

Angular Basics - Lab and Homework

If not already installed, install NodeJS from the NodeJS website.

Once installed verify the installation by opening a command prompt and verifying the version.

```
> node -v  
  
> npm -v
```

Next, install the TypeScript system.

```
> npm install -g typescript
```

And verify the TypeScript compiler is present:

```
> tsc -v
```

Create a basic html file template that includes several `<div>` sections with ids.

In another location, create a TypeScript file. It can be called whatever you want but the shorter the better. Make sure the file extension is ".ts" such as "someFile.ts".

Recreate the TypeScript function from the slide deck and perform the same document updates:

```
function sayHello(person: string): string {  
    return "Hello, " + person;  
}  
  
var user = "Super Student";  
  
document.getElementById("para").innerHTML = sayHello(user);
```

Compile the file by to the proper folder in the command prompt and compiling with the `tsc` command. Note the name of the produced JavaScript file.

Reference the JavaScript file from the html page, then launch the html page in a browser to verify the output.

Next, create a class called `Person` in the file below the `getElementById()` method. Give it the properties of first name, last name, age, phone number, state, zip code, and occupation. Create a constructor to initialize these properties. Also create several methods to return combinations of interest, such as Full name, Name and Phone Number, Location, etc.

Compile the file again with `tsc` to ensure the syntax is correct.

Next, create several people and, mimicing the method above, assign the output of some of their information to various `<div>` sections in your page.

Add a `hourlyWage` property to the person (and if you want, rename the class to `Employee`). Add a function that calculates a weekly wage and takes the number of hours as an optional parameter. If it is not presented, use a typical 40 hours to calculate the wage. Display this in a area that states the name of the person and their weekly wage.

NOTE Remember that after each edit/save to the .ts file, you will have to compile and refresh the web page in the browser.

Create a property for a person's certifications/degrees that is an array. These could be things such as MBA, CSM, MCSD, etc. Next create a function called `addCerts()` that takes an unknown number of arguments and adds them to the array.

Create an interface called `EmployeeOptions` (`PersonOptions` if you didn't rename to `Employee`) where the name values are required but the other parameters are optional. Create a method in the class that takes an option parameter and returns an `Employee` (`Person`) based on the options given.

Implement this with a few different options to observe how it works. For additional reference you may need to look at: <https://www.typescriptlang.org/docs/handbook/interfaces.html#optional-properties>

Examine the compiled JavaScript file to see how the `tsc` changes TypeScript into JavaScript.

Angular ecosystem

Back in the command prompt, issue the following command to install the Angular CLI

```
npm install -g @angular/cli
```

and again, to verify:

```
ng -version
```

Run the help command to get an overview of the number of commands available.

Then run help on a few of the key commands. This is done by typing something similar to `ng new -help`.

Examine the following commands:

- new

- add
- generate
- build
- serve
- config
- run

Create a new empty folder and change to that folder.

Run the ng command to create a new app.

Note that this could take *several* minutes depending on your connection speed.

Optional - but suggested: Install Visual Studio Code. This is actually a free, lightweight pluggable IDE that is actually written in TypeScript.

Once that is installed you can simply type in the command "code ." from the application directory (you may have to cd into that folder from where you ran new) and VS Code will open the folder.

next type ng serve --open to run the basic application shell.

Notice that the address by default is localhost:4200.

in the window press Ctrl-C to stop the server.

type in the command "npm start".

Notice the server will start but you have to manually open the browser to localhost:4200.

Open the package.json file and look at the "scripts" section. Change the line for "start" to include the "--open" option.

Now type "npm start" again and see what happens.

Without shutting down the server, do the following:

Edit the html page to removed the links and save the file.

Change the title property in app.component.ts and save the file.

Notice in both cases the application will rebuild and redisplay.

Shut down the server and add a new component.

Put the tag for the new component in the original html file where the links were. Re-run the application to see the basic page information.