

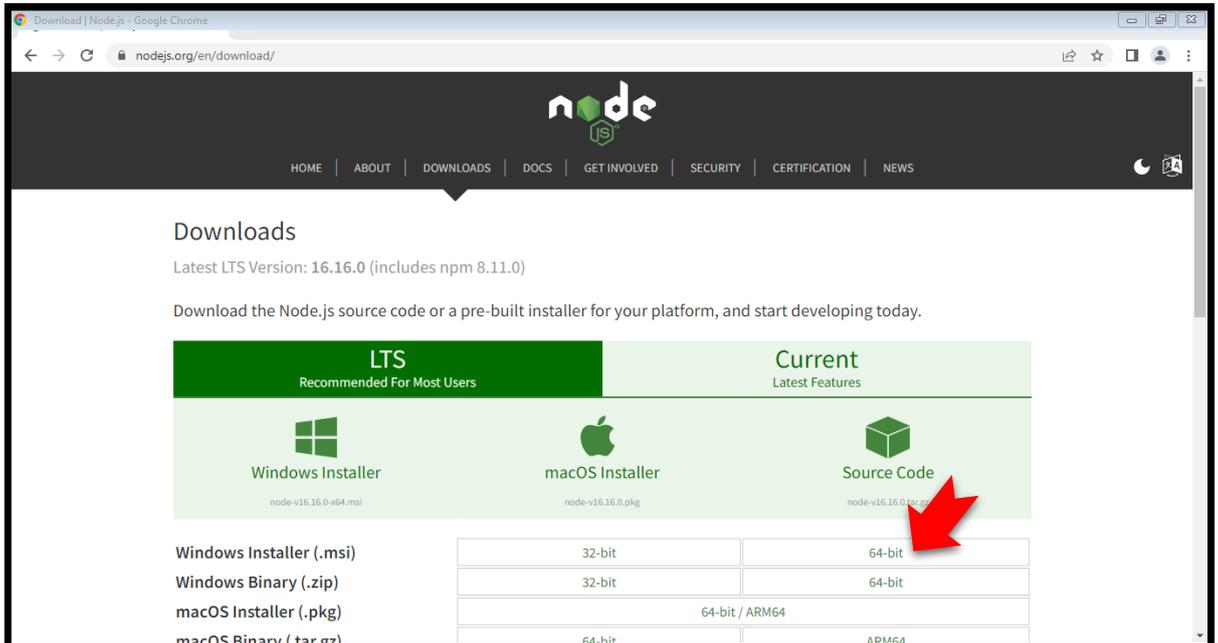
# Setup Guide

## Table of Contents:

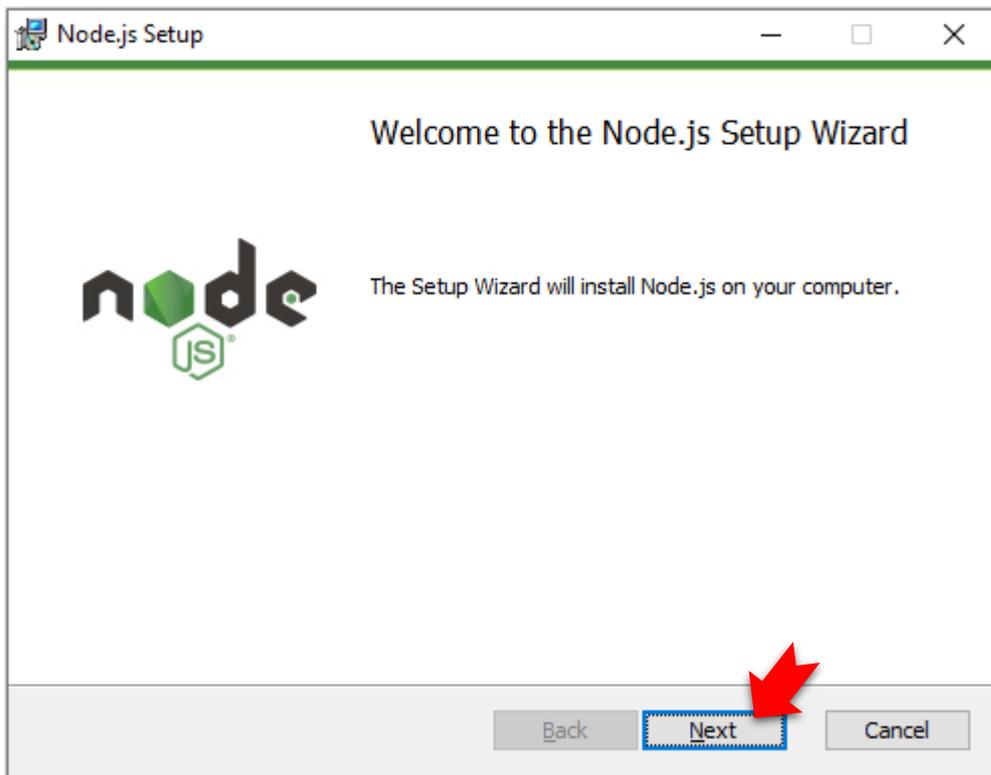
- **Install NodeJS**
- **Install Visual Studio Code**
- **Adding Extensions to Visual Studio Code**
- **Install TypeScript Compiler**
- **Test Setup**

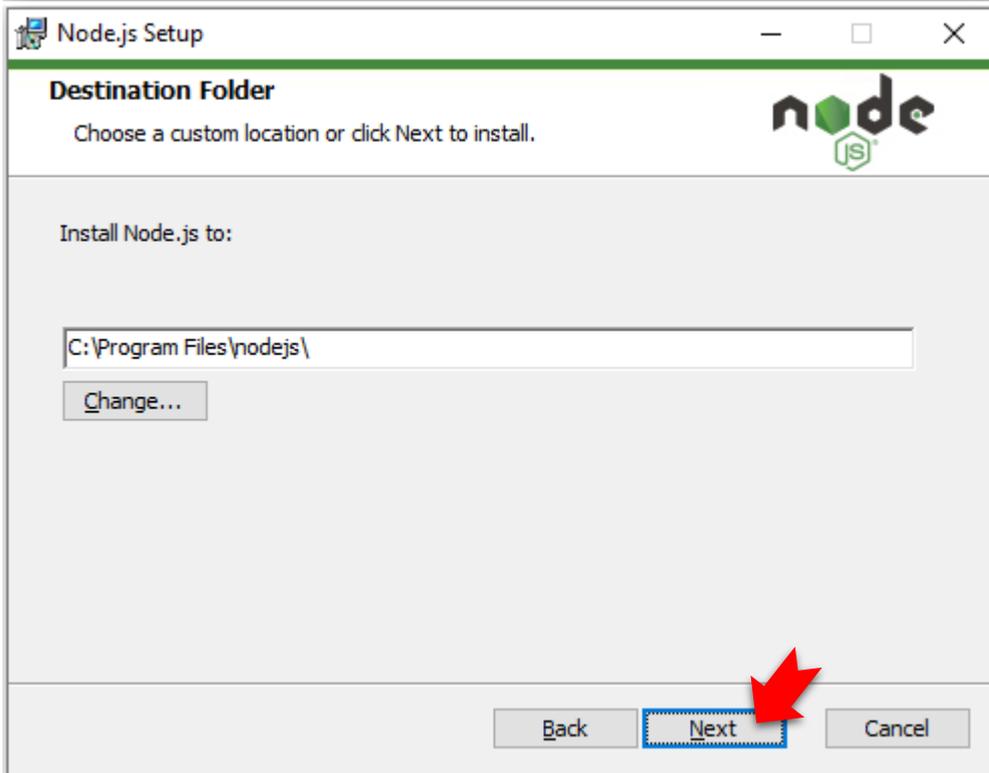
# Install NodeJS

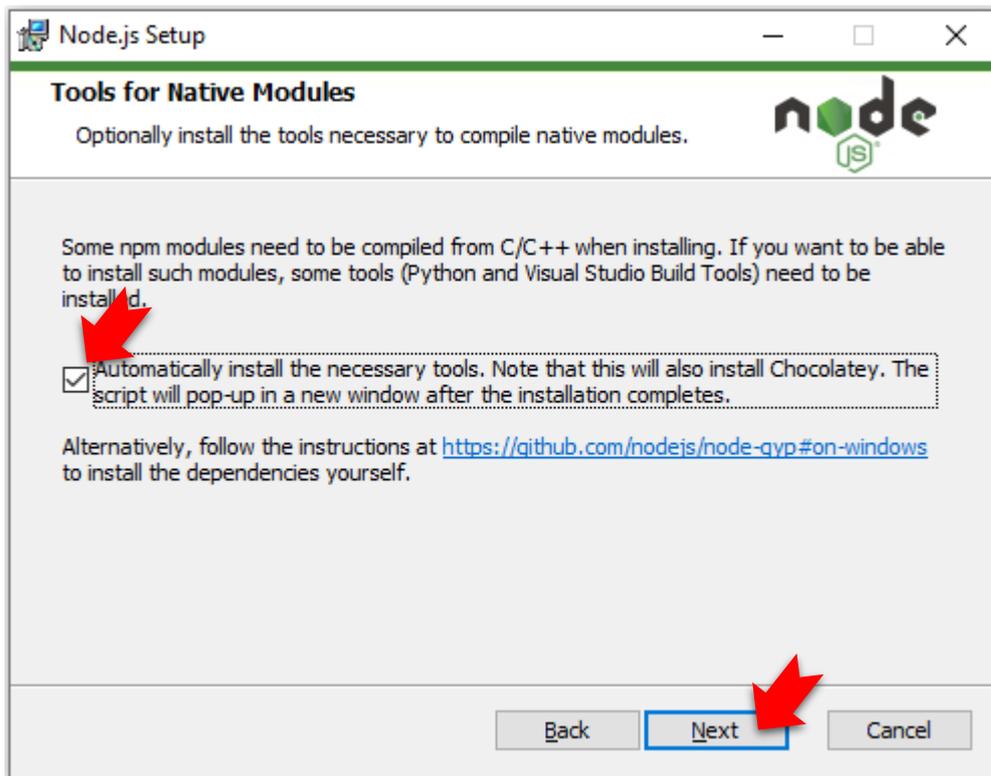
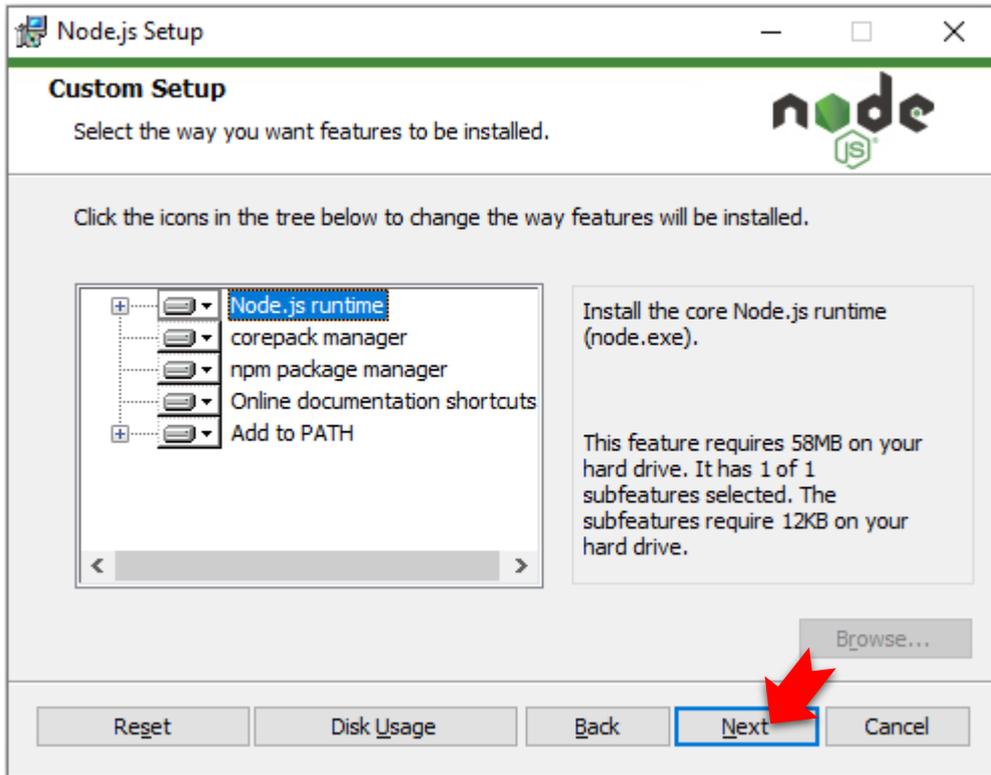
- Go to web-site: <https://nodejs.org/en/download>

