrono)

- slika se dekodira u pozadini, dok preglednik nastavlja prikazivati ostatak stranice.
- Korisno za slike koje nisu odmah potrebne (npr. u članku niže na stranici).

auto (zadano)

Preglednik sam odlučuje kako će dekodirati sliku (najčešće koristi kombinaciju sync + async ovisno o važnosti slike).

Zašto je decoding koristan?

Pomaže u optimizaciji performansi stranice.

Sprječava "zastajkivanje" pri učitavanju.

Brže učitavanje vidljivog sadržaja).

```

```

- **src, alt** – obavezni i osnovni.
- **width, height, title** – često korišteni za izgled i dodatne informacije. Width i height koristimo u CSS-u
- **loading, decoding** – moderni atributi za performanse.

11. Linkovi (HYPERLINK)

HYPERLINK (često se kaže i link) je element web stranice na kojeg kad se klikne, pristupa drugoj web stranici, ili generalno, drugoj web adresi. Linkovi su često u formi teksta, ali su popularni i image(slike) linkovi.

11.1. Tag i atributi linkova

Da bi kreirali link, potreban nam je otvoreni i zatvoreni `<a>...</a>` tag. Sav sadržaj između otvorenog i zatvorenog taga smatra se linkom.

U primjeru,

```
<a>YAHOO!</a>
```

11.1.1. Atribut src

Osnovni, **obavezni atribut** koji definira odredište linka (URL adresa - putanja ili sidro na istoj stranici).

Također možemo koristiti **relativnu adresu**- pišemo samo putanju unutar projekta.

```
<a href="kontakt.html">Kontakt</a>
```

ili **apsolutna adresa** koja sadrži cijeli URL (protokol, domen i poddirektorije) do stranice.

```
<a href="https://www.example.com/clanci/ucenje.html">Učenje HTML-a</a>
```

11.1.2 Atribut target

gdje će se link otvoriti:

- `_self` → u istom prozoru (zadano)
- `_blank` → u novom tabu/prozoru
- `_parent` → u roditeljskom okviru
- `_top` → u cijelom prozoru
-

```
<a href="https://openai.com" target="_blank">Open u novom tabu</a>
```

11.1.3. Atribut title

Dodatna informacija, prikazuje se kad korisnik zadrži pokazivač iznad linka.

```
<a href="https://skola.hr" title="Službena stranica škole">Škola</a>
```

11.1.4. Atributi rel

za sigurnost i ponašanje opisuje odnos linka i stranice.

- **nofollow** → kaže tražilicama da ne prate link
- **noopener** → sigurnost kada koristimo **target="_blank"**. Sprječava odredišnu stranicu da manipulira prozorom iz kojeg je otvorena. Sigurnosna zaštita.
- **noreferrer** → sprječava slanje "referrer" informacija (odakle korisnik dolazi).
-

```
<a href="https://nepoznato.com" target="_blank" rel="noopener noreferrer">Nepoznata  
stranica</a>
```

11.1.5. Atributi za preuzimanje i za resurse

download – umjesto otvaranja linka, preuzima datoteku.

```
<a href="dokumenti/uputstvo.pdf" download>Preuzmi PDF</a>
```

hreflang – označava jezik stranice na koju vodi link.

```
<a href="https://example.fr" hreflang="fr">Francuska verzija</a>
```

type – MIME tip resursa. je oznaka koja opisuje vrstu sadržaja (resursa) koji se prenosi preko interneta, tj. oznaka koja govori pregledniku o kojoj se vrsti datoteke radi i kako je treba prikazati ili obraditi.

```
<a href="video.mp4" type="video/mp4">Pogledaj video</a>
```

- **href** → adresa linka (obavezno)
- **target** → kontrola otvaranja (novi tab/prozor)
- **rel** → sigurnost i SEO
- **title** → dodatna informacija
- **download** → preuzimanje datoteke

11.1. linkovi za e-mail adrese

```
<a href="mailto:support@adatum.com">Contact Us</a>
```

12. Tablice

Tablice su namijenjene prikazivanju strukturiranih podataka. (npr. cijene, rasporedi, rezultati).

Osnovni tagovi:

<table> — početak i kraj tablice

<caption> — naslov/tablice (opis, prikazan korisnicima i koristan za pristupačnost).

<thead> — zaglavlje tablice (redovi <tr> s <th> ćelijama).

<tbody> — glavno tijelo tablice (redovi sa <td>).

<tfoot> — podnožje (npr. zbrojevi, napomene).

<tr> — red (table row).

<th> — zaglavna ćelija— po zadanom semantički označava naslov stupca/retka i obično je tekst prikazan podebljano i centrirano.

<td> — podatkovna ćelija

<colgroup> i **<col>** — definiraju grupu stupaca i omogućuju primjenu CSS stilova na stupce (rjeđe se koriste, ali su korisni za definiranje izgleda tablice).

```

<table>
<caption>Prodaja po kvartalu, 2025.</caption>
<thead>
  <tr>
    <th scope="col">Kvartal</th>
    <th scope="col">Prihod</th>
    <th scope="col">Trošak</th>
    <th scope="col">Profit</th>
  </tr>
</thead>

<tbody>
  <tr>
    <th scope="row">Q1</th>
    <td>€12,000</td>
    <td>€7,500</td>
    <td>€4,500</td>
  </tr>
  <tr>
    <th scope="row">Q2</th>
    <td>€14,000</td>
    <td>€8,000</td>
    <td>€6,000</td>
  </tr>
</tbody>

<tfoot>
  <tr>
    <td colspan="3"> Ukupno:</td>
    <td>€10,500</td>
  </tr>
</tfoot>
</table>

```

12.1 Atribut scope taga <th>

Atribut **scope** se postavlja unutar **<th>** elemenata kako bi čitač ekrana (za slijepe osobe) i preglednici jasnije razumjeli na koje se ćelije **zaglavlje** odnosi.

```
<table>
  <caption>Raspored sati</caption>
  <thead>
    <tr>
      <th scope="col">Dan</th>
      <th scope="col">Predmet</th>
      <th scope="col">Učionica</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <th scope="row">Ponedjeljak</th>
      <td>Matematika</td>
      <td>12A</td>
    </tr>

    <tr>
      <th scope="row">Utorak</th>
      <td>Fizika</td>
      <td>10B</td>
    </tr>
  </tbody>
</table>
```

Atribut	Vrijednost	Opis
scope	col	Određuje da je ćelija zaglavlje za stupac
scope	row	Određuje da je ćelija zaglavlje za red

12.1 Atribut taga <table>

U HTML5 jeziku svi atribut koji se koriste unutar taga table su zastarjeli i koristi se CSS za oblikovanje tablica..

