Introduction

1. Angular 2 is an open source JavaScript framework to build web applications in HTML and JavaScript.

or

Angular JS is an open source framework built over JavaScript

1. It was built by the developers at Google.
2. This framework was used to overcome obstacles encountered while working with Single Page applications. Also, testing was considered as a key aspect while building the framework. It was ensured that the framework could be easily tested
3. The initial release of the framework was in October 2010.

Features of Angular 2

1. **TypeScript** − The newer version of Angular is based on TypeScript. This is a superset of JavaScript and is maintained by Microsoft.
2. **Services** − Services are a set of code that can be shared by different components of an application.

## Components of Angular 2

1. **Modules, Component** , **Templates**− This is used to define the views of an Angular JS application, **Metadata, Service**
2. The official site for Angular is <https://angular.io/>
3. Npm(node package manager)🡺 Angular JS as a framework has dependencies on other components. And npm can be used to download these dependencies and attach them to your project.
4. The official site for npm is <https://www.npmjs.com/>

# **Angular 2 - Hello World**

npm install

npm start

1. **Note** − Visual Studio Code will automatically compile all your files and create JavaScript files for all your typescript files.

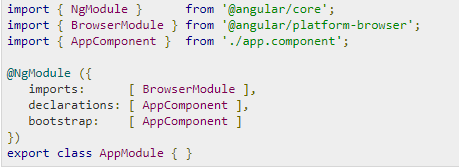
## Deployment

1. The deployment which we had used is quite different for the protytype

# **Angular 2 - Modules**

1. Modules are used in Angular JS to put logical boundaries in your application.

**Hence, instead of coding everything into one application, you can instead build everything into separate modules to separate the functionality of your application**

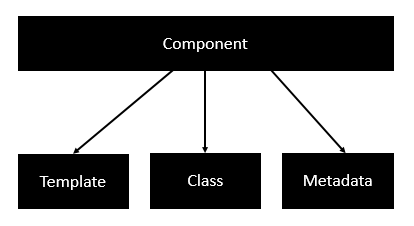
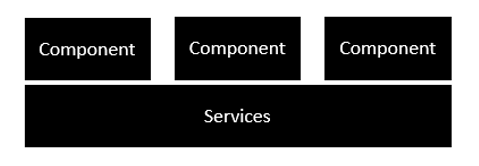
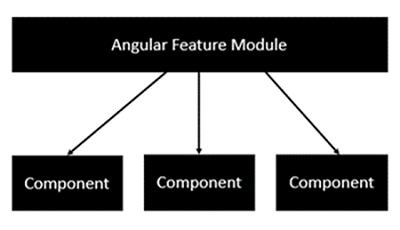
1. app.module.ts is the root module
2. 

Explanation🡺The NgModule decorator is used to later on define the imports, declarations, and bootstrapping options.

The BrowserModule is required by default for any web based angular application.

The bootstrap option tells Angular which Component to bootstrap in the application.

# **Angular 2 - Architecture**

1. 
2. 

Each Angular feature module can then have multiple components to separate the functionality.

# **Angular 2 - Components**

1. Component consists of the following🡺
2. **Template** − This is used to render the view for the application
3. **Class** − This is like a class defined in any language such as C. This contains properties and methods. This has the code which is used to support the view. It is defined in TypeScript.
4. **Metadata** − This has the extra data defined for the Angular class. It is defined with a **decorator.**
5. Example🡺

export class AppComponent {

appTitle: string = 'Welcome';}

# **Angular 2 - Templates**

template: **'**

<div>

<h1>{{appTitle}}</h1>

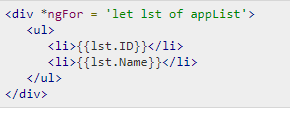
<div>To Tutorials Point</div>

</div>

**'**

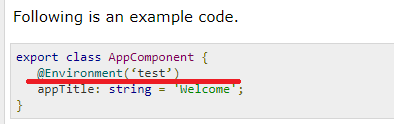
1. here are other ways to define a template and that can be done via the templateURL command🡺**templateURL:viewname.component.html**

# **Angular 2 - Directives**

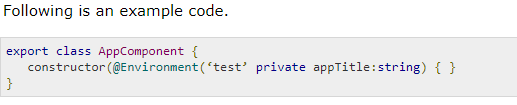
1. Ms 🡺 is used to extend/customize the View
2. Angular 2 has the following directives that get called as part of the BrowserModule module🡺ngif ,ngFor
3. <div \*ngIf = 'appStatus'>{{appTitle}} Tutorialspoint </div>
4. \*ngFor = 'let variable of variablelist'
5. 

# **Angular 2 - Metadata**

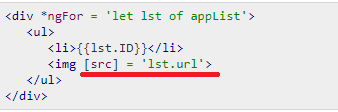
1. **Annotations** − **These are decorators at the class level.** This is an array and an example is @Component and @Routes decorator
2. **propMetadata** − This is the metadata which is applied to the properties of the class.



