**Angular**

**Syllabus – All topics:**

1. **Fundamentals :**
2. **Angular CLI :**
3. **NgModules :**
4. **Components :**

* **Class.**
* **Template.**
* **MetaData.**

1. **Template Syntax :**

* **Interpolation.**
* **Property Binding.**
* **Event Binding.**
* **Two Way Binding.**
* **Pipes.**
* **Template Reference Variable.**

1. **Directives :**
2. **Component.**
3. **Structural :**
   * **ngIf**
   * **ngSwitch**
   * **ngFor**
4. **Attribute :**
   * **NgClass**
   * **NgStyle**
5. **Component Styling :**

* **Class Binding - NgClass**
* **Style Binding – NgStyle**

1. **Component Interaction :**

* **Input**
* **Output**

1. **Lifecycle Methods.**
2. **Advanced :**
3. **Services :**

* **Dependency Injection.**
* **Providing Services.**

1. **Observables :**

* **Creating Observables.**
* **Subscribing to Observables.**
* **Executing Observables.**
* **Disposing Observables.**
* **Operators.**

1. **HTTP Client :**

* **Http Client.**
* **GET , POST , PUT & DELETE**
* **Typed Response.**
* **Error Handling.**
* **Interceptors.**

1. **Forms :**
2. **Template Driven Forms :**

* **Forms Module.**
* **NgForms.**
* **NgModel.**
* **NgModelGroup.**
* **Tracking State & Validity.**
* **Validation & Error messages.**
* **ngSubmit.**

1. **Reactive Form :**

* **Reactive Forms Module.**
* **FormControl.**
* **Form Group.**
* **setValue & patchValue**
* **FormBuilder service.**
* **Validator Functions.**
* **FormArray.**
* **ngSubmit.**

1. **Routing :**

* **Router Module.**
* **Configuring Routes.**
* **Router Outlet.**
* **Router Link.**
* **Wildcard Routes.**
* **Redirecting Routes.**
* **Route Parameters.**
* **Nested Routed.**
* **Relative Paths.**
* **Lazy Loading Routes.**
* **Route Guards.**

1. **Animation.**
2. **Ecosystems :**

* **State Management – ngRX.**
* **UI library – Angular Material.**
* **Server Side Rendering – Angular Universal.**
* **Testing :**
  + **Jasmine & Karma**
  + **Protractor**
* **Miscellaneous :**
  + **I18n.**
  + **Accessibility.**

**Ashok IT – Course Content**

**Part 1 :**

1. What is web application
2. Multi Page web application
3. Single Page web application (SPA)
4. Angular intro
5. Angular js vs Angular framework
6. Angular framework version history
7. Advantages of angular
8. Angular architecture

**Part 2 :**

1. Typescript intro
2. Advantages of typescript
3. Java script vs typescript
4. Env setup to work with typescript
5. Variables
6. Datatypes
7. OOPs
8. Class
9. Object
10. Constructor
11. Inheritance
12. Access modifier
13. Interfacess - contracts
14. Enumerations
15. Modules
16. Examples using typescript

**Part 3 :**

1. Revise angular introduction
2. Revise angular architecture
3. Environment setup (NPM , Angular CLI , Vs code IDE)
4. First application creation using angular
5. Understanding angular application folder structure
6. Live server to run angular apps
7. Components in angular ( main )
8. Modules
9. Data bindings
10. Login app using angular
11. Registration app using angular
12. Directives in angular
13. Pipes
14. Form based applications
15. Form validations : Template driven form , Reactive forms
16. Routings / Navigating

**Part 4 :**

1. http client – 2 ,3 rest api’s using spring boot
2. consuming backend rest api’s using angular
3. interview questions of angular.
4. What is web application

* The application which runs on the internet is called as the web application.

1. Multi Page web application

* 2 types of web applications , single page and multi-page
* **Multi page web application** means , for every client request , server is responding with a html webpage response.so for every client request , response will be a webpage so this is called as multi page web application. even though the content is same still its reloading all the content again.
* **Single Page Web Application means** ,Web page will be sent only for the first request , from second request , only data will be sent from servers to clients. It will not load entire page again , this is called as single page web application. Common content will be loaded only once and later data will be keep loading as per requirement.

