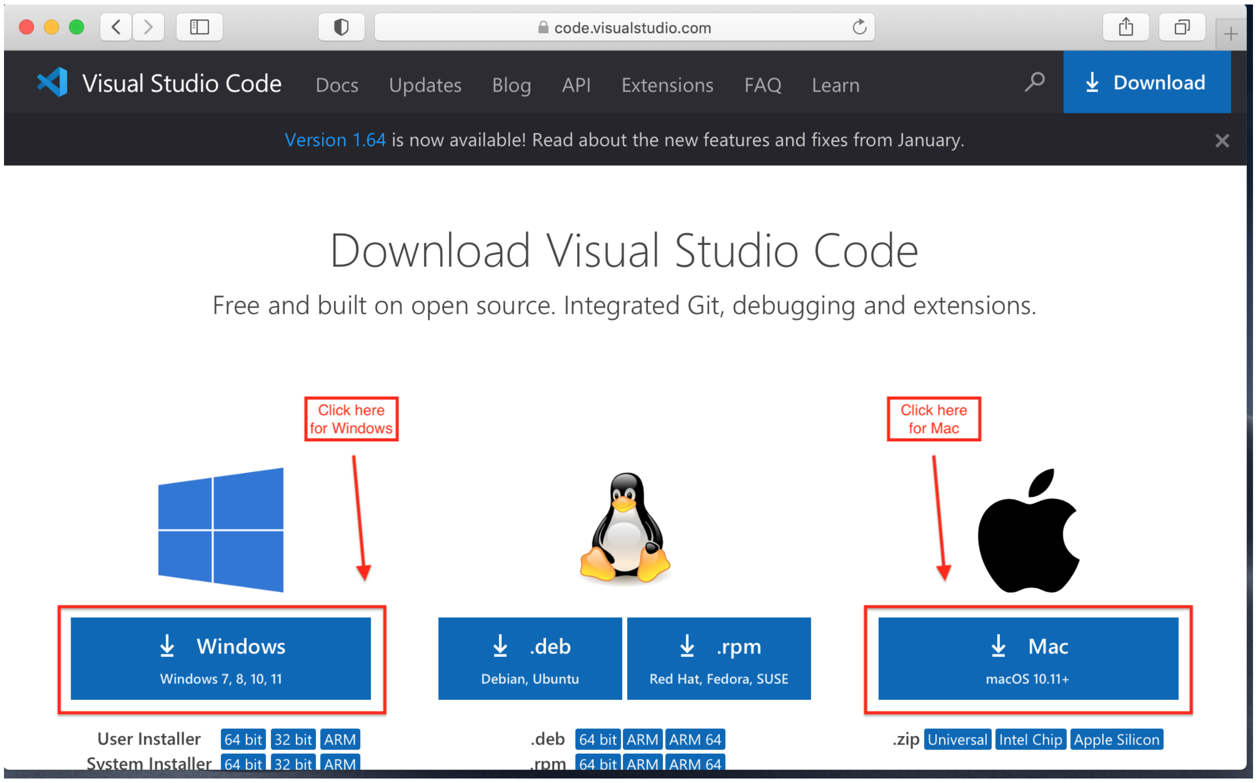
School of Computing and Information Systems

The University of Melbourne

Visual Studio Code Installation

1. Ensure you have first installed GCC as described in the other guides (You will need to follow the WSL guide for Windows and the GCC for MacOS guide for MacOS).

