# Teorijske osnove

## Frontend programiranje

Frontend programiranje se odnosi na proces razvoja korisničkog interfejsa i funkcionalnosti koje korisnici direktno vide i interaguju sa njima prilikom korišćenja web aplikacija, mobilnih aplikacija ili drugih vrsta softvera. Ovo uključuje sve što se prikazuje u veb browser-u ili na ekranu mobilnog uređaja, uključujući tekst, slike, forme za unos podataka, dugmad, menije i druge elemente.

Frontend programeri koriste jezike kao što su HTML (HyperText Markup Language) za strukturu veb stranice, CSS (Cascading Style Sheets) za stilizovanje i izgled stranice, i JavaScript za interaktivnost i dinamičnost. Ovi jezici rade zajedno kako bi kreirali funkcionalne i estetski prijatne korisničke interfejse.

Takođe, frontend programiranje uključuje rad sa različitim alatima i okvirima (npr. React, Angular, Vue.js) koji olakšavaju razvoj kompleksnih korisničkih interfejsa i omogućavaju bolje upravljanje stanjem aplikacije.

Kroz frontend programiranje, programeri se fokusiraju na obezbeđivanje pozitivnog korisničkog iskustva tako što pravilno organizuju informacije, olakšavaju navigaciju i osiguravaju da aplikacija bude odzivna i brza.

## HTML (HyperText Markup Language)

HTML (HyperText Markup Language) je standardni jezik za označavanje i strukturiranje sadržaja veb stranica. To je osnovni jezik koji se koristi za izgradnju i oblikovanje elemenata koji se prikazuju na veb stranicama. Kroz HTML, možemo definisati različite delove veb stranice, kao što su naslovi, paragrafi, slike, veze, forme i druge komponente.

HTML koristi "oznake" ili "tagove" da označi različite delove teksta i elemenata na stranici. Oznake se obično sastoje od otvarajućeg i zatvarajućeg dela, gde otvarajuća oznaka označava početak elementa, dok zatvarajuća označava kraj. Na primer, <h1> označava početak naslova prvog nivoa (heading level 1), a </h1> označava kraj tog naslova. Pravilno strukturiran HTML je ključan za pravilno prikazivanje veb stranica na različitim uređajima i veb preglednicima.

## CSS (Cascading Style Sheets)

CSS (Cascading Style Sheets) predstavlja jezik za stilizovanje veb stranica i omogućava definisanje izgleda i prezentacije elemenata koje smo strukturirali koristeći HTML. U suštini, CSS se koristi za stilizovanje elemenata veb stranice, kao što su boje, fontovi, raspored, veličine i druge vizuelne karakteristike.

CSS omogućava odvajanje dizajna veb stranice od njenog sadržaja, čime se olakšava održavanje i ažuriranje veb stranica. Umesto da stilizujemo svaki element pojedinačno u HTML-u, koristimo CSS da definišemo stilove koji se primenjuju na različite elemente. Na primer, možemo definisati stil za sve naslove (<h1>, <h2>, <h3>, itd.) koji će važiti na svim stranicama na sajtu.

Princip "kaskadnog" (Cascading) u CSS-u se odnosi na to da stilovi mogu biti nasleđeni od roditeljskih elemenata, ali ih takođe možemo izmeniti ili nadjačati specifičnim stilovima za određene elemente.

CSS takođe omogućava prilagodljivost i odzivnost veb stranica na različite uređaje i veličine ekrana, što je ključno za kreiranje korisničkih interfejsa koji su funkcionalni i lepo izgledaju na svim platformama.

Ukratko, CSS je jezik koji omogućava oblikovanje i stilizovanje veb stranica kako bi se postigao željeni izgled i korisničko iskustvo.

## JavaScript

JavaScript je programski jezik koji se često koristi za razvoj dinamičkih i interaktivnih veb stranica. To je jezik koji se izvodi na klijentskoj strani (u veb preglednicima korisnika), što znači da se kod izvršava na računarima korisnika, a ne na serverima.

JavaScript omogućava programerima da kontrolišu ponašanje elemenata na veb stranici, reaguju na korisničke akcije (kao što su klikovi i unos teksta) i dinamički ažuriraju prikaz stranice bez potrebe za ponovnim učitavanjem celokupne stranice. Takođe se koristi za validaciju podataka unetih u forme, manipulaciju dokumentima i animaciju elemenata.

