Angular

(formerly Angular 4)

One framework.
Mobile & desktop.

Agenda

- ❑ Introduction of Angular.
- ❑ Features of Angular.
- ❑ Development environment setup.
- ❑ Project setup
- ❑ Structure of Angular project.
- ❑ How to install packages.
- ❑ What is TypeScript .
- ❑ How Angular application start.
- ❑ Components.
- ❑ How to create component.
- ❑ Styling component.
- ❑ Component selector.
- ❑ Databinding

Angular

- Angular is an open source JavaScript framework that is used to build single page based web applications.
- Developed by Google
- Release Date - March 2017
- One framework. Mobile & Desktop.

Features of Angular

Cross Platform

Speed and Performance

Productivity

Development

Cross Platform

❑ Angular use modern web platform capabilities to deliver app-like experiences.

❑ High performance and zero-step installation.

❑ Build native mobile apps with Ionic Framework, NativeScript, and React Native.

❑ Create desktop - installed apps across Mac, Windows, and Linux.

Speed and Performance

❑ Angular turns our templates into code that's highly optimized for today's JavaScript machines.

❑ Serve the first view of your application on node.js, .NET, PHP, and other servers for rendering in just HTML and CSS. .

❑ Angular apps load quickly with the new Component Router.

Productivity

- ❑ Quickly create UI views with simple and powerful template syntax.
- ❑ Command line tools:
  - ❑ Start building fast
  - ❑ Add components and tests
  - ❑ Then instantly deploy.
- ❑ Get intelligent code completion, instant errors in popular editors and IDEs.

Full Development Story

- ❑ Karma for unit tests.

- ❑ Protractor makes our scenario tests run faster and in a stable manner.

- ❑ Create high-performance, complex choreographies and animation timelines with very little code through Angular's intuitive API.

Development Environment Setup

Node.js

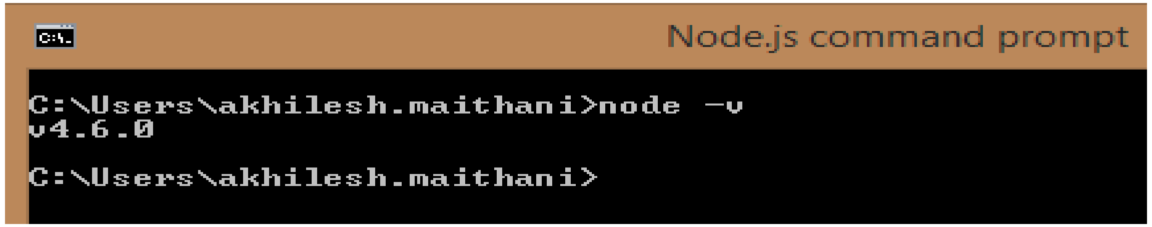
https://nodejs.org/en/download/

- **Node.js** is an open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side.
- Download latest version i.e. node v6.10.3

Check node.js version

Syntax : - **node -v**

- This command is used for checking current installed version of node.



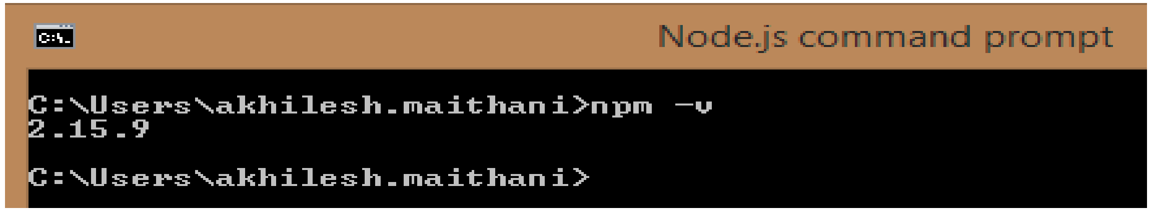
```
C:\Users\akhilesh.maithani>node -v
v4.6.0

C:\Users\akhilesh.maithani>
```

Check npm version

Syntax : - **npm -v**

- This command is used for checking current installed version of Node Package Manager (npm).



```
C:\Users\akhilesh.maithani>npm -v
2.15.9

C:\Users\akhilesh.maithani>
```

Text Editor

- Visual Studio Code, WebStrome , Sublime or any other text editor IDE

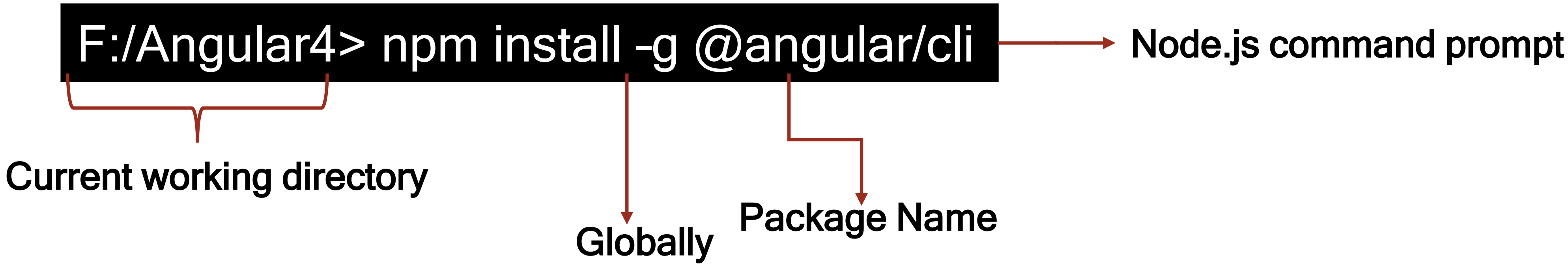
Project Setup

Install Angular CLI

Angular CLI is command line interface for Angular

❖ Open node js command prompt in admin mode.

