Angular

Template ⇒ is a form of HTML that tells Angular how to render the component.

Selector ⇒ used to identify each component uniquely into the component tree, and it also define how the current component is represent in HTML.

@Directive ⇒ are classes that add additional behaviour to element in our angular applications.

Types of directive
- Component directive
- Structural directive - ngIf, ngFor, NgSwitch
- Attribute directive
  - ngclass, NgStyle, NgModel
- Custom directive.

Module in Angular ⇒ A place where we can group the components, directives, libe and services, which are related to the application.
for eg In case you are developing a website, the header, footer, left, center and the right section become part of a module

ngDoCheck, NgAfterContentInit, ngAfterContentChecked, ngAfterViewChecked, NgOnDestroy

Angular Framework

Angular is TypeScript-based open source front-end platform that makes it easy to build applications with in web application / mobile / desktop.

@NgModule ⇒ take a metadata object that describe how to compile a component's template and how to create an injector at runtime.

import {NgModule} from @angular/core.

app.component.ts
app.component.html
app.component.css

@Component ⇒ Component are the most basic UI building block of an Angular app which form a tree of angular component. These components are subset of directives. Unlike directive, components always have a template and only one component can be instantiated per an element in a Template.

ngOnChanges ⇒ When the value of a data bound property changes, than this method is called.

ngOnInit ⇒ This is called whenever the initialization of directive / component after angular first displays the data bound properties hap

Angular JS	Angular.
It is based on MVC architecture	It is based on service / controller.
It uses JavaScript to build the application.	Introduced the TypeScript to write the application.
Based on controllers concept	This is a component based UI approach.
Not mobile friendly framework	Developed considering mobile platform
Difficult in seo friendly application development.	Ease to create SEO friendly applications
Data binding allow to define communication between a component and the DOM. 3 categories (1) Interpolation ⇒ Adds the value of a property from the component. Name : {{user.name}} property binding = [property] = "value" : The value is passed from the component to the specified property of simple HTML attributes. (2) From the DOM to the component: Event binding : (event) = "function" : When a specific DOM event happens (eg click, change, keyup, keydown) etc.	Two way binding ⇒ [(ngModel)] = "value".. allow to have data flow both ~~side~~ ways.
	Difference b/w constructor and ngOnInit constructor = initialize class member. ngOnInit = is a place to put the code that we need to execute at very first as soon as the class is instantiated.
	Service ⇒ A service is used when a common functionality needs to be provided to various modules.

Pipe ⇒ A pipe takes in data as input and transforms it to a desired output.

{{ birthday || date }}

birthday = new Date(1987, 6, 18);

parameterized

{{ birthday | date : dd/MM/yyyy }}

birthday = new Date (1987, 6, 18).

Angular Router.

~~is a~~ in which navigation happens from one view to the next as user perform Application tasks.

Router Outlet ⇒ Display the components for that outlet.

<router-outlet></router-outlet>

RouterLink ⇒ router control over those elements.

Observable = is a features that provide support for delivering messages. b/w different part of ~~the~~ our single page application. It is an interface to handle a variety of common asynchronous operation.

HTTP module use observable to handle HTTP request and response.

subscribe := is a method in angular that connects the observer to observable events.

Synchronous ⇒ Synchronous code runs in sequence. This means that each operation must wait for the previous one to complete before executing. First it complete one and then it move to next step.

Asynchronous = code runs in parallel. This means that an ~~element~~ operation can occur while another one is still being processed.

Form Builder ⇒ The FormBuilder provides syntactic sugar that shortens creating instances of a FormControl, FormGroup or FormArray. It reduces the amount of bailerplate needed to build complex forms.

FormGroup ⇒ aggregates the value of each child FormControl into one object with each control name as the key.

FormControl ⇒ classes that can holds both the data value and the validation information of any form element. reactive form should be bounded by a formControl.

Template Driven forms based on template directive.

Reactive Form = are programmatically at the level of the component class.

Guards ⇒ allow to grant or remove access to certain parts of the navigation.

Interceptors = Interceptors allow us to interrupt incoming and outgoing HTTP request using the HttpClient. They can handle both HttpRequest as well as HttpResponse.

Observer in RxJS ⇒ An observer is a consumer of value delivered by an observable. Observer are simple set of call backs, one for each type of notification delivered by the observable: next; error or complete;

Observables

while observable handle a sequence of asynchronous event over a period of time.

Are lazy

They are not executed until we subscribe to them using the subscribe method.

next, error, complete.

Promise

Promise deal with one asynchronous event at a time.

Are not lazy execute immediately after creation.

resolve, & refuse.

var	let	const
The scope of var variable is fonutional scope	The scope of a let variable is block scope.	The scope of a const variable is block scope.
It can be updated and re-declared into the scope.	It can be updated but can not be re-declared into the scope.	It can not be updated on re-declared into the scope.
It can be declared without initialization	It can be declared without initializatio	It can not be declared without intialization
It can be accessed without initializatio as its default value is "undefined".	It can not be accessed without initialization as it return an error.	It can not be accessed without initialization as it can not be declared without intializa

Advantage of TypeScript over Javascript

- TypeScript always highlights errors at compilation time. during the time of development where as JavaScript paints out error at runtime.
- Typescript supports strongly typed or static typing, whereas this is not in Javascript.
- Typescript runs on any browser or Javascript engine.
- Great tooling supports with IntelliSense which provides active hints as the code is added.
- It has a namespace concept by defining a module.

Disadvantage of Typescript over Javascript.

- Typescript takes a long time to compile the code.
- Typescript does not support abstract class.
- If we run the typescript applicatio in ~~Java~~ Browser, a compilation step is required to transform TypeScript into Javascript.

Features of ES6.

- The let keyword.
  const keyword.
  Arrow function.
  For/Of
  Map objects
  Set objects
  classes
  Promises
  Symbol
  Default Parameters
  Function Rest Parameters

Java Script	Tyhe Script
1. It does't support stronly tyhed or static tyhing.	It support strongly tyhed or static tyhing feature.
2. Netscape developed on 1995.	Andros Hejlsberg developed in 2012
3. JavaScript source file is in .JS extension.	Tyhescript source file in .tS ext.
4. It is directly ron on the browser.	It is not directly ron on browser.
5. It is just a scripting language.	5. It support OOP concept like duss, interfaces, inheritance, generics etc.
6. It doesn't support optional parameters.	6. It support optional parameters
7. It is interpreted language that why it highlighted the errors at runtime.	7. It compile the code and highlighted errors during the development time.
8. Javascript doesn't support modules.	8. Tyhescript give support for modubs.
9. In this, number, string are the object.	9. In this nomber, string are the interface.
10. JavaScript doesn't support generics.	10. Tyhescript support generics