Neki od atributa koji su se koristili u prethodnim verzijama HTML jezika su:

border — podešava širinu obruba tablice. (numerička vrijednost atributa)

cellspacing — razmak između ćelija. (numerička vrijednost atributa)

cellpadding — unutarnji razmak unutar ćelija (numerička vrijednost atributa)

```
<table border="1" cellpadding="8" cellspacing="4">

  <caption>Raspored sati</caption>
  <thead>
    <tr>
      <th scope="col">Dan</th>
      <th scope="col">Predmet</th>
      <th scope="col">Učionica</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <th scope="row">Ponedjeljak</th>
      <td>Matematika</td>
      <td>12A</td>
    </tr>

    <tr>
      <th scope="row">Utorak</th>
      <td>Fizika</td>
      <td>10B</td>
    </tr>
  </tbody>
</table>
```

12.2. Spajanje ćelija

a) colspan

spaja više ćelija **stupaca** u jednu ćeliju. Vrijednost ovog atributa je brojčana oznaka koja kaže koliko je stupaca spojeno. Koristi se u tagovima **<td>** ili **<th>**:

```
<td colspan="2">Ova ćelija se prostire preko 2 stupca</td>
```

Dakle, colspan = horizontalno spajanje ćelija.

b) rowspan

spaja više ćelija **redaka** u jednu ćeliju. Vrijednost ovog atributa je brojčana oznaka koja kaže koliko je stupaca spojeno. Koristi se u tagovima **<td>** ili **<th>**:

```
<td rowspan="2">Ova ćelija se prostire preko 2 reda</td>
```

Dakle, rowspan = vertikalno spajanje ćelija.

td rowspan="2"		
	td colspan="2"	

```
<table border="1" cellpadding="8" cellspacing="0">
```

```
<tr>
```

```
<th rowspan="2">Dan</th>
```

```
<th colspan="3">Predmeti</th>
```

```
</tr>
```

```
<tr>
```

```
<th>1. sat</th>
```

```
<th>2. sat</th>
```

```
<th>3. sat</th>
```

```
</tr>
```

```
<tr>
```

```
<td>Ponedjeljak</td>
```

```
<td>Matematika</td>
```

```
<td rowspan="2">Hrvatski</td>
```

```
<td>Informatika</td>
```

```
</tr>
```

```
<tr>
```

```
<td>Utorak</td>
```

```
<td>Povijest</td>
```

```
<td>Engleski</td>
```

```
</tr>
```

```
</table>
```

13. Forme

HTML forma (**<form>...</form>**) je dio web-stranice koji omogućuje unos podataka od korisnika i slanje tih podataka serveru (backendu) na obradu.

Forma dopušta da se uspostavi komunikacija između **korisnika** i **servera**: korisnik popunjava formu i šalje je serveru. Polazeći od podataka iz forme vrši se odgovarajuća obrada na serveru, a o rezultatima korisnik eventualno biva obaviješten.

Preko formi mogu se realizirati različite aplikacije kao što su:

- upiti nad bazama podataka da bismo pronašli npr. odgovarajuću stranicu (Google);
- smještaj korisničkih podataka u bazu;
- elektronska trgovina (prijem narudžbine i reguliranje načina plaćanja);

13.1. Tag **<form>** i atributi

Forma se implementira preko taga **<form>** čiji je opći oblik:

<form> dolaze elementi jedne forme **</form>**

```
<form id="prva_forma" method="post" action="obrada.php">  
.....  
</form>
```

Tag **<form>** sadrži sljedeće atribute:

Atribut **action** gdje poslati ispunjene podatke obrasca prilikom predaje (klikom na dugme submit). Na koju adresu (URL)

Atribut **method** označava na koji način će se poslati podaci obrasca. Ovaj atribut može imati vrijednosti GET (prikazuje podatke forme u URL-u) ili POST. (skriva podatke forme i sigurniji je)

Atribut **id** koristi se za jedinstveno označavanje HTML elementa radi povezivanja, stiliziranja ili manipulacije.. To znači da element koji ima određeni id na stranici može postojati samo jednom. Ako imamo više formi na html stranici, dati ćemo različita imena formi.

Unutar taga **<form>...</form>** mogu se navesti i drugi tagovi koji definiraju elemente forme. Tagovi **<form>** se ne mogu umetati jedan u drugi.

13.2.Dugme submit

Tag i obavezni atribut dugmeta submit je **<input type="submit">** - klikom na ovaj element šaljemmo vrijednosti (podatke) elemenata forme

```
<input type="submit" id="submit" value="Potvrditi">
```

- atribut **type="submit"** element obrasca - dugme za slanje podataka
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **value** predstavlja tekst koji će se prikazati na dugmetu

Možemo koristiti i tag **<button>**

```
<button type="submit" id="submit" >Potvrditi</button>
```

13.3.Dugme reset

Tag i obavezni atribut dugmeta reset je **<input type="reset">**- klikom na element resetiramo definirane (odabrane) vrijednosti elemenata forme i svi se unosi vraćaju na početne vrijednosti

```
<input type="reset" id="reset" value="Poništi">
```

- atribut **type="reset"** govori pregledniku da je element obrasca dugme reset
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **value** predstavlja tekst koji će se prikazati na dugmetu

Možemo koristiti i tag **<button>**

```
<button type="reset" id="reset" >Potvrditi</button>
```

13.4. Jednoredno polje za unos teksta

Naredba i obavezni atribut jednorednog polja za unos teksta je **<input type="text" >**

(u ovom slučaju tag lebel ima for koji ga veže za tag input putem istog id – klik na njega fokusira polje input)

```
<label for="ime">Ime:</label>
<input type="text" id="ime" name="ime" placeholder=" vaše ime upišite"
autocomplet="on" required>
```

ili

(u ovom slučaju lebel nema for jer se unutar taga label ugnijezdio tag input)

```
<label >Ime:
<input type="text" id="ime" name="ime" placeholder=" vaše ime upišite"
autocomplet="on" required>
</label>
```

- atribut **type="tekst"** govori pregledniku da je element obrasca jednoredno polje za unos teksta
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s upisanim podatkom u jednorednom polju.
- atribut **placeholder** navodi nas što se traži od nas da upišemo u zadano polje
- atribut **autocomplet** pomaže korisnicima da ispunjavaju obrasce na temelju ranijeg unosa. Vrijednosti su on ili off
- **required** određuje da se prije slanja obrasca mora popuniti polje za unos
- atribut **size** koristi se za definiranje širine polja za unos teksta (ne ograničava broj znakova unosa)
- atribut **minlength** koristi se za definiranje minimalnog broja znakova koje korisnik može upisati u polje za unos. Na primjer kod OIB minlength iznositi će 11.
- atribut **maxlength** koristi se za definiranje maksimalnog broja znakova koje korisnik može upisati u polje za unos. Na primjer kod OIB maxlength iznositi će 11.
- **tabindex** određuje redoslijed fokusa kada se pritisne tab na tipkovnici. Manji brojevi dobivaju fokus prvo, a veći kasnije

13.5. Jednoredno polje za unos lozinke

Naredba i obavezni atribut jednorednog polja za unos teksta je `<input type="password" >`

```
<label for="lozinka">Lozinka:</label>
<input type="password" id="lozinka" name="lozinka" placeholder=" vaša lozinka
required>

<label>Lozinka:
<input type="password" id="lozinka" name="lozinka" placeholder=" vaša lozinka
required>
</label>
```

- atribut **type="password"** govori pregledniku da je element obrasca jednoredno polje za unos lozinke
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s upisanim podatkom u jednorednom polju.
- atribut **placeholder** navodi nas što se traži od nas da upišemo u zadano polje atribut
- **autocomplete="off"** onemogućava spremanje lozinke, **autocomplete="new-password"** koristi se za registraciju, **autocomplete="current-password"** koristi se za login
- **required** određuje da se prije slanja obrasca mora popuniti polje za unos
- atribut **minlength** koristi se za definiranje minimalnog broja znakova koje korisnik može upisati u polje za unos.
- atribut **maxlength** koristi se za definiranje maksimalnog broja znakova koje korisnik može upisati u polje za unos.
- **pattern** validiranje pomoću kojeg definiramo pravila unosa (koliko znakova, koliko velikih slova, ... (npr `pattern="(?!.*[A-Z]).{8,}"`)
- **tabindex** određuje redoslijed fokusa kada se pritisne tab na tipkovnici. Manji brojevi dobivaju fokus prvo, a veći kasnije

13.6. Jednoredno polje za unos email-a

Naredba i obavezni atribut jednorednog polja za unos teksta je `<input type="email">`

```
<label >Email:
<input type="email" id="email" name="email" placeholder=" vaša email adresa
required>
</label>

<label for="email">Lozinka:</label>
<input type="email" id="email" name="email" placeholder=" vaša email adresa
required>
```

