

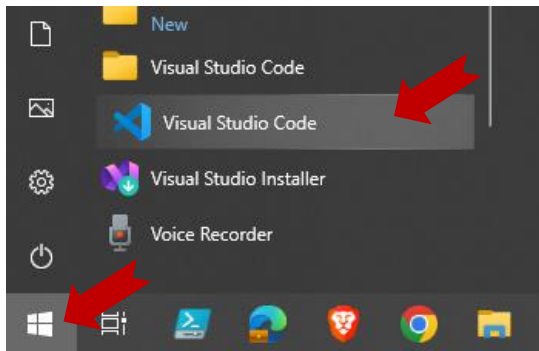
Setup TypeScript Environment

Note: This setup assumes the following software already installed

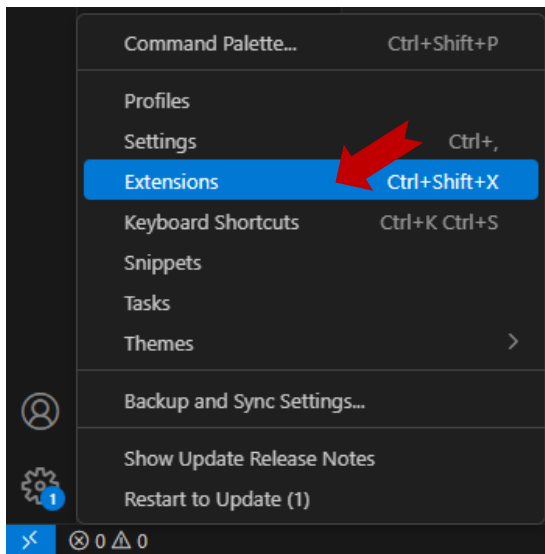
- ✓ Web Browser (Edge, Chrome, Brave, etc...)
- ✓ Latest NodeJS
- ✓ Latest Visual Studio Code

1) Adding Extensions to Visual Studio Code

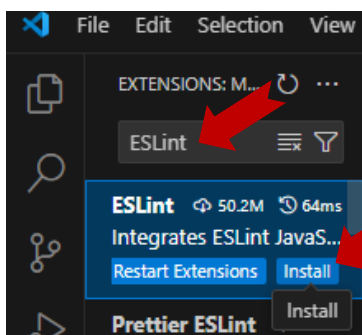
a. Start Visual Studio Code



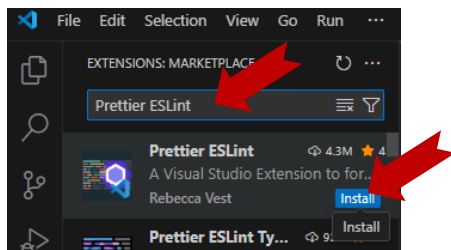
b. In Visual Studio Code, Select Extensions



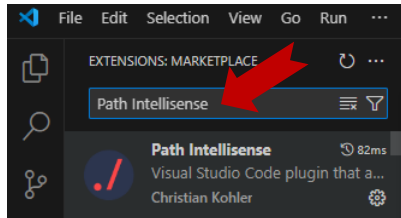
c. Install **ESLint**.



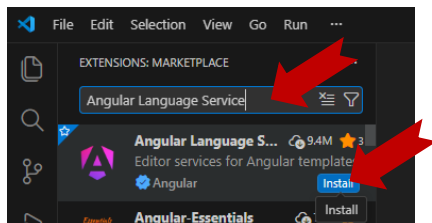
d. Install **Prettier ESLint**.



e. Install **Path Intellisense**.



f. Install **Angular Language Service** (This is only for Angular development)



