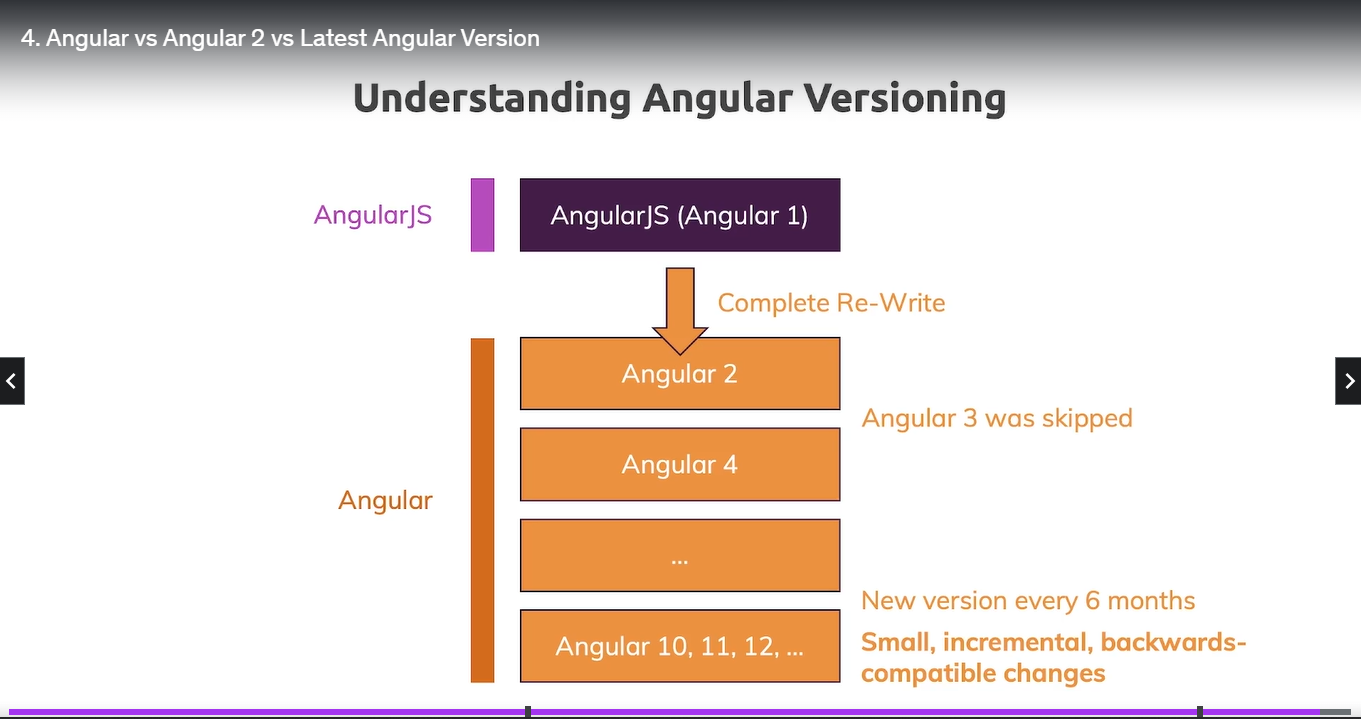
**Angular Tutorials**

What is Angular?

Angular is a JavaScript Framework which allows you to create reactive Single-Page-Applications [SPAs].

Angular Versions.



