

# Intro

# Node.js

- Provera trenutne verzije Node.js
  - `$ npm -v`
- Ukoliko verzija nije prikazana ili ukoliko dođe do greške, instalirajte Node.js sa sledećeg linka <https://nodejs.org/en/>
- Napomena - potrebno je da imate Node.js u verziji 8.9.x ili višoj (poželjna verzija - 10.2.1)



# Angular-Cli

- Omogućava korisnicima da kreiraju i upravljaju projektima iz komande linije
- Automatizuje procese kreiranja projekata, dodavanja novih komponenti, modula itd.
- Instalacija
  - `$ npm install -g @angular/cli@latest`
- Provera verzije
  - `$ ng --version`



# Kreiranje i pokretanje aplikacije

- `$ ng new hello-world`
- `$ cd hello-world`
- `$ ng serve`
- Aplikacija će biti pokrenuta po defaultu na portu 4200
- `$ ng serve --port 9001`

# Angular Components

# Kreiranje komponente

- HTML poseduje predefinisani skup tagova koji mogu da se koriste
- Ideja komponenti jeste u tome da se definiše novi skup tagova koji će biti korišćeni od strane browsera
- Nova komponenta će prilikom generisanja automatski biti dodata u `app.module.ts`

```
1 $ ng generate component hello-world
2 installing component
3   create src/app/hello-world/hello-world.component.css
4   create src/app/hello-world/hello-world.component.html
5   create src/app/hello-world/hello-world.component.spec.ts
6   create src/app/hello-world/hello-world.component.ts
```

# hello-world.component.ts

- Import sekcija
  - ideja jeste da se povuku zavisnosti iz drugih modula
- Component dekoratori
  - Deklaracija komponente
- Pozivanje komponente
  - `<app-hello-world> </app-hello-world>`



# app.component.html

- Index.html je statička .html stranica
- `<app-root>` poziva app.component koja je root komponenta Angulara





# Interpolation

- Pomoću `{{ }}` možemo prikazivati vrednost promenljive iz komponente u template-u
- Takođe, unutar zagrada se može upisati bilo koja vrednost izraza koja treba da bude izračunata

```
1  <p>  
2    Hello {{ name }}  
3  </p>
```



# Sharing data

# Parent => Child

- Angular omogućuje razmenjivanje podataka između komponenti
- Child komponenta
  - Deklariše promenljivu pomoću `@Input()` dekoratora
  - `@Input() childMessage: string;`
- Parent komponenta
  - Prilikom pozivanja template child komponente mora da joj prosledi vrednost koju ona definiše kao input
  - `<app-child [childMessage]="parentMessage"`



# Child => Parent via Output() + EventEmitter

- Drugi način razmene podataka je emitovanjem podataka iz child komponente
- Child komponenta
  - Deklarišemo messageEvent promenljivu pomoću @Output() dekoratora
  - @Output() messageEvent = new EventEmitter<string>();
  - Nad deklarisanom promenljivom poziva se emit() metoda sa porukom koju želimo da pošaljemo
- Parent komponenta
  - Parent komponenta se subscribe-uje na događaje koji se emituju iz child komponente
  - <app-child1 (messageEvent) = "receiveMessage(\$event)">



# Child => Parent via ViewChild()

- ViewChild omogućava da jedna komponenta bude injektovana u drugu
- Ovo omogućava parent-u da pristupi svim atributima i metodama child komponente
- Parent komponenta
  - Deklarišemo kojoj child komponenti pristupamo pomoću @ViewChild(ComponentName) dekoratora
  - Pristup je omogućen tek nakon inicijalizacije Child komponente





# Forms

# Forme

- `<form #f="ngForm" (ngSubmit)="onSubmit(f.value)">`
- `#f="ngForm"` kreiramo lokalnu promenljivu za ovaj view
- `ngForm` dolazi iz `NgForm` direktive
- `(ngSubmit)="onSubmit(f.value)"`
- `(ngSubmit)` dolazi iz `NgForm` direktive
- `onSubmit()` je funkcija koja će biti definisana u `.ts` fajlu komponente
- `f.value` će vratiti `key/value` objekte iz `FormGroup`-e

# Routing



# Rutiranje

- Omogućuje:
  - Navigaciju između različitih prikaza
  - Razmenu parametara između komponenti koje upravljaju prikazima
- Obzirom da je aplikacija stablo komponenti, rutiranje se svodi na zadavanje komponenti za rute
- Rutiranje se postiže pravilima za rutiranje
- Kako bi omogućili rutiranje u Angularu potrebno je:
  - Importovati RouterModule
  - Instalirati rute korišćenjem RouterModule.forRoot(routes) unutar importa NgModule-a



# Rutiranje

```
const routes: Routes = [  
  // basic routes  
  { path: '', redirectTo: 'home', pathMatch: 'full' },  
  { path: 'home', component: HomeComponent },  
  { path: 'about', component: AboutComponent },  
  { path: 'contact', component: ContactComponent },  
  { path: 'contactus', redirectTo: 'contact' },
```

# <router-outlet>

- <router-outlet> direktiva definiše gde sadržaj svake od ruta treba biti renderovan
- Pozivanje druge rute može biti izvršeno na sledeći način
  - <a [routerLink]="['/home']">Home</a>