Syntax : - npm install –g <packagename>



```
F:/Angular4> npm install –g @angular/cli
```

Node.js command prompt

Current working directory

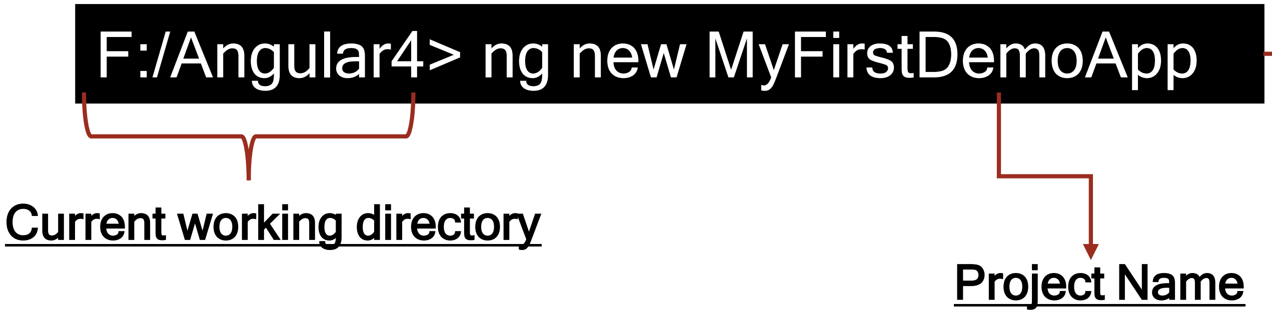
Globally

Package Name

Step 2

Creating project using Angular CLI

Syntax : - ng new <project_name>

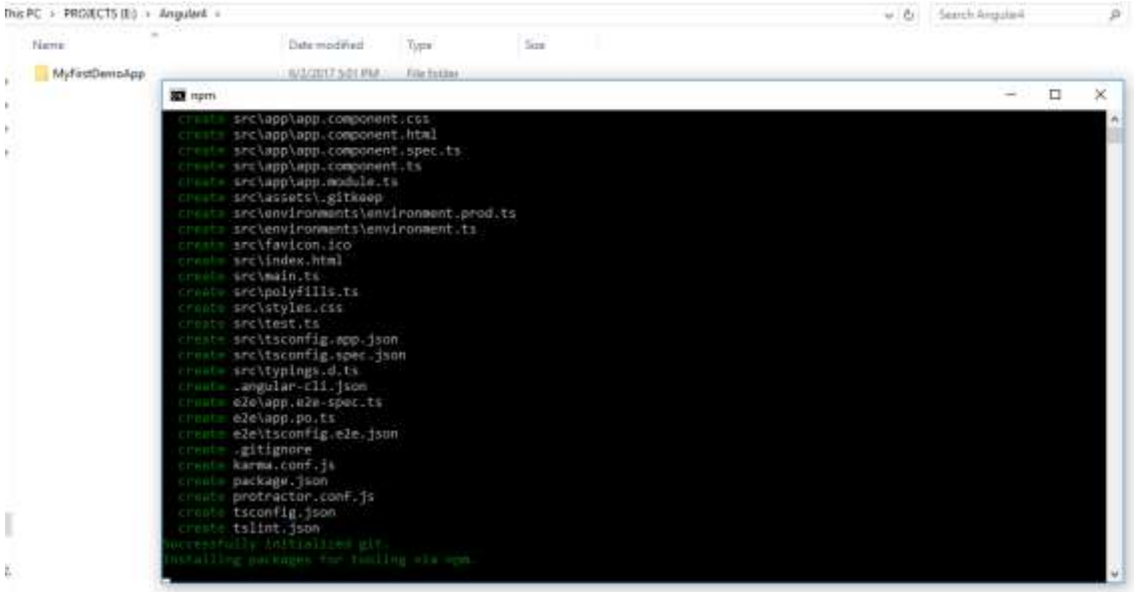


```
F:/Angular4> ng new MyFirstDemoApp
```

