Data Types

Difficulty: Easy

Create the following variable using appropriate data types in typescript:

studentName, age

Difficulty: Medium

Create the following variable using appropriate data types in typescript:

studentName, age, contact. Contact detail can be a mobile number or email id or address. It should allow any type.

Difficulty: Hard

Create the following variable using appropriate data types in typescript:

studentName, age, standard. standard may be a number or string. ex: 10 (or) X (only number or string allowed).

TypeScript Interfaces

Difficulty: Easy

Allow the user to create an object of employee which must contain employeeId, firstName, lastName. The employeeId must be a number and firstName, lastName must be a string. (Use interface to restrict the object to only specified data members).

Difficulty: Medium

Allow the user to create an object of student which must contain rollNo, firstName, lastName and standard. The employeeId must be a number and firstName, lastName must be a string. The standard can be a number or string (ex: 10 (or) X). (Use interface to restrict the object to only specified data members).

Difficulty: Hard

Allow the user to create an object of student which must contain rollNo, firstName, lastName and standard, contactInfo. The employeeId must be a number and firstName, lastName must be a string. The standard can be a number or string (ex: 10 (or) X). The contactInfo can be any type (ex: mobileNo (or) email (or) address). (Use interface to restrict the object to only specified data members).

TypeScript Classes

Difficulty: Easy

Create a class Employee with properties employeeId, firstName, dateOfJoining. Create a constructor that accepts these values as parameters and initialize the properties. Compile the ts file and verify the js file.

Difficulty: Medium

Create a class Employee with properties employeeId, firstName, dateOfJoining. Create a constructor that accepts these values as parameters and initialize the properties.

Create an object of Employee class and give values to the properties in the constructor. Call the toString method. Compile the ts file and execute the js file.

Difficulty: Hard

Create a class Employee with properties employeeId, firstName, dateOfJoining. Create a constructor that accepts these values as parameters and initialize the properties. Create a method toString() to display the values using console.log().

Create an object of Employee class and give values to the properties in the constructor. Call the toString method. Compile the ts file and execute the js file.

TypeScript Namespaces and Modules

Difficulty: Easy

Create a typescript function findMax() that accepts 2 numbers and return the bigger one.

Difficulty: Medium

Create a typescript function findAvg() that accepts 2 or 3 arguments and return the average of them.

Difficulty: Hard

Create a typescript function findAvg2() that accepts 2 or 3 arguments and return the average of min and max of them.

TypeScript Functions

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Progressive Part Practice Assignments

Difficulty: Easy

Create a typescript program that creates an object of employee with employeeId, firstName and lastName. Initialize the object using a constructor and display the values using a toString method. Compile the typescript program in command line and execute the javascript file generated.

Difficulty: Medium

Difficulty: Hard

Dive into Angular 7

Difficulty: Easy

Create an angular project 'angularapp' using command line interface.

Difficulty: Medium

Difficulty: Hard

Creating Custom Component

Difficulty: Easy

Inside the angularapp project, create a component called "login-component".

Difficulty: Medium

Inside the angularapp project, create a component called "login-component". In the template, create the html code to accept user name and password with submit button.

Difficulty: Hard

Inside the angularapp project, create a component called "login-component". In the template, create the html code to accept user name and password with submit button.

In the component class, declare variables userName and password.

Component Templating and Styling

Difficulty: Easy

Create the input form controls inside the login-component template ie., html file to capture userName and password. Give background and text color, font size, font family, alignment in the css file.

Difficulty: Medium

Put the user interface elements inside a html table and give cell padding, color, alignment, font family, size etc. in the css file for the login-component.

Difficulty: Hard

Put the user interface elements inside a html table and give cell padding, color, alignment, font family, size etc. in the css file for the login-component. Use id, class attributes to provide style in css. Make alternate row colors and make the form look vibrant.

Understanding New Component Hierarchy

Difficulty: Easy

Install bootstrap in command line and update the module with the bootstrap path

Difficulty: Medium

Install bootstrap in command line and update the module with the bootstrap path

Difficulty: Hard

Install bootstrap in command line and update the module with the bootstrap path

Adding Bootstrap to the Angular Project

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Progressive Part Practice Assignments

Difficulty: Easy

Create an angular project "my-angular-project" and create a "login" component. Create a form to capture user inputs like user name and password in html template or template Url. Add css styles in the appropriate file and add variable to the component class. Apply bootstrap after installing it in command line.

Difficulty: Medium

Difficulty: Hard

User Interactivity and Dynamic Elements

Difficulty: Easy

In order to allow links to refer to ths login component using simple url (ex: login), configure the routes variable in app-routing.module.ts file.

Difficulty: Medium

In order to allow links to refer to ths login component using simple url (ex: login), configure the routes variable in app-routing.module.ts file.

For Home, the path should be "/" and for login component, the path should be "login".

Difficulty: Hard

In order to allow links to refer to ths login component using simple url (ex: login), configure the routes variable in app-routing.module.ts file.

For Home, the path should be "/" . For login component, the path should be "login". For about compionent, the path should be "about".

Working with User Inputs

Difficulty: Easy

Handle the ng-submit to capture the user input and use console.log to display the same.

Difficulty: Medium

Handle the ng-submit to capture the user input and use console.log to display the success or failure based on the criteria. Success if user name and password are not same. Else failure.

Use \*ngIf directive to display appropriate div for success and failure.