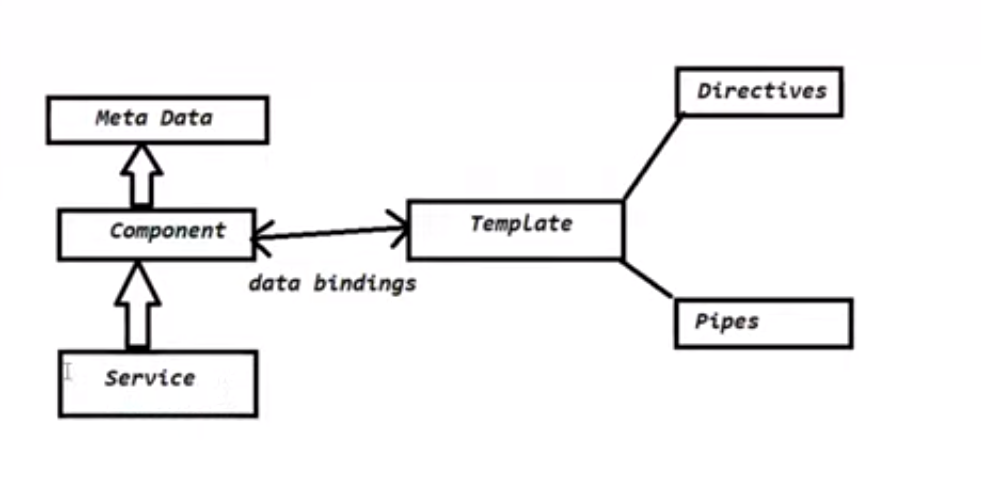
1. Why to use angular?

* It’s a client side framework.
* It is used to create web applications.
* Its mainly used to develop Single page applications.
* Angular have cross platform support, It supports for all platforms like mobile,Desktop.
* Angular is free and Open source.
* Angular have cross browser compatibility
* Two way data binding possible with angular..

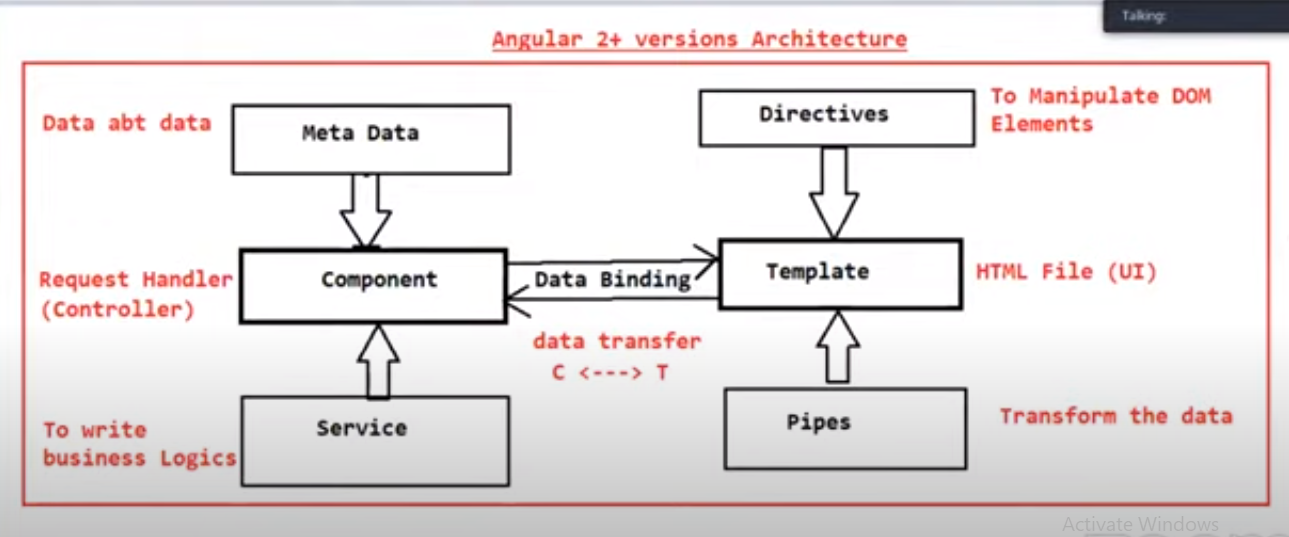
1. Angular framework version history :