Current working directory

Project Name

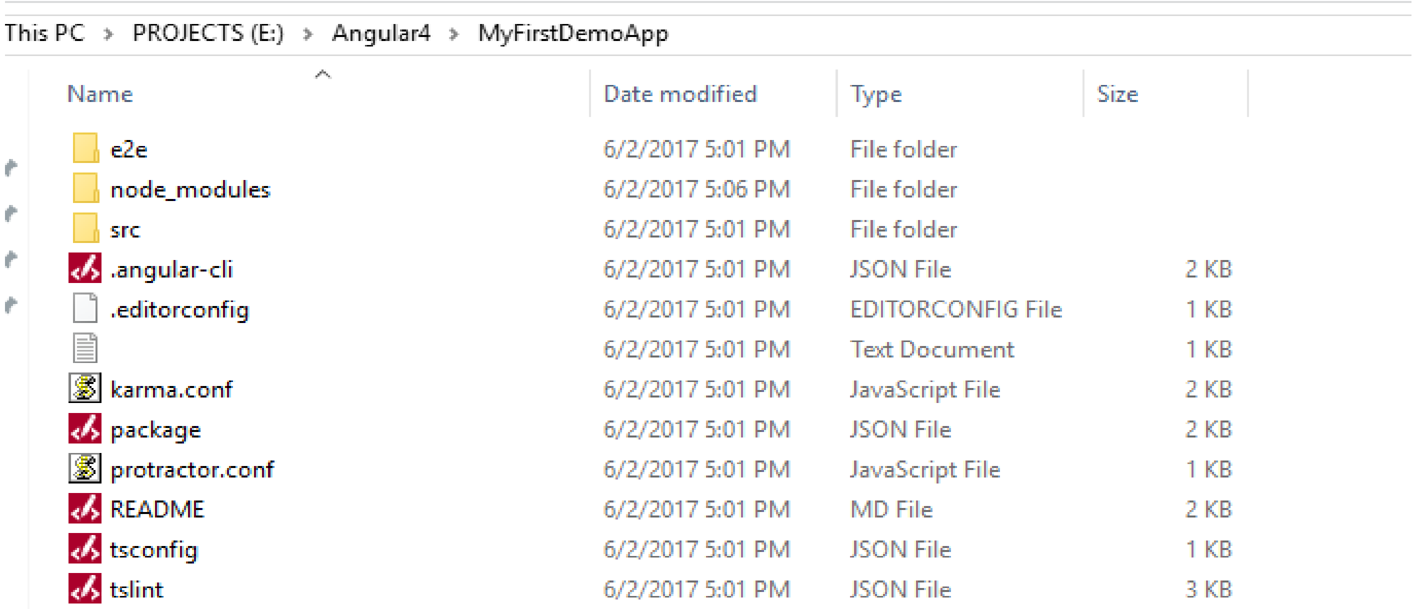
Node.js command prompt



```
create src\app\app.component.css
create src\app\app.component.html
create src\app\app.component.spec.ts
create src\app\app.component.ts
create src\app\app.module.ts
create src\assets\.gitkeep
create src\environments\environment.prod.ts
create src\environments\environment.ts
create src\favicon.ico
create src\index.html
create src\main.ts
create src\polyfills.ts
create src\styles.css
create src\test.ts
create src\tsconfig.app.json
create src\tsconfig.spec.json
create src\typings.d.ts
create .angular-cli.json
create e2e\app.e2e-spec.ts
create e2e\app.po.ts
create e2e\tsconfig.e2e.json
create .gitignore
create karma.conf.js
create package.json
create protractor.conf.js
create tsconfig.json
create tslint.json
Successfully initialized git.
Installing packages for tooling via npm.
```

Step 3

Enter into newly created project folder i.e. MyFirstDemoApp



This PC › PROJECTS (E:) › Angular4 › MyFirstDemoApp

Name	Date modified	Type	Size
e2e	6/2/2017 5:01 PM	File folder	
node_modules	6/2/2017 5:06 PM	File folder	
src	6/2/2017 5:01 PM	File folder	
.angular-cli	6/2/2017 5:01 PM	JSON File	2 KB
.editorconfig	6/2/2017 5:01 PM	EDITORCONFIG File	1 KB
	6/2/2017 5:01 PM	Text Document	1 KB
karma.conf	6/2/2017 5:01 PM	JavaScript File	2 KB
package	6/2/2017 5:01 PM	JSON File	2 KB
protractor.conf	6/2/2017 5:01 PM	JavaScript File	1 KB
README	6/2/2017 5:01 PM	MD File	2 KB
tsconfig	6/2/2017 5:01 PM	JSON File	1 KB
tslint	6/2/2017 5:01 PM	JSON File	3 KB

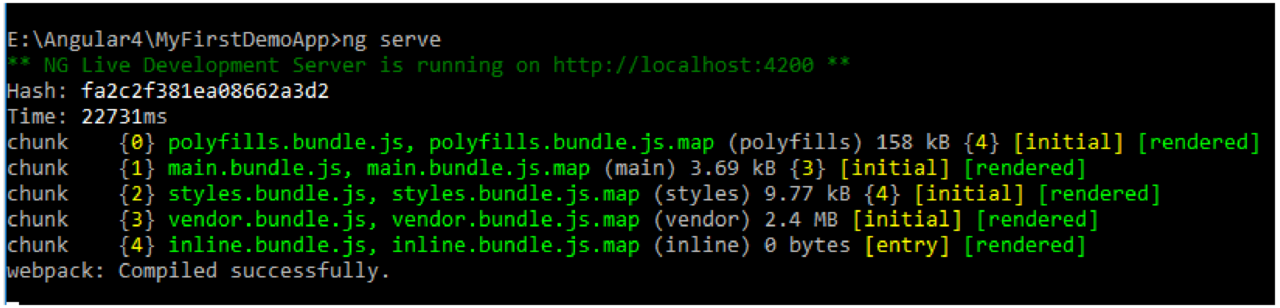
Step 4

Run application

❖ ng serve command is used for to run application.