- atribut **type="email"** govori pregledniku da je element obrasca jednoredno polje za unos email-a
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s upisanim podatkom u jednorednom polju.
- atribut **autocomplete** pomaže korisnicima da ispunjavaju obrasce na temelju ranijeg unosa. Vrijednosti su on ili off ili autocomplete="email" za login, kontakt, registracija
- **required** određuje da se prije slanja obrasca mora popuniti polje za unos
- atribut **minlength** koristi se za definiranje minimalnog broja znakova koje korisnik može upisati u polje za unos.
- atribut **maxlength** koristi se za definiranje maksimalnog broja znakova koje korisnik može upisati u polje za unos.
- **pattern** validiranje pomoću kojeg definiramo pravila unosa
- **tabindex** određuje redoslijed fokusa kada se pritisne tab na tipkovnici. Manji brojevi dobivaju fokus prvo, a veći kasnije

13.7. Radio dugme

Tag i obavezni atribut polja za potvrdu je **<input type="radio">**

```
<input type="radio" name="spol" id="musko" value="musko" >
<label for="musko">Muško</label>
<input type="radio" name="spol" id="zensko" value="zensko" checked>
<label for="zensko">Žensko</label>
```

- atribut **type="radio"** govori pregledniku da je element obrasca radio dugme
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s odabranim podatkom u jednom radio dugmetu.
- atribut **value** predstavlja izabranu vrijednost
- **checked** govori pregledniku da je u postavkama izaberano to dugme

13.8. Polje za potvrdu

Tag i obavezni atribut polja za potvrdu je **<input type="checkbox">**

```
<input type="checkbox" name="operativni_sustav" id="windows" value="windows">
<label for="windows"> Windows </label>
<input type="checkbox" name="operativni_sustav" id="vista" value="vista">
<label for="windows"> Vista </label>
<input type="checkbox" name="operativni_sustav" id="linux" value="linux"
checked="checked">
<label for="windows"> Linux</label>
```

- atribut **type="checkbox"** govori pregledniku da je element obrasca radio dugme
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati različito ime..
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s odabranim podatkom ili podacima u polju. checkbox-a

- atribut **value** predstavlja izabranu vrijednost
- **checked** govori pregledniku da je u postavkama izaberano to dugme

13.9. Višeredno polje unos teksta

Tag višerednog polja za unos teksta je `<textarea> </textarea>`

```
<textarea name="poruka" id="poruka" cols="40" rows="5" placeholder="komentar"
required ></textarea>
```

- atribut **id** koristi se za dodjelu imena polju. Svaki id u kodu mora imati različito ime.
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s upisanim tekstom u polju..
- atribut **cols** koristi se za definiranje broja znakova koji se mogu unijeti u jedan red
- atribut **rows** koristi se za definiranje broja redova.
- atribut **placeholder** navodi nas što se traži od nas da upišemo u zadano polje
- **required** određuje da se prije slanja obrasca mora popuniti polje za unos

13.10. Padajuće liste

Tag padajuće liste je `<select>... </select>`. Da bismo dodali stavke (opcije) u padajućoj listi između taga `<select>... </select>` dodajemo tag `<option>... </option>`

```
<label for="godine">Dob </label>
<select name="godine" id="godine" size="1">
  <option value="manje20" selected>manje od 20 </option>
  <option value="20do29">20 do 29 </option>
  <option value="30do39">30 do 39 </option>
  <option value="40do49">40 do 49 </option>
  <option value="više50">50 i više </option>
</select>
```

Atributi tag <select>

- atribut **size** koliko je vidljivih opcija prikazano u padajućem izborniku bez otvaranja liste. Ako je size="1" rezultat je isti kao da nema size (klasični padajući izbornik).
- atribut **id** koristi se za dodjelu proizvoljnog imena elementu. Svaki id u kodu mora imati
- *različito ime..*
- atribut **name** preglednik serveru prosljeđuje vrijednost atributa name zajedno s odabranim podatkom u padajućoj listi..

Atributi tag <option>

- atribut **value** predstavlja izabranu vrijednost
- atribut **selected** govori pregledniku da prikaže tu vrijednost kao prvi izbor

14. HTML blok i linijski (inline) elementi

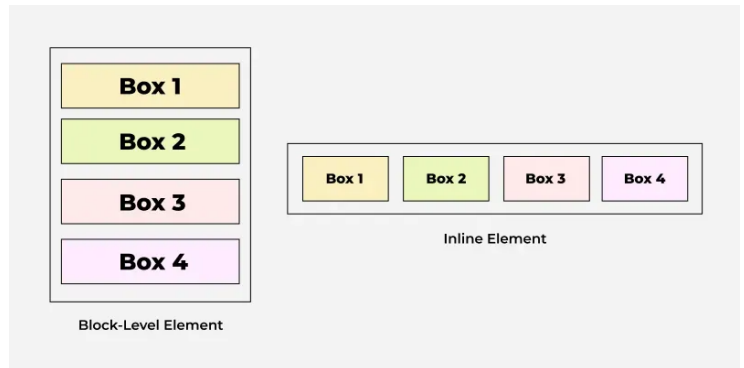
Svaki HTML element ima zadanu vrijednost za prikaz, ovisno o vrsti elementa.

Dvije vrijednosti prikaza su

- blok i
- inline (u liniji).

HTML elementi koji su na **razini bloka**, strukturiraju izgled i protežu se **punom širinom** (100%), a preglednici automatski dodaju malo prostora (marginu) prije i poslije elementa.

HTML elementi u liniji stiliziraju sadržaj bez prekidanja toka (ne počinje u novom retku), a širina ovisi o širini sadržaja koji je sadržan u njima.



HTML elementi bloka

<address>	<article>	<aside>	<blockquote>	<canvas>	<dd>
<div>	<dl>	<dt>	<fieldset>	<figcaption>	<figure>
<footer>	<form>	<h1>-<h6>	<header>	<hr>	
<main>	<nav>	<noscript>		<p>	<pre>
<section>	<table>	<tfoot>		<video>	

HTML elementi linije

<a>	<abbr>	<acronym>		<bdo>	<big>
 	<button>	<cite>	<code>	<dfn>	
<i>		<input>	<kbd>	<label>	<map>
<object>	<output>	<q>	<samp>	<script>	<select>
<small>			<sub>	<sup>	<textarea>
<time>	<tt>	<var>			

`<p>Hello World</p>`
`<div>Hello World</div>`

Element `<p>` je element na razini bloka.

Element `<div>` je element na razini bloka.

`<p>`Ovo je inline span element`<span>Hello World</span>` unutar odlomka`</p>`

`<p>`Element SPAN je inline element i neće počinjati u novom retku te zauzima samo onoliko širine sadržajem koji se nalazu unutar elementa.`</p>`

Ovo je inline span element `Hello World` unutar odlomka.

Element SPAN je inline element i neće počinjati u novom retku te zauzima samo onoliko širine koliko je potrebno.

15. Osnove CSS-a

Kako je HTML prezentacijski jezik i njegova namjena nije bila ciljana na izgled web stranice, HTML opisuje strukturu stranice (što se prikazuje). CSS (engl. **Cascading Style Sheets**) je **jezik za oblikovanje izgleda web stranica**

Temelj CSS-a čine stilovi - pravila (rules) koja određuju kako će se HTML elementi prikazivati u pregledniku.

Ti stilovi - pravila sastoje se od **selektora** i **deklaracija**.

ime (selektor),

deklaracija

- skup određenih **svojstava**
- koja imaju neku **vrijednost**.

```
selektor {  
svojstvo: vrijednost;  
svojstvo: vrijednost;  
svojstvo: vrijednost;  
}
```

The diagram illustrates the components of a CSS rule. It shows a code snippet: `h1 { font-size: 24px; color: #23a8b4; }`. Three arrows point from labels to parts of the code: an orange arrow from 'ime (selector)' to 'h1', a blue arrow from 'svojstva' to 'font-size', and a green arrow from 'vrijednosti' to '24px'.

```
ime (selector) → h1 {  
svojstva → font-size: 24px;  
color: #23a8b4;  
} ← vrijednosti
```

Selektor — kaže koji element(e) na web stranici želimo urediti.