* **Angular JS / Angular 1.0 :** it was the initial version of angular, developed by Google using JS , some performance issue was there as was developed by JS , type safety was not there.
* **Angular 2.0 :** now onwards it is called as **Angular** only , was developed with the help of **Type Script**., Google developers completely written Angular 2.0 using Type Script , so it was not the enhancement of Angular 1.0.
* **Type Script** supports **Data Types.**

1. Angular Architecture :



1. **Component** - data processing – Request Handler – Controller.
2. **Meta Data** – data about data – mapping between component and template – Annotations – Decorators.
3. **Service** - Bussiness logic , this class will be injected in component class. – Typescript class.
4. **Data Bindings** – data transfer between component and template.
5. **Template** – html – view file – contains presentation logic
6. **Directives**  - manipulate data of view file. – manipulate DOM elements.
7. **Pipes** – transform the data from one form to another .



**TypeScript**

* Typescript is developed using javascript
* It is a superset of javascript which adds

Data types + classes + interfaces etc,

* TypeScript is developed by Microsoft in 2012.
* **Features of TypeScript :**
* TypeScript is a general purpose programming language.
* It supports all features of javaScript and also providing some additional features.
* JS can be executed on browser directly , but typescript can’t be executed on browser directly , they should be converted into javascript.
* The process of converting typescript file into JS is called as “**Transpilation**”, its is converted using typescript compiler.

File.ts 🡪 tsc 🡪 file.js 🡪 browser 🡪 output.

* **TypeScript Versions :**
* Typescript 0.8 : 2012 – initial version.
* Typescript 4.x : 2020
* **Static Typong & Dynamic Typing :**
* Whenever we can fix a data type for the variable while declarations of the variable and we can’t change it’s datatype through program ,then its called as a “**static typing**”.

Int age = 20; // valid

Age = “20 years”; // in-valid

Ex : **c , c++ , java , c# .Net are** **Static Typed**

* If we can’t fix data type of variable while declaration and data type will be automatically taken by the runtime engine at the time of program execution then its is called as “**Dynamic Typing**”.

Var age;

Age = 20; // valid

Age = “20 years”; // valid

Ex : **JavaScript** and **Python** are examples for **Dynamic Typing.**

* Typescript supports “**optional static typing**”, so it supports both **static** and **dynamic** typing.
* Typescript maintains **Type Safety**.If we specifity data type while declaration and if we assign wrong type of value then it will show error. either it should static typing of dynamic typing at the declaration.

**Type Script Installation**

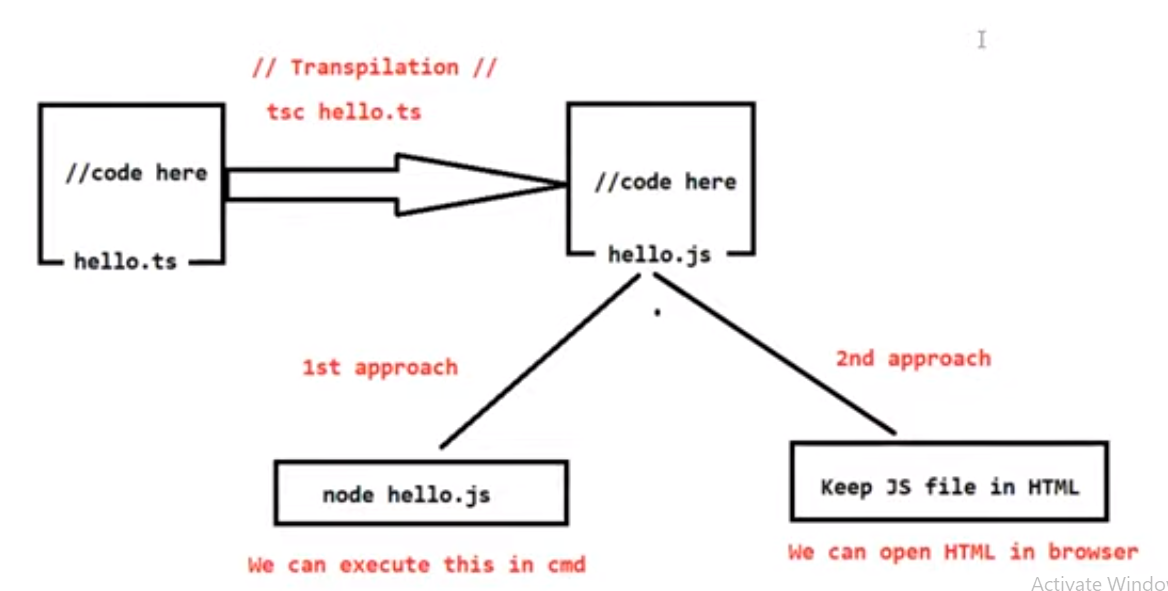
1. Install Node : **node -v**
2. Install TypeScript : **npm install –g typescript** (node package manager)
3. Typescript version : **tsc –v**
4. Typescript basic code :

