CHIANG MAI UNIVERSITY

Bachelor of Science (Software Engineering)

College of Arts, Media and Technology

2nd Semester / Academic Year 2017

SE 331 Component Based Software Development

Quick Start Angular

Name ……..…… ID ……

Objective In this session, you will experience how the application design by separate the view from the control

Suggestion you should read the instructions step by step. Please try to answer a question by question without skipping some questions which you think it is extremely difficult.

Hint The symbol + and – in front of the source code is to show that you have to remove the source code and add the source code only. There are not the part of the source code

0. Setting

0.1. Open the the virtual machine the virtual machine application, select the SE331 virtual machine.

Find the latest version of the given application and install in the virtual machine.

- Git for windows
- Node js LTS version
- Notepad++

0.2. Create the folder `C:\lab`

0.3. Open the command window and then go to the folder `C:\lab\`

0.4. Type `git clone https://github.com/angular/quickstart  lab01`

0.5. What happens to the folder?

……

……

0.6. Remove the .git folder by using the command `rd .git /S/Q`

0.7. List what are the folders in the `lab1` directory

……

……

0.8. Run the command `npm install`

What folder has been generated? And what are the content?

……

……

0.9. Run `npm start`

Show what happen in the web browser

Lecture's Signature _____

1. Understand Angular2

1.1. To understand how the code link together, delete every file in the `/src/app` folder.

And then create a new file name `main.ts` in the `app` folder with the given information

```
+import {Component} from '@angular/core';
+
+@Component({
+       selector: 'my-app',
+       template: '<h1>Hello Angular</h1>'
+})
+class AppComponent{}
+
```

1.2. This component can be used however, the module must be declared. In the same file, import the `NgModule` as shown

```
-import {component} from '@angular/core';
+import {NgModule,Component} from '@angular/core';
```

1.3. And then create a new module app at the end of the file

```
+@NgModule({
+       declarations: [ AppComponent]
+})
+class AppModule{}
```

1.4. Add the required module which are used to generate the angular application

```
+import {BrowserModule} from '@angular/platform-browser';
+import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';
```

Update the `NgModule` at the end of the file

```
@NgModule({
-       declarations: [ AppComponent]
+ declarations: [ AppComponent],
+ imports: [BrowserModule],
+ bootstrap: [AppComponent]
 })
class AppModule_{}
+
+platformBrowserDynamic()
+       .bootstrapModule(AppModule);
```

Now your application is ready.

See the browser and show the result

Lecture's Signature _____

Are there any files created here?

.....

.....

2. The data member in the class can be shown in the web page. Update the `main.ts` as given

```
@Component({
  selector: 'my-app',
- template: '<h1>Hello Angular</h1>'
+ template: '<h1>Hello {{name}}</h1>'
 })
-class AppComponent {}
+class AppComponent {
+       name = 'SE331';
+}
```

See how the browser change?

Create a new attribute `courseName` as this course name, and show the new attribute value on the web page.

Lecture's Signature _____

3. The object can be added to the web site, update the app component as given.

```
@Component({
  selector: 'my-app',
- template: '<h1>Hello {{name}}</h1>'
+ template: `<h1>Hello {{name}}</h1>
+                       <h2>{{student.studentId}}</h2>
+                       <p> {{student. name}} {{student.surname}}</p>
+                       <p> gpa = {{student.gpa}}</p>`
 })
 class AppComponent {
        name = 'SE331';
+       student = {
+               "id": 1,
+               "studentId":"562110507",
+               "name":"Prayuth",
+               "surname":"Tu",
+               "gpa":4.00
+       };
 }
```

Notice that the graveaccent or backtick (`) is used for the multi line html code, you can type it by hold the Alt key and press 96 on the key pad.

See what has been changed?

4. The array of object can be added either, the array in Javascript, and also in typescript is defined under the block bracket ('[', ']'), each object is defined under the curly braces ('{', '}'). The array can be provided in the AppComponent class as given

```
class AppComponent {
        name = 'SE331';
-       student = {
+       students = [{
                "id": 1,
                "studentId":"562110507",
                "name":"Prayuth",
                "surname":"Tu",
                "gpa":4.00
-       };
+       },{
+               "id": 2,
+               "studentId":"562110509",
+               "name":"Pu",
+               "surname":"Priya",
+               "gpa":2.12}];
 }
```

And then add the structural directive `ngFor` to read all the data in the array in the HTML template.

```
@Component({
  selector: 'my-app',
  template: `<h1>Hello {{name}}</h1>
+       <ul>
+               <li *ngFor="let student of students">
                        <h2>{{student.studentId}}</h2>
                        <p> {{student. name}} {{student.surname}}</p>
-                       <p> gpa = {{student.gpa}}</p>`
+                       <p> gpa = {{student.gpa}}</p>
+               </li>
+       </ul>`
 })
```

See in the html file.