6.Project Setup & First App:

**https://angular.io/cli**

**https://nodejs.org/en/download/**

(CLI=Command Line Interface)

GO TO CMD Make it as administrative and run below commands in your machine

To install npm:(Windows) npm install -g @angular/cli@latest

(MAC) sudo npm install -g @angular/cli@latest

Go to Angular folder: cd My-System, cd AngularProjects

Creating a project: ng new my-angular-app

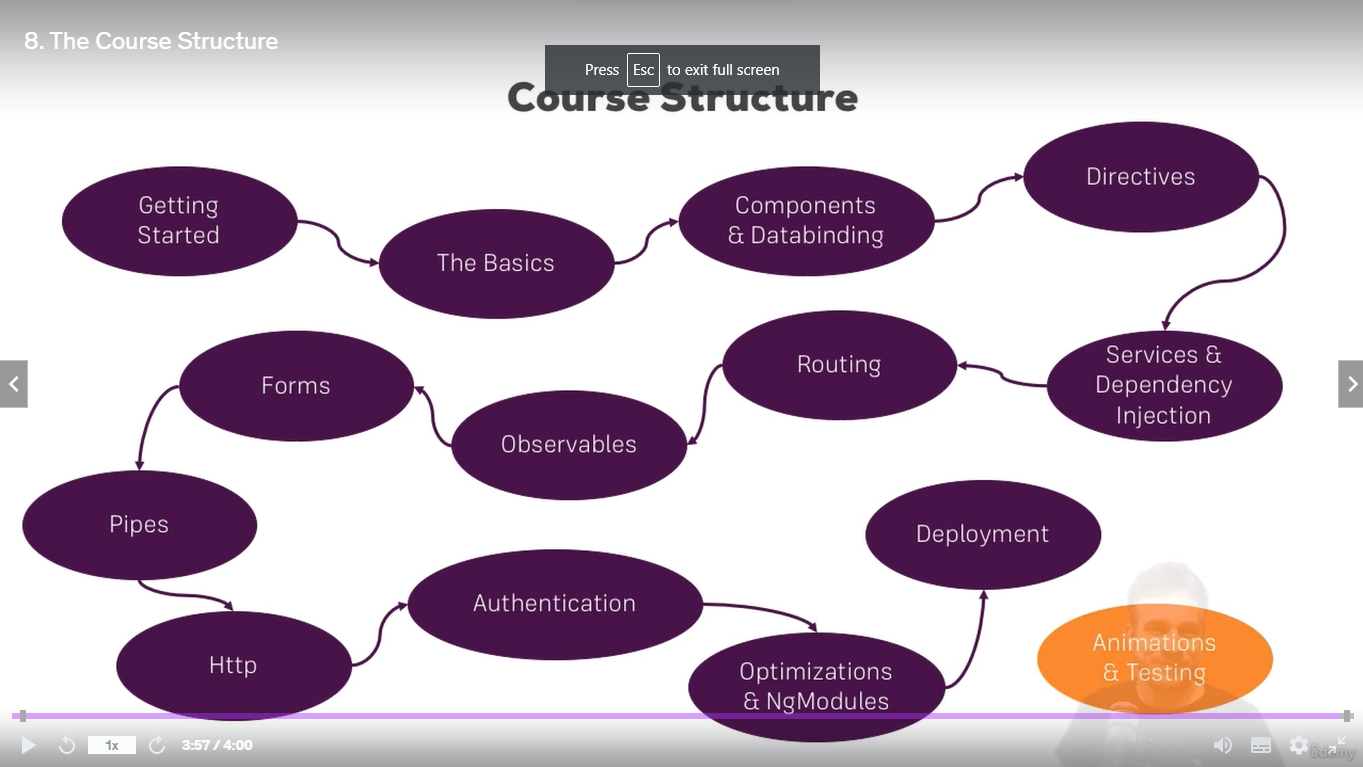
**Graphical user interface, website

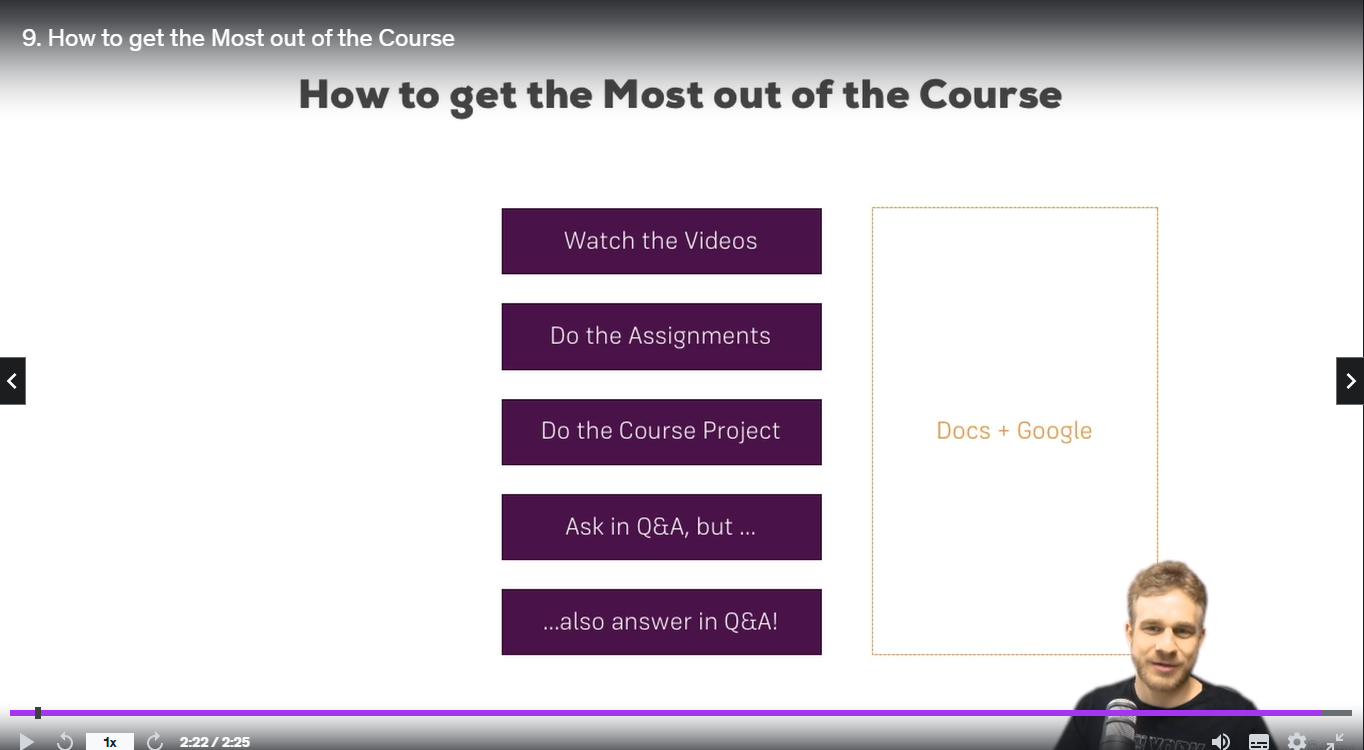
Description automatically generated**

Important folders In Angular Projects:

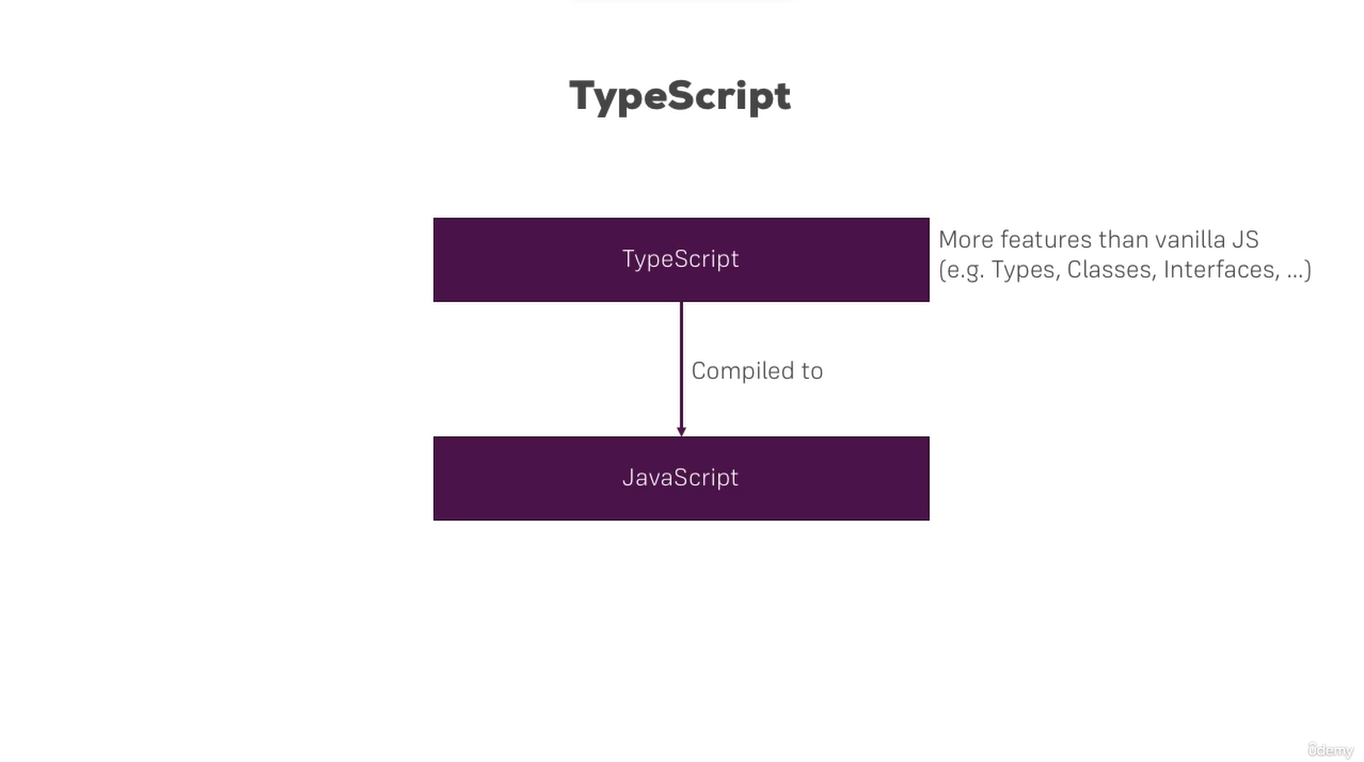
\*package.json: you can see Angular versions

\*src>app>





What is TypeScript?

****

A Basic Project setup using Bootstrap for Styling:

Open in terminal run the command to install Bootstrap to project:

npm install --save bootstrap@5 (@5 is the version of bootstrap installed locally).

Setting up Bootstrap CSS to project in Angular.json:

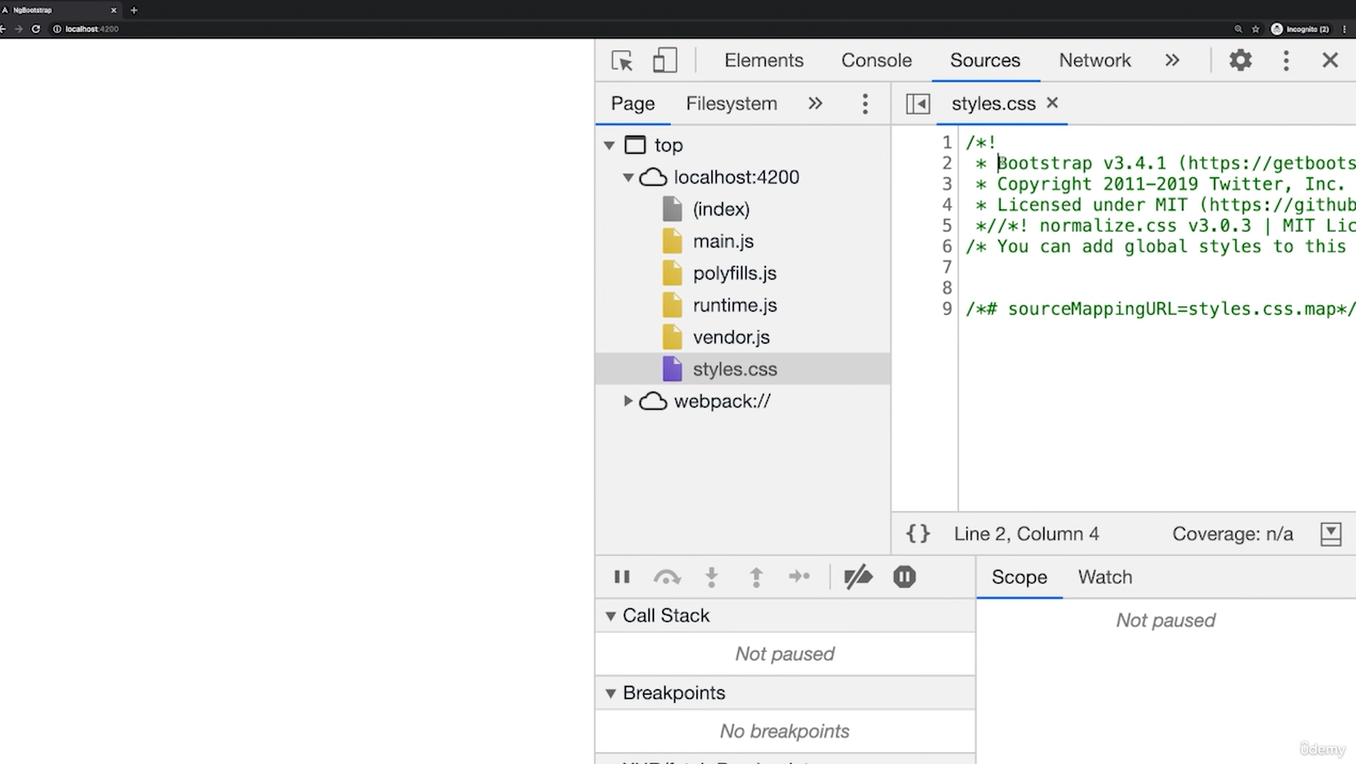
Open>Angular.json

Goto> architect {} section > “styles”: [

“node\_modules/bootstrap/dist/css/bootstrap.min.css”,

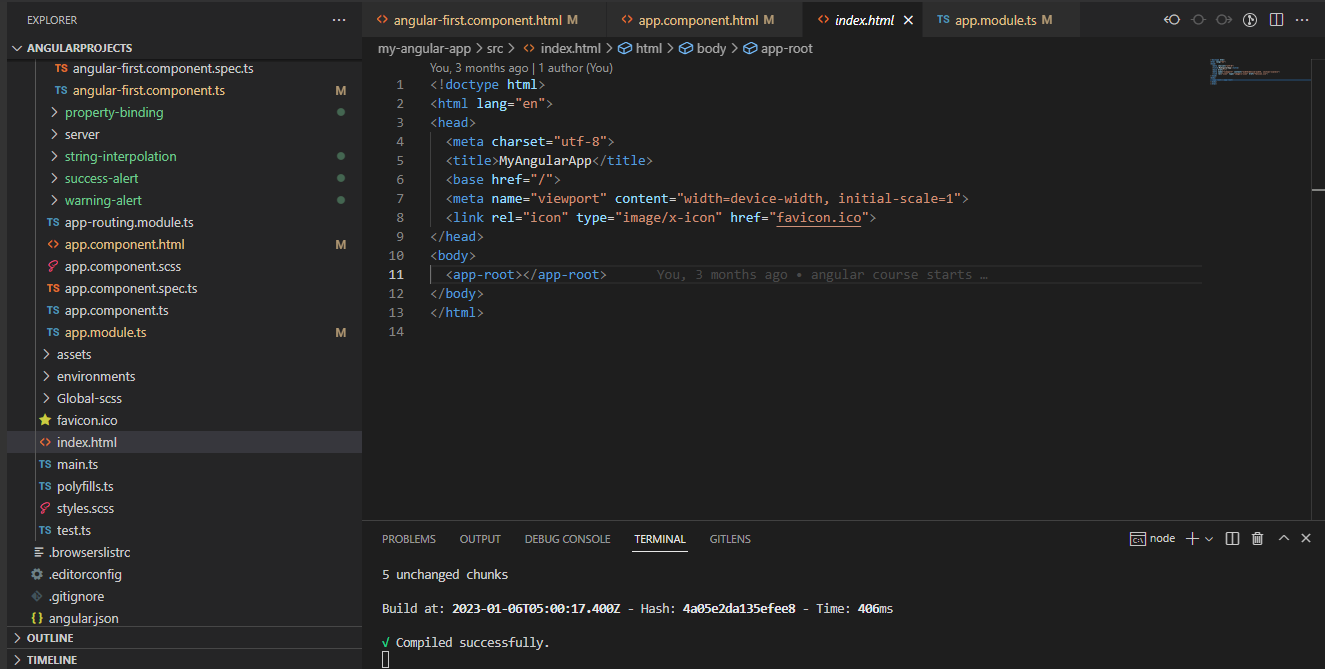
“src/styles.scss” (default style)