Syntax : - ng serve

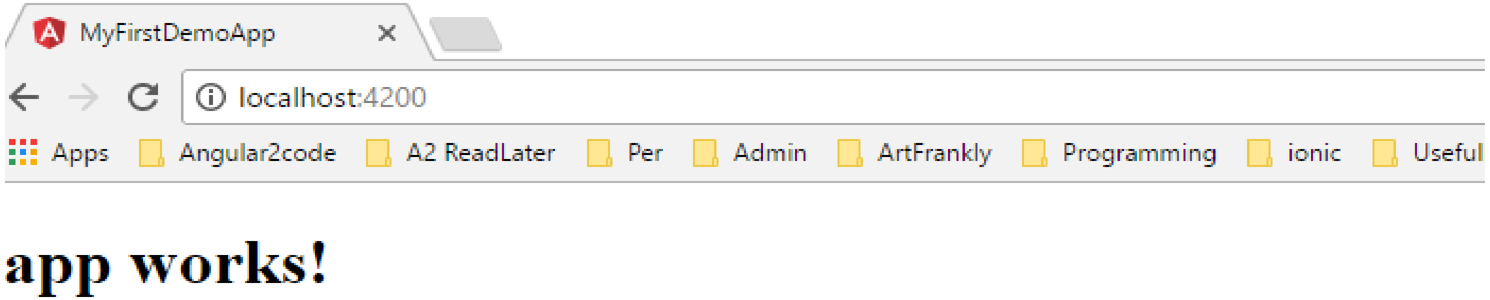
```
F:/Angular4/MyFirstDemoApp> ng serve
```



```
E:\Angular4\MyFirstDemoApp>ng serve
** NG Live Development Server is running on http://localhost:4200 **
Hash: fa2c2f381ea08662a3d2
Time: 22731ms
chunk    {0} polyfills.bundle.js, polyfills.bundle.js.map (polyfills) 158 kB {4} [initial] [rendered]
chunk    {1} main.bundle.js, main.bundle.js.map (main) 3.69 kB {3} [initial] [rendered]
chunk    {2} styles.bundle.js, styles.bundle.js.map (styles) 9.77 kB {4} [initial] [rendered]
chunk    {3} vendor.bundle.js, vendor.bundle.js.map (vendor) 2.4 MB [initial] [rendered]
chunk    {4} inline.bundle.js, inline.bundle.js.map (inline) 0 bytes [entry] [rendered]
webpack: Compiled successfully.
```

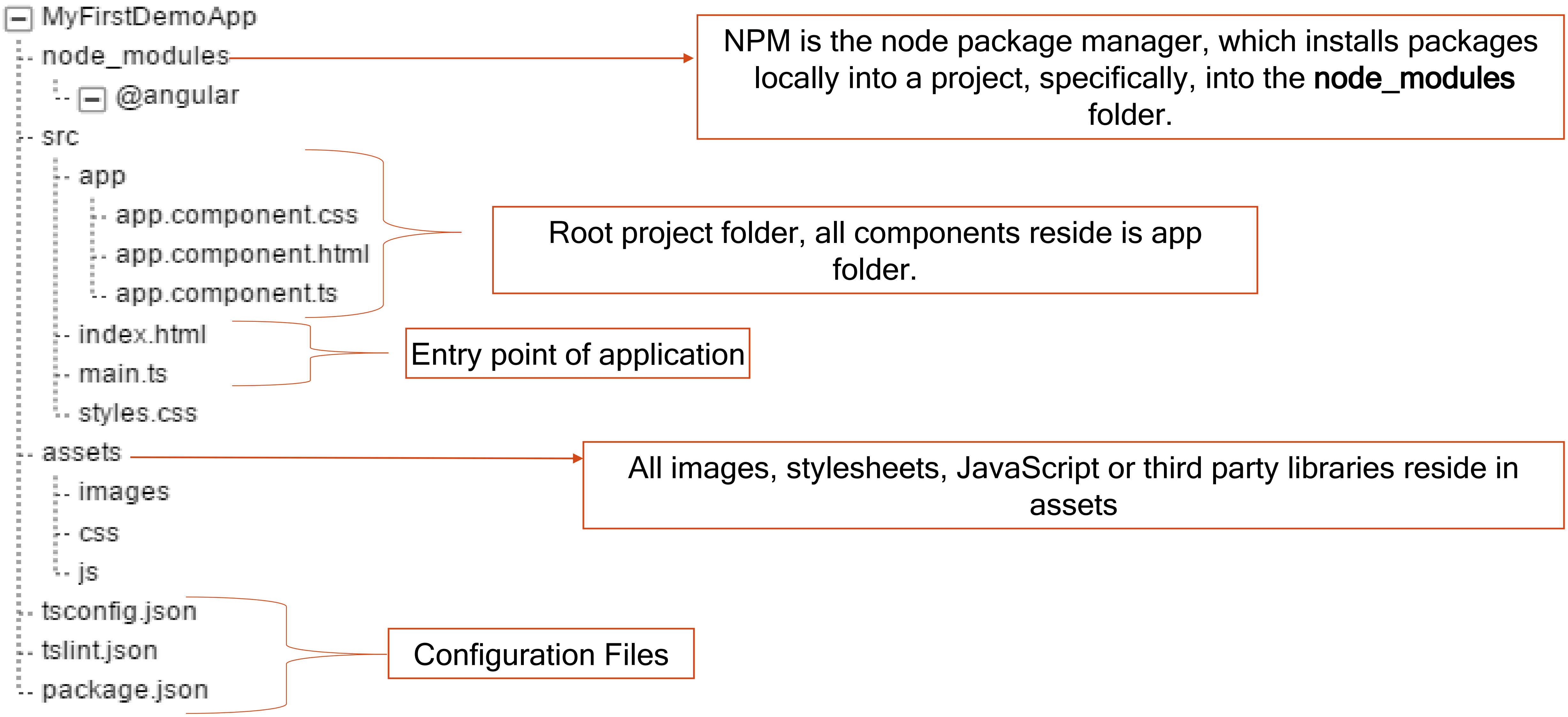
Step 5

Open web browser - *localhost:4200*



app works!

