**What is Angular and Why Angular?**

Angular (2/4/5) is not a programming language like JavaScript or Typescript. Angular is a front-end or client-side Framework, which needs a [programming](https://www.greycampus.com/programming) language like Typescript (developed by Microsoft).

**What do you mean “Angular is a client-side framework”?**

Angular runs on client side or user’s browser. It will not run in server. There are an angular universal runs Angular in server. But primarily it was developed in client side.

It was developed by techie giant Google for SPA (Single Page Application)

**What’s new in Angular 5?**

* Events like ActivationStart and ActivationEnd are introduced in Router
* Angular 5 supports Typescript version 2.4
* Ahead of Time (AOT) is updated to be on by default
* Angular 5 supports RxJS 5.5 which has new features like Pipeable Operators
* A build tool to make the js bundles (files) lighter

**Name the building blocks of Angular?**

* Modules
* Component
* Templates
* Directives
* Data Binding
* Services
* Dependency Injection
* Routing

### What is Transpiling in Angular?

The process of converting typescript to JavaScript using js compiler or any other compiler is called transpiling.

Typescript is used to write code in angular, the code internally converts into JavaScript.

### Which of the Angular life cycle component execution happens when a data-bound input value updates?

ngOnChanges is the life hooks gets executed if there is any change in input data.

### Differentiate between Components and Directives in Angular 5.

In Angular we have

* Components
* Structural Directive
* Attribute Directive

# Component:

It is a type of attribute with template and styles.It is famous attribute from angular 2.

The structure of it will be as below.

@Component({

Selector:”app”

directives:[Custom directive]

})

Use the directive as below in view

<app></app>

# Structural Directives:

Like \*ngFor and \*ngIf or some other ng modules will be used to update the dom Layout by adding HTML Element and removing DOM elements.

# Attribute directives

It is used to apply some styles to existing HTML elements by applying ngStyle.

import {Directive, ElementRef, Renderer, Input} from 'angular2/core';

@Directive({

selector: '[Icheck]',

})

export class RadioCheckbox{

custom logic here ...,,,

}

<span Icheck>HEllo Directive</span>

Components have their own view (HTML and styles). Directives are just "behavior" added to existing elements and components.

Because of that there can only be one component on a host element, but multiple directives.

### What is the use of @Input and @Output?

We will use these parameters to interact with two components. We use @Input in Child component when we are passing data from Parent to Child and @Output will be used to pass data from child to Parent.

Suppose we have 2 components as Parent and Child

import { Component, Input } from '@angular/core';

import { Hero } from './hero';

@Component({

selector: 'app-hero-child',

template: `

<h3>{{hero.name}} says:</h3>

<p>I, {{hero.name}}, am at your service, {{masterName}}.</p>

`

})

export *class* HeroChildComponent {

@Input() hero: Hero;

@Input('master') masterName: *string*;

}

import { Component } from '@angular/core';

import { HEROES } from './hero';

@Component({

selector: 'app-hero-parent',

template: `

<h2>{{master}} controls {{heroes.length}} heroes</h2>

<app-hero-child \*ngFor="let hero of heroes"

[hero]="hero"

[master]="master">

</app-hero-child>

`

})

export *class* HeroParentComponent {

heroes = HEROES;

master = 'Master';

}

### What is ng-content Directive?

If we have some data between the custom tags and text in HTML.it will not work. But in angular it will be possible by passing data in ng-content tag as below.

<ng-content>{{data}}<ng-content/>

### What does a router.navigate do?

It will be use to navigate from one component to another using the below syntax.

this.httpRoute.navigate(["./path"])

### What is ViewEncapsulation?

“ViewEncapsulation” decides whether the styles defined in a component can affect the entire application or not. There are three ways to do this in Angular:

Emulated: styles from other HTML spread to the component.

Native: styles from other HTML do not spread to the component.

None: styles defined in a component are visible to all components.

### What are Services in Angular and what command is used to create a service?

These are the used not to repeat the same code. We can use in multiple components by importing the file and function.

Using the below command line we create service.

Ng g service Name or ng g s Service Name.

### What is Dependency Injection in Angular 4?

When a component depends on another component then the dependency will be injected/provided during runtime.

### What is Dependency Injection in Angular 4?

When a component is dependent on another component the dependency is injected/provided during runtime.

### What is Routing in Angular 5?

Routing helps a user in navigating to different pages using links.

### How to handle Events in Angular 5?

Any activity (button click, mouse click, mouse hover, mouse move, etc) of a user on a frontend/web screen is termed as an event. Such events are passed from the view (.HTML) page to a typescript component (.ts).

### What is a RouterOutlet?

### RouterOutlet is a substitution for templates rendering the components. In other words, it represents or renders the components on a template at a particular location.

