

# *Angular Introduction*

---

Kristoffer Hendricks



## What is Angular?

- Developed by Google
- Component-based framework
- Libraries for routing, forms, client-server coms, and more
- Suite of tools for developers
- Open Source
- Current Version 12.2.8





# *Why Angular???*

<https://www.theladders.com/jobs/searchresults-jobs?keywords=Angular>

---

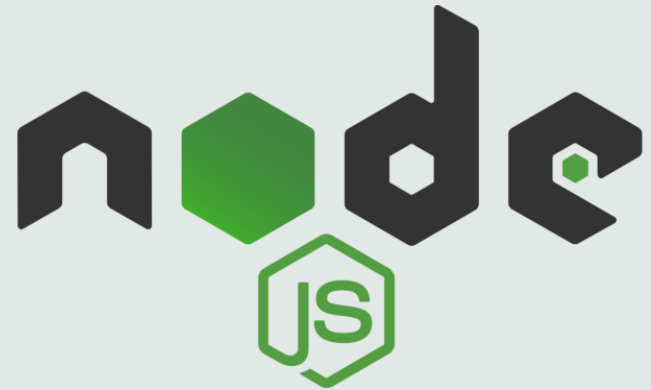
# *How to get started with Angular*

- Install Node and NPM
- Install Angular CLI
- Create first Angular App



# *How to install Node.js*

- Goto <https://nodejs.org/en/download>
- Download the version for your operating system
- Open cmd or terminal
- To check for installation type: `node -v`



# *How to install Angular and Angular CLI*

- Open cmd or terminal
- Type: `npm install -g @angular/cli`
- Check version: `ng version`



# *Create an Angular application*

- Create angular-workshop folder on desktop
- Open cmd or terminal and navigate to the folder you just created
- Type: `ng new <application name>`
- Navigate to app folder
- To see app:
  - `ng serve`
  - navigate to localhost:4200 in browser
- Folder structure
- Hello Angular



# *Interpolation* `{{}}`

Things we can do:

- `{{ title }}`
- Functions
- Operations
- Comparisons

Things we cannot do:

- Assign values
- `typeof`
- Increment
- Assignment





# Angular CLI

- <https://angular.io/cli>
- ng serve
- ng generate
- ng generate component <yourComponent>
- ng g c <yourComponent>
- ng generate module <yourModule>
- ng generate service <yourService>
- ng generate class <yourClass>
- ng g c folderStructure/<yourComponent>
- ng build



# *Components*

- Why do we use components?
- Create a component:
  - `ng g c <componentFile/componentName>`
- Creates 4 files
  - `componentName.component.ts`
  - `componentName.component.html`
  - `componentName.component.spec.ts`
  - `componentName.component.css`
- How to use the components?
  - `<app-myComponent></app-myComponent>`



# *Other Angular features to explore*

- Modules
- Services
- Forms & Reactive Forms
- Logic and loops
- Authorization and Authentication
- Pipes
- Directives
- Routing
- APIs
- Testing



*Questions?*

---

