**Angular 2**

Angularjs 1 released in 2010

* Performance :faster initial loads, change detection, improved rendering time
* Improved modularity
* Dependency injection
* Mobile support(without 3rd party)
* Component based development(means greater code reuse and testable)
* Build on typescript(typescript has support of ECMAScript6).

What is ECMAScript:

* Javascript official language standard is called ECMAScript.
* Has several releases(1-7).
* Most modern browsers available today support till ECMS5
* Browser support for ECMA6 is still incomplete.
* So we have **transpilation** compiles ECMA6 to ECMA5
* ECMA6 is officially known as ECMA2015.
* New features in ECMA2015- Classes, Modules, arrow functions etc.

What is typescript?

* TypeScript is a free and open-source programming language developed by Microsoft.
* It is a superset of JavaScript and compiles to JavaScript through a process called transpilation.
* Using TypeScript to build angular applications provides several benefits.  
  1. Intellisense   
  2. Autocompletion   
  3. Code navigation   
  4. Advanced refactoring   
  5. Strong Typing   
  6. Supports ES 2015 (also called ES 6) features like classes, interfaces and inheritance.
* If you have any experience with object oriented programming languages like C# and Java, learning TypeScript is easy.
* Because of all these benefits writing, maintaining and refactoring applications can be an enjoyable experience. So obviously TypeScript has become the number one choice of many developers for developing Angular applications.

Running Angular2 on Visual studio 2015

1. Intall node
2. Install typescript
3. Make sure visual studio 2015 update 3.
4. Download quick start template for anngular2 from github
5. Select src, package.json, tsocnfig.json, lint.json and include in the project.
6. Rightclick on package.json and restore packages.
7. Now open cmd window and type npm start
8. Or change in index.html base href=”/src”, script src=”/node…….” And in systemsjs.config.js ‘npm’:’/node\_modules’.

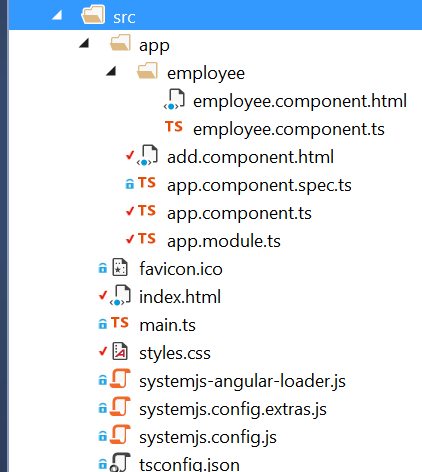
**Component:**

A component in angular is a class with a template and decorator  
Template:defines UI,contains HTML, directives and databindings  
Class : contains required code for template,  
 contains methods(contains logic for the view) and properties(data that we want to display in the view template).  
Decorator: provided by angular to add metadata to the class, A class becomes an Angular component, when it is decorated with the Component decorator.

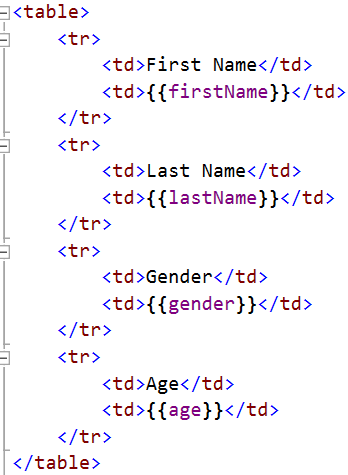


**Template vs TemplateURL:**

Template: we can replace a pair of back tick chars with single or double quotes, when we have html in one line(inline template). If we have more than oneline we need only back tick chars.With inline template, we loose intellisense, code completing and formatting features.  
TemplateURL: when we move html code to separate file, and specify that file name/path,

**Nested Components**:

In this commit, what we gonna do is render a component inside another component. For this we create a folder inside app folder called employee which has .html file and .ts file.

  
  
In app.component.ts write this: 

In app.module.ts write this:



Here we turned class Appmodule into Module by decorating with @NgModule attribute.

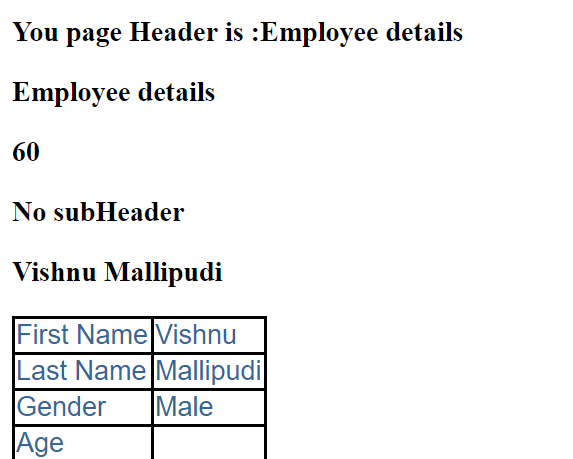
What is a module: a mechanism to group of components, directives, pipes that are related,in such a way that can be combined with other module to create a application.  
This attribute takes 3 properties:  
Imports: an array of modules that we import into current module. Here we imported BrowserModule which means this is the RootModule, There are 2 types of Modules rootmodule and Commonmodule, which will be 1 for entire application, submodules must import from Commonmodule,.  
declarations: array of components, directives,and pipes, that are part of that of module.  
Bootstrap: an array that tells the root component of the module, 99% of the time, there will be only one component. For more info about module see this: https://angular-2-training-book.rangle.io/handout/modules/introduction.html

If you want you can ad styles in style.css

**Adding styles: there are several ways to add styles,**

1. Adding all styles in style.css file
2. Inline styles
3. Specify both,
4. In style property of the @component attribute(but we loose intelliense,autoformatting here).
5. In a separate file and add that file path to styleUrl property of @component attribute.