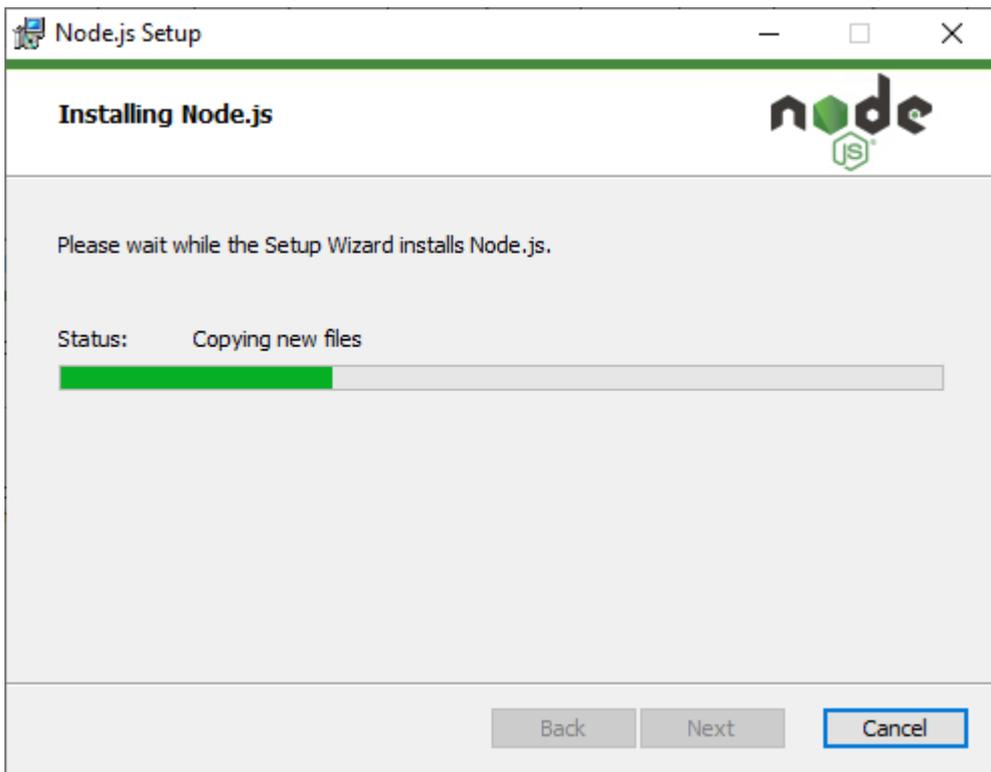
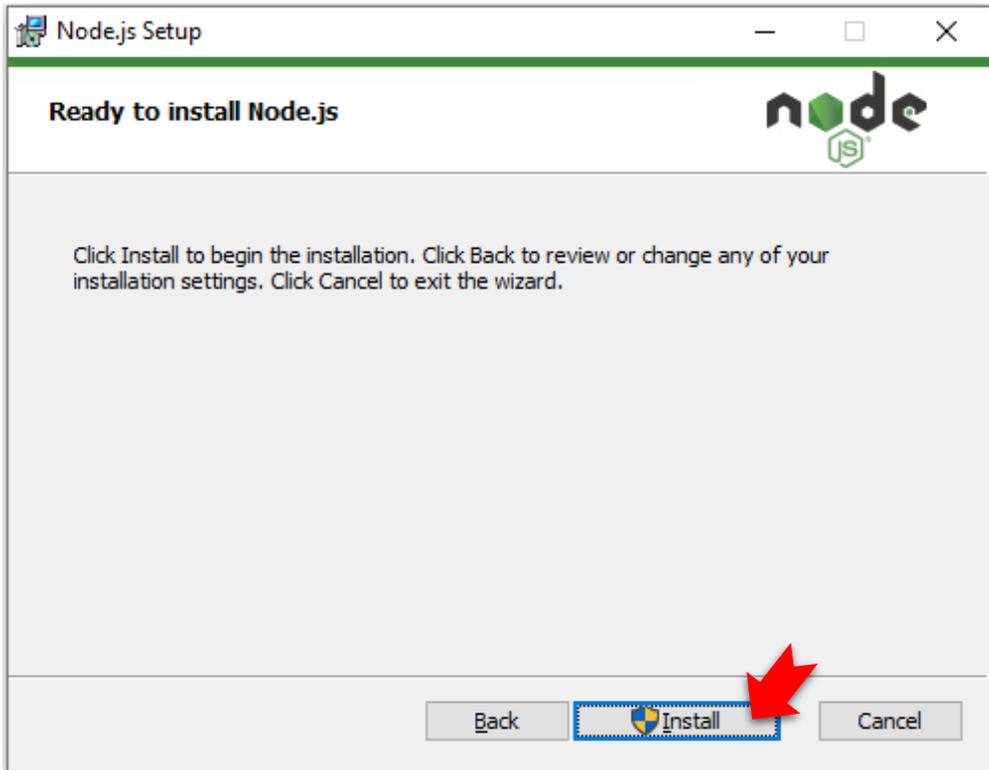


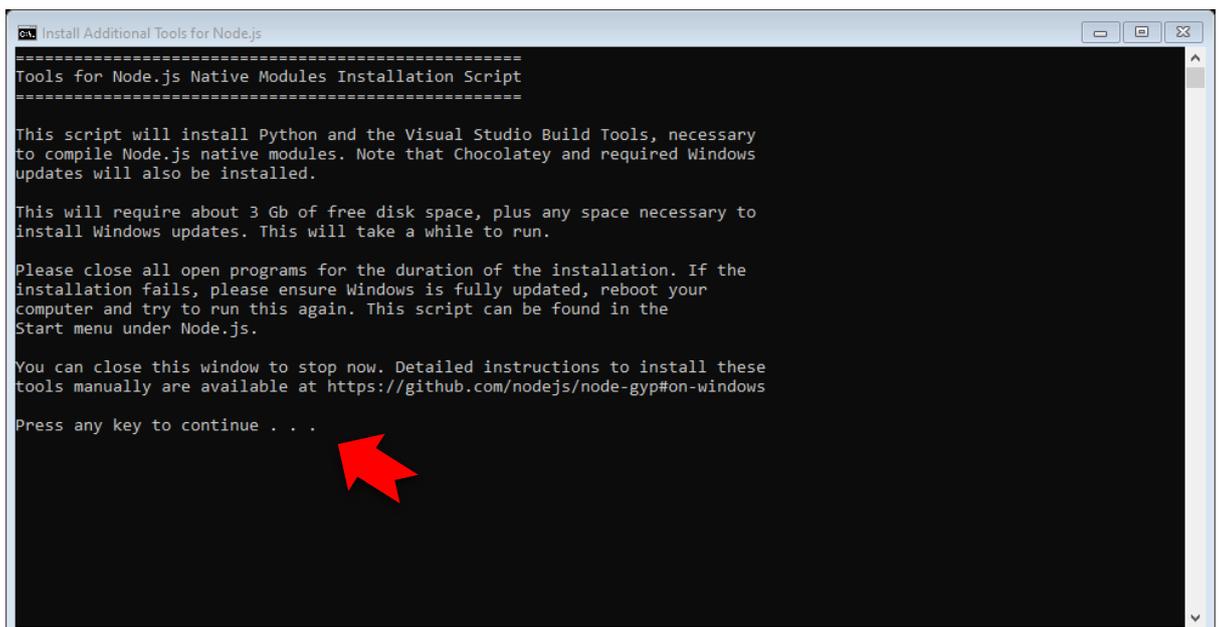
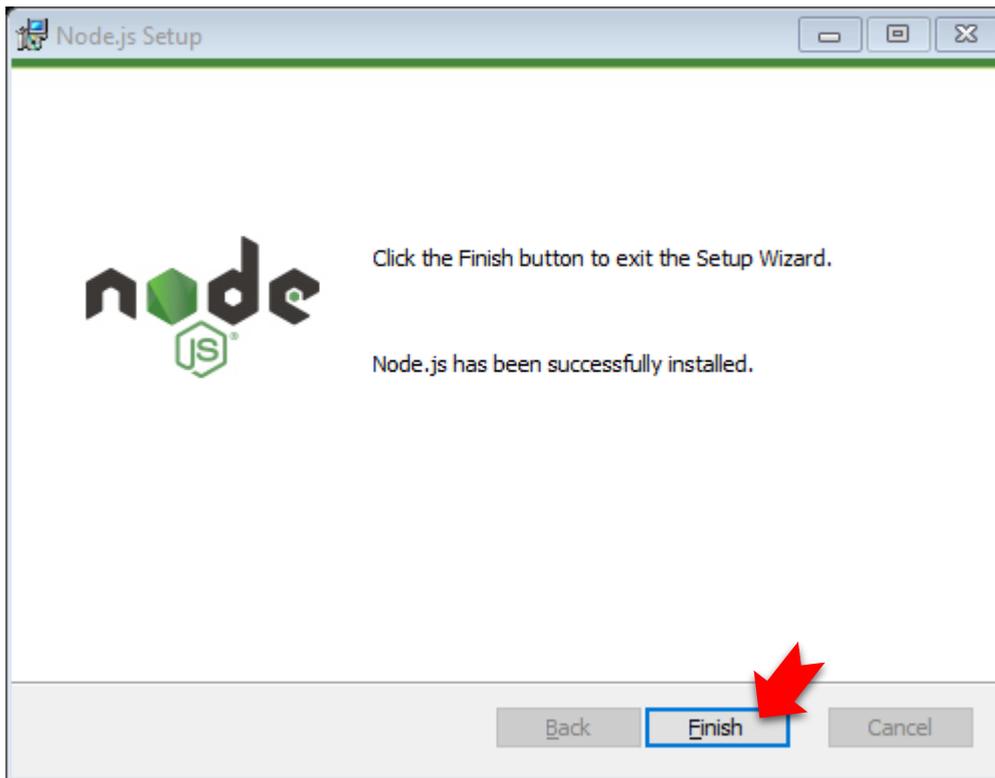
- Download and run the installer











Press any key to continue...

```
Install Additional Tools for Node.js
Using this script downloads third party software
-----
This script will direct to Chocolatey to install packages. By using
Chocolatey to install a package, you are accepting the license for the
application, executable(s), or other artifacts delivered to your machine as a
result of a Chocolatey install. This acceptance occurs whether you know the
license terms or not. Read and understand the license terms of the packages
being installed and their dependencies prior to installation:
- https://chocolatey.org/packages/chocolatey
- https://chocolatey.org/packages/python
- https://chocolatey.org/packages/visualstudio2019-workload-vctools

This script is provided AS-IS without any warranties of any kind
-----
Chocolatey has implemented security safeguards in their process to help
protect the community from malicious or pirated software, but any use of this
script is at your own risk. Please read the Chocolatey's legal terms of use
as well as how the community repository for Chocolatey.org is maintained.

Press any key to continue . . .
```



Press any key to continue...

```
Setup.docx - Word
Getting latest version of the Chocolatey package for download.
Not using proxy.
Getting Chocolatey from https://community.chocolatey.org/api/v2/package/chocolatey/1.1.0.
Downloading https://community.chocolatey.org/api/v2/package/chocolatey/1.1.0 to C:\Users\WINDOW-1\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip
Extracting C:\Users\WINDOW-1\AppData\Local\Temp\chocolatey\chocoInstall\chocolatey.zip to C:\Users\WINDOW-1\AppData\Local\Temp\chocolatey\chocoInstall
Installing Chocolatey on the local machine
Creating chocolatey\install as an environment variable (targeting 'Machine')
Setting Chocolatey\install to 'C:\ProgramData\chocolatey'
WARNING: It's very likely you will need to close and reopen your shell
before you can use choco.
Restricting write permissions to Administrators
We are setting up the Chocolatey package repository.
The packages themselves go to 'C:\ProgramData\chocolatey\lib'
(i.e. C:\ProgramData\chocolatey\lib\yourPackageName).
A shim file for the command line goes to 'C:\ProgramData\chocolatey\bin'
and points to an executable in 'C:\ProgramData\chocolatey\lib\yourPackageName'.

Creating Chocolatey folders if they do not already exist.

WARNING: You can safely ignore errors related to missing log files when
upgrading from a version of Chocolatey less than 0.9.9.
'batch file could not be found' is also safe to ignore.
'The system cannot find the file specified' - also safe.
chocolatey.nupkg file not installed in lib.
Attempting to locate it from bootstrapper.
PATH environment variable does not have C:\ProgramData\chocolatey\bin in it. Adding...
WARNING: Not setting tab completion: Profile file does not exist at 'C:\Users\window
10\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1'.
Chocolatey (choco.exe) is now ready.
You can call choco from anywhere, command line or powershell by typing choco.
Run choco /? for a list of functions.
You may need to shut down and restart powershell and/or consoles
first prior to using choco.
Ensuring Chocolatey commands are on the path
Ensuring chocolatey.nupkg is in the lib folder
Chocolatey v1.1.0
Upgrading the following packages:
python3 visualstudio2019-workload-vctools
By upgrading, you accept licenses for the packages.
python is not installed. Installing...
Progress: Downloading python3 3.10.5... 80%
```

```
Setup.docx - Word
[25d8:0001][2022-07-17T15:31:58] Releasing singleton lock.
[25d8:0001][2022-07-17T15:31:58] Singleton lock does not exist. Releasing singleton lock skipped.
[25d8:0001][2022-07-17T15:31:58] Closing the installer with exit code 0
[25d8:0001][2022-07-17T15:31:58] Exit code: 0
[25d8:0001][2022-07-17T15:31:58] Cleared previous session ID.
[25d8:0001][2022-07-17T15:31:59] Trying to remove channel manifest: C:\Users\Window 10\AppData\Local\Microsoft\VisualStudio\Packages\
Channels\A0FE1051\installChannelManifest.json
[25d8:0001][2022-07-17T15:31:59] Trying to remove product manifest: C:\Users\Window 10\AppData\Local\Microsoft\VisualStudio\Packages\
Channels\A0FE1051\install_catalog.json
visualstudio2019-workload-vctools has been installed.
visualstudio2019-workload-vctools may be able to be automatically uninstalled.
The upgrade of visualstudio2019-workload-vctools was successful.
Software install location not explicitly set, it could be in package or
default install location of installer.

Chocolatey upgraded 18/18 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

Upgraded:
- chocolatey-dotnetfx.extension v1.0.1
- visualstudio2019buildtools v16.11.16.0
- kb3033929 v1.0.5
- python3 v3.10.5
- chocolatey-windowsupdate.extension v1.0.4
- vcredist140 v14.32.31332
- kb2999226 v1.0.20181019
- visualstudio-installer v2.0.3
- kb2919355 v1.0.20160915
- chocolatey-core.extension v1.4.0
- kb2919442 v1.0.20160915
- chocolatey-visualstudio.extension v1.10.2
- vcredist2015 v14.0.24215.20170201
- chocolatey-compatibility.extension v1.0.0
- dotnetfx v4.8.0.20220524
- visualstudio2019-workload-vctools v1.0.1
- kb3035131 v1.0.3
- python v3.10.5

Packages requiring reboot:
- vcredist140 (exit code 3010)

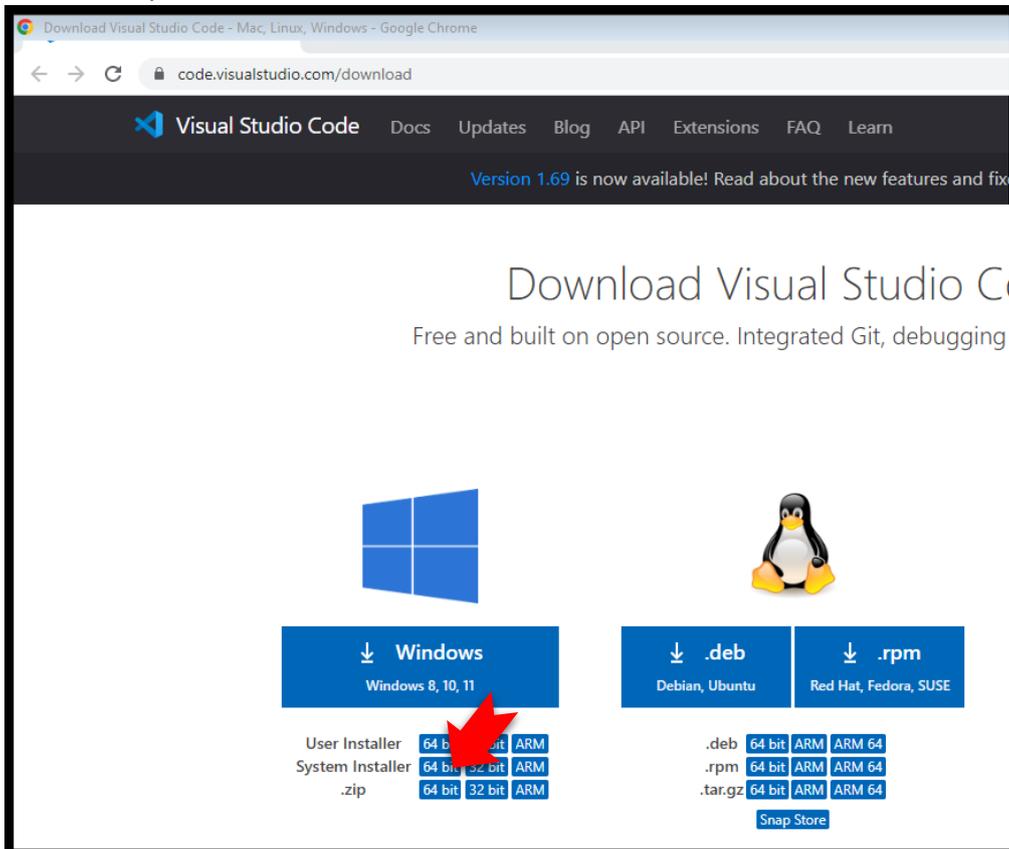
The recent package changes indicate a reboot is necessary.
Please reboot at your earliest convenience.
Type ENTER to exit:
```