Structure of project



```
MyFirstDemoApp
├── node_modules
│   └── @angular
├── src
│   ├── app
│   │   ├── app.component.css
│   │   ├── app.component.html
│   │   └── app.component.ts
│   ├── index.html
│   ├── main.ts
│   └── styles.css
├── assets
│   ├── images
│   ├── css
│   └── js
├── tsconfig.json
├── tslint.json
└── package.json
```

NPM is the node package manager, which installs packages locally into a project, specifically, into the **node_modules** folder.

Root project folder, all components reside is app folder.

Entry point of application

All images, stylesheets, JavaScript or third party libraries reside in assets

Configuration Files

Installing Packages

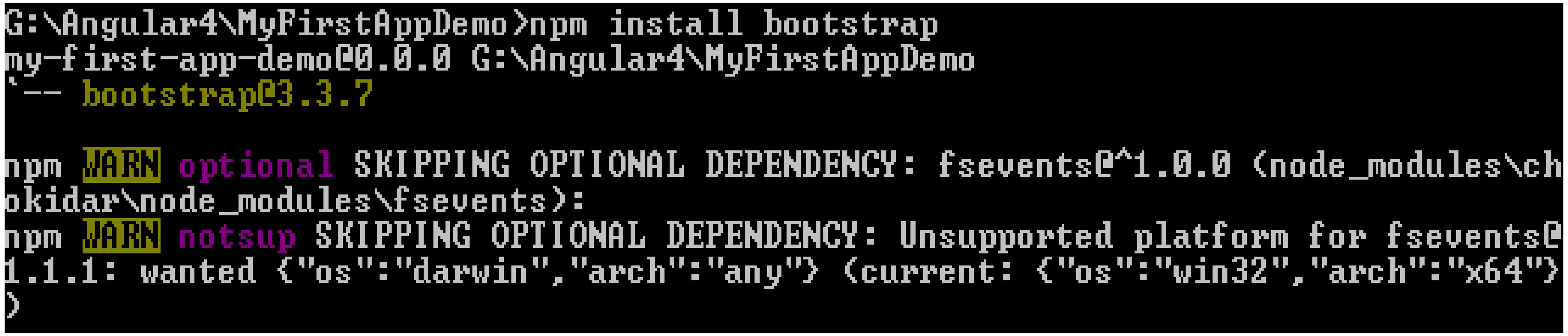
- npm install command is used for installing packages.

Syntax

npm install <package_name>

Example

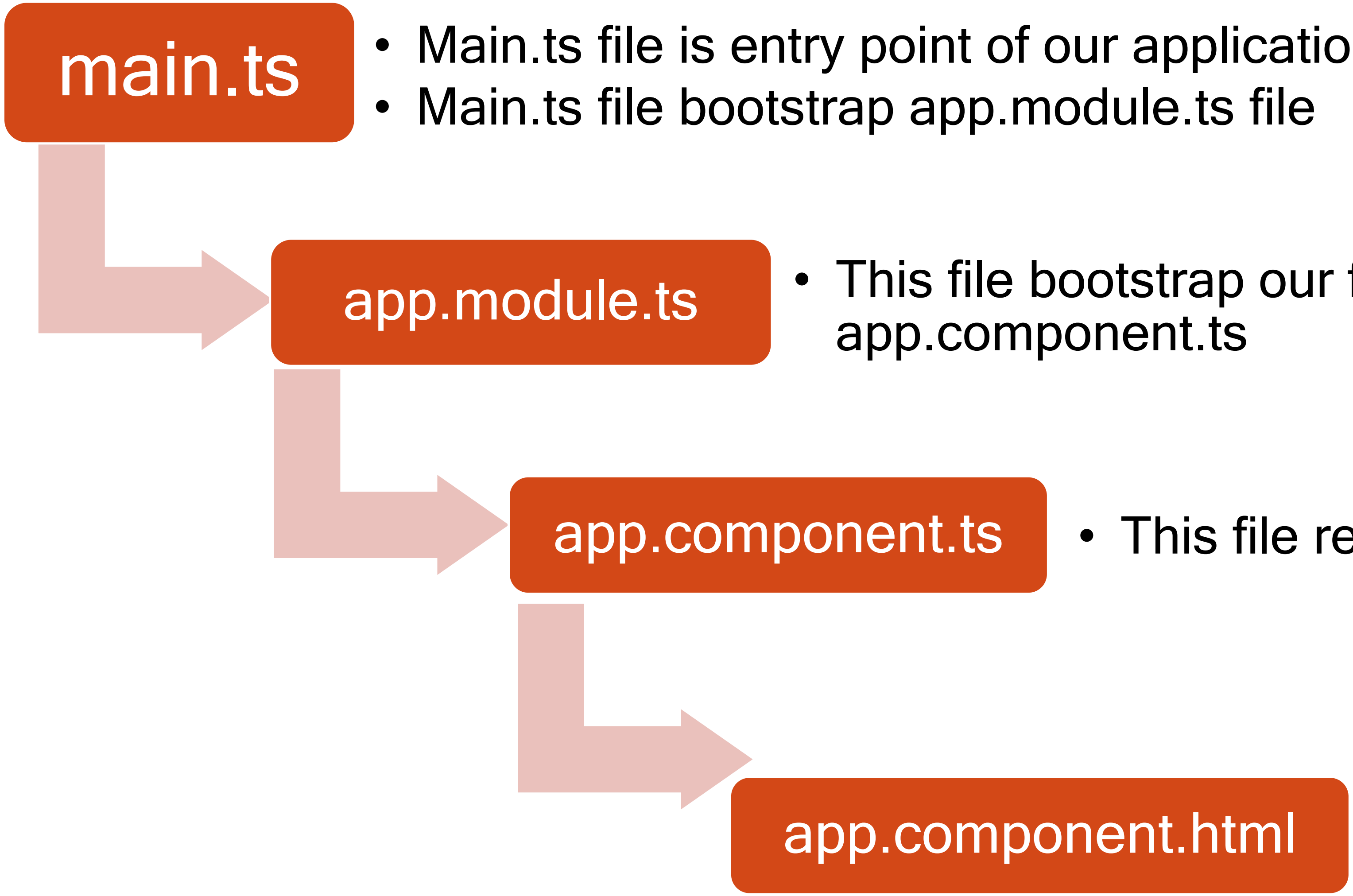
F:/Angular4/MyFirstDemoApp> npm install bootsrap



```
G:\Angular4\MyFirstAppDemo>npm install bootstrap
my-first-app-demo@0.0.0 G:\Angular4\MyFirstAppDemo
`-- bootstrap@3.3.7

npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@^1.0.0 (node_modules\ch
okidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.1.1: wanted {"os":"darwin","arch":"any"} (current: {"os":"win32","arch":"x64"}
)
```