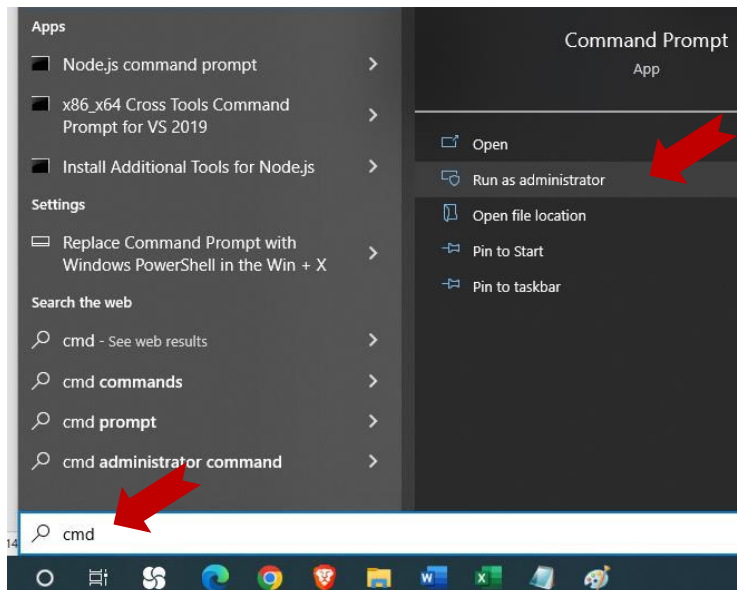
g. Close Visual Studio Code

2) Install TypeScript Compiler

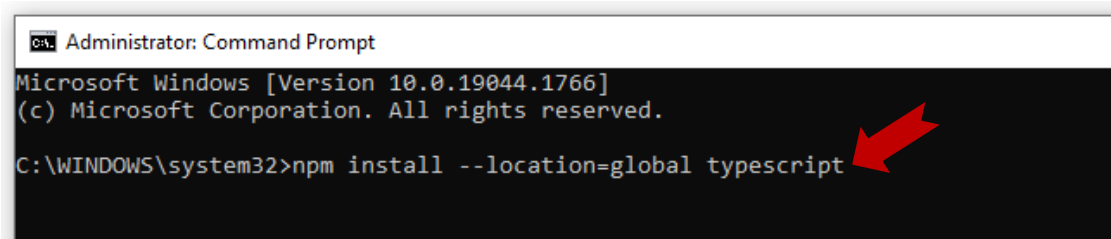
a. Click Window **Start** button.



b. When the menu appears, just type **cmd** and select **Run as Administrator**.



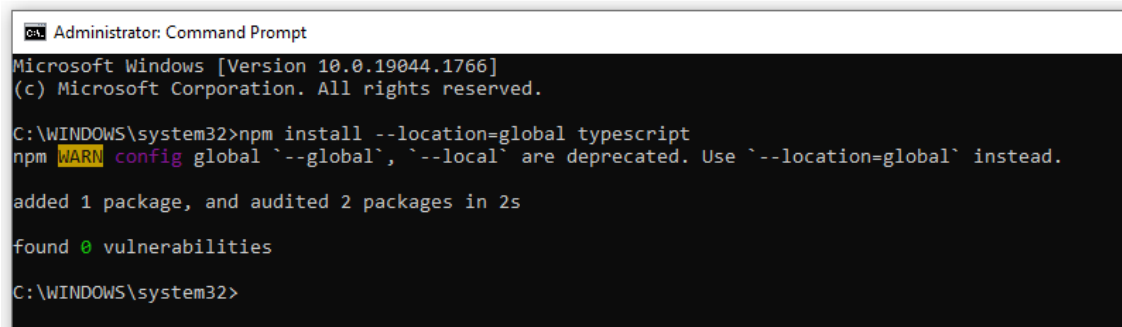
- c. Under the command box, type **npm install --location=global typescript**.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>npm install --location=global typescript
```

Wait until the following appears.



```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>npm install --location=global typescript
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