Svojstvo (property) — kaže što želimo promijeniti.

Vrijednost (value) — kaže kako želimo da to izgleda.

16. Ubacivanje CSS stilova u HTML stranicu

Stilovi se mogu ubaciti na tri načina:

- direktno **u kodu** (inline css) unutar HTML taga
- unutrašnja lista (interni css) u HTML dokumentu unutar taga <head></head>
- vanjska lista (eksterni css) u posebnoj datoteci s nastavkom .css (npr. stil.css)

16.1. U kodu (inline css)

U kodu možemo definirati svojstva i vrijednosti CSS stila za pojedine HTML tagove. Ova svojstva i vrijednosti CSS stila odnose se **samo** na **HTML oznaku** u kojoj je definiran style="...".

```
<body>
<h1 style="font-size:18px; color:#000099; font-family:Arial, Helvetica, sans-serif">Naslov</h1>
<h1>Naslov1 </h1>
</body>
```

Prednosti:

- Nadjačava interni i eksterni CSS
- Ponekad praktično za vrlo male promjene (samo npr.boja)
- **Ne koristi se u modernom web dizajnu**

Nedostaci:

- Promjene treba raditi na svakom elementu posebno
- HTML postaje nečitljiv i neuredan i teško održavanje koda

16.2. Unutrašnja lista (interni css)

U HTML dokumentu i to unutar taga <head> </head> . Svojstva i vrijednosti stilova se primjenjuju samo na dotičnu html datoteku i ne utječu na ostale html datoteke.

Unutrašnje liste započinju s tagom <style>

<style type = "text/css">

.css pravila.....

</style>.

Unutar taga **<style>** nalazi se selektori sa svojim svojstvima i vrijednostima

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS</title>
  <style type="text/css">

    h1 {
      font-size: 24px;
      color: aquamarine;
    }
    p {
      font-size: 14px;
      color: #52776b;
    }

  </style>
</head>
```

Prednosti:

- Jednostavno za **manje projekte**
- Može nadjačati stilove iz vanjske CSS datoteke
- **Ograničeno korištenje internog css-a**

Nedostaci:

- Stilovi vrijede samo za jednu HTML stranicu i nepraktično za više web stranica (promjene treba raditi na svakoj stranici zasebno)
- HTML datoteka postaje preopterećena stilovima
- Povećava veličinu dokumenta (može usporiti učitavanje kod većih stilova)

16.3. Vanjska lista

Kreiramo vanjsku datoteku sa **proizvoljnim imenom** i ekstenzijom **.css** (npr. tomo.css).



Svojstva i vrijednosti stilova iz vanjske css datoteke mogu se primjenjivati na sve HTML datoteke u koje je pozvana vanjska css datoteka. Da bismo pozvali vanjsku CSS datoteku potrebno je u HTML datoteku unutar oznaka

<head> ... </head> napisati

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>CSS</title>

  <link href="tomo.css (url adresa putanja do vanjske css datoteke)" rel="stylesheet"
  type="text/css">

</head>
```

Prednosti:

- HTML sadrži samo strukturu i sadržaj
- Kôd je pregledniji i čitljiviji
- Promjene se rade **na jednom mjestu**
- Idealno za veće projekte
- Ista CSS datoteka može se koristiti na više web stranica
- **Preporuča se korištenje vanjske css datoteke**

Nedostaci:

- Jedna pogreška u CSS datoteci može pokvariti izgled cijele web-stranice
- Zahtijeva oprez i testiranje

17.CSS selektori

CSS selektori određuju **na koje se HTML elemente primjenjuju CSS stilovi**. Oni su temelj za povezivanje HTML strukture i CSS dizajna.

17.1. Oznake ili tag

Svaka HTML oznaka može se koristiti kao ime stila. Takav stil **automatski** se **primjenjuje** na **sve** one elemente web stranice **koji su pod tim tagom**.

Primjer:

```
p {  
  color:#ffffff;  
  background: #000000;  
}
```

Ovaj css stil primjenjuje se na dijelove teksta koji su u HTML kodu obuhvaćeni tagom `<p>....</p>`

Primjer:

```
h1 {  
  text-align: center;  
}
```

Svi naslovi koji su u HTML kodu obuhvaćeni tagom `<h1>....</h1>` bit će centrirani.

Primjer:

```
ul {  
  list-style-type: square;  
}
```

Sve neuređene liste koji su u HTML kodu obuhvaćeni tagom `<ul>....</ul>` imat će kvadratne oznake..

Takoder možemo grupirati više HTML tagova koje imaju ista svojstva i vrijednosti, **te time izbjegavamo ponavljanje**

Primjer:

```
p, h2, ul {  
color:ffffff;  
background: #000000;  
}
```

17.2. Klase

Ako ne želimo promijeniti npr. veličinu teksta **svakom paragrafu** na web stranici, već samo **jednom ili nekima** tada su nam oznake ili tagovi kao css selektori ograničavajući jer se automatski primjenjuju na sve elemente pod tom oznakom.

U ovakvim slučajevima možemo koristiti **klase** koje možemo **proizvoljno primijeniti** na jedan ili više tagova - elementa na web stranici.

Svaka klasa započinje **točkom** i koristimo proizvoljno **asocijativno ime**.

Primjer:

<pre>.tekst { color:ffffff; background: #000000; }</pre>	ili	<pre>p.tekst { color:ffffff; background: #000000; }</pre>
--	-----	---

Klasa se **ne primjenjuje automatski** već je moramo pozvati na svaki html tag na koji želimo da se primjeni.

<p class="tekst">Ovo je tekst</p>

Višestruke klase

HTML element može imati **više klasa istovremeno**.

```
.okvir {  
  border: 2px solid black;  
  padding: 10px;  
}
```

```
.istaknuto {  
  color: red;  
  font-weight: bold;  
}
```

```
<p class="istaknuto okvir">Tekst</p>
```

Pravila korištenja klase

- Klasa počinje točkom
- Ime klase **ne smije** sadržavati razmake
- Može se koristiti više puta na stranici
- Moramo je pozvati u HTML tag

17.3. ID oznake

Svaka ID oznaka započinje # i koristimo proizvoljno asocijativno ime. Primjena je veoma slična klasama i glavna razlika je što istu klasu možemo primijeniti više puta unutar jednog HTML dokumenta, a istu **ID oznaku** možemo primijeniti samo **jedanput** unutar jedne HTML stranice.

ID selektor služi za stiliziranje jednog, jedinstvenog HTML elementa na stranici.