Now add a new object to the array. It should be the student object with your own information. And then show the result in the browser.

Lecture's Signature _____

5. The other directive to select whether to show the HTML element or not is `ngIf` try the given template and see the result.

```
                        <h2>{{student.studentId}}</h2>
                        <p> {{student. name}} {{student.surname}}</p>
-                       <p> gpa = {{student.gpa}}</p>
+                       <p *ngIf="student.gpa > 2.5">Good Student who get grade
                                {{student.gpa}}</p>
+                       <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
                                {{student.gpa}}</p>
                </li>
```

Update the template to shown only students whose `gpa` is more than 2.2

Lecture's Signature _____

6. The output can be decorated. The pipe symbol ('|') is used to change the output display value. Update the given template

```
                <li *ngFor="let student of students">
-                       <h2>{{student.studentId}}</h2>
-                       <p> {{student. name}} {{student.surname}}</p>
-                       <p *ngIf="student.gpa > 2.5">Good Student who get grade
                                {{student.gpa}}</p>
-                       <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
                                {{student.gpa}}</p>
+                       <h2>{{student.studentId }}</h2>
+                       <p> {{student. name | uppercase}}
                                {{student.surname}}</p>
+                       <p *ngIf="student.gpa > 2.5">Good Student who get grade
                                {{student.gpa | number:'1.2-2'}}</p>
+                       <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
                                {{student.gpa | number:'1.2-2'}}</p>
                </li>
```

Explain what have been changed?

.....

.....

7. We can create the class method to show the value to the template file.

Create a method in a class with default return value. As given

```
                "name":"Pu",
                "surname":"Priya",
                "gpa":2.12}];
+
+       averageGpa(){
+               return 0;
+       }
 }
```

Then add the template html to show the method return value

```
-       </ul>`
+       </ul>
+
+       <p> The average gpa is {{averageGpa()}} </p>`
```

See what have been changed in the browser.

Then implement the `averageGpa()` method in order that the method will return the correct average value. Update the code as given

```
        averageGpa(){
-               return 0;
+               let sum = 0;
+               for(let student of this.students){
+                       sum += student.gpa;
+               }
+               return sum/this.students.length;
        }
```

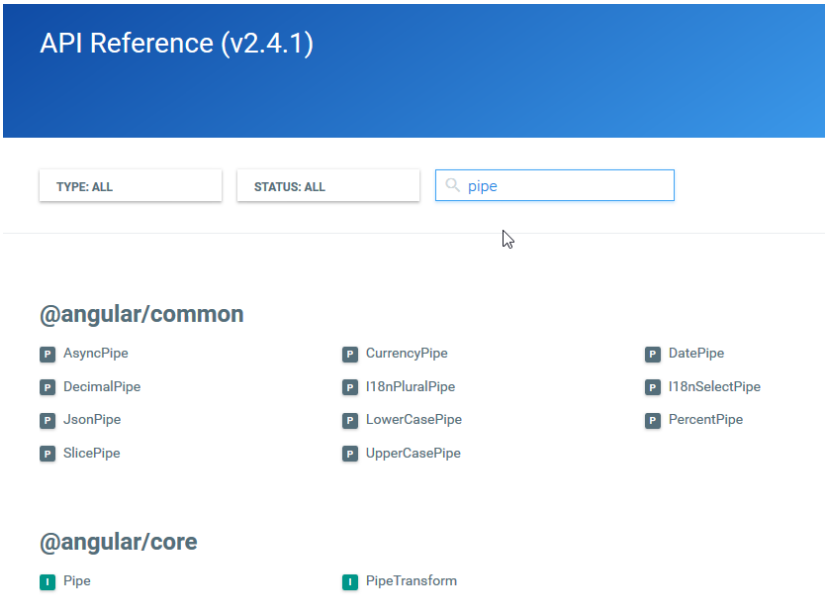
See what have been changed in the browser.

Update the template to show the gpa value with 4 digit precision. You can set the precision by reading the pipe API.

To get the API, you must google for the angular2 API then you will get this URL

https://angular.io/docs/ts/latest/api/

Then in filter type `pipe`



You will see the link to all pipe API, you can click on the `DecimalPipe` to see how to set the output.

Lecture's Signature _____

8. Put everything in one file is not good. We need to modularize the project in order that it is easy to maintain the project.