2. Download VSCode from https://code.visualstudio.com/download and install.



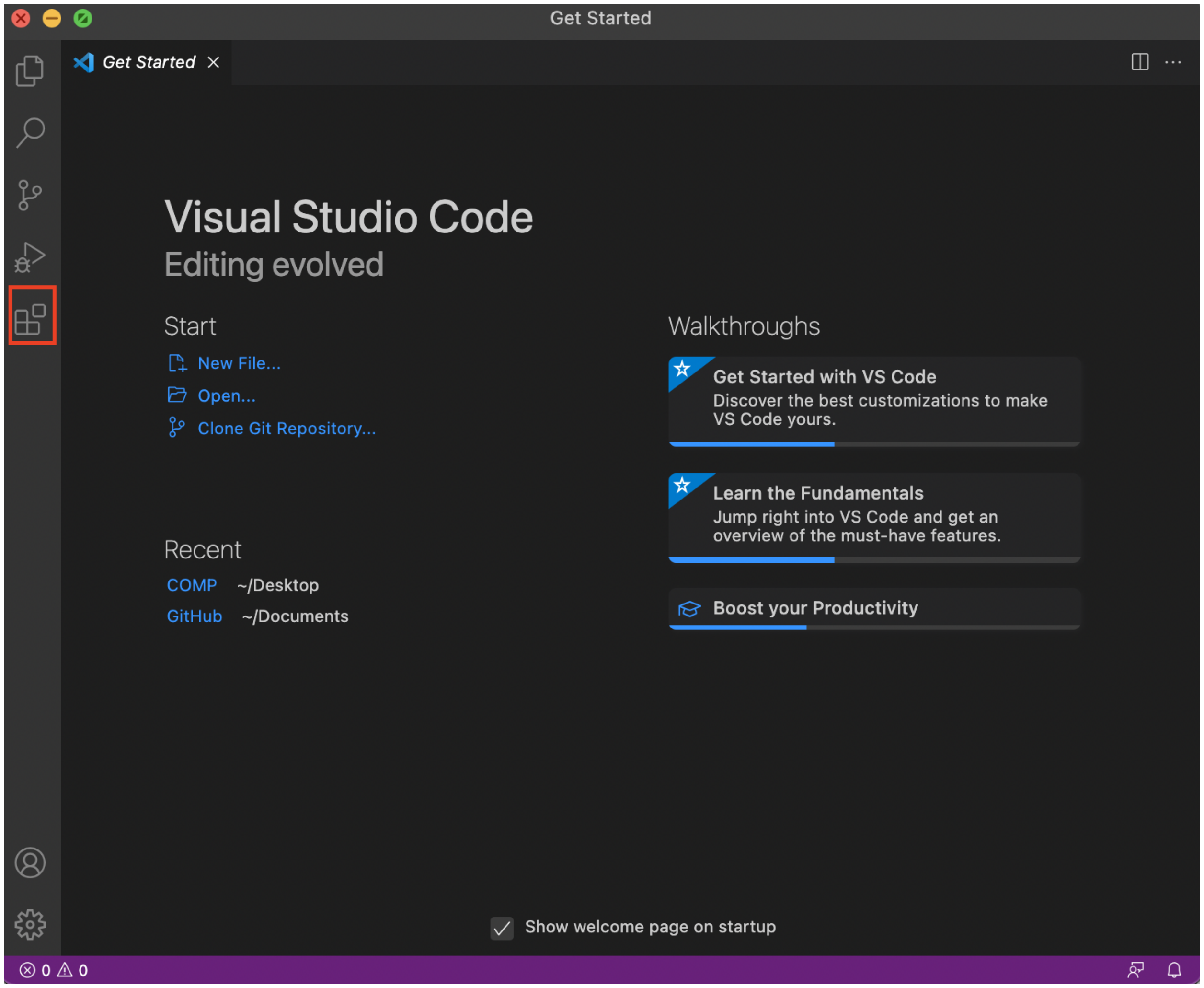
Version 1.64 is now available! Read about the new features and fixes from January.