Primjer:

#header {		div#header {
color:ffffff;	ili	color:ffffff;
background: #000000;		background: #000000;
}		}

ID oznaka se ne primjenjuje automatski već je moramo pozvati za svaki elemente na koji želimo da se primjeni.

```
<div id="header"><h1>Ovo je naslov</h1></div>
```

Pravila korištenja ID selektora

- ID mora biti **jedinstven** na web stranici
- Jedan HTML element može imati **samo jedan ID**
- ID se u CSS-u označava znakom #

17.4. Kontekstualni selektori

Primjena kontekstualnih selektora jest da oblikuje određene „segmente“ HTML stranice. Ovi selektori imaju razmake unutar svog imena. Omogućuju odabir elemenata **prema njihovom međusobnom odnosu u HTML strukturi**, a ne samo prema nazivu, klasi ili ID-u.

a) Selektor potomaka (descendant selector)

Primjer:

```
ul strong {  
  color:blue;  
}
```

```
<ul>  
  <li>Tekst</li>  
  <li><strong>Tekst</strong></li>  
  <li><strong>Tekst</strong></li>  
  <li>Tekst</li>  
</ul>
```

Ovaj stil odabire sve tagove koji se nalazi unutar taga

Primjer:

```
div p {  
  color: blue;  
}
```

```
<div>  
  <p>Tekst</p>  
  <section>  
    <p>Tekst</p>  
  </section>  
</div>
```

Ovaj stil odabire sve <p> tagove koji se nalazi unutar taga <div>

b) Selektor djeteta (child selector)

```
div > p {  
  color: red;  
}
```

```
<div>  
  <p>Tekst</p>  
  <section>  
    <p>Tekst</p>  
  </section>  
</div>
```

Ovaj stil odabire sve <p> tagove koji se nalazi unutar taga <div>

c) Selektor susjedni (adjacent sibling)

```
h2 + p {  
  color: blue;  
}
```

```
<h2>Naslov</h2>  
<p>Ovaj paragraf će biti plav</p>  
<p>Ovaj paragraf neće biti zahvaćen.</p>
```

d) Opći susjedni (general sibling)

```
h2 ~ p {  
  color: blue;  
}
```

```
<h2>Naslov</h2>  
<p>Ovaj paragraf će biti plav.</p>  
<p>I ovaj paragraf će biti plav.</p>  
<div>Nije paragraf</div>  
<p>I ovaj paragraf će biti plav.</p>
```

Kombinacija ID-a ili klase

Primjer:

```
ul.podebljano b {  
  color:blue;  
}
```

```
<ul class="podebljano">  
  <li>Ovo je <b>važno</b></li>  
  <li>Još jedan <b>primjer</b></li>  
</ul>  
  
<ul>  
  <li>Ovo je <b>obično</b></li>  
  <li>Ovo je <b>obično</b></li>  
  <li>Ovo je <b>obično</b></li>  
</ul>  
  
<ul class="podebljano">  
  <li>Ovo je <b>važno</b></li>  
  <li>Još jedan <b>primjer</b></li>  
</ul>
```

Ovaj stil djeluje samo na tagove koji se nalaze unutar taga na koju je **primijenjena klasa** podebljano. Ne primjenjuje se na sve već samo na odabrane

Primjer:

ul#podebljano b {

color:blue;

}

```
<ul id="podebljano">
  <li>Ovo je <b>važno</b></li>
  <li>Još jedan <b>primjer</b></li>
</ul>

<ul>
  <li>Ovo je <b>obično</b></li>
  <li>Ovo je <b>obično</b></li>
  <li>Ovo je <b>obično</b></li>
</ul>

<ul>
  <li>Ovo je <b>obično</b></li>
  <li>Ovo je <b>obično</b></li>
  <li>Ovo je <b>obično</b></li>
</ul>
```

Ovaj stil djeluje samo na tagove koji se nalaze unutar taga na koju je **primijenjen id** podebljano. Ne primjenjuje se na sve već **samo na jedan** odabrani tag

17.5. Pseudo css selektori

Pseudo selektori vam omogućuju da definirate izgled pojedinih HTML elemenata u **određenim stanjima**.

Razlikujemo

- pseudo elementi i
- pseudo klase

17.5.1 Pseudo elementi

Pseudo-elementi stiliziraju (oblikuju) određeni dio odabranih HTML tagova - elemenata. Pseudo-elementi se **odvajaju dva puta s dvotočkom**

Standardni pseudo-elementi

<code>::after (:after)</code>	<code>::first-letter (:first-letter)</code>	<code>::placeholder</code> 🚩
<code>::backdrop</code> 🚩	<code>::first-line (:first-line)</code>	<code>::selection</code>
<code>::before (:before)</code>	<code>::grammar-error</code> 🚩	<code>::slotted()</code>
<code>::cue</code>	<code>::marker</code> 🚩	<code>::spelling-error</code> 🚩
<code>::cue-region</code>	<code>::part()</code> 🚩	

Primjer:

```
p::first-letter{  
color: blue;;  
}
```

Ova tip selektora pseudo-elementa nam kaže da će prvo slovo svakog odlomka biti u plavoj boji.

Pseudo element možemo koristiti s ostalim selektorima npr klasama ili id oznakama

Primjer:

```
p.tekst::first-letter {  
color: blue;  
}
```

Ova tip selektora pseudo-elementa nam kaže da će prvo slovo paragrafa koji ima **klasu tekst** biti u plavoj boji.

Ili

Primjer:

```
P#tekst::first-letter {  
  color: blue;  
}
```

Ova tip selektora pseudo-elementa nam kaže da će prvo slovo paragrafa koji ima **id tekst** biti u plavoj boji. Možemo za razliku od klasa primijeniti samo jednom na web stranici

::before i

::after

```
h2::before {  
  content: "★ ";  
  color: gold;  
}
```

Dodavanje zvijezde u zlatnoj boji ispred svakog naslova <h2>. Možemo isto kombinirati s klasama i id-om.

17.5.2. Pseudoklase

Pseudoklase se odvajaju dvotočkom i ima ih više nego pseudo-elemenata

<code>:active</code>	<code>:host</code>	<code>:only-child</code>
<code>:any-link</code> 	<code>:host()</code>	<code>:only-of-type</code>
<code>:blank</code> 	<code>:host-context()</code> 	<code>:optional</code>
<code>:checked</code>	<code>:hover</code>	<code>:out-of-range</code>
<code>:current</code> 	<code>:indeterminate</code>	<code>:past</code> 
<code>:default</code>	<code>:in-range</code>	<code>:placeholder-shown</code> 
<code>:defined</code>	<code>:invalid</code>	<code>:read-only</code>
<code>:dir()</code> 	<code>:is()</code> 	<code>:read-write</code>
<code>:disabled</code>	<code>:lang()</code>	<code>:required</code>
<code>:drop</code> 	<code>:last-child</code>	<code>:right</code>
<code>:empty</code>	<code>:last-of-type</code>	<code>:root</code>
<code>:enabled</code>	<code>:left</code>	<code>:scope</code>
<code>:first</code>	<code>:link</code>	<code>:state()</code> 
<code>:first-child</code>	<code>:local-link</code> 	<code>:target</code>
<code>:first-of-type</code>	<code>:not()</code>	<code>:target-within</code> 
<code>:fullscreen</code> 	<code>:nth-child()</code>	<code>:user-invalid</code> 
<code>:future</code> 	<code>:nth-col()</code> 	<code>:valid</code>
<code>:focus</code>	<code>:nth-last-child()</code>	<code>:visited</code>
<code>:focus-visible</code> 	<code>:nth-last-col()</code> 	<code>:where()</code> 
<code>:focus-within</code>	<code>:nth-last-of-type()</code>	
<code>:has()</code> 	<code>:nth-of-type()</code>	

Primjer:

```
a:link {  
color: green;  
}
```

Ova tip selektora pseudoklasa nam kaže da će **izgled** svih linkova imati zelenu boju.

```
a:visited {  
color: yellow;  
}
```

Ova tip selektora pseudoklasa nam kaže da će **izgled** svih **posjećenih** linkova imati žutu boju.

```
a:hover {  
color: gray;  
}
```

Ova tip selektora pseudoklasa nam kaže da kada mišom pređemo preko svih linkova koji imaju doći će do promjene boje linka u sivu.

```
a:active {  
color: red ;  
}
```

Ova tip selektora pseudoklasa nam kaže da kada kliknemo mišem na bilo koji link oni će promijeniti boje linka u crvenu

Primjer:

```
ul li:nth-child(2n) {  
background-color: lime;  
}
```

Ova tip selektora pseudoklasa nam kaže da će sve parne stavke u svim listama unutar html dokumenta imati lime boju pozadine