added 1 package, and audited 2 packages in 2s

found 0 vulnerabilities

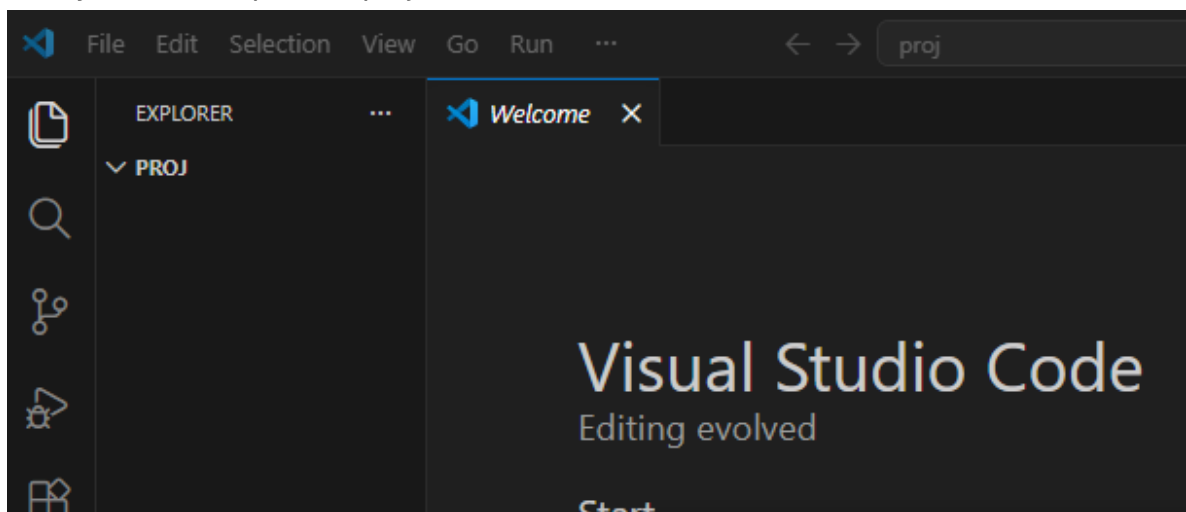
C:\WINDOWS\system32>
```

- d. Leave this command window remains open.

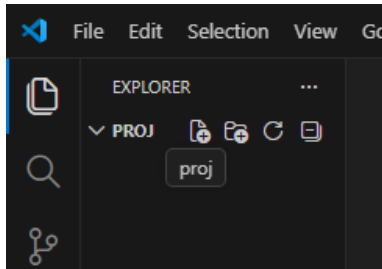
3) Setting up new project environment

- a. Type the following commands in the previous command window. (In case no command window, start a new one with admin right again).
- b. Type and run the following commands one by one:
- `cls`
 - `cd\`
 - `md proj`
 - `cd proj`
 - `code .` *(There is an empty space in between code and dot)*

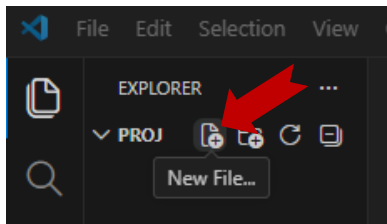
Now you should open the project folder from Visual Studio Code.



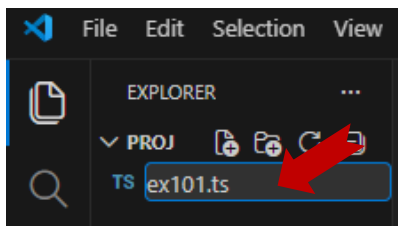
- c. Move the mouse cursor to the PROJ folder under the Explorer, you should see few icons appear



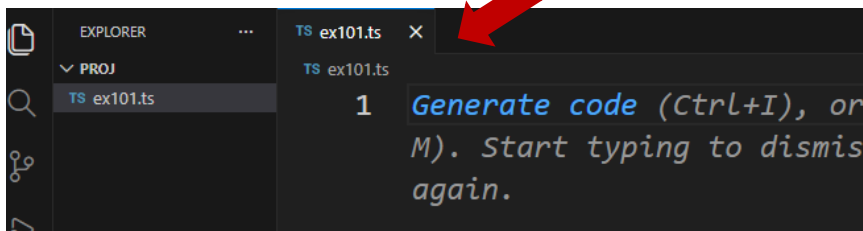
- d. Click New File.



- e. Give the file name as **ex101.ts**. Press **Enter**.



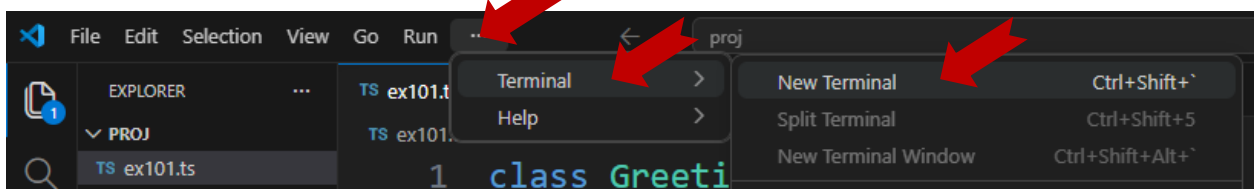
- f. A new empty TypeScript file was created.



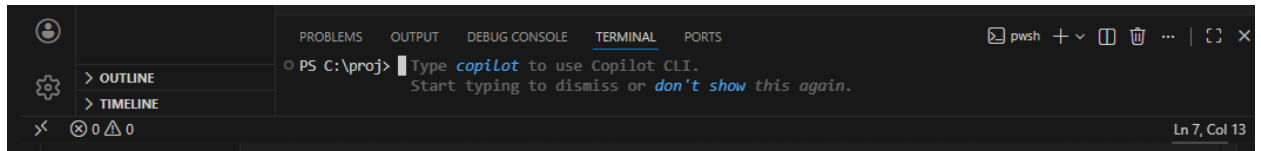
- g. Type the following code and save the file:

```
class Greeting{
  greet():void{
    console.log("Hello World!!!")
  }
}
var obj = new Greeting();
obj.greet();
```

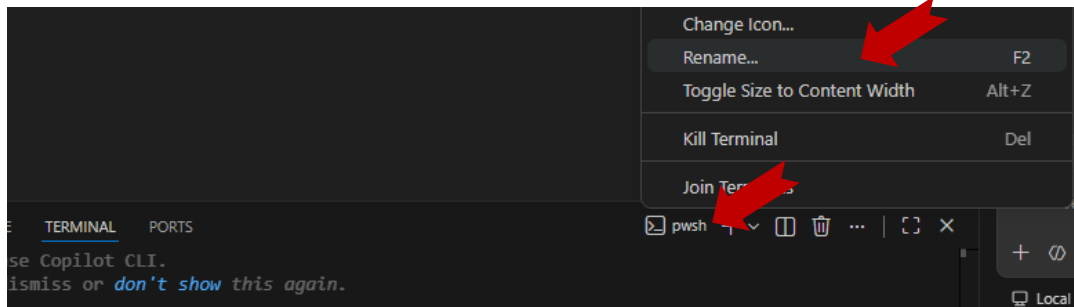
- h. Create new Terminal window.



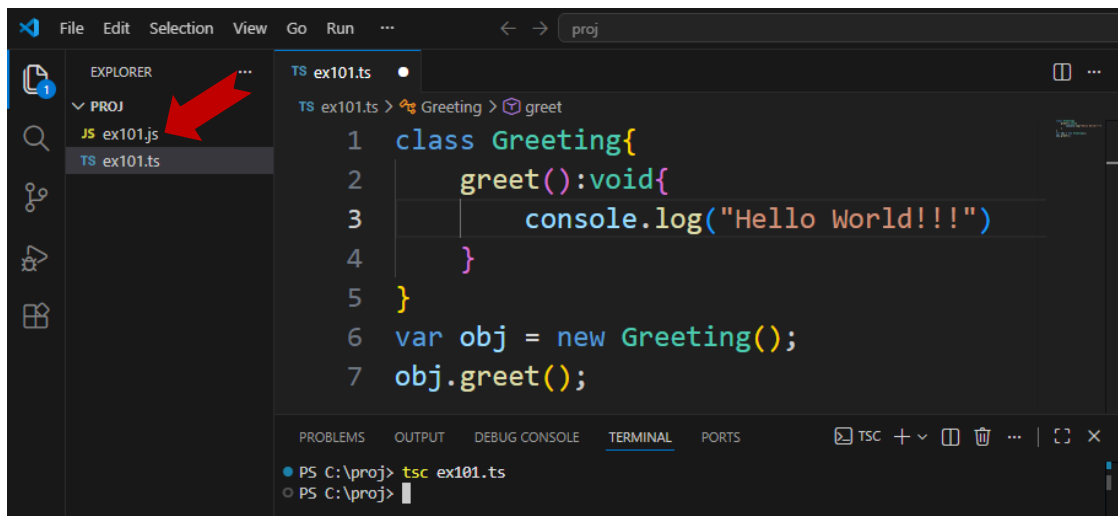
A terminal window will appear at the bottom of the editor



- i. Right-click to rename the terminal window as **TSC** (This window is for the TypeScript compiler).



- j. Type the command **tsc ex101.ts** in the terminal Window. Upon successful compilation, there will be a new file **ex101.js** generated.



At this point, TypeScript compiler is confirmed installed and work properly.

Notes: In case you are having problem with running script under VS Code, follow the following steps:

- Close the Visual Studio Code
- Press the windows-button on your keyboard
- Type 'PowerShell'
- Right-click Windows PowerShell and Run as Administrator
- Run the following command and confirm with 'Y'

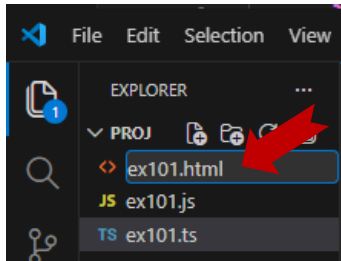
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine

- Close the PowerShell Window
- Restart the Visual Studio and repeat the step that caused error.

for more information: <https://www.roelpeters.be/solved-running-scripts-is-disabled-on-this-system-in-powershell/>

4) Testing Program

- a. Create a new **ex101.html** file.

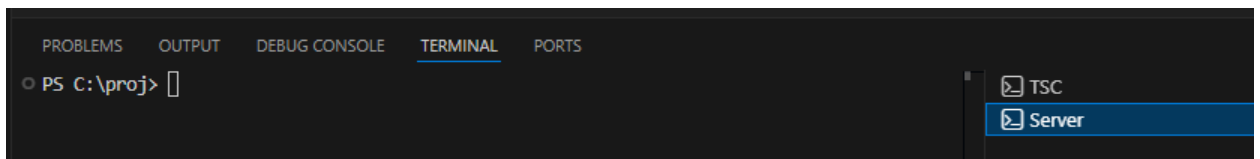


- b. In the ex101.html editor, type **html:5** then press **Enter**. HTML script is auto generated for you. Change and add the following:

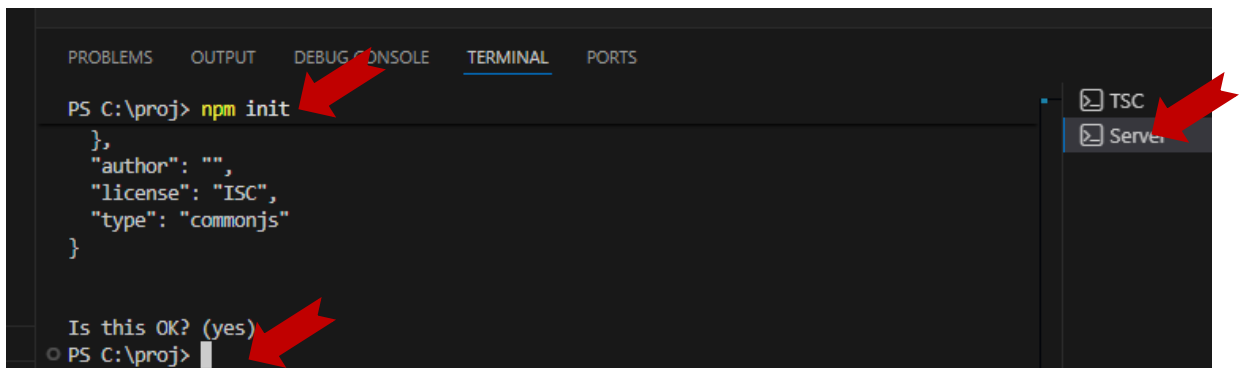


Save the file

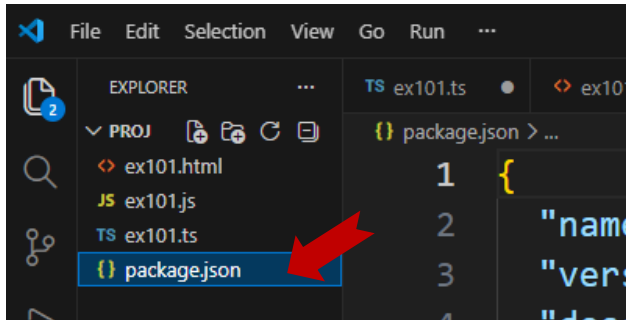
- c. Start a new terminal and rename it as **Server**. Now you should have 2 terminal windows.



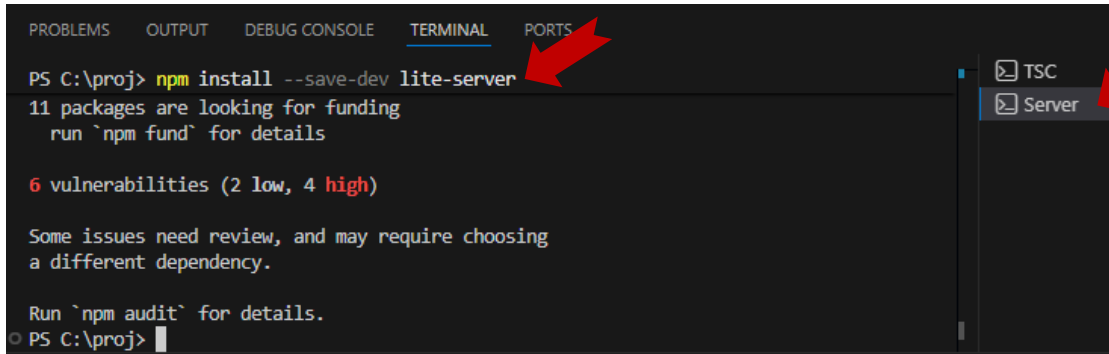
- d. Under the “Server” terminal, type the command **npm init**. Just press **Enter** to use all default values.



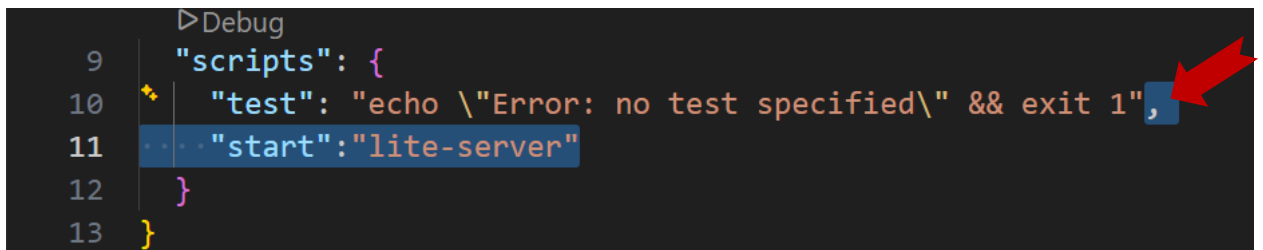
A new **package.json** file is generated



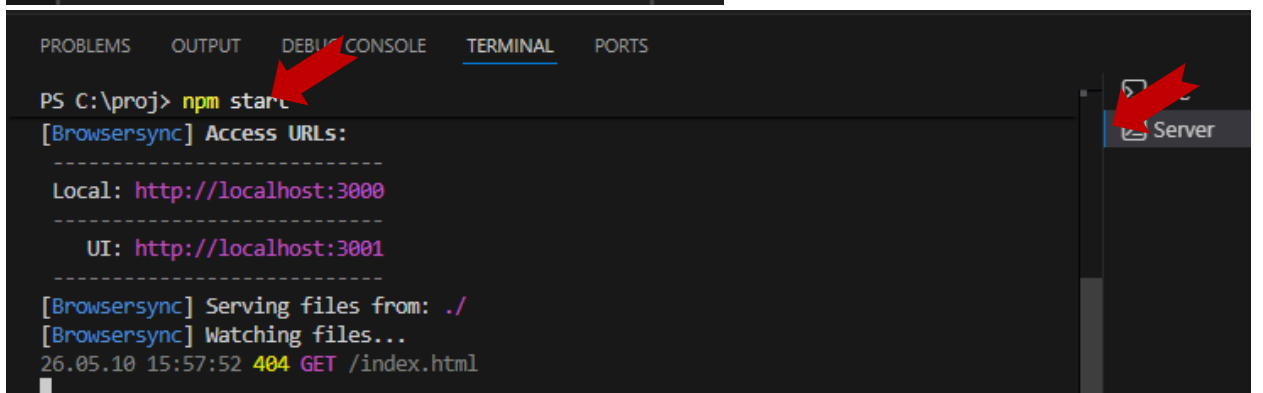
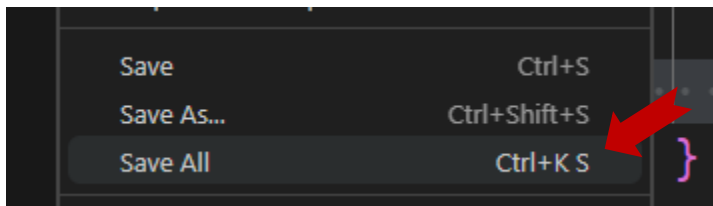
- e. Under the Server terminal, type **npm install --save-dev lite-server**.



- f. Under generated file **package.json**, add the following

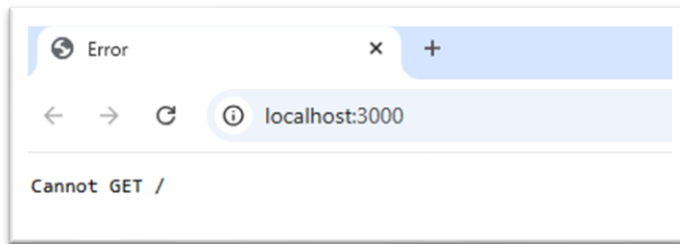


- g. Save all files and run the command **npm start** under Server terminal.

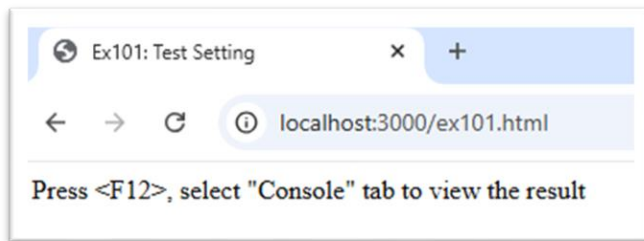


The Lite Server is running now.

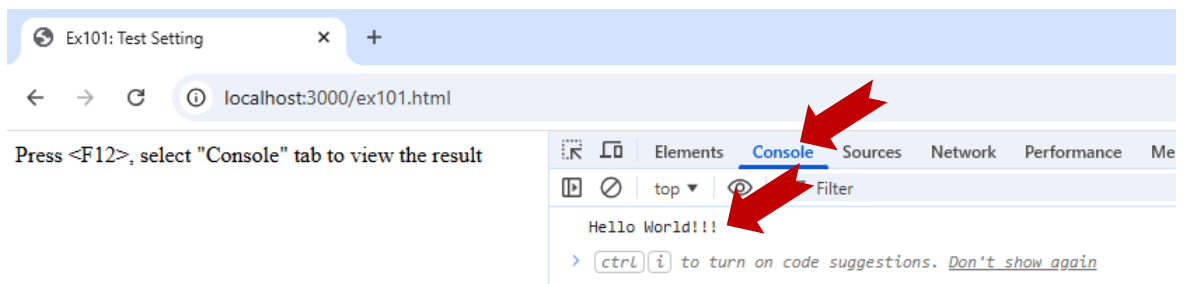
Meanwhile, the default web browser is showing up



- h. Don't worry about the error now. Just manually change the URL to:
<http://localhost:3000/ex101.html>.

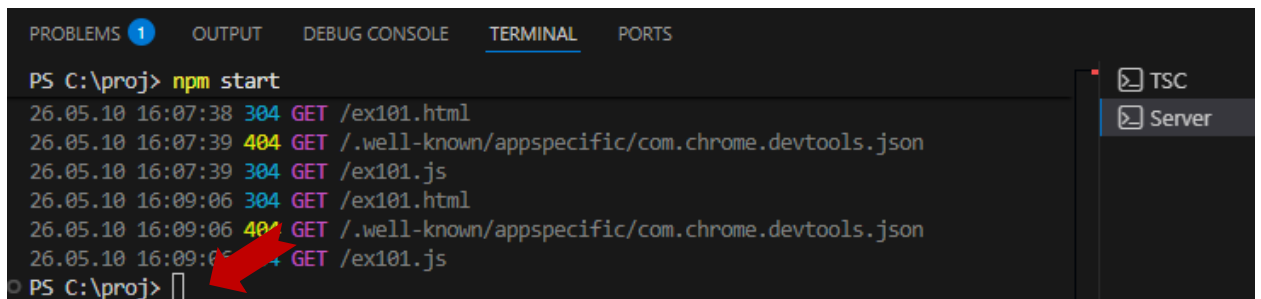


- i. From browser, press **F12** to show the **Developer Pane**.



Notes: You can use F12 key to toggle the developer pane. The developer pane for most browsers using F12. If your browser can't get the Developer Pane with F12, you need to find out which is the right key for it.

- j. Under Server terminal, press **<Ctrl>+C** to stop the server



5) Add a new **index.html** file

- a. In explorer, add new file. Name the file as **index.html**, and prepare the following script:

```
TS ex101.ts 1 JS ex101.js index.html ex101.html package.json
index.html > html > body
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Project Home Page</title>
7 </head>
8 <body>
9   <table border="1">
10     <tr><th colspan="2">Exercises</th></tr>
11     <tr><th>Code</th><th>Description</th></tr>
12     <tr><td><a href="ex101.html">ex101</a></td><td>Test Setting</td></tr>
13   </table>
14 </body>
15 </html>
```

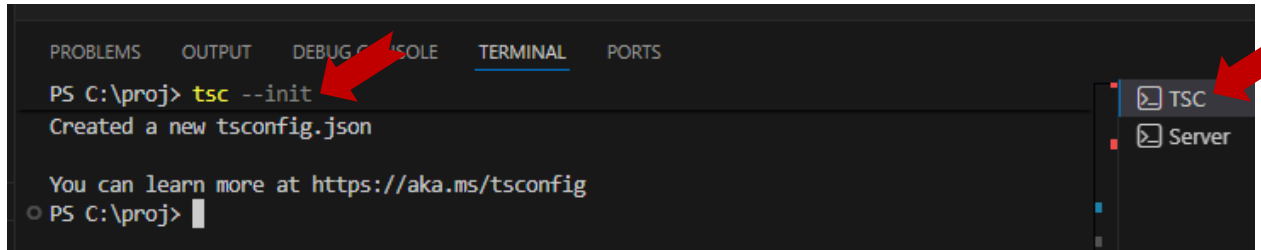
- b. In **package.json**, change the main attribute:

```
TS ex101.ts 1 JS ex101.js index.html ex101.html package.json X
package.json > abc main
1 {
2   "name": "proj",
3   "version": "1.0.0",
4   "description": "",
5   "license": "ISC",
6   "author": "",
7   "type": "commonjs",
8   "main": "index.html",
9   "scripts": {
10     "test": "echo \"Error: no test specified\" && exit 1",
11     "start": "lite-server"
12   },
13   "devDependencies": {
14     "lite-server": "^2.6.1"
15   }
}
```

- c. Save all files
- d. Under Server terminal, use up arrow key or type **npm start** to run the server again.
Now the browser launches index.html Home Page.

6) Configure TypeScript compiler auto detect changes

- a. Select the TSC terminal window, and type command **tsc --init**



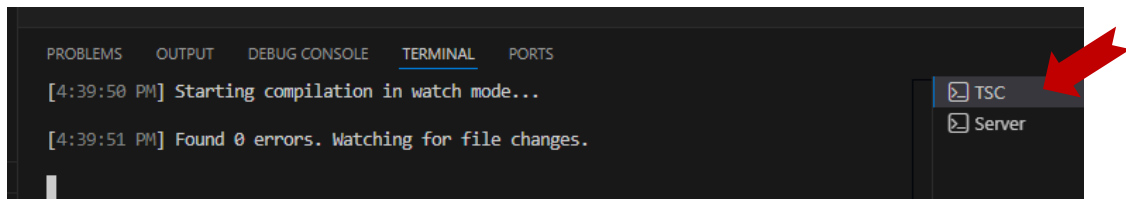
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\proj> tsc --init
Created a new tsconfig.json

You can learn more at https://aka.ms/tsconfig
PS C:\proj>
```