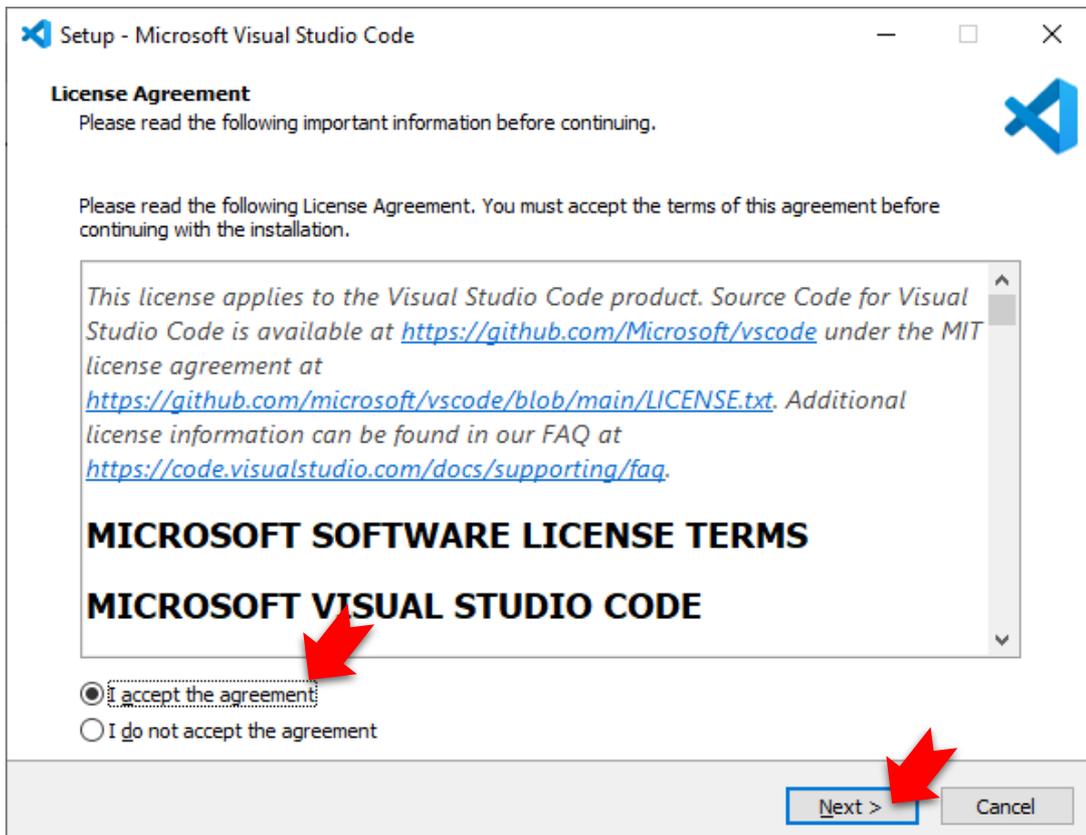
Press **Enter** to exit

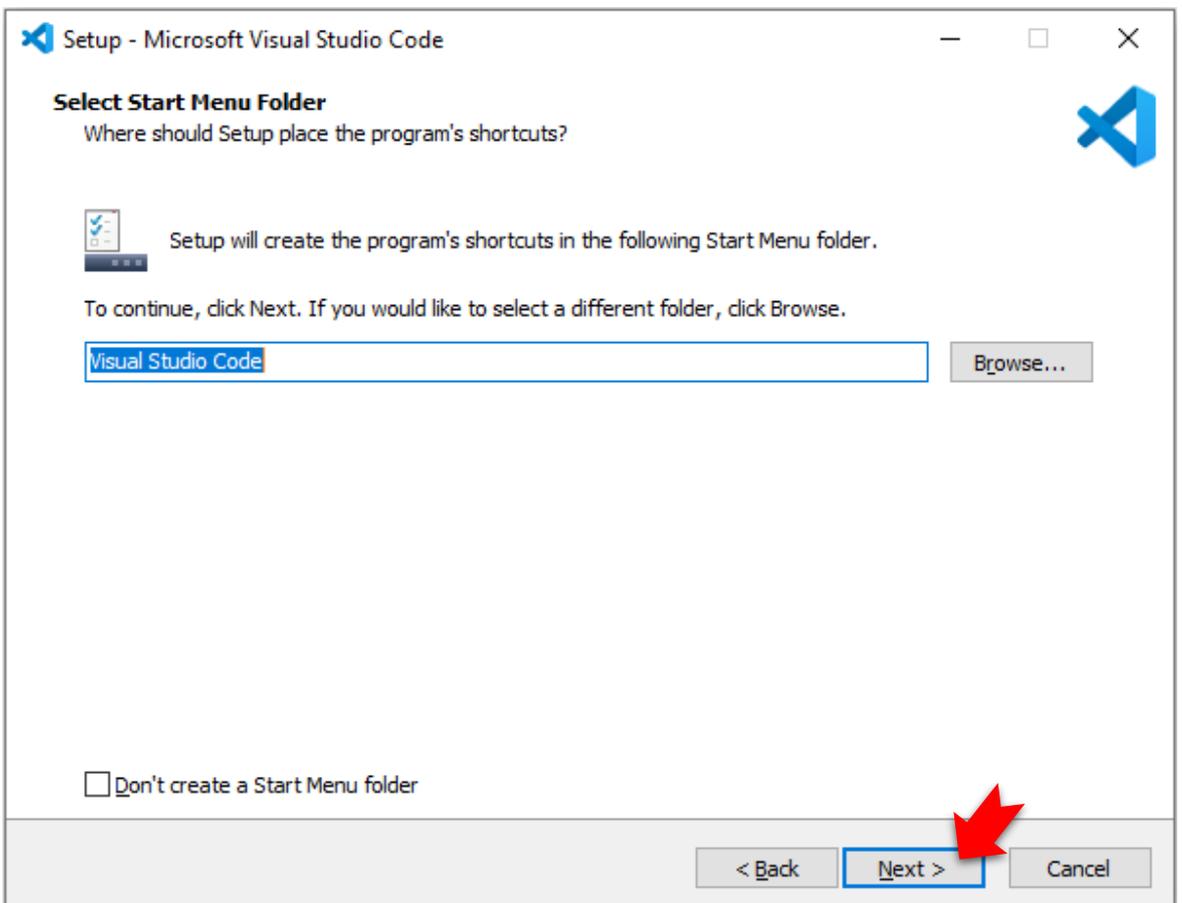
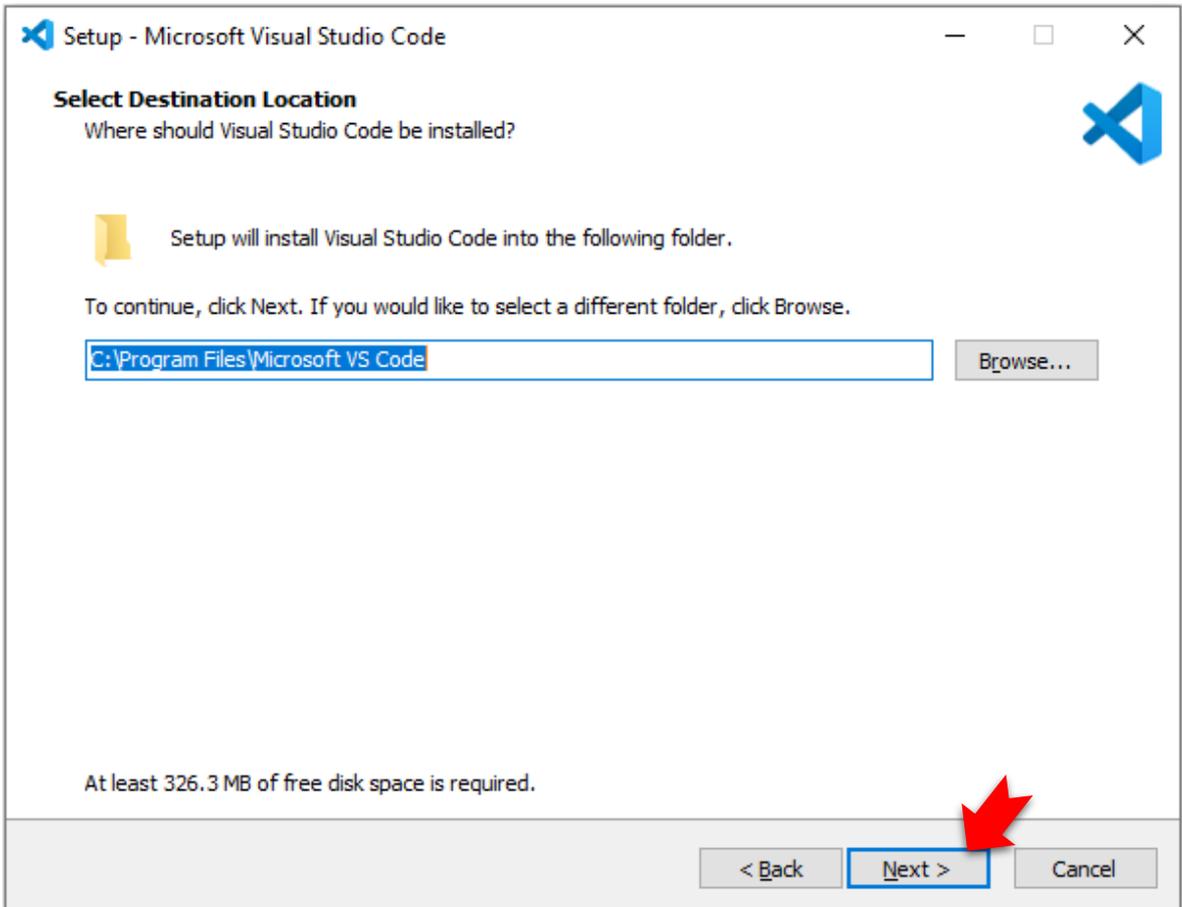
## Install Visual Studio Code