1. **Parameters** − This is set by the decorators at the constructor level.

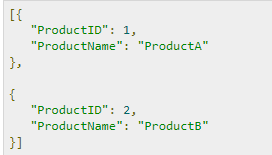


# **Angular 2 - Data Binding**

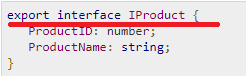
1. 

# **Angular 2 - CRUD Operations Using HTTP**

2. Step1🡺products.json



1. Create a  products.ts



1. Create App.module.ts
2. Create products.service.ts



Explanation🡺the Observable framework is used to detect any changes in the http response which can then be sent back to the main application.

Once we get the data from the data source, we then use the JSON.stringify(data) command to send the data to the console in the browser.

1. 

# **Angular 2 - Error Handling**

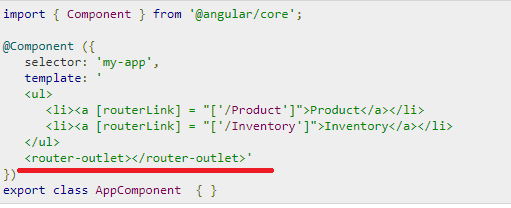
1. Angular 2 applications have the option of error handling. This is done by including the **ReactJS** catch library and then using the catch function.



Explanation🡺The catch function contains a link to the Error Handler function.In the error handler function, we send the error to the console. We also throw the error back to the main program so that the execution can continue.

# **Angular 2 – Routing**

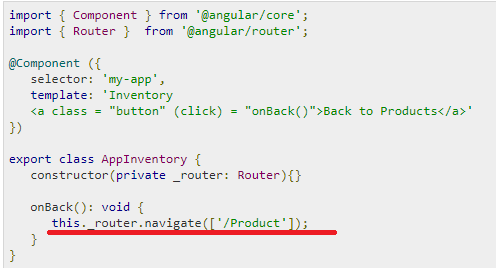




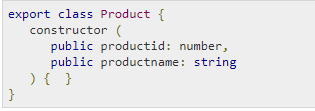
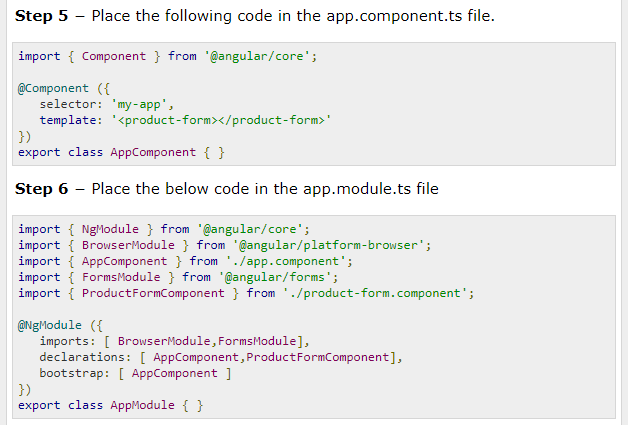
<router-outlet></router-outlet> is the placeholder to render the component based on which option the user chooses.

# **Angular 2 - Navigation**

1. In Angular 2, it is also possible to carry out manual navigation. As shown below,



# **Angular 2 - Forms**

2. Create a file called **products.ts** file. And Place the following code in the file🡺
3. 
4. 
5. 

# **Angular 2 - CLI**

1. Command Line Interface (CLI) can be used to create our Angular JS application.
2. The official site for Angular CLI is <https://cli.angular.io/>

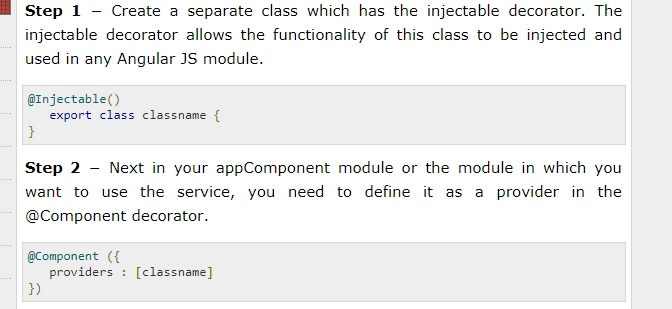
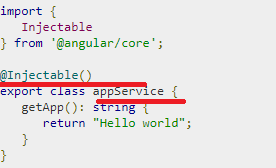
The first step is to install the CLI. 🡺npm install –g angular-cli

1. Now, create a new folder called angularCLI in any directory and issue the above command.

## Creating a Project Using CLI

1. ng new Project\_name
2. To run the project, you need to issue the following command 🡺”ng serve”

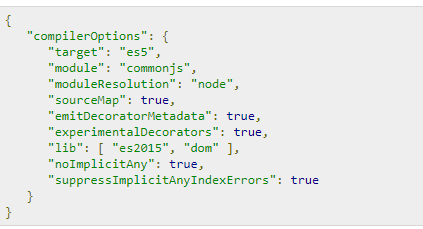
# **Angular 2 - Dependency Injection**

1. 
2. Example
3. 
4. 