- b. You will realize that **tsconfig.json** file is generated. Replace it with following script:

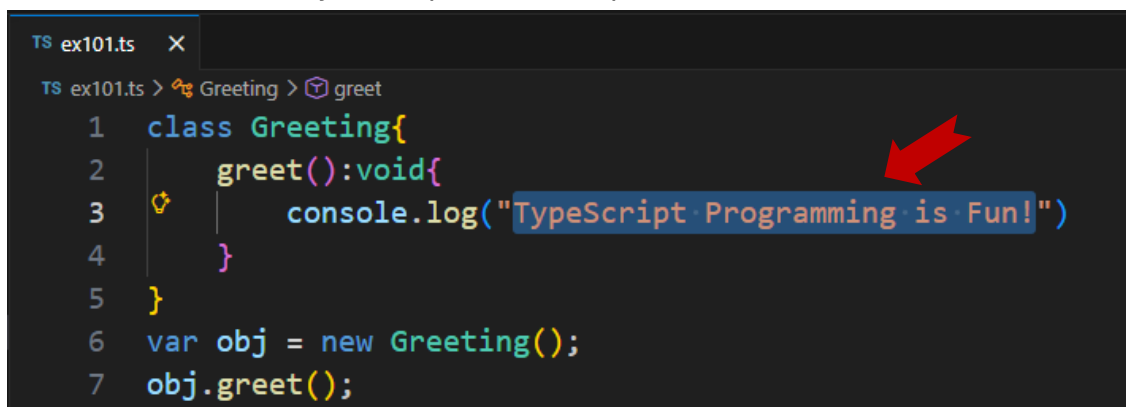
```
{
  "compilerOptions": {
    "module": "commonjs",
    "target": "es2016",
    "noImplicitAny": false,
    "sourceMap": true,
    "paths": {
      // "*": ["node_modules/*", "src/types/*"]
    }
  },
  "exclude": [
    "node_modules",
    "*.specs.ts"
  ]
}
```

- c. Save all files, and type the command **tsc -w** under TSC terminal window:



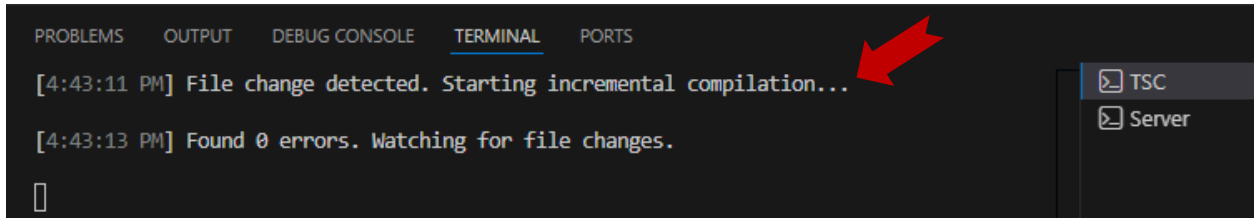
```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[4:39:50 PM] Starting compilation in watch mode...
[4:39:51 PM] Found 0 errors. Watching for file changes.
```

- d. Now, the TypeScript compiler is now observing any TypeScript source file change, and it will automatically recompile them. To prove it works, edit the ex101.ts file:



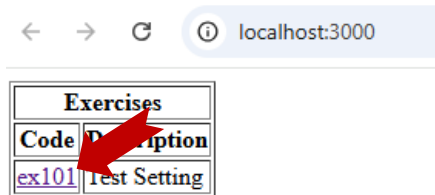
```
TS ex101.ts X
TS ex101.ts > Greeting > greet
1 class Greeting{
2   greet():void{
3     console.log("TypeScript Programming is Fun!")
4   }
5 }
6 var obj = new Greeting();
7 obj.greet();
```

- e. Save ex101.ts and observe the messages in TSC terminal window. Fix any problem until you get the following messages:

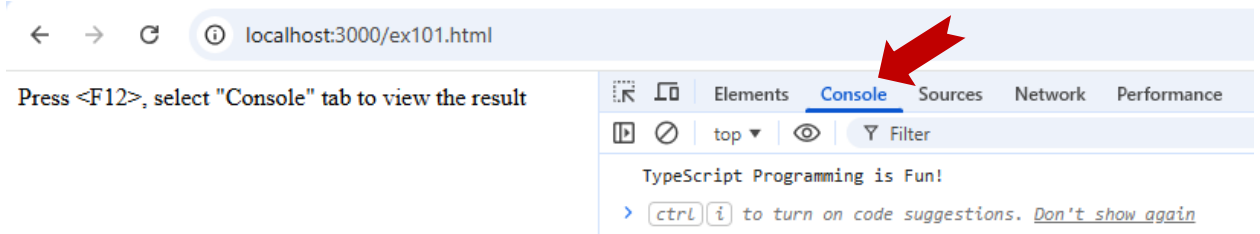


```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[4:43:11 PM] File change detected. Starting incremental compilation...
[4:43:13 PM] Found 0 errors. Watching for file changes.
[]
```

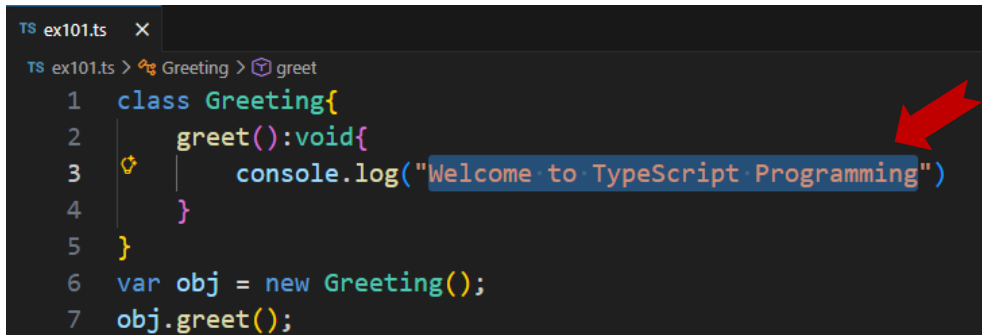
- f. Switch to **Server** terminal. Type **npm start** to start the server again. In the home page click the ex101.html link



- g. Press **F12**. You will see the updated message in console tab.

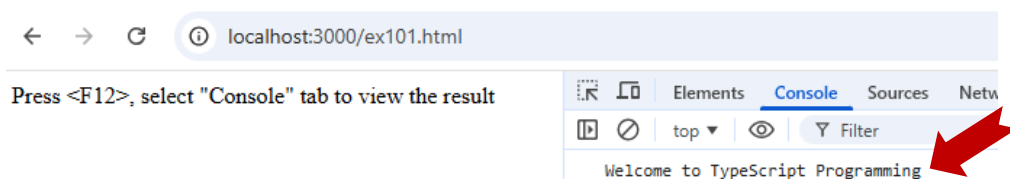


- h. Switch back to ex101.ts editor, change the following, and save the file.



```
TS ex101.ts x
TS ex101.ts > Greeting > greet
1 class Greeting{
2   greet():void{
3     console.log("Welcome to TypeScript Programming")
4   }
5 }
6 var obj = new Greeting();
7 obj.greet();
```

- i. Switch back to browser, you will get updated message



- j. Close the browser, Shutdown the lite-server, close the Visual Studio Code