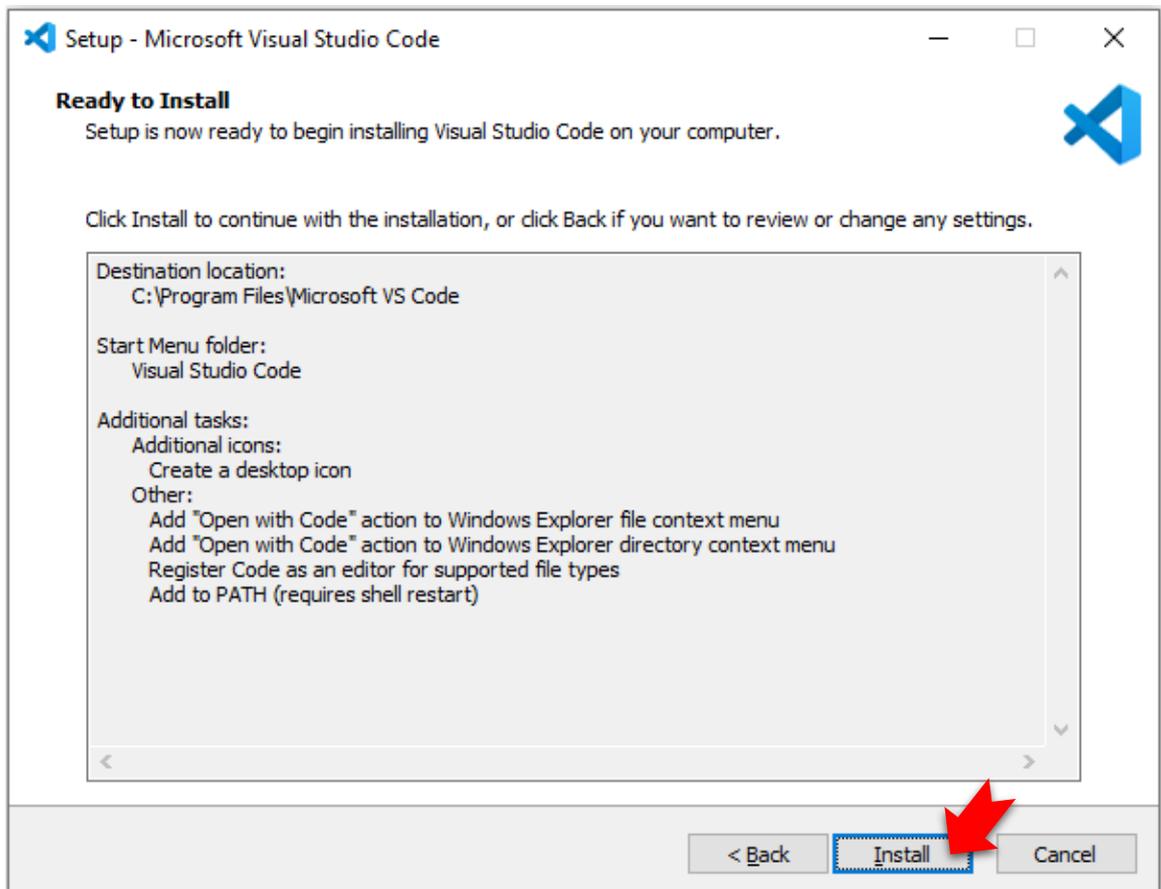
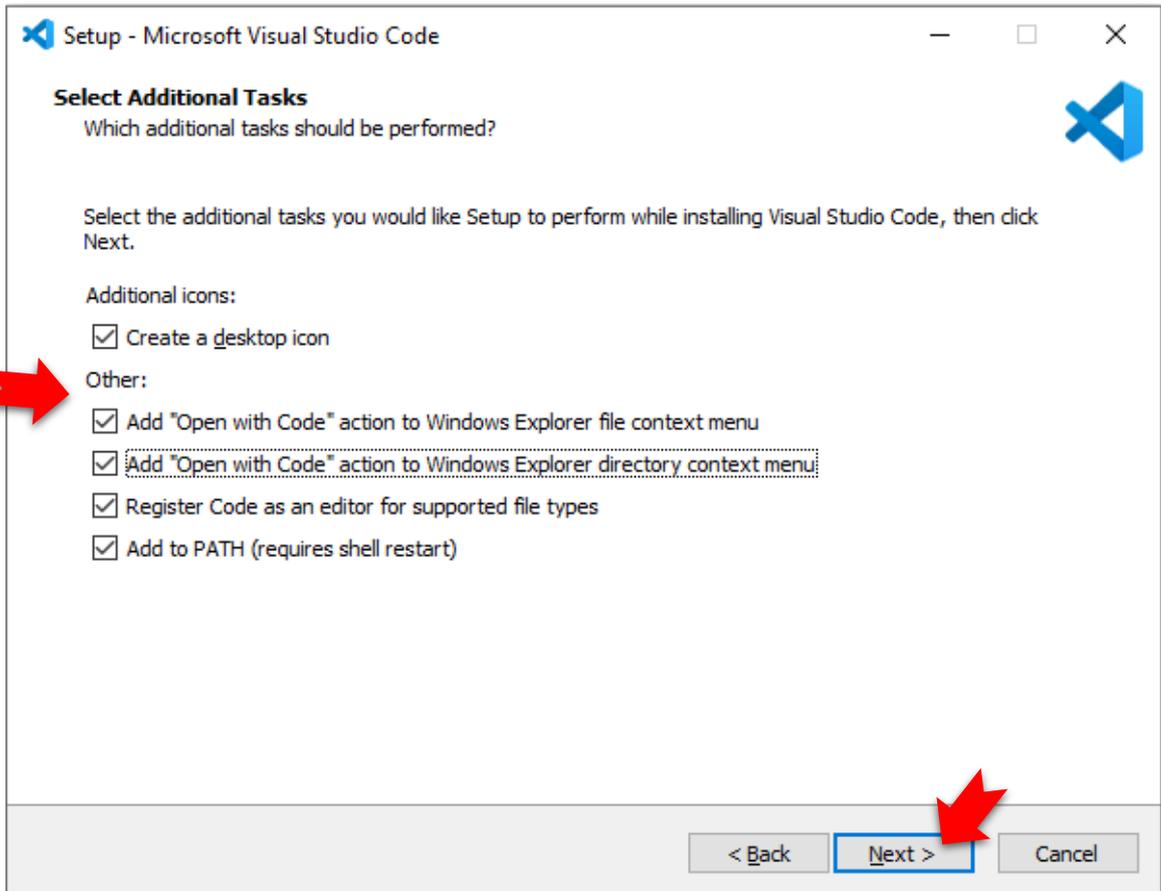
- Go to website: <https://code.visualstudio.com/download>
- Download System installer

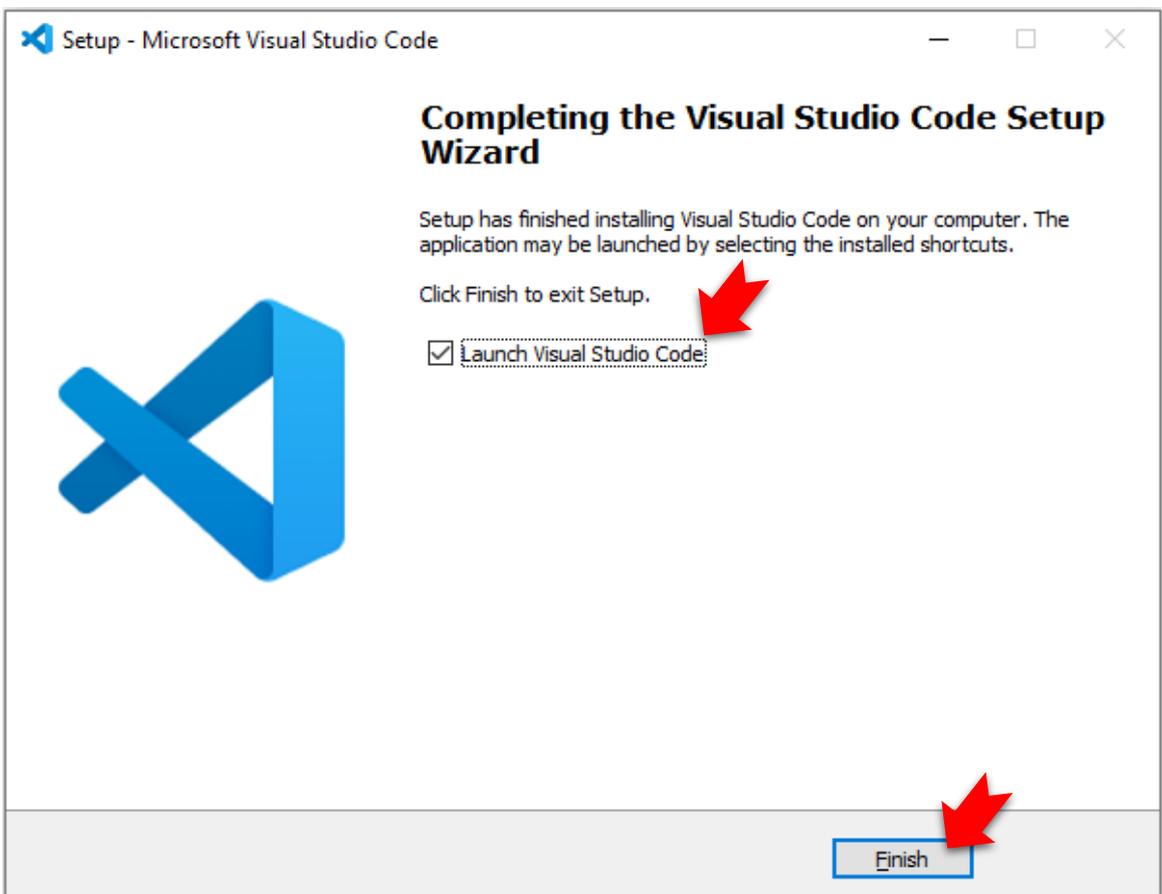
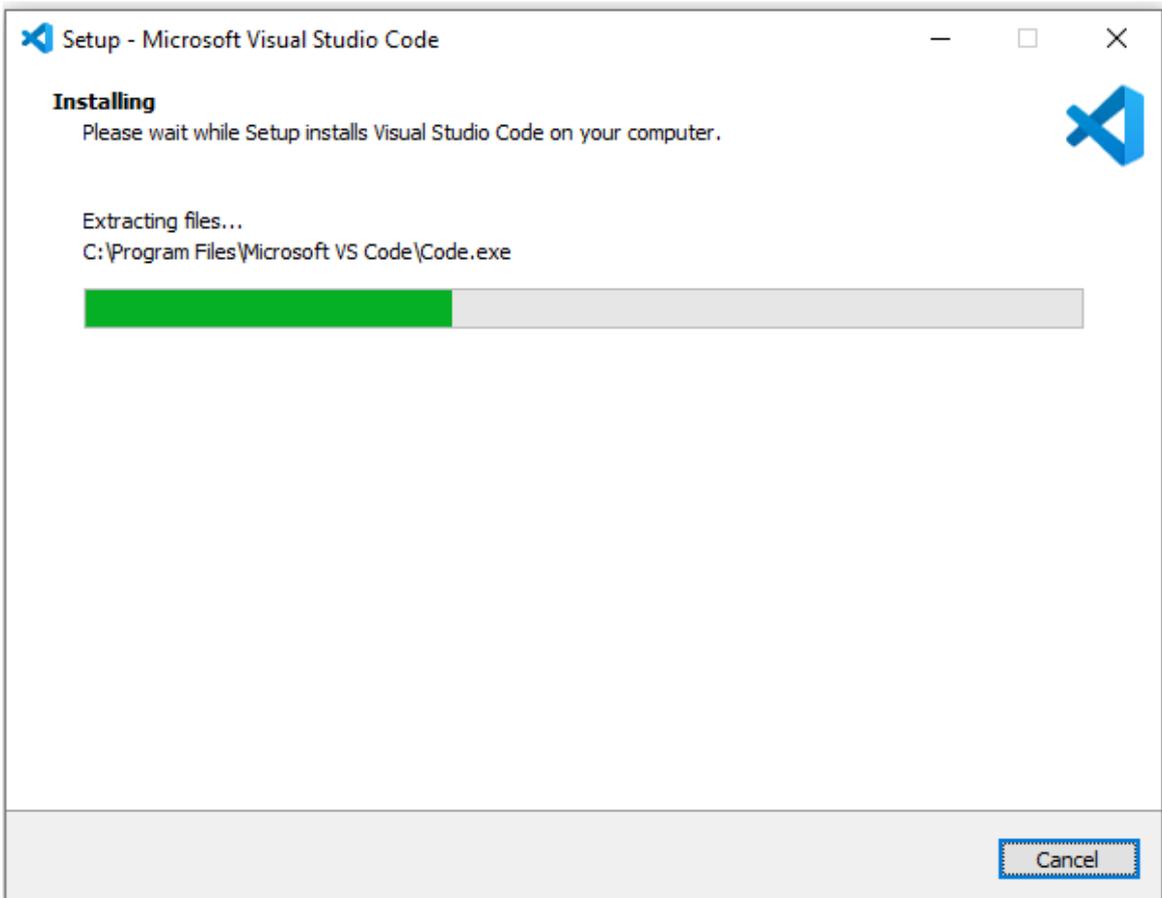


- Follow the installation wizard:









File Edit Selection View Go Run Terminal Help Get Started - Visual Studio Code [Administrator]

Get Started x

< Get Started

### ⚡ Get Started with VS Code

Discover the best customizations to make VS Code yours.