* Create typescript file with .ts extension , and write below code

Var s:String = “Hello World!”;

Console.log(s);

1. Open cmd and compile typescript file.
   1. **tsc filename.ts**
   2. this will convert ts file into JS file.
2. Run JS file using node with cmd.
   1. **node filename.js**

****

1. In VS code install “Live Server” extension to run codes in IDE.
2. Running js file using html file.

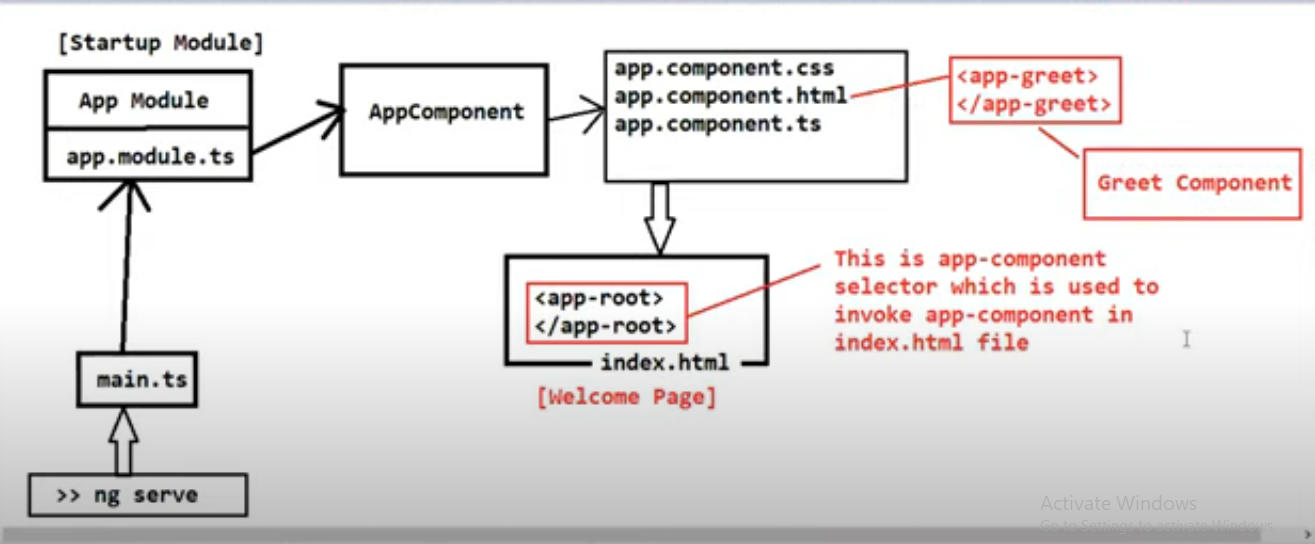
* Compile typescript file and get javascript file : **tsc filename.js**
* Create html file and inside that use javascript file as a Script in Head.
* <html>
* <head>
* <script src="HelloWorld.js"></script>
* </head>
* </html>
* Run html file using live server.

**Type Script Data Types :**

1. **Number :**

**First Angular App from visual studio :**

This way angular app will be started :



This way angular component will be loaded :