# **Angular 2 - Advanced Configuration**

1. Lets see them below

## tsconfig.json

1. This file is used to give the options about TypeScript used for the Angular JS project
2. 

Explanation🡺

The target for the compilation is es5 and that is because most browsers can only understand ES5 typescript.

The sourceMap option is used to generate Map files, which are useful when debugging. Hence, during development it is good to keep this option as true.

The "emitDecoratorMetadata": true and "experimentalDecorators": true is required for Angular JS decorators. If not in place, Angular JS application will not compile.

## package.json

1. Notes

## systemjs.config.json

1. This file contains the system files required for Angular JS application. This loads all the necessary script files without the need to add a script tag to the html pages. The typical files will have the following code.
2. 

Explanation🡺

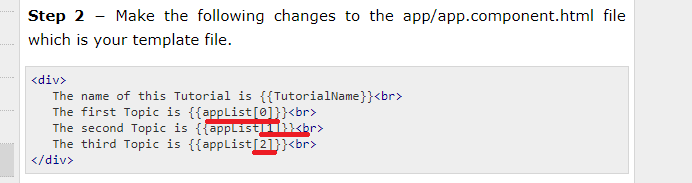
**'npm:': 'node\_modules/'** tells the location in our project where all the npm modules are located.

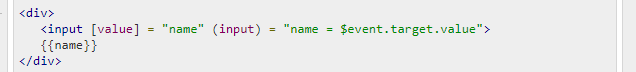
The mapping of **app: 'app'** tells the folder where all our applications files are loaded.

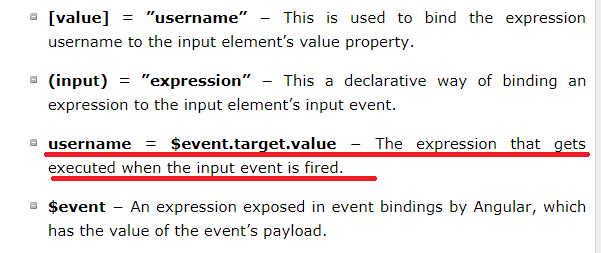
# **Angular 2 - Third Party Controls**

# **Angular 2 - Data Display**

1. 



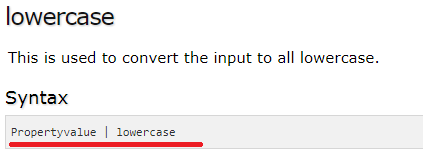
1. **Another simple example, which is binding on the fly is the use of the input html tag. It just displays the data as the data is being typed in the html tag.as shown below**
2. 



# **Angular 2 - Handling Events**

1. Simple click even see the notes itself

# **Angular 2 - Transforming Data**

1. Angular 2 has a lot of filters and pipes that can be used to transform data.
2. 
3. Upper case

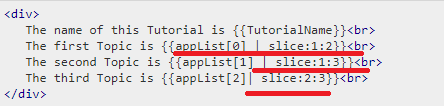
### **Syntax**

Propertyvalue | uppercase

## slice

1. This is used to slice a piece of data from the input string🡺

Propertyvalue | slice:start:end

1. Example🡺

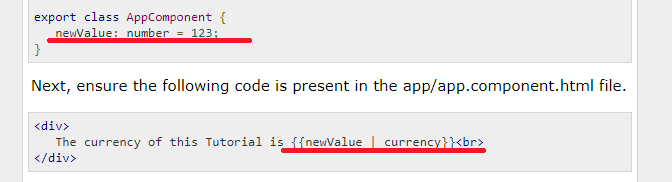
## date

This is used to convert the input string to date format. 🡺Propertyvalue | date:”dateformat”

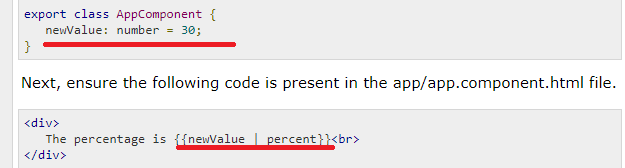
1. 

## currency

This is used to convert the input string to currency format. 🡺Propertyvalue | currency

1. 

## percentage

1. This is used to convert the input string to percentage format.
2. 
3. Similarly we have 🡺

* **minIntegerDigits** − This is the minimum number of Integer digits.
* **minFractionDigits** − This is the minimum number of fraction digits.
* **maxFractionDigits** − This is the maximum number of fraction digits.

# **Angular 2 - Custom Pipes**