**Choose the look you want**

The right color palette helps you focus on your code, is easy on your eyes, and is simply more fun to use.

[Browse Color Themes](#)

Tip: Use keyboard shortcut: (Ctrl+K Ctrl+T)

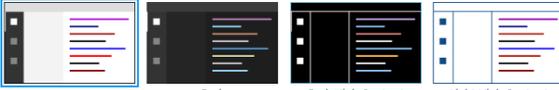
Sync to and from other devices

One shortcut to access everything

Rich support for all your languages

Open up your code

[Mark Done](#) [Next Section →](#)



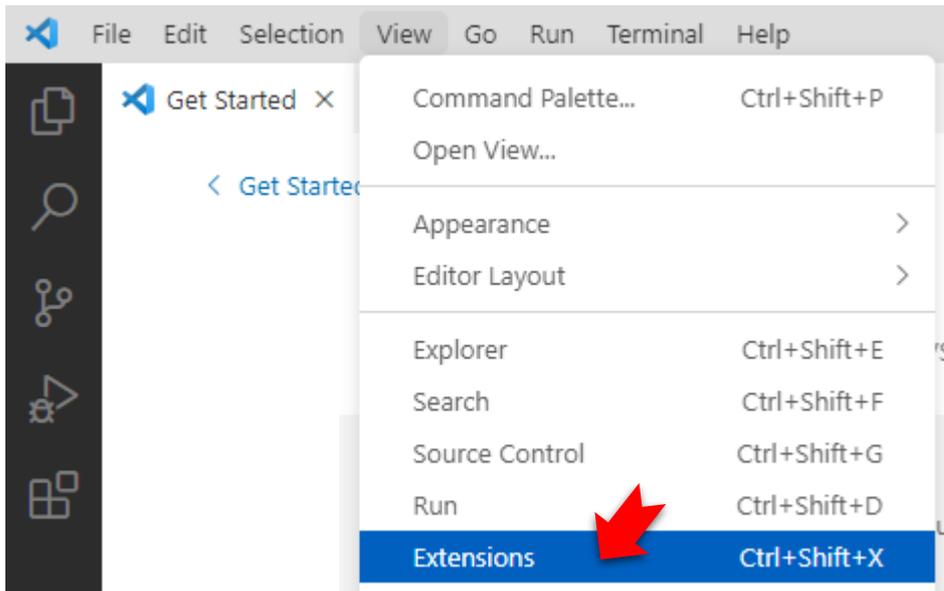
Light Dark Dark High Contrast Light High Contrast

[See More Themes...](#)

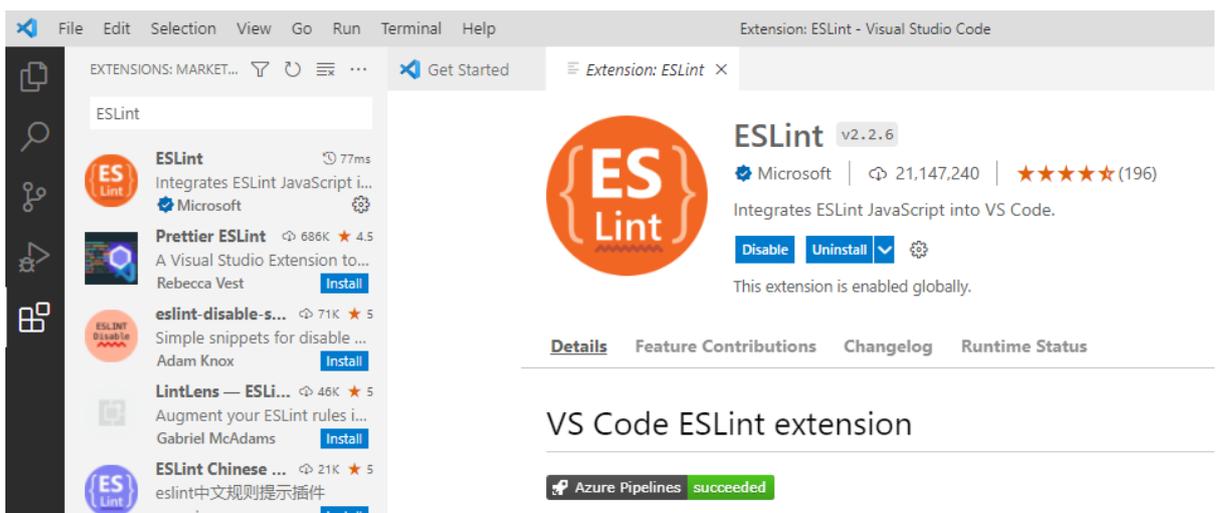
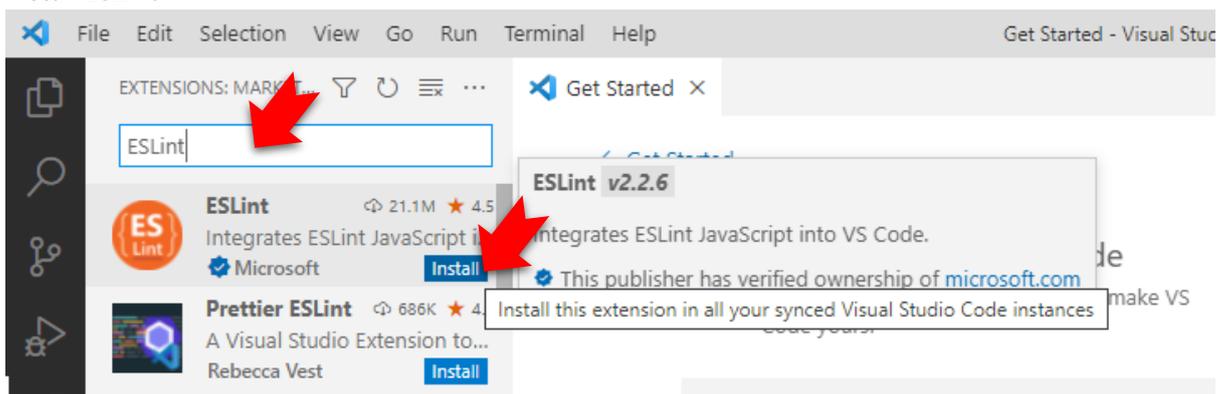
Code collects usage data. Read our [privacy statement](#) and learn how to [opt out](#).

## Adding Extensions to Visual Studio Code

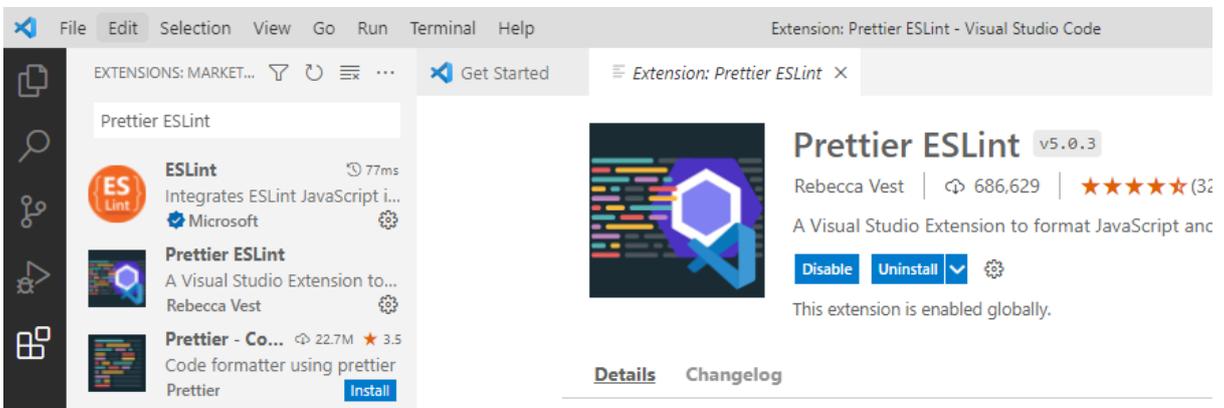
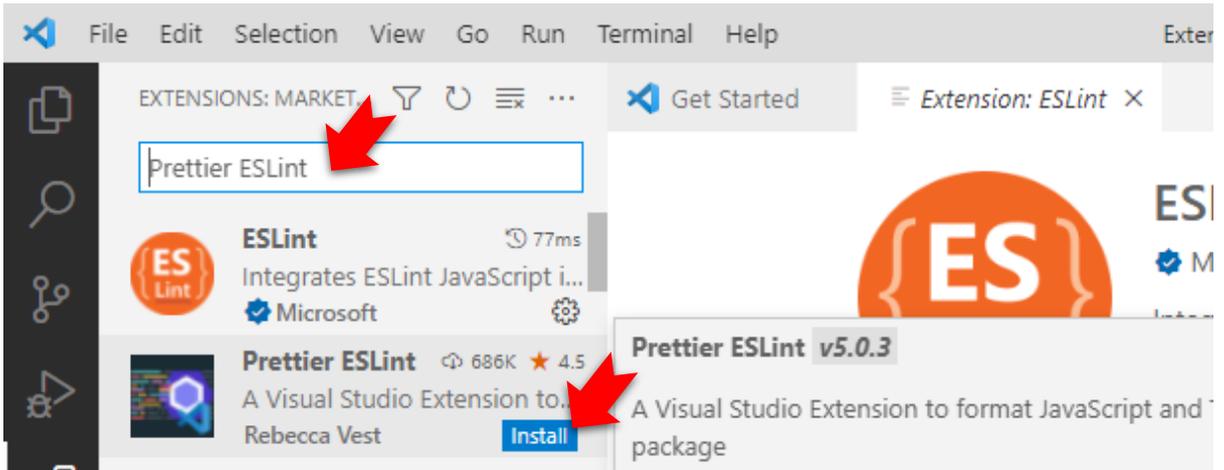
- Select **Extensions** command



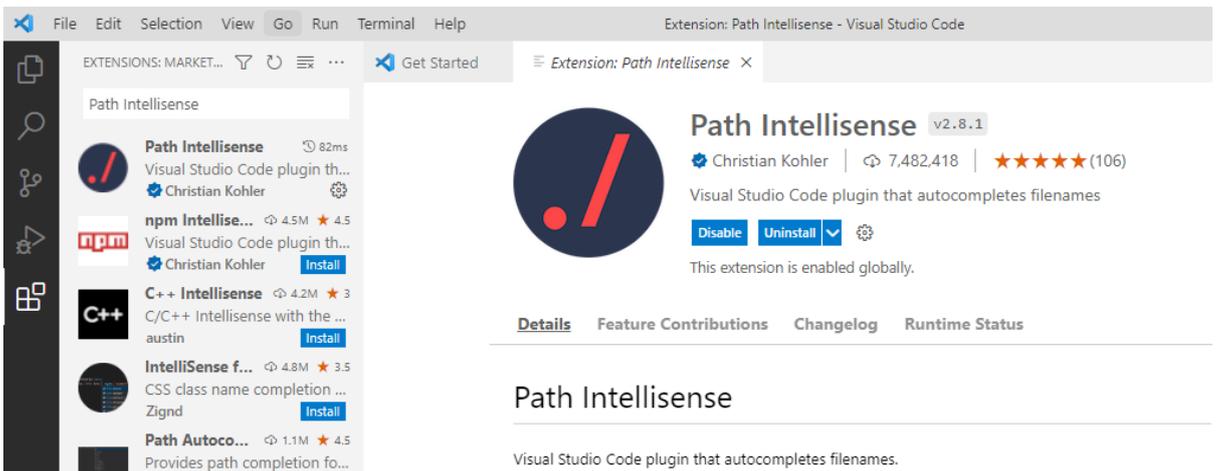
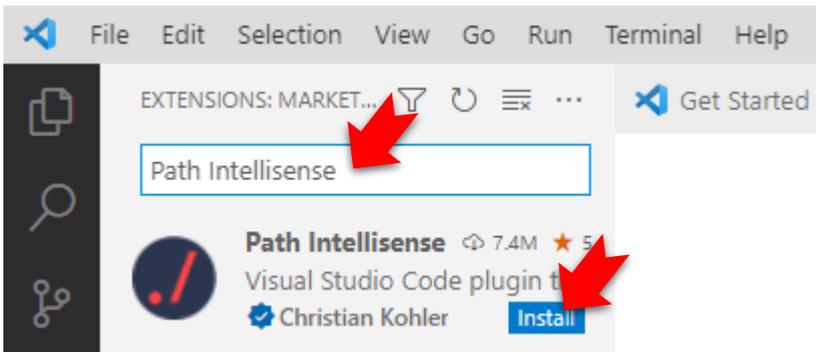
- Install **ESLint**