### *Explain the usage of {{}}?*

The set of brackets {{}} when used with an HTML tag, represent data from a component. For example, on a HTML page which has <h1>{{variableName}}</h1>, here the ‘variableName’ is actually typescript (component) data representing its value on the template; i.e., HTML. This entire concept is called String Interpolation.

### *In how many ways the Data Binding can be done?*

Data Binding happens between the HTML (template) and typescript (component). Data binding can be done in 3 ways:

(i) Property Binding (ii) Event Binding (iii) Two-Way Data Binding.

### *What is the sequence of Angular Lifecycle Hooks?*

OnChange()  -  OnInit()  -  DoCheck()  -  AfterContentInit()  -  AfterContentChecked()  -  AfterViewInit()  -  AfterViewChecked()  -  OnDestroy().

### *What is the purpose of using package.json in the angular project?*

With the existence of package.json, it will be easy to manage the dependencies of the project. If we are using typescript in the angular project then we can mention the typescript package and version of typescript in package.json.

### *How is SPA (Single Page Application) technology different from the traditional web technology?*

In traditional web technology, the client requests for a web page (HTML/JSP/asp) and the server sends the resource (or HTML page), and the client again requests for another page and the server responds with another resource. The problem here is a lot of time is consumed in the requesting/responding or due to a lot of reloading. Whereas, in the SPA technology, we maintain only one page (index.HTML) even though the URL keeps on changing.

### *What is Component in Angular Terminology?*

A web page in Angular has many components involved in it. A Component is basically a block in which the data can be displayed on HTML using some logic usually written in typescript.

### *What are ngModel and how do we represent it?*

ngModel is a directive which can be applied on a text field. This a two-way data binding. ngModel is represented by [()]

### *What does a Subscribe method do in Angular 4?*

It is a method which is subscribed to an observable. Whenever the subscribe method is called, an independent execution of the observable happens.

### *Differentiate between Observables and Promises.*

Observables are lazy, which means nothing happens until a subscription is made. Whereas Promises are eager; which means as soon as a promise is created, the execution takes place. Observable is a stream in which passing of zero or more events is possible and the callback is called for each event. Whereas, promise handles a single event.icon

### 

### *What is an AsyncPipe in Angular?*

When an observable or promise returns something, we use a temporary property to hold the content. Later, we bind the same content to the template. With the usage of AsyncPipe, the promise or observable can be directly used in a template and a temporary property is not required.

### *Explain Authentication and Authorization.*

Authentication: The user login credentials are passed to an authenticate API (on the server). On the server side validation of the credentials happens and a JSON Web Token (JWT) is returned. JWT is a JSON object that has some information or attributes about the current user.  Once the JWT is given to the client, the client or the user will be identified with that JWT.

Authorization: After logging in successfully, the authenticated or genuine user does not have access to everything. The user is not authorized to access someone else’s data,  he/she is authorized to access some data.

### *What is AOT Compilation?*

Every angular application gets compiled internally. The angular compiler takes javascript code, compiles it and produces javascript code again. Ahead-of-Time Compilation does not happen every time or for every user, as is the case with Just-In-Time (JIT) Compilation.

### *What is Redux?*

It is a library which helps us maintain the state of the application. Redux is not required in applications that are simple with the simple data flow, it is used in Single Page Applications that have complex data flow.

### *What are Pipes?*

This feature is used to change the output on the template; something like changing the string into uppercase and displaying it on the template. It can also change Date format accordingly.

### *Differentiate between ng-Class and ng-Style.*

In ng-Class, loading of CSS class is possible; whereas, in ng-Style we can set the CSS style.

### *Why Typescript with Angular?*

Typescript is a superset of Javascript. Earlier, Javascript was the only client side language supported by all browsers. But, the problem with Javascript is, it is not a pure Object Oriented Programming Language. The code written in JS without following patterns like Prototype Pattern, becomes messy and finally leading to difficulties in maintainability and reusability. Instead of learning concepts (like patterns) to maintain code, programmers prefer to maintain the code in a OOP approach and is made avilable with a programming language like Typescript was thus developed by Microsoft in a way that it can work as Javascript and also offer what javascript cannot ie;

* pure OOPS as Typescript offers concepts like Generics, Interfaces and Types (a Static Typed Language) which makes it is easier to catch incorrect data types passing to variables.
* TS provides flexibility to programmers experienced in java, .net as it offers encapsulation through classes and interfaces.
* JS version ES5 offers features like Constructor Function, Dynamic Types, Prototypes. The next version of Javascript ie ES6 introduced new feature like Class keyword but not supported by many browsers.
* TS offers Arrow Functions (=>) which is an ES6 feature not supported by many browsers directly but when used in TS, gets compiled into JS ES5 and runs in any browser.
* TS is not the only alternative to JS, we have CoffeScript, Dart(Google).
* Finally, it is like, TS makes life easier when compared to JS.