How App Start



main.ts

- Main.ts file is entry point of our application.
- Main.ts file bootstrap app.module.ts file

app.module.ts

- This file bootstrap our first component i.e app.component.ts

app.component.ts

- This file render app.component.html file.

app.component.html

- Final HTML template

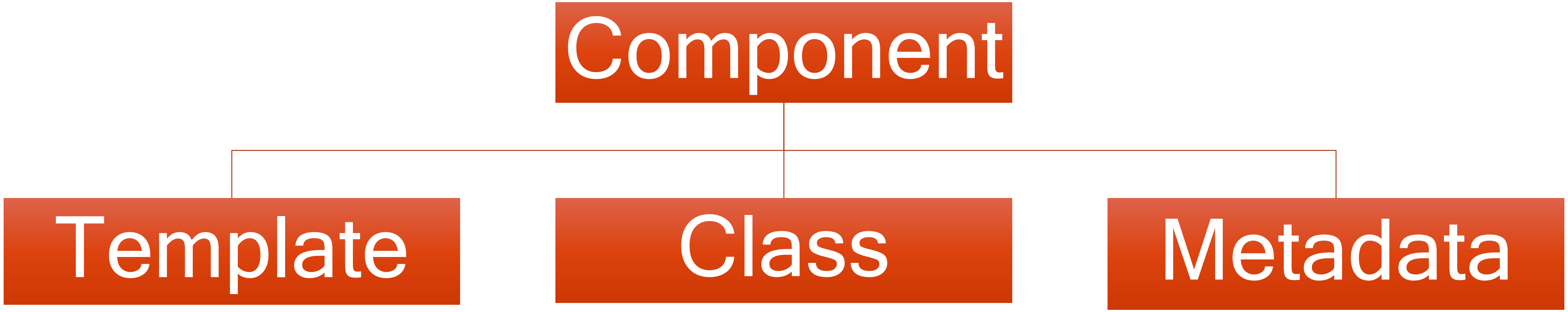
TypeScript

- TypeScript is a free and open source programming language.
- It is developed and maintained by Microsoft.
- It is syntactical superset of JavaScript and adds optional static typings and class based object oriented programming to the language.

Components

- ❖ Components are a logical piece of code for Angular application.