U pseudoklasama možemo koristiti također klase ili id oznake

Primjer:

```
ul.lista li:nth-child(2n) {  
    background-color: lime;  
}
```

Ova tip selektora pseudoklasa nam kaže da će sve parne stavke u listama koje u imaju **klasu *lista*** unutar html dokumenta imati lime boju pozadine

Primjer:

```
ul li:first-child {  
    font-weight: bold;  
}
```

Ova tip selektora pseudoklasa nam kaže da će sve prve stavke u svim listama unutar html dokumenta imati podebljanu vrijednost

CSS (engl. **Cascading Style Sheets**), riječ cascading (kaskadno) opisuje pravila prema kojima preglednik odlučuje koji će se CSS stil primijeniti kada **više istih CSS svojstava** cilja isti HTML element. *(Ako postoji istih stilova za isti element, CSS ima "pravila prioriteta".)*

Tri glavna kriterija kaskadnosti

1. Način pozivanja CSS-a

Inline CSS (najveći prioritet)

Eksterni CSS (najmanji prioritet)

2. Specifičnost selektora – ako su stilovi iz istog izvora (internog ili eksternog)

HTML oznaka (p) (najslabiji prioritet)

Klasa (.tekst) (jači prioritet)

ID oznaka (#naslov) (najači prioritet)

CSS kod

```
p { color: green; }
```

```
.tekst { color: blue; }
```

```
#glavni { color: red; }
```

Boja paragrafa je **crvena** jer ID ima najveći prioritet

HTML kod

```
<p id="glavni" class="tekst">Tekst</p>
```

3. Redoslijed pravila - Ako su selektori istis istim svojstvom, **primjenjuje se** zadnje napisano pravilo (vrijednost).

CSS kod

```
p { color: blue; }
```

```
p { color: red; }
```

Boja paragrafa je **crvena** jer je color:red (svojsvo i vrijednost) zadnje napisano

HTML kod

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

!important – iznimka (koristiti oprezno)

```
p {
```

```
  color: purple !important;
```

```
}
```

- Nadjačava sva druga pravila
- Narušava kaskadnost
- Oprezno koristiti

18.Svojstva i vrijednosti CSS stila za tekst

18.1.Vrsta pisma – font

U CSS definiramo **vrstu pisma** pomoću svojstva

```
{  
  font-family: Verdana, Arial, Helvetica, sans-serif;  
}
```

Na webu koristimo 5 familija fontova:

- serif
- sans-serif
- cursive
- fantasy
- monospace

Na većini web stranica za odlomke koriste se najčešće dvije familije fontova:

- **serifni fontovi** – kao što su Times, Garamond, Georgia,...
- **bez serifa fontovi (sans serif)**- kao što su Arial, Helvetika, Verdana, ...

Problem kod definiranja fontova je taj što font kojim smo izabrali za web stranicu, se neće prikazati ako korisnik na svom računalu nema instaliran taj font.

Danas je taj problem riješen preko npr,google fonts ili korištenjem css @font-face)

Zbog toga se definira veći broj fontova i na kraju se navede sans-serif ili serif da se kaže računalu da ako korisnik nema niti jedan od prije navedenih fontova, prikaži bilo koji bez serifni ili serifni font koji postoji na korisnikovom računalu.

Npr: font-family: Verdana, Arial, Helvetica, sans serif; govori pregledniku neka nam prikaže Verdana font, a ako ne postoji u računalu prikaži Arial font, ako Arial ne postoji prikaži Helvetiku font, a ako ne postoji Helevetica prikaži bilo koji bez sans serifni font koji postoji na računalu.

18.2.Veličina pisma

U CSS definiramo **veličinu** pisma pomoću svojstva

```
{  
  font-size: 10px;  
}
```

Veličina se može zadavati u pikselima(px), tipografskim točkama (pt) centimetrima (cm), milimetrima (mm), postocima(%), emovima (em), exovima (ex).

18.3. Stil pisma

U CSS definiramo stil pisma pomoću svojstva

```
{  
  font-style: normal;  
}
```

Vrijednosti mogu biti

normal,

italic (različiti verzija fonta koji postoji) i

oblique (normalni font koje je računalo samo udesno ukosilo).

18.4. Razmak između redova

U CSS definiramo razmak između redova pomoću svojstva

```
{  
  line-height: 20px;  
}
```

Vrijednosti se zadaju brojčanim vrijednostima, a veličine mogu biti isto kao kod veličine pisama(px, pt, in, cm, mm, %, em, ex). Razmak između redova treba biti jednak ili veći od veličine pisama.

18.5.Položaj crte u odnosu na tekst

U CSS definiramo različit položaj crte u odnosu na tekst pomoću svojstva

```
{  
  text-decoration: underline;  
}
```

Vrijednosti mogu biti

underline (podcrtan tekst),

overline (crta iznad teksta),

line-through (precrtan tekst),

blink (blinkanje teksta),

none (nema crte)

18.6.Debljina slova

U CSS definiramo **debljinu slova** pomoću svojstva

```
{  
  font-weight: bold;  
}
```

Vrijednosti mogu biti bold(podebljan tekst), a također vrijednosti se mogu zadavati 100, 200, 300, 400, 500, 600, 700, 800, 900, gdje je 900 najdeblje slovo

18.7.Zamjena slova velikim umanjenim

U CSS definiramo zamjenu slova **velikim umanjenim** pomoću svojstva

```
{  
  font-variant: small-caps;  
}
```

Vrijednosti mogu biti

small-caps (velika umanjena slova),

normal

18.8.Zamjena slova velikim ili malima

U CSS definiramo **zamjenu slova** pomoću svojstva

```
{  
  text-transform: lowercase;  
}
```

Vrijednosti mogu biti

capitalize (prvo slovo u riječi veliko),

uppercase (sva slova velika)

lowercase (sva slova mala)

18.9.Boja

U CSS definiramo **boju** teksta pomoću svojstva

```
{  
  color: #006666;  
}
```

Boje na ekranu se definiraju s tri boje: **crvenom, zelenom i plavom**. Tako definirane boje nazivaju se RGB od red, green i blue (aditivni sistem miješanja boja, jer se boje zbrajaju i miješanjem svih tri boja u punom intenzitetu dobivamo bijelu boju).

Svaka od te tri boje može imati 256 različitih nijansi tj. tonova. Ukupno boja koje RGB model može reproducirati je $256 \times 256 \times 256 = 16\,777\,216$ nijansi.

U **heksadecimalnom sustavu** u css-u koristimo šest znamenki s prefiksom „ # “(može i tri). Prve dvije znamenke su heksadecimalna vrijednost crvene boje, druge dvije su za zelenu boju i zadnje dvije za plavu boju. Svaka boja može imati raspon od (00 - ff)

Primjer:

color: #00ffaa ili color: #0fa

color: #abc1ea

U **dekadskom sustavu** u css-u koristimo raspon boja od (0 – 255). Prva brojčana vrijednost je za crvenu boje, druga brojčana vrijednost za zelenu boju i treća brojčana vrijednost za plavu boju.

Primjer:

color: rgb(0, 115, 172)

Vrijednost css svojstva color može biti ime neke boje. Međutim, u tom slučaju imamo na raspolaganju manji opseg boja, jer sve nijanse boja nemaju svoje ime.