],

****

15.How Angular get loaded & started?

The index.html file is the main html file, we can see <app-root> that mentioned in app-component.ts file selector , redirecting the index to app component.



16.Components are Important,Creating a Component

Command: ng g c componentname (ng generate component componentname)

For creating the own component file:

Create a folder ‘server’

Create html file: server.component.html

Create ts file: server.component.ts

In ts file under component section

@Component({

  selector: 'app-server',

  templateUrl: './server.component.html',

  styleUrls: ['./server.component.scss']

})

After that adding this component in app.module.ts

import { ServerComponent } from './server/server.component';

@NgModule({

  declarations: [

    AppComponent,

    AngularFirstComponent,

    ServerComponent,

    WarningAlertComponent,

    SuccessAlertComponent,

    PropertyBindingComponent,

    StringInterpolationComponent

  ],

To checkout this element declares in app.component.html as

<app-server></app-server>

Working with component templates:

If we have two components like

<app-server> & <app-servers>

In <app-servers> ts file, we can change the html file in ts file

templateUrl: './server.component.html',

by replacing with

template: ‘<app-server></app-server><app-server></app-server>’

we can control any html file in a component we can configure by template.

template: `

<app-server></app-server>

<app-server></app-server>`, (with Javascript quotation also we can use)

25.What is Databinding

Output Data

Template (HTML)

Typescript code

(Business Logic)

String Interpolation ( { { data }} )

Property Binding ( [property] = “data” }

React to (user) Events

Event Binding ( (event)=”expressions”)

Combination of Both: Two-Way-Binding ([(ngModel)] = “data”)

Understanding Databinding: The communication between the ts code to html.

Databinding = Communication

26.String Interpolation: {{}}

Adding data Dynamically, without writing hardcoated.

Eg:HTML:

Server or {{‘Server’}} with Id {{ serverId }} and status is {{serverStatus}};

Server or {{‘Server’}} with Id {{ serverId }} and status is {{ getServerStatus () }};

TS: serverId: number =10;

serverStatus: string = “offline”;

Calling from a method also we can do the same

getServerStatus() {

return this.serverStatus;

}

27.Property Binding: [property]=””

|  |  |
| --- | --- |
| HTML | TS |
| <button [disabled]= allowNewServer>Add Server</button> | allowNewServer: false;  constructor() {  setTimeout(() => {  this.allowNewServer = true;  }, 2000);  } |

28.Property Binding Vs String Interpolation:

|  |  |
| --- | --- |
| HTML | TS |
| <button [disabled]= allowNewServer>Add Server</button>  <p>{{ allowNewServer }}</p> | allowNewServer: false;  constructor() {  setTimeout(() => {  this.allowNewServer = true;  }, 2000);  } |