Download Visual Studio Code

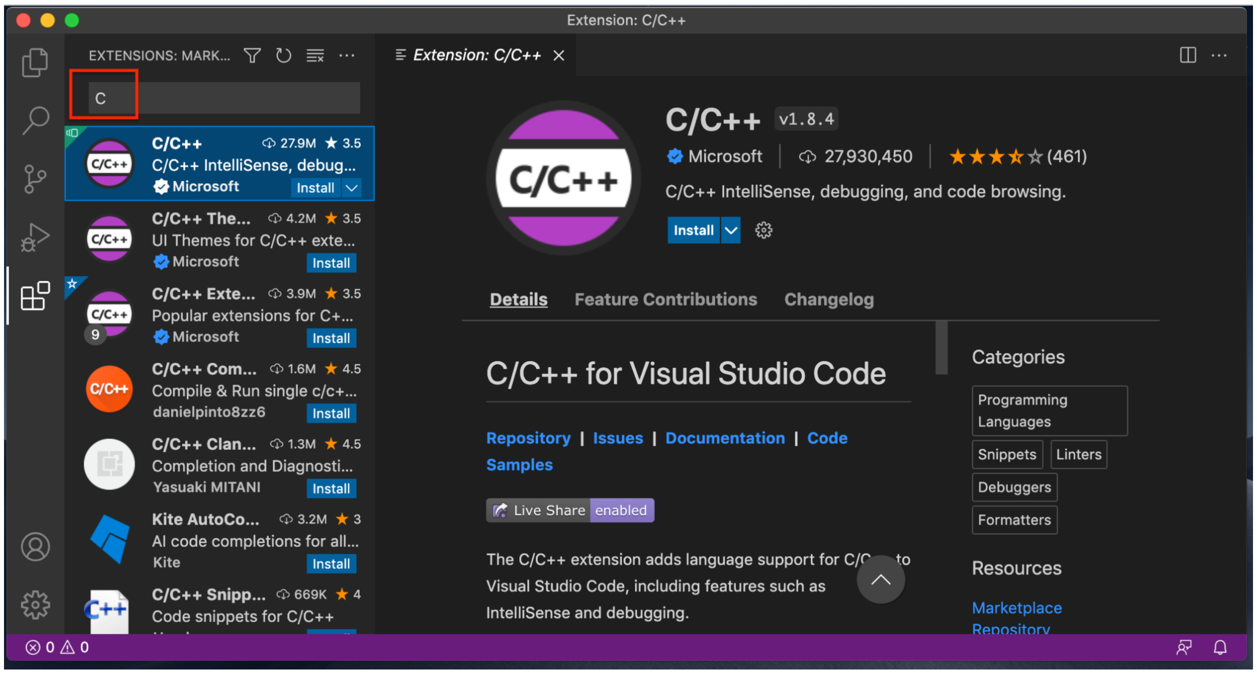
Free and built on open source. Integrated Git, debugging and extensions.

Note, you may be prompted to accept the download and verify before you install it.

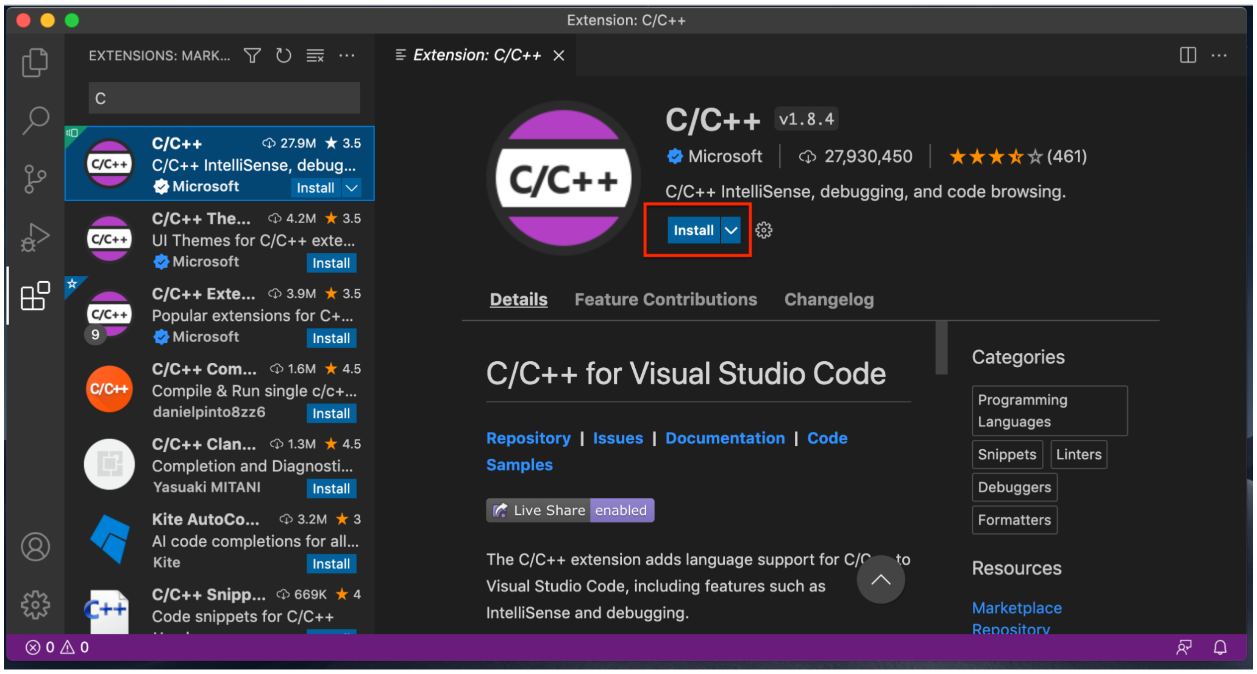
3. Open VSCode and click the “Extensions” tab on the left.