Primjer:

color: lightblue

18.10. Razmak između riječi

U CSS definiramo razmak između riječi pomoću svojstva

```
{  
  word-spacing: 2px;  
}
```

Vrijednosti se zadaju brojčanim vrijednostima, a veličine mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

18.11. Razmak između slova

U CSS definiramo razmak između slova pomoću svojstva

```
{  
  letter-spacing: 2px;  
}
```

Vrijednosti se zadaju brojčano, a mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

18.12 Poravnanje teksta

U CSS definiramo poravnanje teksta pomoću svojstva

```
{  
  text-align: left;  
}
```

Vrijednosti mogu biti

- **left** (lijevo poravnanje),
- **center** (centralno),
- **right** (desno) i
- **justify** (obostrano)

18.13 Uvlaka

U CSS definiramo uvlaku između slova pomoću svojstva

```
{  
  text-indent: 20px;  
}
```

Vrijednosti se zadaju brojučano, a mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

18. Layout – dimenzioniranje i pozicioniranje elemenata

Postoje tri osnovne css tehnike za izradu rasporeda elemenata na web stranici:

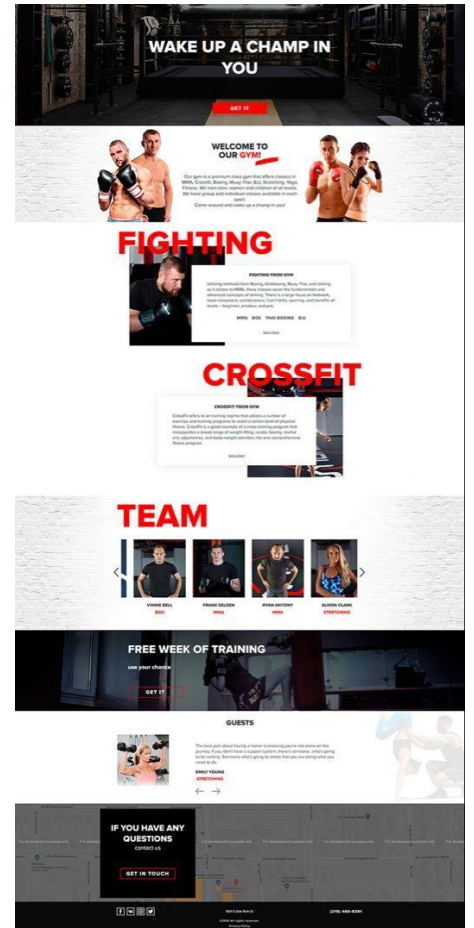
1. Flotanje
2. Flexbox
3. CSS grid

Float layout je starija tehnika za izradu rasporeda elemenata, osobito korisna za razumijevanje temelja naprednijih CSS layouta.

Flotanje nam omogućuju postavljanje elemenata **na razini bloka jedan pored drugog** umjesto jedan na drugi. Omogućuje nam izradu svih vrsta layouta, stranice s više stupaca, mreže i članke u stilu časopisa s tekstom koji teče oko slike.

Flotanjem konačno počinjemo stvarati prave web stranice.

Rasporedi temeljeni na float elementima uglavnom su zamijenjeni flexboxom i css gridom na modernim web stranicama. Više od desetljeća, float elementi su služili kao temelj layouta većine web stranica na internetu, što znači da ćete se s njima sigurno susretati.



Primjer – dva stupca

HTML kod

```
<div class="roditelj">
  <div class="lijevo">...</div>
  <div class="desno">...</div>
</div>
```

Obavezno imamo odnos „**roditelj-dijete**“. Roditelj je `<div class="roditelj">`unutar kojeg se nalaze djeca, u ovom slučaju `<div class="lijevo">...</div>` i `<div class="desno">...</div>`

Osnovni CSS kod

```
.roditelj {  
  background-color: #ddd;  
  width: 1200px;  
  margin-left: auto;  
  margin-right: auto;  
}
```

```
.lijevo {  
  float: left;  
  width: 500px;  
}
```

```
.desno {  
  float: right;  
  width: 600px;  
}
```

Problem:

Roditelj nema visinu, jer su oba djeteta floatana i u našem slučaju roditelj nema boju pozadine. Problem rješavamo uvođenjem pseudo elementa kojem je pridružena klasa.

```
.clearfix::after {  
  content: "";  
  display: block;  
  clear: both;  
}
```

i pozivanjem klase u <div> koji je roditelj

```
<div class="roditelj clearfix">  
<div class="lijevo">...</div>  
<div class="desno">...</div>
```

</div>

Na što treba paziti kod flotanja

Djeca imaju svojstvo float i width

Zbroj širina djece je jednak ili manji od širine roditelja

roditelj ima pozvanu klasu clearfix

19. Model kutija (box-model)

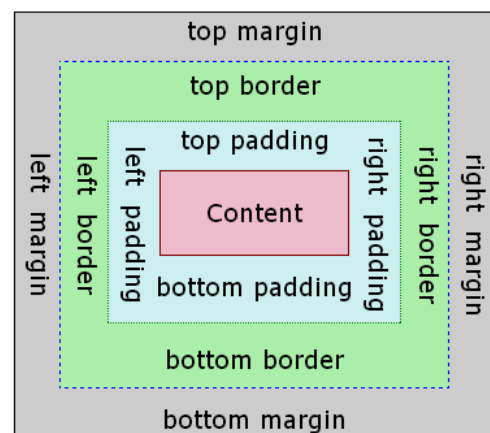
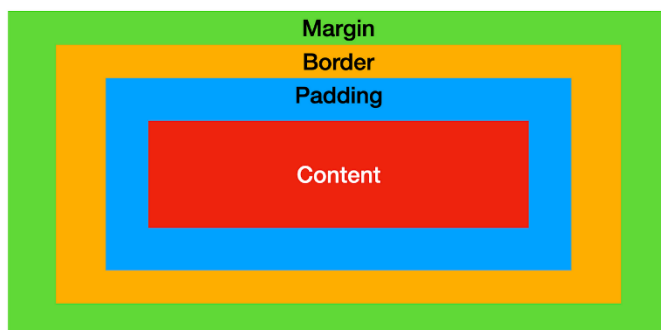
Box model u CSS-u opisuje kako će svaki HTML element biti prikazan na stranici i koje su i njegove dimenzije i međusobni razmak susjednih HTML elemenata.

Svaki element se sastoji od **četiri dijela**:

- Sadržaj (content)
Stvarni sadržaj elementa (tekst, slika itd.)
Na njega se odnose **width i height**
- Unutarnji razmak (padding)
Prostor između sadržaja i obruba
Povećava ukupnu veličinu elementa
- Obrub (border)
Linija oko paddinga i sadržaja
- Vanjski razmak (margin)
Prostor izvan obruba, odvaja element od drugih elemenata

Zašto je box model važan?

- Precizno kontrolira veličinu elemenata



19.2 Svojstva i vrijednosti CSS stila za obrub-border

Preko CSS-a možemo ubaciti rub-liniju svakom elementu. Rub možemo zadavati za :

- **top** (vrh)
- **right** (desno)
- **bottom** (dno)
- **left** (lijevo)

Kod svakog ruba-linije moramo definirati tri svojstva:

- vrstu ruba-linije
- debljina okvira- linije
- boju okvira-linije

19.2.1.Vrsta ruba-linije

U CSS definiramo vrstu ruba-linije pomoću svojstva

```
{  
  border-style: dashed;  
}
```

Vrijednosti mogu biti

solid(obična),

dotted (točkasta),

dashed (crti) i

double (dvije),

itd .

Ako želimo zadati vrstu ruba samo sa jedne strane (npr vrh) definiramo

```
{  
  border-top-style: dashed;  
}
```

19.2.2. Debljina ruba- linije

U CSS definiramo **debljinu ruba** pomoću svojstva

```
{  
  border-width: 2px;  
}
```

Također kao vrijednosti mogu se zadavati brojke. Mjere mogu biti isto kao kod veličine slova (px, pt, in, cm, mm, %, em, ex).