Difficulty: Hard

Handle the ng-submit to capture the user input and use console.log to display the success or failure based on the criteria. Success if user name is 'john' and password is 'abraham'. Else failure.

Use \*ngIf directive to display appropriate div for success and in case of failure, use ng-template for elseBlock.

Angular Directives

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

ngSwitch & ngFor Directives

Difficulty: Easy

Display the multiplication table of 5 from 1 to 15. Display the table in alternate row color using ngSwitch and ngFor Directives

Difficulty: Medium

Display the multiplication table of the user choice. Display the table in alternate row color using ngSwitch and ngFor Directives

Difficulty: Hard

Display the multiplication table of the user choice. Display the table in alternate row color using ngSwitch and ngFor Directives. All the user to choose those 2 alternate row colors.

Angular Services

Difficulty: Easy

Create a login-service to validate the user login. From login component, call the login service method.

Difficulty: Medium

Create a login-service with a validateLogin function to validate the user login. Alternately, try From login component, call the login service method.

On successful login, display the post title and body using the http path given in the tutorial.

Difficulty: Hard

Create a login-service with a validateLogin function. Call this service from login component. Get the validation status as return value of the function and display the result in a div.

On successful login, display the post title and body using http path given in the tutorial. Also handle life cycle events to destroy the component after completion of the function.

HttpClient Library

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Component Life Cycle Hooks

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Progressive Part Practice Assignments

Difficulty: Easy

Create an angular project HRMS with employee module. Allow users to add, remove employees from the employees array declared in the component.ts and display the list of employees in a tabular form. Allow users to navigate to Employee form and Employee list using routing. Use appropriate directives and services to achive this.

Difficulty: Medium

Difficulty: Hard

Angular Forms

Difficulty: Easy

Create employee component to the HRMS project. Import form module and bind the form input elements with the component class properties. Import FormGroup and use FormBuilder to group the form elements in the component class.

Difficulty: Medium

Create employee component to the HRMS project. Import form module and bind the form input elements with the component class properties.

Import FormGroup and use FormBuilder to group the form elements in the component class.

Display the error messages using div with appropriate messages.

Difficulty: Hard

Create employee component to the HRMS project. Import form module and bind the form input elements with the component class properties.

Import FormGroup and use FormBuilder to group the form elements in the component class.

Use \*ngIf directive to display appropriate messages in case of valid or invalid submission.

Reactive Forms

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Form Validations

Difficulty: Easy

In HRMS project, Allow users to enter new Employee Details. Include an array 'skillset' in the component and allow the user to choose number of skills they would like to add. Accordingly, create textboxes dynamically in the form.

Difficulty: Medium

In HRMS project, Allow users to enter new Employee Details. Include an array 'skillset' in the component and allow the user to choose number of skills they would like to add. Accordingly, create textboxes dynamically in the form.

Validate the employee firstName and lastName that should not contain any special characters except a '.'

Difficulty: Hard

In HRMS project, Allow users to enter new Employee Details. Include an array 'skillset' in the component and allow the user to choose number of skills they would like to add. Accordingly, create textboxes dynamically in the form.

Validate the firstName and lastName that should not contain any special characters except a '.' and the firstName and lastName cannot be same.

Dynamic Forms

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Progressive Part Practice Assignments

Difficulty: Easy

Create an angular project HRMS with employee module. Allow users to add, remove employees from the employees array declared in the component.ts and display the list of employees in a tabular form. Allow users to navigate to Employee form and Employee list. In the employee registration form, include skillset which is an array in the component. Allow users to enter noOfSkills and based on that create as many number of user inputs dynamically. Validate the user inputs that name should not contain special characters except '.'

Difficulty: Medium

Difficulty: Hard

Angular Animations

Difficulty: Easy

Create an animation by importing necessary angular modules to the project module.

Difficulty: Medium

Create an animation as per the instruction given below, by importing necessary angular modules to the project module.

Change the background color, width and height of the div on clicking a toggle button.

Export the component for reuse.

Difficulty: Hard

Create an animation as given below, by importing necessary angular modules to the project module.

Create a toggle accordion menu, when clicked, it should show the menu items and hide when clicked again. Export the component to resue.

Animation States

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Reusable Animations

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Observables and the RxJS

Difficulty: Easy

Use the third party library 'RxJS for the observables to subscribe the data

Difficulty: Medium

Use the third party library 'RxJS for the observables to subscribe the data

Difficulty: Hard

Use the third party library 'RxJS for the observables to subscribe the data

Progressive Part Practice Assignments

Difficulty: Easy

Create an angular project HRMS with employee module. Allow users to add, remove employees from the employees array declared in the component.ts and display the list of employees in a tabular form. Allow users to navigate to Employee form and Employee list using routing. Use appropriate directives and services to achive this. Use Animations to make the page look vibrant. Display Menus which expands as toggle using Angular animations. Export the animations to reuse.

Difficulty: Medium

Difficulty: Hard

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Whole Practice Assignment

Difficulty: Easy

Create Angular Project for Attendance Management. Allow users to choose student from the dropdown, populated from an array declared in the component. The user should be allowed to enter a number and dynamically display as many rows to enter the attendance using dynamic forms, give animations to enhance the presentation. Validate the data and submit the data to the component.

Difficulty: Medium

Difficulty: Hard

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard

Difficulty: Easy

Difficulty: Medium

Difficulty: Hard