- Install **Prettier ESLint**



- Install **Path Intellisense**

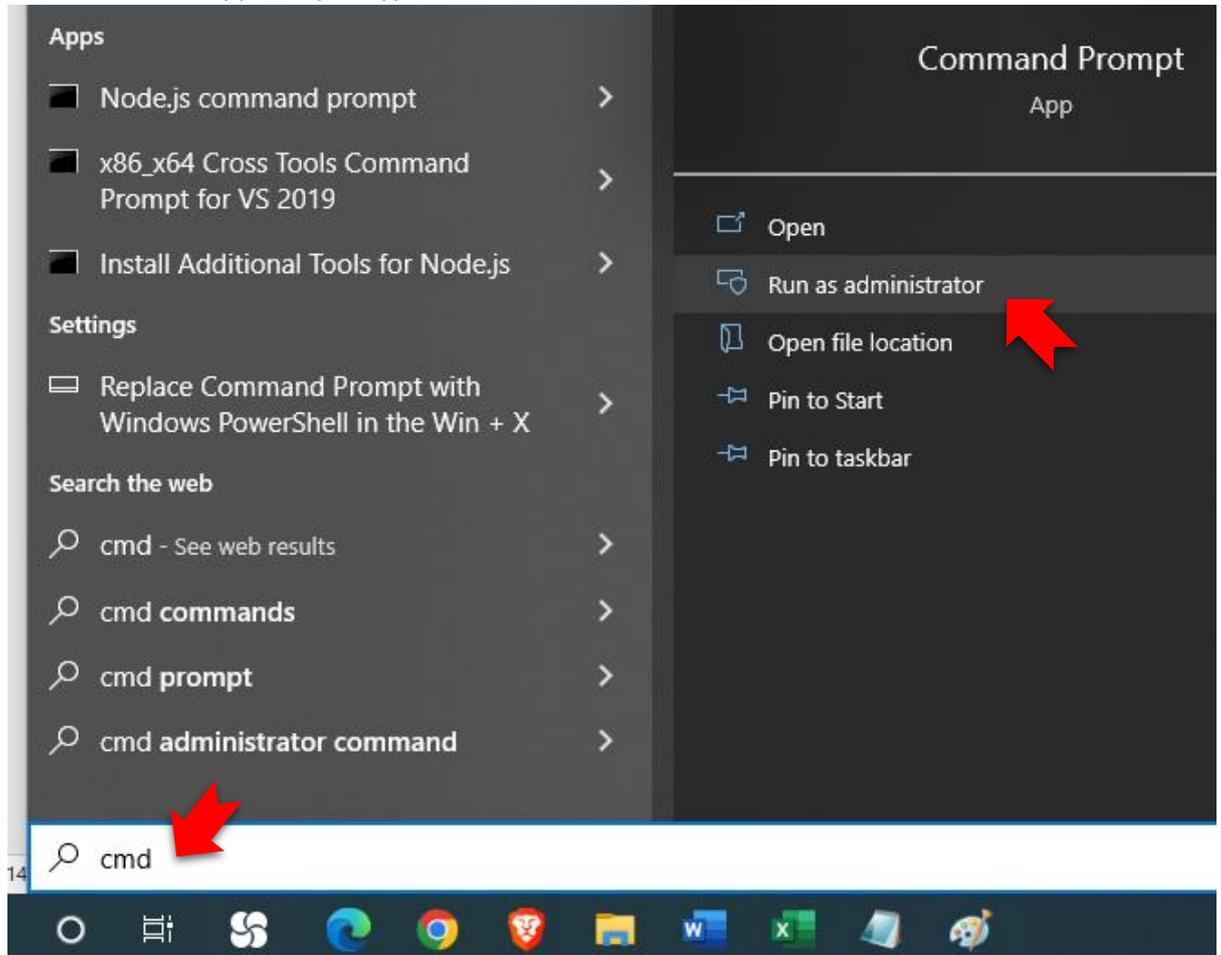


## Install TypeScript Compiler

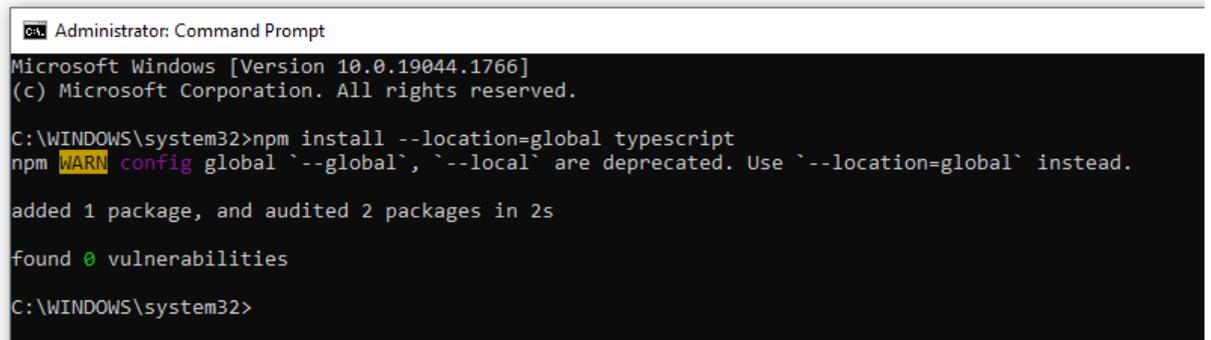
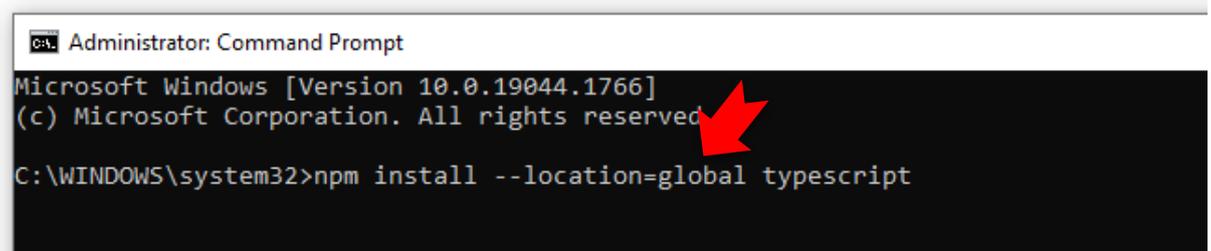
- Click Window **Start** button.



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

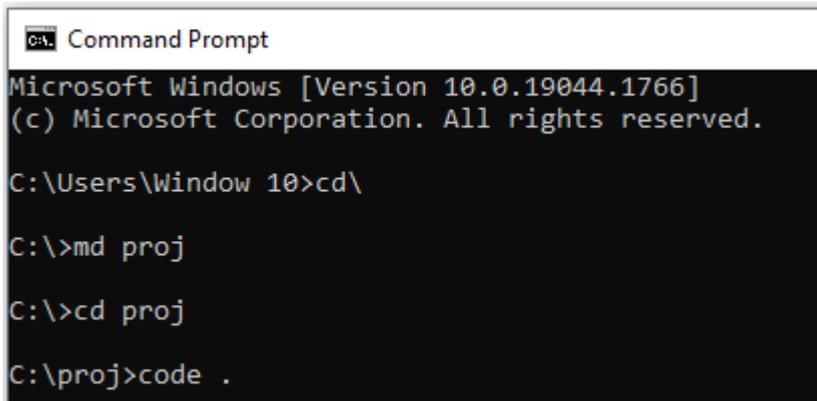


- Under the command box, type **npm install --location=global typescript**



## Test Setup

- Start Command Prompt as Administrator
- Type the following commands:
  - `cls`
  - `cd\`
  - `md proj`
  - `cd proj`
  - `code .`



