Angular Components in-depth:

* Create a new project with **ng new project\_name –style=scss**
* Every Angular component has **@Component** annotation and each component will have an argument which is an object contains information that tells how the component will behave.
* **selector :** The value of this selector can be used in view as a tag.
* To generate a class, **ng generate class class\_name**
* Implementing interface is optional in Angular. If we implement interface methods, then it is not mandatory to use **implements interface\_name** to a class. Angular will execute that method whenever the lifecycle event happens.
* Use Constructor for Dependency Injection to inject services, inject other providers.
* To make an instance of the component we need to much mocking for initialization. The more complex is the constructor is the more complicate to write test cases on the component. So we are initializing the data in **ngOnInit**() instead of initializing in constructor.
* A **pipe** is a feature which lets you pass in values inside our interpolation to a function. To use a pipe we use the following.

<p class="card-text">{{post.summary | truncate}}</p>

* T create a pipe, **ng generate pipe pipe\_name**

@Pipe({

name: 'truncate'

})

export class TruncatePipe implements PipeTransform {

transform(value: any, args?: any): any {

return value.length < args ? value : value.substring(0, args)

+ '...' ;

}

}

* A pipe will be created with the specified name and this pipe will be declared in **app.module.ts** file like a component.

@NgModule({

declarations: [

AppComponent,

BlogPostTileComponent,

BlogListComponent,

TruncatePipe

],

* The pipe function should be stateless. That is, it should process an input to provide an output. But it shouldn’t store or “remember” anything.

<p class="card-text">{{post.summary | truncate:[50]}}</p>