Ako želimo zadati debljinu samo sa jedne strane (npr vrh) definiramo

```
{  
  border-top-width: dashed;  
}
```

19.2.3. Boja ruba-linije

U CSS definiramo **boju ruba** pomoću svojstva

```
{  
  border-color: #0099CC;  
}
```

Vrijednost može biti definirana tekstualnom, heksadecimalnom ili u dekadskom sustavu

Ako želimo zadati boju ruba samo sa jedne strane pišemo:

```
{  
  border-top-color: #993300  
}
```

19.2.3. Sintaksa pisanja

Ako imamo **rub-liniju samo s jedne strane** (npr.vrh) oko nekog html elementa.

```
{
  border-top-width: 1px;
  border-top-style: solid;
  border-top-color: #993300
}
ili jednostavnije
{
  border-top: 1px solid #993300
}
(Redosljed pisanja vrijednosti nije bitan.)
```

Ako imamo **isti rub-liniju sa sve 4 strane** oko nekog html elementa.

```
{
  border-top: 1px solid #993300;
  border-bottom: 1px solid #993300;
  border-left: 1px solid #993300
  border-right: 1px solid #993300
}
ili jednostavnije
{
  border: 1px solid #993300
}
```

Ako imamo **rub-liniju sa sve 4 strane a razlikuju se sa svih strana samo po boji** oko nekog html elementa.

```
{
  border: 1px solid;
  border-top-color: #993300;
  border-right-color: #22aa11;
  border-bottom-color: #aa12ad;
  border-left-color: #33a145;
}
ili jednostavnije
{
  border: 1px solid;
  border-color: #993300 #22aa11 #aa12ad #33a145;
                top      right  bottom  left
}

```

Može vrijediti za bilo koje svojstvo ruba-linije (debljinu, vrstu ili boju)

Ako imamo **rub-liniju** sa sve 4 strane a razlikuju se samo po boji, s time da su gornji i donji rub iste boje, te lijevi i desni rub iste boje oko nekog html elementa.

```
{
border: 1px solid;
border-top-color: #993300;
border-right-color: #22aa11;
border-bottom-color: #993300;
border-left-color: #22aa11;
}
ili jednostavnije
{
border: 1px solid;
border-color: #993300 #22aa11;
               top i bottom   right i left
}
```

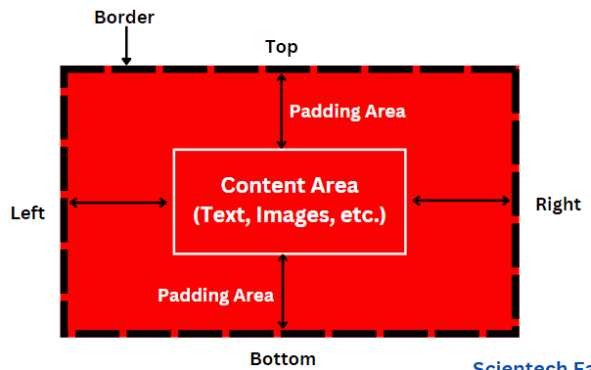
Može vrijediti za bilo koje svojstvo ruba-linije (debljinu, vrstu ili boju)

Ako imamo **rub-liniju** sa sve 4 strane a razlikuju se samo po boji, s time da su lijevi i desni rub iste boje oko nekog html elementa.

```
{
border: 1px solid;
border-top-color: #993300;
border-right-color: #22aa11;
border-bottom-color: #993300;
border-left-color: #22aa11;
}
ili jednostavnije
{
border: 1px solid;
border-color: #993300 #123456 #22aa11;
               top       left i right  bottom
}
```

Može vrijediti za bilo koje svojstvo ruba-linije (debljinu, vrstu ili boju)

19.3 Svojstva i vrijednosti CSS stila za padding



Padding definira razmak između sadržaja HTML elementa i njegovog ruba-bordera.

Imamo 4 vrijednosti:

- vrh (**top**),
- lijevo (**left**),
- desno (**right**) i
- noge (**bottom**)

```
{  
  padding: 10px;  
}
```

U ovom slučaju vrijednosti su **iste** za vrh(**top**), lijevo(**left**), desno(**right**) i noge(**bottom**)

```
{  
  padding-top: 5px;  
  padding-right: 10px;  
  padding-bottom: 15px;  
  padding-left: 20px;  
}  
ili  
padding: 5px 10px 15px 20px;
```

Možemo zadavati padding posebno za vrh(**top**), desno(**right**), noge(**bottom**) i lijevo(**left**). Ako ostavimo samo padding, tada je **prva** vrijednosti za **vrh**, **druga** vrijednost je za **desno**, **treća** vrijednost je za **noge** i četvrta vrijednost je za **lijevo**.

```
{  
  padding: 20px 10px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh i noge**, a **druga** vrijednost za **lijevo i desno**.

```
{  
padding: 20px 10px 30px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh**, **druga** vrijednost je za **lijevo i desno**, a **treća** vrijednost je za **noge**.

19.4 Svojstva i vrijednosti CSS stila za margin

Definira razmak od ruba HTML elemenata.

Imamo 4 vrijednosti:

- vrh (**top**),
- lijevo (**left**),
- desno (**right**) i
- noge (**bottom**)

```
{  
margin: 10px;  
}
```

U ovom slučaju vrijednosti su **iste** za vrh(**top**), lijevo(**left**), desno(**right**) i noge(**bottom**)

```
{  
margin-top: 5px;  
margin-right: 10px;  
margin-bottom: 15px;  
margin-left: 20px;  
}  
ili  
margin: 5px 10px 15px 20px;
```

Možemo zadavati margin posebno za vrh(**top**), desno(**right**), noge(**bottom**) i lijevo(**left**), Ako ostavimo samo padding, tada je **prva** vrijednosti za **vrh**, **druga** vrijednost je za **desno**, **treća** vrijednost je za **noge** i četvrta vrijednost je za **lijevo**.

```
{  
  margin: 20px 10px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh i noge**, a **druga** vrijednost za **lijevo i desno**.

```
{  
  margin: 20px 10px 30px;  
}
```

U ovom slučaju **prva** vrijednosti je za **vrh**, **druga** vrijednost je za **lijevo i desno**, a **treća** vrijednost je za **noge**.

19.5 Svojstva i vrijednosti CSS stila za width i height

U CSS definiramo širinu HTML elementa pomoću svojstva

```
{  
  width: 500px  
}
```

Širina se najčešće definira u **pikselima(px)**, **postocima(%)**, **removima (rem)**, **emovima (em)**...

U CSS definiramo visinu HTML elementa pomoću svojstva

```
{  
  height: 500px  
}
```

Visina se najčešće definira u **pikselima(px)**, **postocima(%)**, **removima (rem)**, **emovima (em)**...

Visina HTML elementa može ograničavati sadržaj. Sadržaj postane veći od definirane visine i može izaći iz elementa. Zbog toga se u mnogim slučajevima izbjegava definiranje visine HTML elementa, a ovisiti će o visini sadržaja + padding vrha + padding dna

19.6 Svojstva i vrijednosti CSS stila za box-sizing

box-sizing: content-box; - drfinirana vrijednost i padding i border ZBRAJAJU SE s širinom i visinom.

```
.box {  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
  box-sizing: content-box;  
}
```

Kako se računa veličina?

širina:: 200px

padding: $20\text{px} \times 2 = 40\text{px}$ (lijevi i desni)

border: $5\text{px} \times 2 = 10\text{px}$ (lijevi i desni)

Stvarna veličina HTML elementa s klasom box:: **ukupna širina: 250px**

```
.box {  
  width: 200px;  
  padding: 20px;  
  border: 5px solid black;  
  box-sizing: border-box;  
}
```

Kako se računa veličina?

Širina :: 200px

padding: $20\text{px} \times 2 = 40\text{px}$ (lijevi i desni)

border: $5\text{px} \times 2 = 10\text{px}$ (lijevi i desni)

Stvarna veličina HTML elementa s klasom box:: **ukupna širina ostaje 200px.** (Smanjuje se širina sadržaja unutar HTML elementa)