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

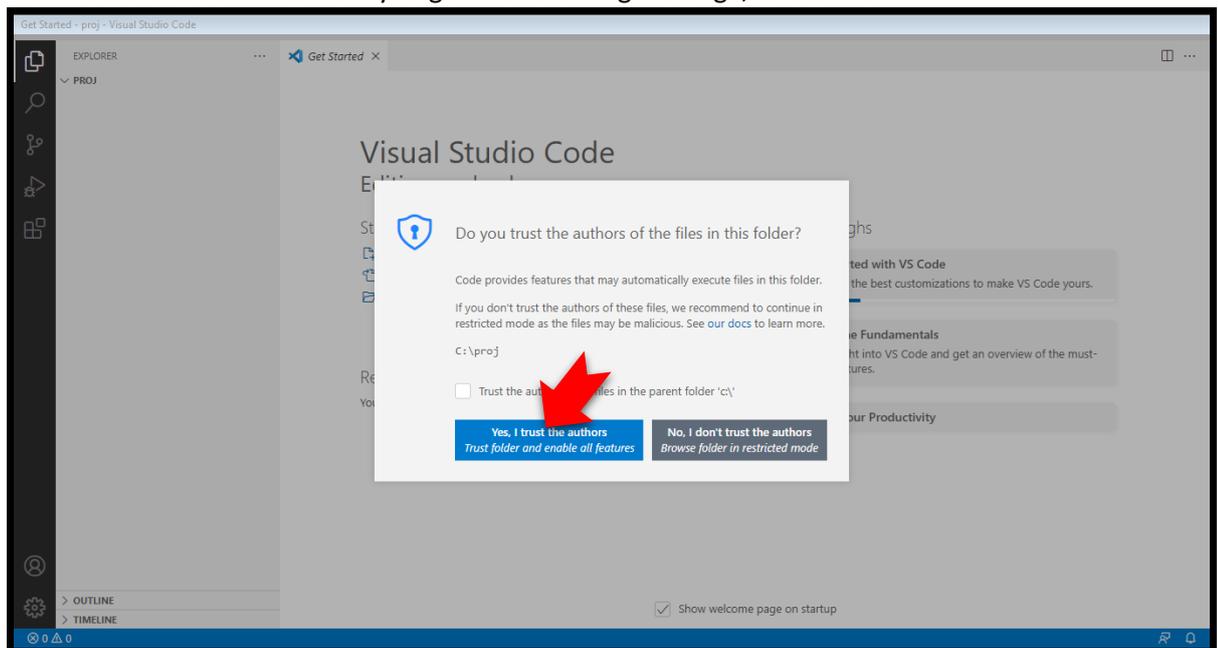
C:\Users\Window 10>cd\

C:\>md proj

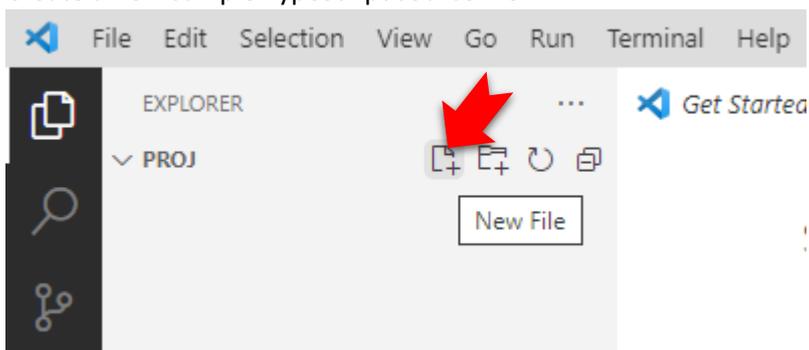
C:\>cd proj

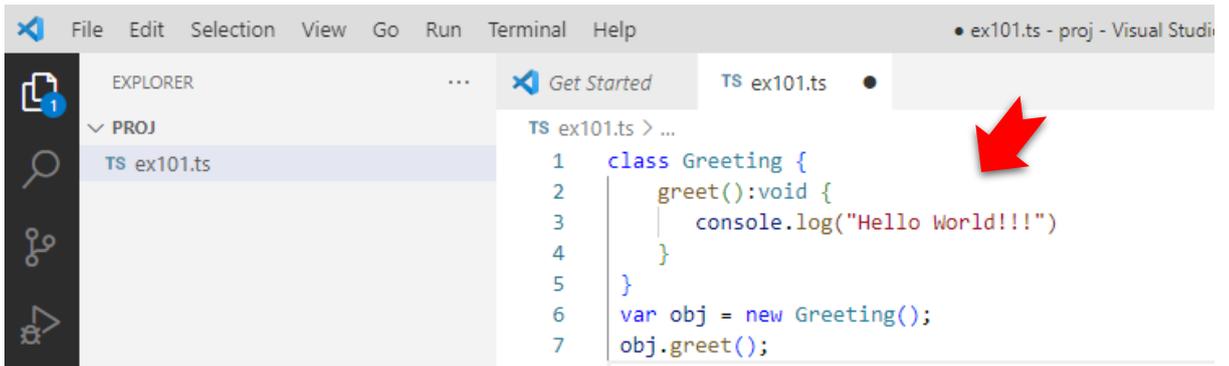
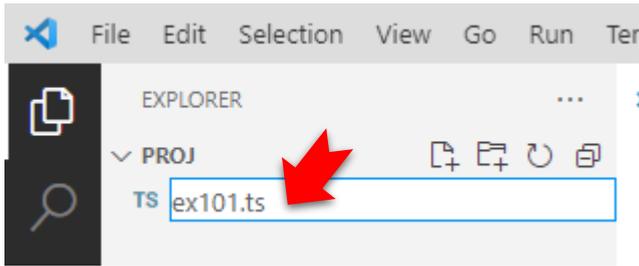
C:\proj>code .
```

Visual Studio Code will start. If you get the following message, select **Yes**

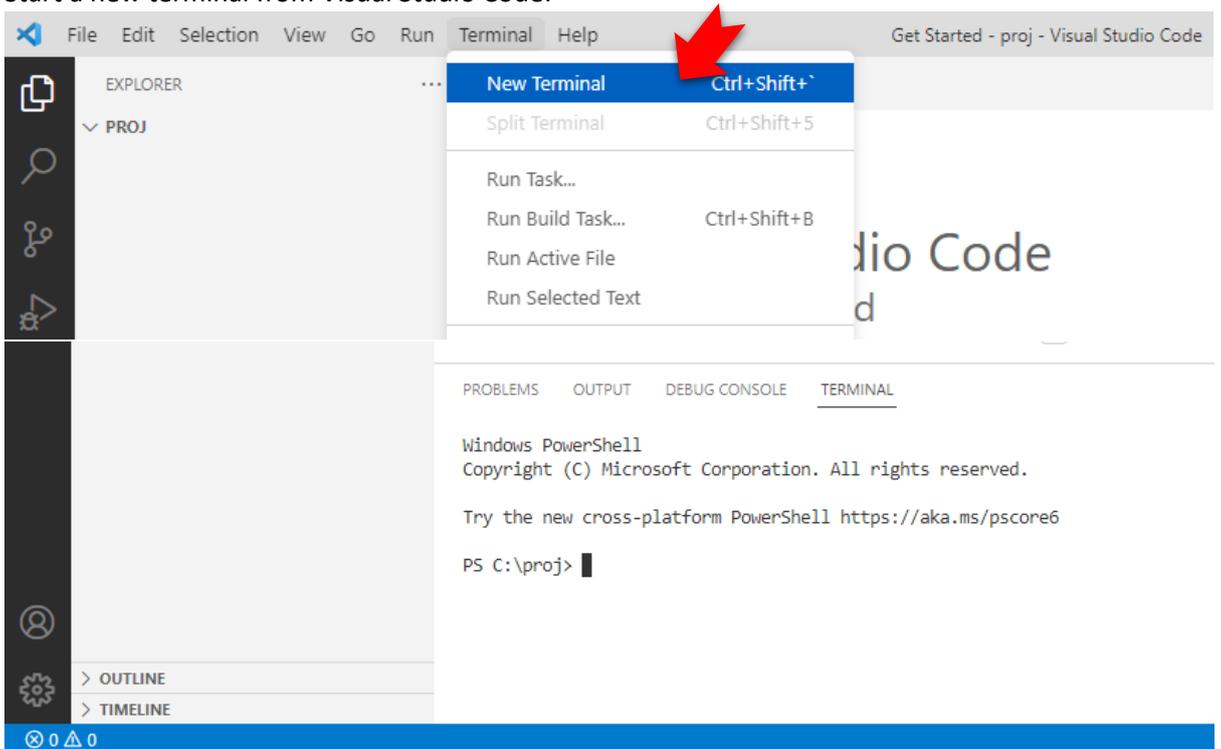


- Create a new sample TypeScript source file:





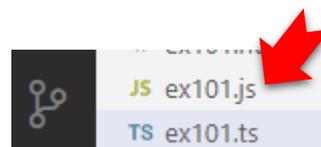
- Save the ex101.ts file
- Start a new terminal from Visual Studio Code:



- Type the command **tsc ex101.ts** in the terminal Window



- If no error, a new ex101.js file will be generated.



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

- 1) Close the Visual Studio Code
- 2) Press the windows-button on your keyboard
- 3) Type 'PowerShell'
- 4) Right-click Windows PowerShell
- 5) Click Run as Administrator
- 6) Run the following command and confirm with 'Y'

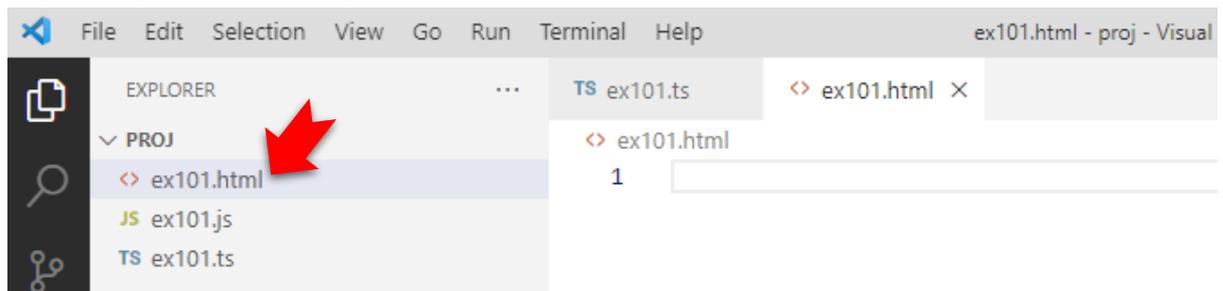
***Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope LocalMachine***

- 7) Close the PowerShell Window

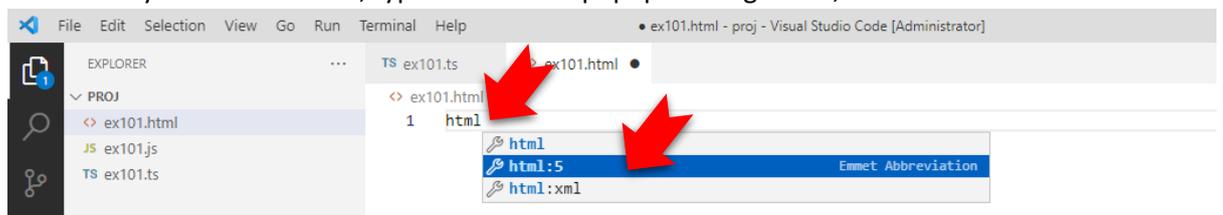
for more information: <https://www.roelpeters.be/solved-running-scripts-is-disabled-on-this-system-in-powershell/>

- 8) Restart the Visual Studio and continue with previous exercise

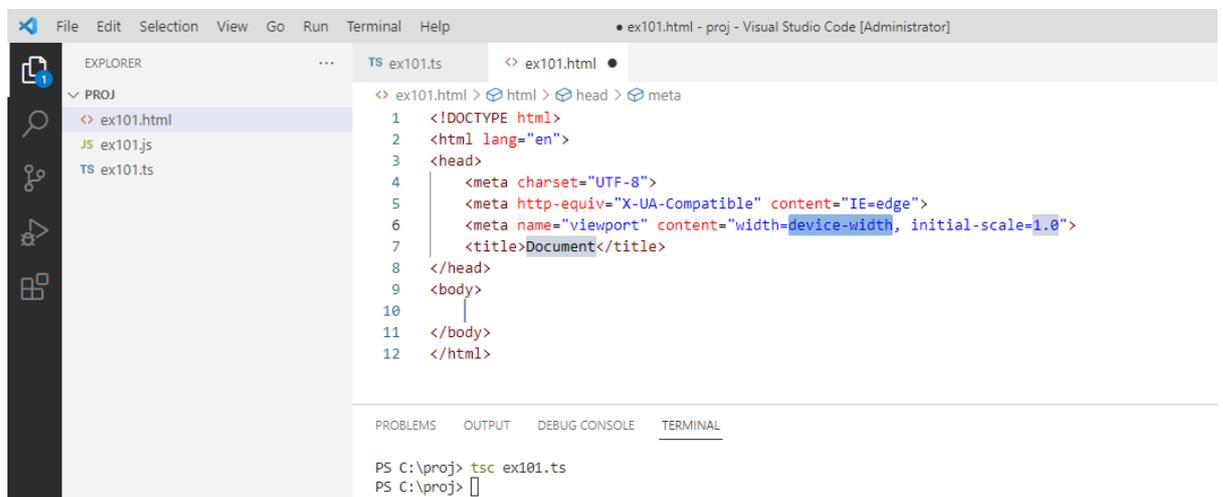
- Create a new **ex101.html** file:



- In the newly created html file, type **html**. In the popup floating menu, select **html:5**



- Press **Enter**



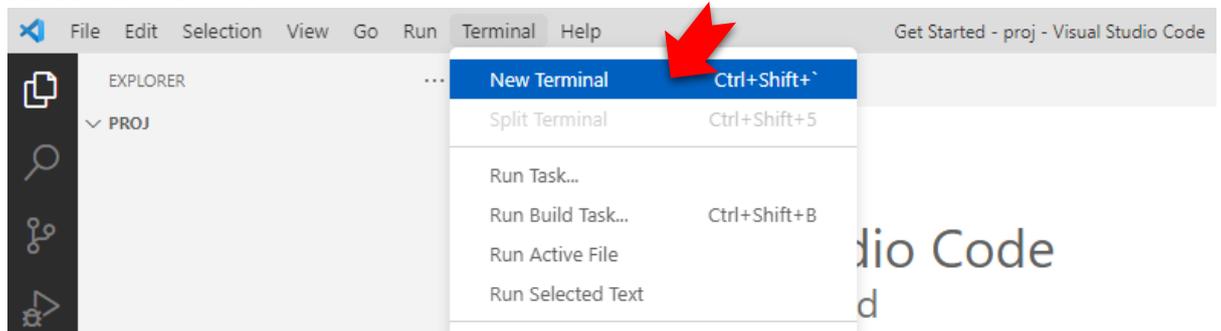
- Add and change the following:

```

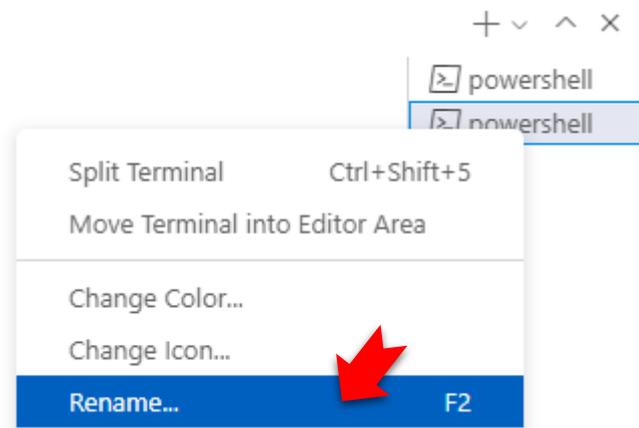
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Ex101: Test Setting</title>
8   <script src="ex101.js" defer></script>
9 </head>
10 <body>
11   Press <F12>, select "Console" tab to view the result
12 </body>
13 </html>

```

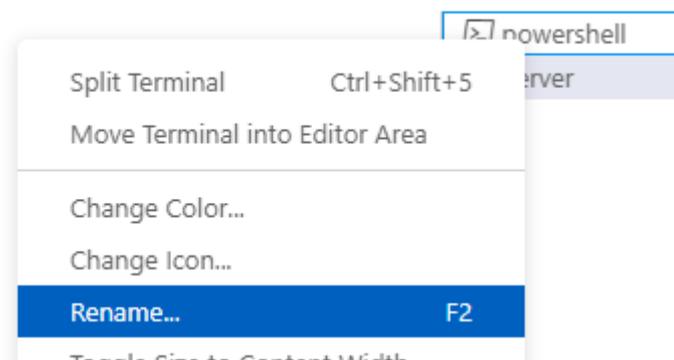
- Save the file
- Start a **New Terminal**



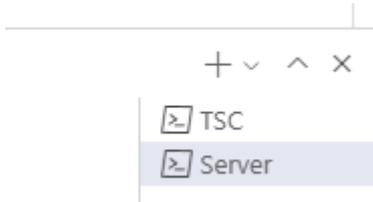
- Now you should have 2 terminal windows. To differentiate them, let's rename them



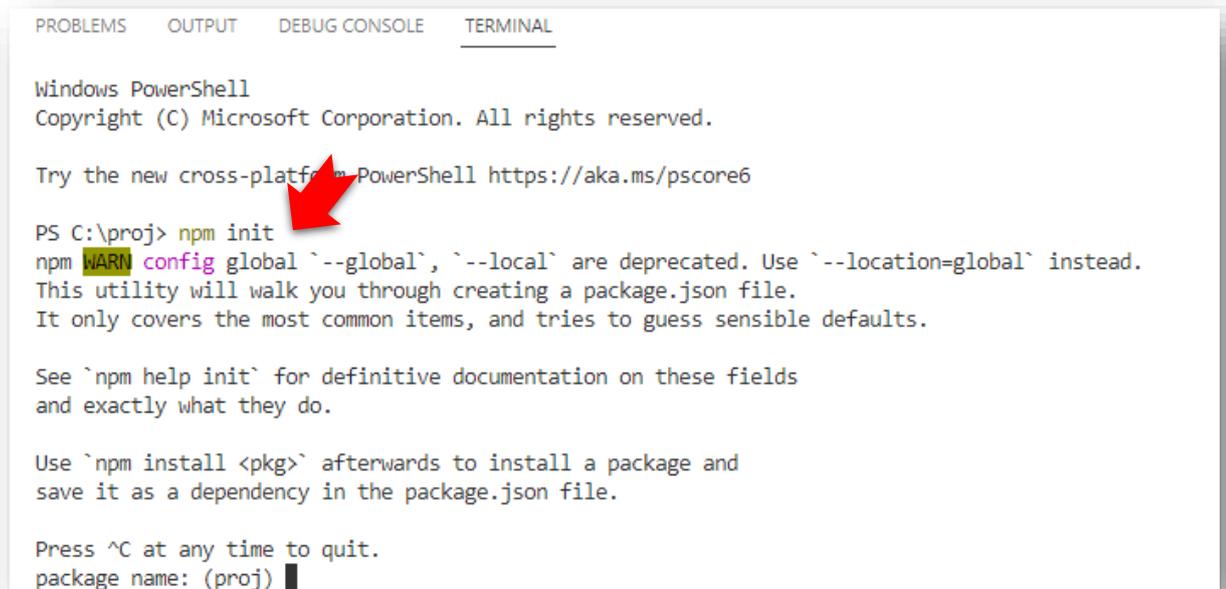
- Rename this Terminal as **Server**
- Let's rename previous Terminal Window as **TSC**



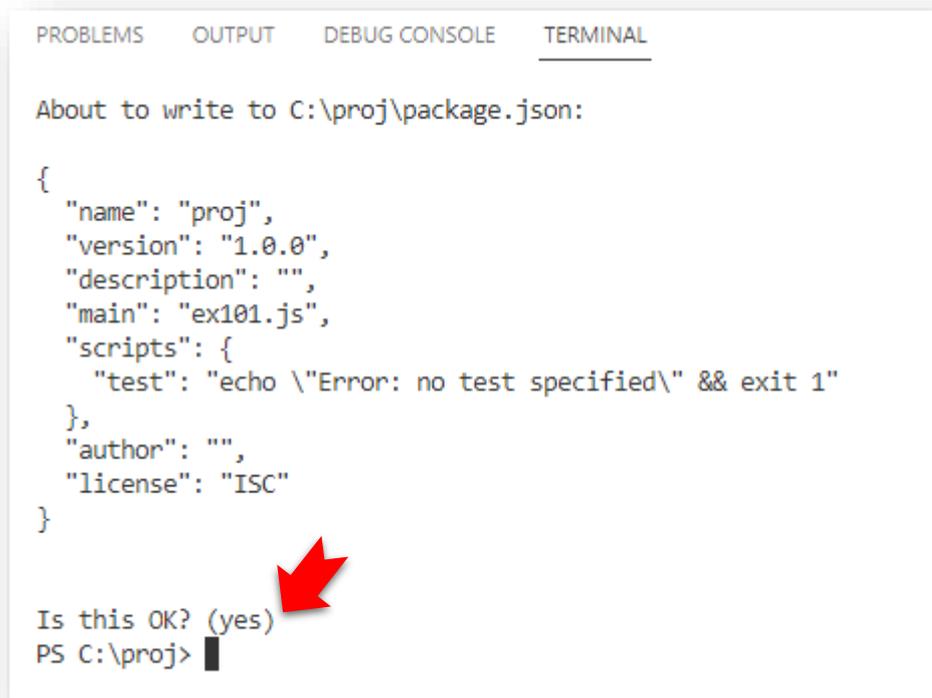
- Now, the Terminal Windows has meaningful names:



- Under the “Server” terminal, type the command **npm init**



- Just press **Enter** to use all default values



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

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\proj> npm install --save-dev lite-server
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 173 packages, and audited 174 packages in 52s

6 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\proj> 
```