A Component consists of the following –



Component

Template

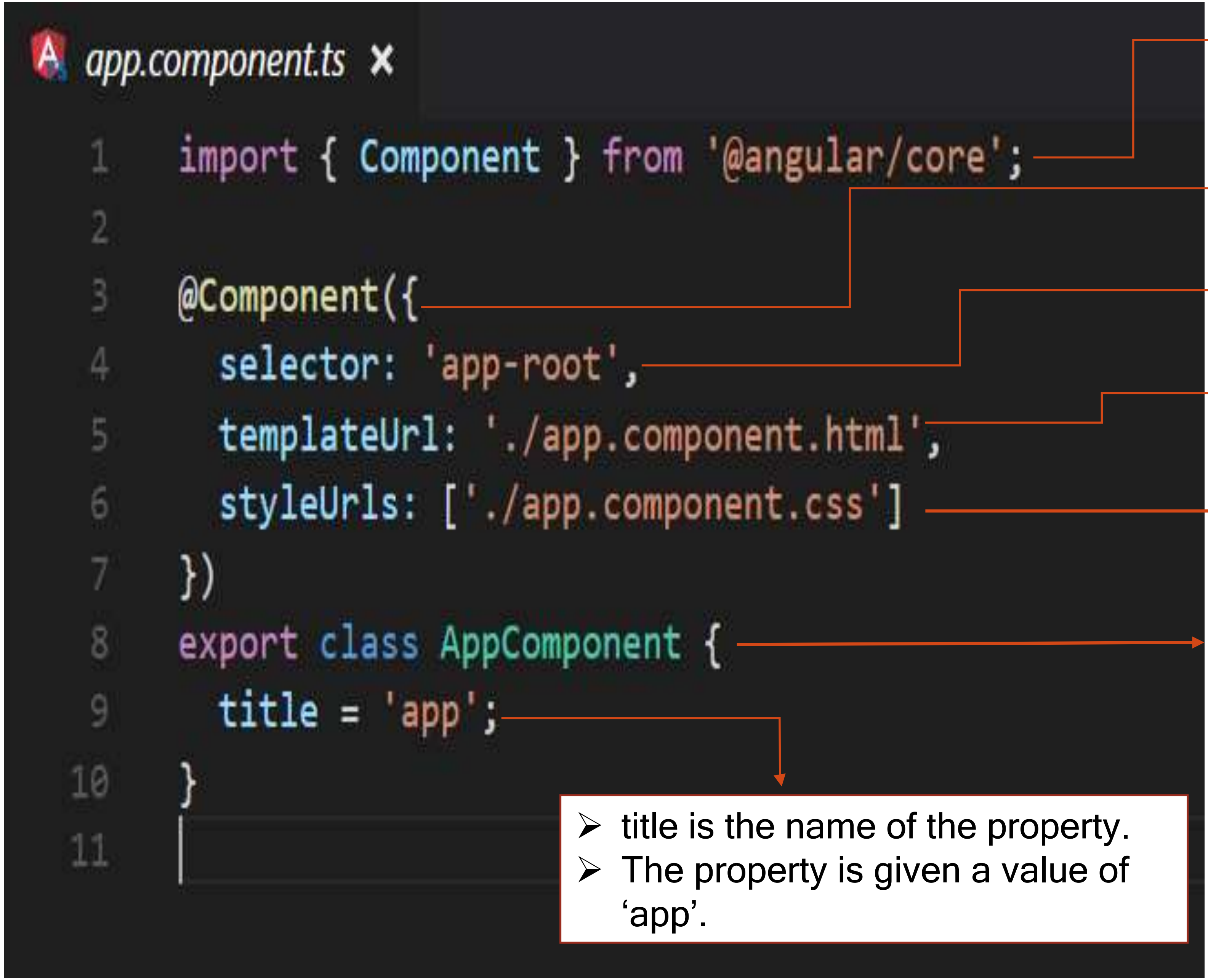
Class

Metadata

- ✓ Template is used to render the view for the application.
- ✓ This contains the HTML that needs to be rendered in the application.

- ✓ This is like a class defined in any language such as C#.
- ✓ This has the code which is used to support the view.
- ✓ It is defined in TypeScript.

- ✓ This has the extra data defined for the Angular class.
- ✓ It is defined with a decorator.



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

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app';
}
```

We are using the import keyword to import the ‘Component’ decorator from the angular/core module.

Decorator

Defines the name of the HTML tag.

It holds our HTML template.

The styles option is used to style a specific component

- Defining a class called AppComponent.
- The export keyword is used so that the component can be used in other modules in the application.

- title is the name of the property.
- The property is given a value of ‘app’.

Creating Component

❖ ng generate command is used for create component.

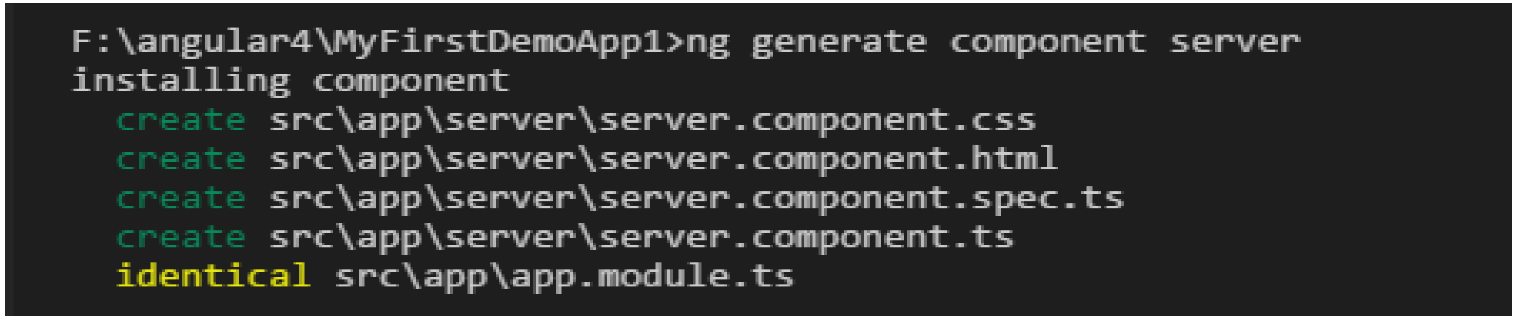
Syntax

ng generate component <component_name>
or
ng g c <component_name>

Example

F:/Angular4/MyFirstDemoApp>

ng g c server



