**Angular JS:**

1. **Installation Angular Steps:**
2. Download and install Node : URL: <https://nodejs.org/en/> Note: after installation, verify node version $ node -v
3. Install type script $ npm install –g typescript $ tsc –v
4. Install angular CLI $ npm install @angular/cli –g $ ng v
5. Download and install VS code IDE URL: <https://code.visualstudio.com/download> Note: We are done with angular setup... let’s start building angular applications
6. Create angular application EX: $ ng new app1
7. Start the node server then we have use command in VS Code like: $ ng serve
8. **Angular Introduction:** 1. Angular is client side framework 2. Angular Framework developed by Google 3. Angular developed by using typescript 4. Angular is mainly used for SPA (single page app). 5. Angular supports multiple browsers. 6. Angular is free and open source. Note: Angular JS & Angular framework both are not same. 7. Angular JS developed using java script. (Angular 1.x). 8. Google identified some performance issues in Angular JS 1.x version then re-developed angular is typescript which is called as angular framework. Note: From 2.x version onwards it is called as angular framework.
9. **Angular Building Blocks:**  1. Components 2. Metadata 3. Template 4. Data Binding 5. Modules 6. Services 7. Dependency Injection 8. Directives 9. Pipes.

=> **Template** is a view page (html file)

=> **Component** represents small portion in web page

EX: header component menu component etc.…

Note: Every component will have its own template.

=> **Component** & **Template** relation will be represented using Meta data

=> **Data binding** is the process of sending data from component to template and vice versa.

=> Note: Angular support two way data binding.

=> **Service** is a typescript class which contains business logic.

=> **Directives** are used to manipulate DOM elements. EX: if – else, loops etc.

=> **Pipes** are used to transform the data which as you required format. Ex: lower case to uppercase .etc.

=> **Dependency Injection** means injecting one class to another class object EX: Inject service object into component.

=> **Modules** represents collection of components, Services, directives…

=> Note: Modules are used for logical grouping.

=> In angular application by default “app-component” will be created. It is called as parent component.

=>Note: app-component will have a selector, which is used to invoke the component.

=> ### app-component selector is ‘app-root’

=> app-component will access using its selector in index.html page.

=> index.html page is called as welcome page in angular application

=> When we run angular application, index.html page will be loaded and it will invoke app-component hence we will get response from app.component.html.

=> If we want create app component then you have to run below command.

Ex: ng generate component component name

(OR)

ng g c < component – name>.

1. **Data Binding:**

**=>** It is used to establish relation between “component” and template.

**=>** In angular we can perform data binding in 4 ways.

1. Interpolation

2. Property Binding

3. Event Binding

4. Two way Data Binding (property binding + event binding).

**1. Interpolation:**

=> It used to access component variable/ property in template.

Syntax: {{property Name}}

//java variable syntax:

String welcome =”welcome to ravi it”

// type script syntax

welcome: String=” welcome to ravi it”;

**2.** **Event Binding:**

=> It is used to pass notification from template to component

EX: Button Click

export class AppComponent {

title: ‘app3’;

message: string=”welcome to angular application”.

display () {

this.message= “welcome to event binding”;

}

<h1> {{message}} </h1>

<input type=”button” value=”change” (click) =” display ()”>

Whenever click on button then it will change the content of message like above example

**3. Two Way Data Binding:**

=> It is combination of both property binding and event binding.

=> When we change the value of the property in component then automatically it will be updated in template.

=> When we change the value in template then automatically it will be updated in component.

=> To work with two-way-data-binding we will use “ngModel” directive.

=> Two way data binding is applicable only for <input/> and <select/> tags

Note: Forms Module should be imported to work with two way data binding.