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

```
TS ex101.ts  <> ex101.html  {} package.json X

({}) package.json > ({} ) scripts > [start] start
1  {
2  |   "name": "proj",
3  |   "version": "1.0.0",
4  |   "description": "",
5  |   "main": "ex101.js",
6  |   > Debug
7  |   "scripts": {
8  |     "test": "echo \"Error: no test specified\" && exit 1",
9  |     "start": "lite-server"
10 |   },
11 |   "author": "",
12 |   "license": "ISC",
   |   "devDependencies": {
```

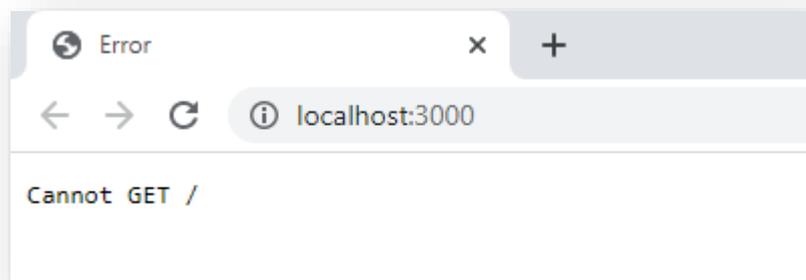
- Save all files, and run the command **npm start** under **Server** terminal,

```
PS C:\proj> npm start
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.

> proj@1.0.0 start
> lite-server

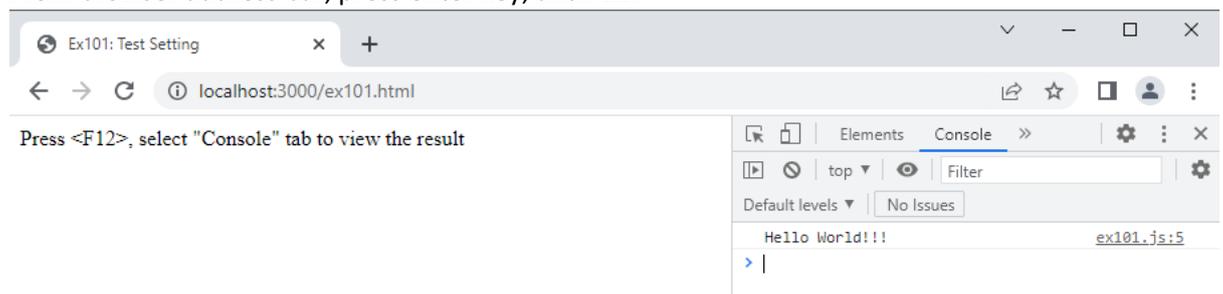
Did not detect a `bs-config.json` or `bs-config.js` override file. Using lite-server defaults...
** browser-sync config **
{
  injectChanges: false,
  files: [ './**/*.html,css,js' ],
  watchOptions: { ignored: 'node_modules' },
  server: {
    baseDir: './',
    middleware: [ [Function (anonymous)], [Function (anonymous)] ]
  }
}
[Browsersync] Access URLs:
-----
Local: http://localhost:3000
External: http://192.168.1.104:3000
-----
UI: http://localhost:3001
UI External: http://localhost:3001
-----
[Browsersync] Serving files from: ./
[Browsersync] Watching files...
22.07.17 18:05:22 404 GET /index.html
```

- The server is running and lunch the browser as shown below:



**Notes:** This setup guide assumes the default browser is Google Chrome. If your default browser is others, the developer pane might not use F12 key.

- Don't worry about the error at the moment. Just change the URL to: <http://localhost:3000/ex101.html>
- From browser address bar, press enter key, and F12:



- Under **Server** terminal, press **Ctrl-C** to stop the server

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
22.07.17 18:13:54 304 GET /ex101.html
22.07.17 18:13:54 304 GET /ex101.js
Terminate batch job (Y/N)? y
PS C:\proj> █
```

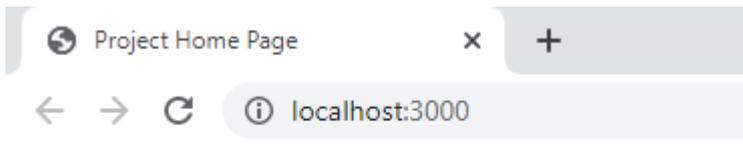
- Add a new index.html file

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

- Under package.json, change the main attribute:

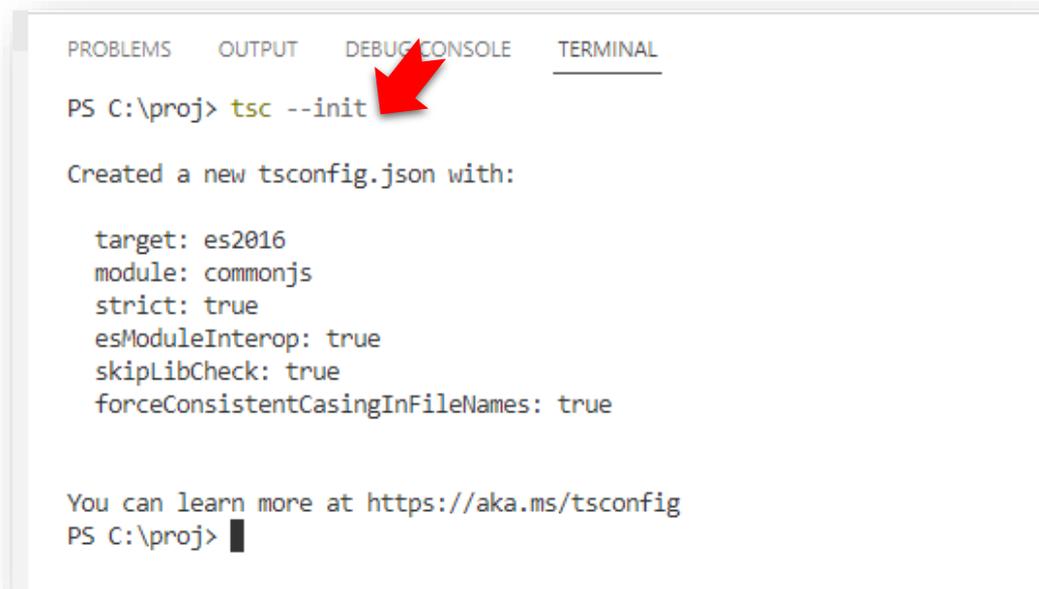
```
{ } package.json > main
1 {
2   "name": "proj",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.html",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "start": "lite-server"
9   },
10  "author": "",
11  "license": "ISC",
12  "devDependencies": {
13    "lite-server": "^2.6.1"
14  }
15 }
16
```

- Save all files
- Under **Server** terminal, type **npm start** to run the server again. Now the browser launch with the following content

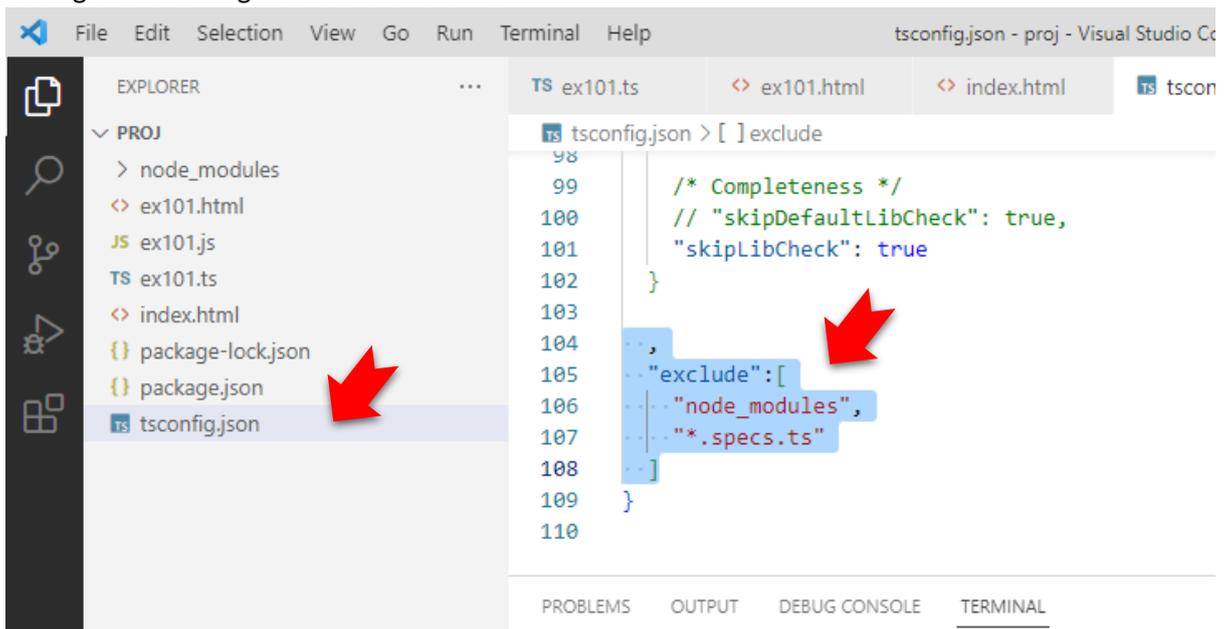


Exercises	
Code	Description
<a href="#">ex101</a>	Test Setting

- To make the TypeScript compiler auto detect changes to the TypeScript files, go to **TSC** terminal window, and type command **tsc -init**



- You will realize that **tsconfig.json** file is generated. Edit this newly generated JSON file by adding the following to the end of the file:



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

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\proj> tsc -w
```

TSC  
Server

```
109 }
110

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[7:12:17 PM] Starting compilation in watch mode...
[7:12:20 PM] Found 0 errors. Watching for file changes.
```

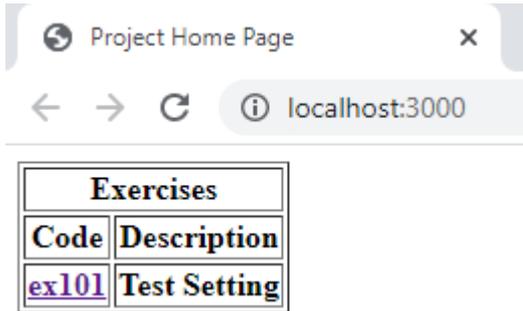
- Now, the TypeScript compiler is now is observing any TypeScript source file change, and it will automatically recompile them. To prove it works, edit the ex101.ts file:

```
TS ex101.ts x ex101.html index.html tsconfig.json
TS ex101.ts > Greeting > greet
1 class Greeting {
2   greet():void {
3     console.log("TypeScript Programming is Fun");
4   }
```

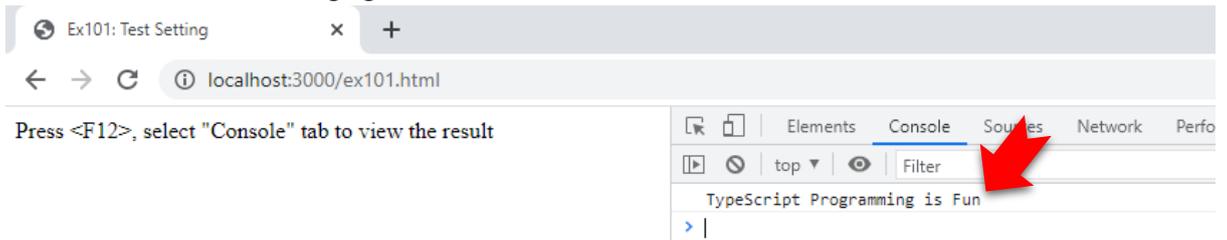
- Save it and observe the messages in **TSC** terminal window. Fix any problem until you get the following messages:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
[7:17:38 PM] File change detected. Starting incremental compilation...
[7:17:38 PM] Found 0 errors. Watching for file changes.
```

- Now, go back to the home page, and click **ex101** hyperlink.



- Press F12 to see the message generated in console tab:



- If you can get the output as shown, the installation and set-up is completed successfully.