```
F:\angular4\MyFirstDemoApp1>ng generate component server
installing component
  create src\app\server\server.component.css
  create src\app\server\server.component.html
  create src\app\server\server.component.spec.ts
  create src\app\server\server.component.ts
  identical src\app\app.module.ts
```

Component Selector

By Element

Define by square brackets [ ] in selector name.

Syntax

```
@Component({
selector: '[selector-name]',
templateUrl: ‘html - template',
styleUrls: [‘stylesheet']
})
```

```
<div selector_name>  </div>
```

Example

```
@Component({
selector: '[app-root]',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
```

```
<div app-root>  </div>
```

By Class

Define by dot (.) in selector name.

Syntax

```
@Component({
selector: ‘.selector-name',
templateUrl: ‘html - template',
styleUrls: [‘stylesheet']
})
```

```
<div class =“selector-name”>
```

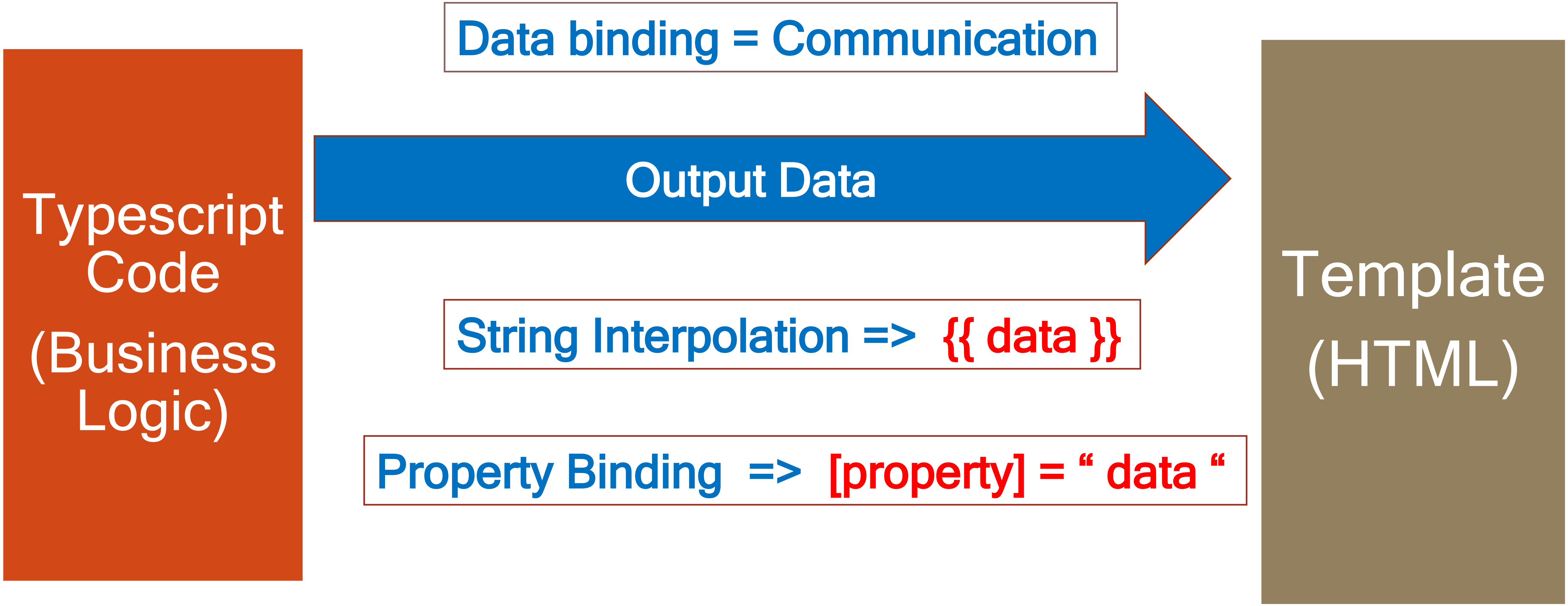
Example

```
@Component({
selector: ‘.app-root',
templateUrl: './app.component.html',
styleUrls: ['./app.component.css']
})
```

```
<div class=“app-root”>  </div>
```

Data binding

- Data binding is communication between business logic and views.



Data binding = Communication

Typescript Code

(Business Logic)

Output Data

Template (HTML)

String Interpolation => {{ data }}

Property Binding => [property] = “ data “

String Interpolation

Interpolation markup is used to provide data-binding to text and attribute values.

Syntax

```
export class <class_name>
 {
      variableName = ‘any string';
}
```

```
<div {{variableName }} >
```

Example

```
export class AppComponent
{
      title = ‘This is my demo app';
}
```

```
<div {{ title }} >  </div>
```

Property Binding

Property binding allow us to bind values to properties of an element to modify their behavior or appearance. This can include properties such as class, disabled, href or textContent.

HTML

```
<button [disabled]="!isActive"
class="btn">ADD</button>
```

Typescript

```
export class ClientComponent {
isActive = false;
constructor() {
setTimeout(() =>
{   this.isActive = true;    }, 2000);  }
ngOnInit() {  }
```

Contact handles

www.cynoteck.com

Contact No: **+1-612-800-9092, +918272014440, +918430155522s**

Website : **www.cynoteck.com**

Email : **sales@Cynoteck.com**

Thank you