Ovaj jezik je ključan u razvoju modernih veb aplikacija i omogućava programerima da kreiraju kompleksne funkcionalnosti koje poboljšavaju korisničko iskustvo. Pored toga, JavaScript je često korišćen i izvan veb preglednika, kao što je u razvoju mobilnih aplikacija koristeći okvire kao što su React Native ili u razvoju serverske aplikacije koristeći Node.js.

U suštini, JavaScript je snažan alat koji omogućava interaktivnost, dinamiku i kompleksne funkcionalnosti na veb stranicama i u drugim vrstama softvera.

## Angular

Angular je popularan open-source okvir (framework) za razvoj veb aplikacija. Razvila ga je kompanija Google i predstavlja moćan alat koji olakšava kreiranje dinamičkih, interaktivnih i skalabilnih veb aplikacija. To je klijentski (frontend) okvir koji se koristi za izgradnju modernih veb aplikacija koje mogu da rade u veb preglednicima.

Glavne karakteristike Angular-a uključuju:

* Komponentnaarhitektura: Angular se bazira na komponentama, koje su nezavisne, ponovno upotrebljive i samostalne delove korisničkog interfejsa. Ovo olakšava organizaciju i upravljanje kodom.
* Dvosmerna veza podataka: Angular omogućava automatsko ažuriranje korisničkog interfejsa kada se podaci u modelu promene, i obrnuto. To olakšava praćenje i upravljanje podacima.
* Injekcija zavisnosti: Angular koristi koncept injekcije zavisnosti kako bi se omogućila bolja kontrola i organizacija komponenata i servisa.
* Modularnost: Kod se organizuje u module, što doprinosi boljoj organizaciji i skalabilnosti aplikacije.
* Rute i navigacija: Angular pruža mehanizam za definisanje ruta i navigaciju unutar aplikacije, omogućavajući korisnicima da prelaze između različitih delova aplikacije.
* Testiranje: Angular ima ugrađene alate za jednostavno testiranje koda, što doprinosi kvalitetu aplikacije.

Angular omogućava razvoj složenih aplikacija kao što su veb aplikacije, progresivne veb aplikacije (PWA) i veb stranice visokih performansi. Kroz svoje napredne funkcionalnosti, Angular je postao popularan izbor za razvoj modernih veb aplikacija.

## Mikrofrontend

Mikrofrontend je arhitektonski pristup u razvoju veb aplikacija gde se aplikacija dekomponuje na manje, autonomne delove (mikrofrontendi) koji mogu da se razvijaju, testiraju, implementiraju i održavaju nezavisno. Ovi mikrofrontendi predstavljaju manje delove korisničkog interfejsa koji rade zajedno kako bi činili celinu aplikacije.

Svaki mikrofrontend je samostalan modul ili aplikacija koji je fokusiran na specifičnu funkcionalnost ili deo korisničkog interfejsa. Oni mogu biti razvijani od strane različitih timova ili pojedinaca, a zatim integrisani u glavnu aplikaciju. Ovaj pristup omogućava agilnost u razvoju, ubrzava implementaciju novih funkcionalnosti i olakšava održavanje kompleksnih veb aplikacija.

Prednosti mikrofrontend arhitekture uključuju:

* Nezavisna razvojna ekipa: Svaki mikrofrontend može da bude razvijan od strane nezavisne ekipe, omogućavajući paralelni razvoj i ubrzanje vremena do implementacije.
* Nezavisna implementacija i održavanje: Svaki mikrofrontend može da se implementira, testira i održava nezavisno, što olakšava upravljanje kodom i brže isporučivanje promena.
* Skalabilnost: Omogućava skalabilnost na nivou komponenata, gde se resursi mogu optimalno koristiti prema zahtevima svake komponente.
* Modularnost i ponovna upotreba: Mikrofrontendi su modularni i mogu se koristiti u različitim delovima aplikacije ili u različitim aplikacijama, što doprinosi ponovnoj upotrebi koda.
* Brže učitavanje stranica: Moguće je optimizovati učitavanje stranica, jer se samo potrebni mikrofrontendi učitavaju za određenu stranicu.

Važno je naglasiti da, iako mikrofrontendi donose brojne prednosti, njihova implementacija zahteva pažljivo planiranje, upravljanje stanjem aplikacije i pravilno upravljanje komunikacijom između mikrofrontenda i glavne aplikacije.