   First, crate the `app.component.ts`, copy information of the `AppComponent` class from `main.ts` to the new file as given

```
+import {Component} from '@angular/core';
+@Component({
+ selector: 'my-app',
+ template: `<h1>Hello {{name}}</h1>
+       <ul>
+               <li *ngFor="let student of students">
+                       <h2>{{student.studentId }}</h2>
+                       <p> {{student. name | uppercase}}
{{student.surname}}</p>
+                       <p *ngIf="student.gpa > 2.5">Good Student who get grade
{{student.gpa | number:'1.2-2'}}</p>
+                       <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
{{student.gpa  | number:'1.2-2'}}</p>
+               </li>
+       </ul>
+
+       <p> The average gpa is {{averageGpa()}} </p>`
+})
+ class AppComponent {
+       name = 'SE331';
+       students = [{
+               "id": 1,
+               "studentId":"562110507",
+               "name":"Prayuth",
+               "surname":"Tu",
+               "gpa":4.00
+       },{
+               "id": 2,
+               "studentId":"562110509",
+               "name":"Pu",
+               "surname":"Priya",
+               "gpa":0}];
+       averageGpa(){
+               let sum = 0;
+               for(let student of this.students){
+                       sum += student.gpa;
+               }
+               return sum/this.students.length;
+       }
+}
```

   So there would be only this information in the `main.ts`

```
import {NgModule} from '@angular/core';
import {BrowserModule} from '@angular/platform-browser';
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';

@NgModule({
 declarations: [ AppComponent],
 imports: [BrowserModule],
 bootstrap: [AppComponent]
})
class AppModule {}

platformBrowserDynamic()
        .bootstrapModule(AppModule);
```

From here, the `AppModule` required `AppComponent` in the declaration part. But the `AppComponent` class is moved to the other file. So we import the `AppComponent` from the `app.component.ts` by adding the import directive as given.

```
+import {AppComponent} from './app.component';
```

But the `AppComponent` could not be imported unless we allowed them to. So allow the `AppComponent` to be used in the different file by adding the `export` keyword as given.

```
- class AppComponent {
+ export class AppComponent {
```

After finish this step, the website should be the same, no error occurred.

Now in `AppComponent`, there are two component, the Hello component, and the students component.

So we extract the students component to `students.component.ts` by creating the `students.component.ts` and copying the Student information from the `app.component.ts` and add some required import as given.

```
+import {Component} from '@angular/core';
+@Component({
+ selector: 'students',
+ template: `<ul>
+               <li *ngFor="let student of students">
+                       <h2>{{student.studentId }}</h2>
+                       <p> {{student. name | uppercase}}
{{student.surname}}</p>
+                       <p *ngIf="student.gpa > 2.5">Good Student who get grade
{{student.gpa | number:'1.2-2'}}</p>
+                       <p *ngIf="student.gpa <= 2.5">Bad Student who get grade
{{student.gpa  | number:'1.2-2'}}</p>
+               </li>
+       </ul>
+
+       <p> The average gpa is {{averageGpa()}} </p>`
+})
+export class StudentsComponent {
+       students = [{
+               "id": 1,
+               "studentId":"562110507",
+               "name":"Prayuth",
+               "surname":"Tu",
+               "gpa":4.00
+       },{
+               "id": 2,
+               "studentId":"562110509",
+               "name":"Pu",
+               "surname":"Priya",
+               "gpa":0}];
+       averageGpa(){
+               let sum = 0;
+               for(let student of this.students){
+                       sum += student.gpa;
+               }
+               return sum/this.students.length;
+       }
+}
```

Note that the select is changed to students in order to be called by another template, and the class is marked as exported.

So in the app.component.ts is now have only this information.

```
import {NgModule,Component} from '@angular/core';
@Component({
 selector: 'my-app',
 template: `<h1>Hello {{name}}</h1>
        <students></students>`
})
export class AppComponent {
        name = 'SE331';
}
```

The part which show the students information is now replaced with `students` tag.

However, to use the students.component, we need to import and declared in the `main.ts`. update the `main.ts` using thegiven snipet.

```
+import {StudentsComponent} from './students.component';

 @NgModule({
- declarations: [ AppComponent],
+ declarations: [ AppComponent,
                 StudentsComponent],
  imports: [BrowserModule],
```

Now see in the browser, there should be the same.

Finally, we extract all the module declaration into one file.

Create app.module.ts and copy the AppModule class from the main.js as given.

```
+import {NgModule} from '@angular/core';
+import {BrowserModule} from '@angular/platform-browser';
+import {AppComponent} from './app.component';
+import {StudentsComponent} from './students.component';
+
+@NgModule({
+ declarations: [ AppComponent,
+                              StudentsComponent],
+ imports: [BrowserModule],
+ bootstrap: [AppComponent]
+})
+export class AppModule {}
```

Do not forget to `export` the class.

So the `main.ts` is now have only this.

```
import {platformBrowserDynamic} from '@angular/platform-browser-dynamic';
import {AppModule} from './app.module';

platformBrowserDynamic()
        .bootstrapModule(AppModule);
```

Now we can add a new module only in an `app.module.ts`, and our file layout is similar to what we have to get at the beginning from the GitHub.

9. Now add the time component to the web application. The time component shows the `time` which the website is loaded as shown in the given picture.

The average gpa is 2

current time 10:25 AM

You can use the `time` pipe to adjust the output.

Lecture's Signature _____