**Angular Basics**

**Components:**

* Components are the most basic UI building block of an Angular app.An Angular app contains a tree of Angular components.
* Angular is always associated with a template only one component can be instantiated per an element in a template.

**How to create a Component in Angular:**

* Generate Component:

ng generate component componentName

* Import component decorator

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

* Use decorator and add metadata:

@Component ({

selector:’my-component’,

template:

`<div>my template</div>`

})

export class myComponent {}

**MODULE:**

* Module is a collection of directives,services,controllers,filters

Anguler.module is used to configure the injecter

* Every angular application has the at least one angular module

**Two Way Binding:**

* Data binding in angular is synchronization between the model and the view.
* When data model changes the view reflect the changes , when the data in view changes the model is also updated.
* This will happens immediately and automatically, this will make sure that view and model is updated all the time.

Ex:

<input type=”text” [(ngModel)]=’name’>

Name is: {{name}}

OUTPUT:

It will bind the same data given in text box to Name property.

**Routing:**

* It allows developers to build Single Page Applications (SAP) with multiple components.
* It allows to use the browser's URL to navigate between Angular components in the same way you can use the usual server side navigation.

Importing Router:

import {RouterModule,Routes} from ‘@angular/router’;

**TEMPLATES:**

* HTML is a language of angular template .All most all HTML syntax are valid template syntax.
* It will eliminate the script injection attacks (XSS).
* The template expression will produce a value .Angular executes the expression and assigns it to a property of a binding target, the target might be a HTML element, component or a directive.
* templateUrl: The module relative address of this component HTML template.

**METADATA:**

* The @component decorator identifies the class immediately below it as a component class, specifies its metadata.
* The metadata for a component tells angular where to get the major building blocks that it needs to create and present the component and its view.
* In addition to containing or pointing to the template, the @component metadata configures.

**DIRECTIVES:**

There are three types of directives in angular

* The component-Directive with a template.
* Structural Directive it will change the DOM layout by adding and removing elements in DOM.
* Attribute Directive it will change the appearance or behaviour of the element ,component or another directive.

**PIPES:**

* pipes are a useful feature in Angular.
* They are a simple way to transform values in an Angulartemplate.
* There are some built in pipes, but it also allow to build own pipes. A pipes takes in a value or values and then returns a value.

Usage ideas of pipes:

1.Return default values

2.Debounce Input

3.Get the position of an element

4.Track user Input

**SERVICES:**

* These are also called as singletons, which are the objects that are instantiated only once per application services are injected using Dependency Injection Mechanism of AngularJs
* Service provide a method for us to keep data around the lifetime of the app and communicate across controllers in a consistent manner.
* A service is a class with a narrow and well-defined purpose.

**DEPENDENCEY INJECTION:**

* Angular has its own DI framework which is used to design of angular application to increase the efficiency and modularity.
* DI is a coding pattern in which a class ask for dependency from external sources rather than creating them itself.
* Creating a Injectable Service class:

ng g service servicename.

import {Injectable} from ‘@angular/core’;

@Injectable({

})

**LIFE CYCLE HOOKS:**

* A component has a life cycle managed by Angular.
* The angular will creates it, renders it, creates and renders its children, checks it when its data-bound property changes and destroy it before removing it from the DOM.
* The directive and component instance has a lifecycle as angular creates, updates,and destroy them.
* **Life cycle Sequence:**

ngOnChanges()

ngOnInt()

ngDoCheck()

ngAfterContentChecked()

ngAfterContentInt()

ngAfterViewChechked()

ngAfterViewInit()

ngOnDestroy()

**HTTP:**

* Http Service will help us fetch external data, post to it, and other operations, We need to import the http module to make use of the http service.

import{HttpModule} from ‘@angular/http’;

imports[HttpModule],

* For fetching the data we need to use the GetAPI:

this.http.get();

it will take the url to be fetched.

**RxJS:**

RxJS or Reactive Extensions for JavaScript is a library for transforming, composing, and querying streams of data.

We mean all kinds of data too, from simple arrays of values, to series of events (unfortunate or otherwise), to complex flows of data.

**Observable from promises:**

import {fromPromises} from ‘rxjs’;  
**Observable from counter**: It will wait for some given time to do an particular event.

import {interval} from ‘rxjs’;

**Using AJAX:**

import {ajax} from ‘rxjs/ajax’;

const apiInfo=ajax(`api/data`);

apiData.subscribe()(res=> console.log (res.status,res.response));