**Interpolation**:



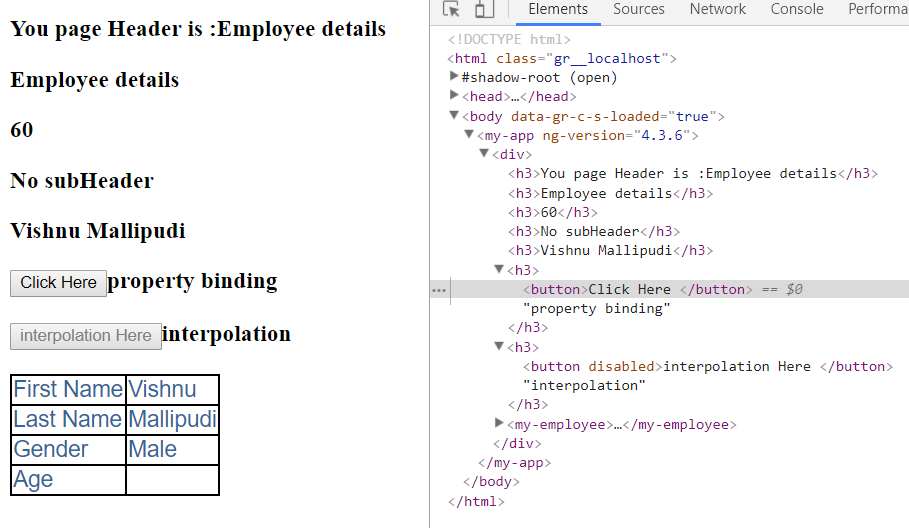
**Property Binding:**

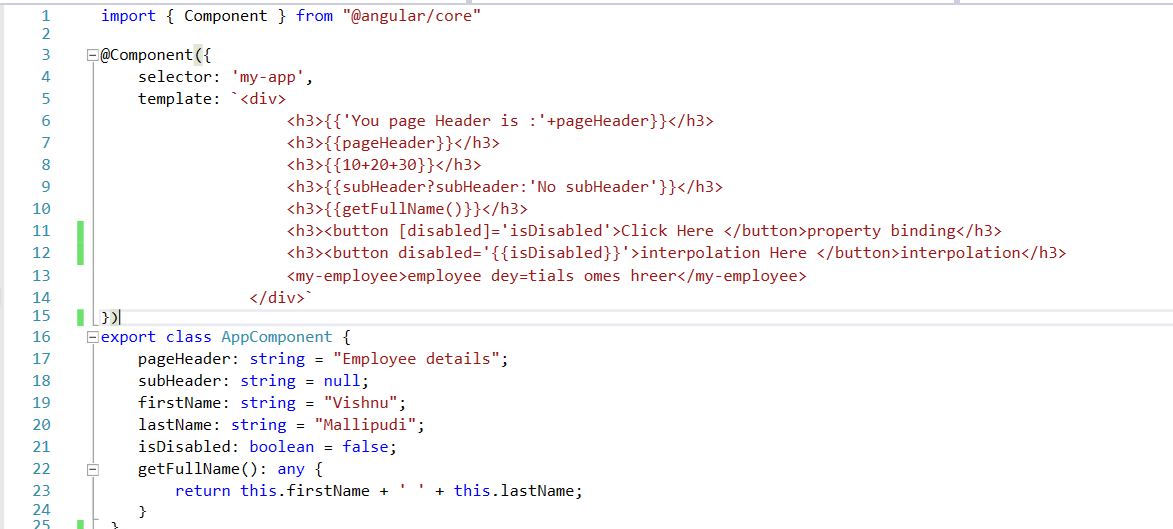
For interpolation we use {{expression}} ,

For property binding we use [property]=”expression”.

To concatenate string we must interpolation ex:



When setting an element property to a non-string data value, Use property binding



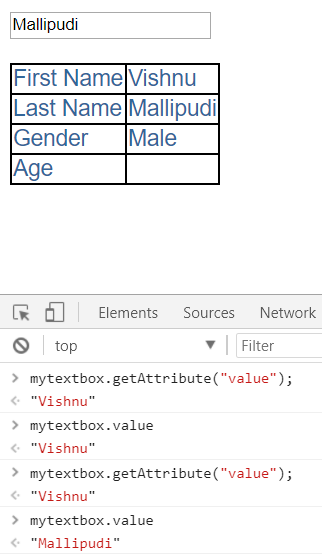
**Html attribute vs DOM property:**

Attributes are defined by HTML, Propeties are defined by DOM,

Attributes initialize DOM properties, once it is complete, their job is done.

Properties value can change, where as attributes value can’t

In Console, the first two lines, are from default initialization, and next two lines are after changing, what we observed is propties can change,but attributes can’t change. Angular binding works with properties, not attributes



**Attribute Binding:**

So we now want to bind values to attributes. We can doo like this,

Interpolation style:

****

Property binding style:



ClassBinding:

We have ngClass directive to add classes to properties



**Styles binding:**



Event Binding:

We use open brackets to bind event in angular 2.



ngIf: see commit event binding



**Two way data binding:**



Or we can use like this:

This ngmodel exists in FormsModule. So import @angular@Forms module in rootModule(app.component.ts).

**NgFor:**

In HTML file we use like this: 

In .ts file define employees variable



**NgFor with TrackBy:**

When ever we use ngfor, a small change in the list, makes the entire Dom (entire rows) will be removed and added newly, if the list is bigger, we got delay. For this reasons, we have trackBy.





With these changes, whenever, you change a employee in the list, only that will be added to the DOM.

Similarly, we have even, odd, first, last

