Angular Provides built-in features for:

* Animation
* HTTP service
* Materials

Some features:

* Auto-complete
* Navigation
* Toolbar
* Menus

The code is written in typescript and compiled. Then the compiler converts the typescript to JS. Then displayed in the browser.

AngularJS was based on MVC.

Angular 4 is based on the **component structure**.

Angular 4 uses TypeScript 2.2.

Component structure and TypeScript 2.2 makes faster and gives better Performance.

Folder structure:

E2e: integration testing purpose.

Node\_modules: npm package available.

Src: workplace.

File Structure:

Karma.conf.js: project requirements and unit testing via Protractor.

Package.json: Add required library and run the npm install command.

Declaration:

reference to the components.

By default, App component.

Imports:

Modules imported.

Providers:

References to the service created.

Bootstrap:

References the default component created.

Css, html, spec.ts, .ts

.ts:

The class definition for the component created.

Processing of HTML such as Connecting DB, Services, Routing, Interacting with other components.

Assets: images and js files are stored.

Index.html: contents here are shown in the browser.

App.component.ts 🡪 Selector: ‘app-root’ 🡪 display contents in app.component.html

Polyfill.ts 🡪 used for backward compatibility.

**Angular 4 – Components:**

Classes that interact with a .html file of the component.

1. Create components.
2. Link using Selector.
3. Selector in app.component.html (Parent) file.

**Angular 4 – Module:**

Place where the Components, directives, pipes, and services which are related to the application are **grouped**.

**Angular 4 – Data Binding:**

Declaration and definition of the variable at app.component.ts.

Use the variable at app.component.html.

/\*app.component.ts\*/

export class AppComponent {

title = 'Sudharsan';

months = ["---Select---", "January", "Feburary", "March", "April", "May",

"June", "July", "August", "September",

"October", "November", "December"];

isAvailable = true;

}

For loop:

<\*ngFor = “let i of variable”> {{i}} </> // at app.component.html file

Variable is referred from app.component.ts.

Egs:

<div>

Months:

<select>

<option \*ngFor="let i of months">

{{i}}

</option>

</select>

</div>

If condition:

<div>

<span \*ngIf = "isAvailable”> Since condition is true, content is displayed.</span>

Condition is false. But content is displayed

</div>

If else condition:

<div>

<span \*ngIf = "isAvailable; else condition1"> Since condition is true, content is displayed.</span>

<ng-template #condition1>Condition is false. But content is displayed.

</ng-template>

</div>

If then else condition:

<div>

<span \*ngIf = "isAvailable; then condition1 else condition2"></span>

<ng-template #condition1>Since condition is true, content is displayed.

</ng-template>

<ng-template #condition2>Condition is false. But content is displayed.

</ng-template>

</div>

**Angular 4 – Event Binding:**

Events: Keyboard movement, mouse click, mouseover.

Ex codes practiced: change and click

Change:

<div>

Months:

<select (change) = "changeMonth($event)">

<option \*ngFor="let i of months">

{{i}}

</option>

</select>

</div>

Export class appcomponent{

changeMonth(event){

alert("Month is selected from the dropdown");

console.log(event);

}

}

Click:

/\*app.component.ts\*/

clickFunction(event){

alert("Button is clicked");

console.log(event);

}

/\*app.component.html\*/

<button (click) = "clickFunction($event)">

click

</button>

**Angular 4 – Template**

<ng-template> in angular 4.

<template> in angular 2

<template> in HTML is used to hide the contents inside the tag and render them only through Javascript and not through HTML.

**Angular 4 – Directives**

@directive

Three types: Component, Structural, Attribute

The structural directive has \* before the name.

**Angular 4 – Pipes**

Convert to the required format.

**Angular 4 – Routing**

**Angular 4 – Services**

It helps to access methods and properties across other components in the entire project.

**Angular 4 – Http service**