Angular

→ Angular is an open-source JavaScript MVW framework for front-end of web application.

→ It can be used to create SPA (Single Page Application)

→ Angular 11.

How to install Angular?

→ Prerequisite: node

→ Install Angular CLI (Command Line Interface)

↳ Used to install libraries, generate, build, run, and deploy Angular application

```
npm install -g @angular/cli@latest
```

How to create an Angular project?

```
ng new <projectname>
```

Run an Angular Application (Prerequisite: npm install)

We have to first build the angular app: `ng build`

Then, we can run it.

```
ng serve -O
```

OR

```
npm start
```

default url: http://localhost:4200

* Angular Component

An Angular view is made up of one or more components.

An Angular component represents a portion of the view.

Angular Component = HTML Template
+ Component class
+ Component Metadata

i) HTML Template (Presentation)

It is the HTML code for rendering the component. It can also have additional Angular specific syntax to communicate with the component class.

ii) Component Class (Logic)

It is a Typescript class that includes properties and methods.

Properties store data.

Methods include the logic for the component.

Eventually, this class will be compiled to JS.

iii) Component Metadata

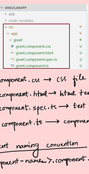
Metadata is some extra data for a component used by Angular API to execute the component.

Examples:-
location of HTML template,
location of CSS files of the component,
etc.

↳ Generating Angular Component using Angular CLI

```
ng generate component component-name
```

The above command will create a new folder for the component under src/app folder. Four files will be created in the folder.



↳ great-component.css → CSS file

↳ great-component.html → html template

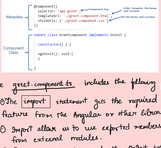
↳ great-component.spec.ts → test file for unit tests

↳ great-component.ts → component class file

Component naming convention

<component-name>.component.<file-type>

→ Component Class



The great-component.ts includes the following part:

i) The import statement gets the required features from the Angular or other libraries.

① Import alias as to use exported members from external modules.

② @Component decorator and the OnInit interface are imported in @angular/core library

ii) Component Metadata: The @Component decorator is used to specify the metadata for the component class defined immediately below it.

It is a function and can include different configs for the component.

It instructs Angular where to get the required files for the component, create and render the component.

All Angular components must have @Component decorator above the component class.

iii) Component class: It contains properties and methods to interact with the view through the Angular API.

It implements the OnInit interface, which is a lifecycle hook.

Bootstrapping the Component

→ Loading our component

→ Before this, we need to host our application and load the root component.

→ This process is called bootstrapping.

→ Angular is a SPA framework. So, we need to host our application in index.html, then we need to define a root module to bootstrap our components.

→ index.html is the only web page in an Angular application. Content changes is injected dynamically by JS. → SPA.

→ Angular CLI generates the following automatically

① index.html

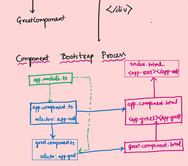
② root component → app.component.ts

③ root module → app.module.ts

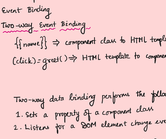
④ HTML template → app.component.html

↳ Appcomponent class = root component

↳ AppModule class = root module



Component Bootstrap Process



* HTML Template

* Event Binding

→ Two-way Event Binding

{{name}} → component class to HTML template

(click)="great()" → HTML template to component class

Two-way data binding performs the following:-

1. Sets a property of a component class
2. Listens for a DOM element changes event.

Angular supports two-way event binding using:-

i) ngModel directive

ii) getters and setters

i) ngModel directive – [( )] (banana-box syntax)

Syncs values from UI to the property of component class and vice-versa.

[( )] = [ ] + ( )

where, [ ] binds attribute

( ) binds an event.