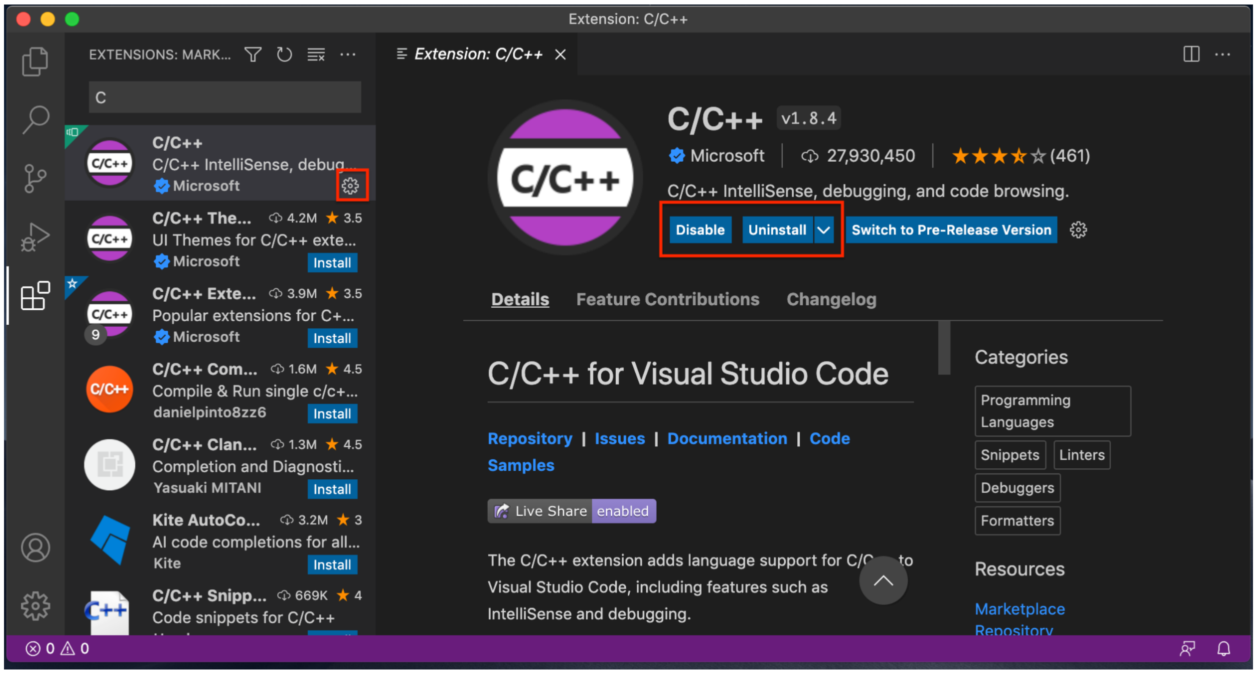
4. Search for ‘C’ in the search bar in the top left.


Extension: C/C++
EXTENSIONS: MARK...
C
C/C++ 27.9M ★ 3.5
C/C++ IntelliSense, debug...
Microsoft Install
C/C++ The... 4.2M ★ 3.5
UI Themes for C/C++ exte...
Microsoft Install
C/C++ Exte... 3.9M ★ 3.5
Popular extensions for C+...
Microsoft Install
C/C++ Com... 1.6M ★ 4.5
Compile & Run single c/c+...
danielpinto8zz6 Install
C/C++ Clan... 1.3M ★ 4.5
Completion and Diagnosti...
Yasuaki MITANI Install
Kite AutoCo... 3.2M ★ 3
AI code completions for all...
Kite Install
C/C++ Snipp... 669K ★ 4
Code snippets for C/C++
Extension: C/C++
C/C++ v1.8.4
Microsoft 27,930,450 (461)
C/C++ IntelliSense, debugging, and code browsing.
Install
Details Feature Contributions Changelog
C/C++ for Visual Studio Code
Repository | Issues | Documentation | Code Samples
Live Share enabled
The C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.
Categories
Programming Languages
Snippets Linters
Debuggers
Formatters
Resources
Marketplace
Repository
0 0

5. The first extension should be called C/C++, select it and click “Install”

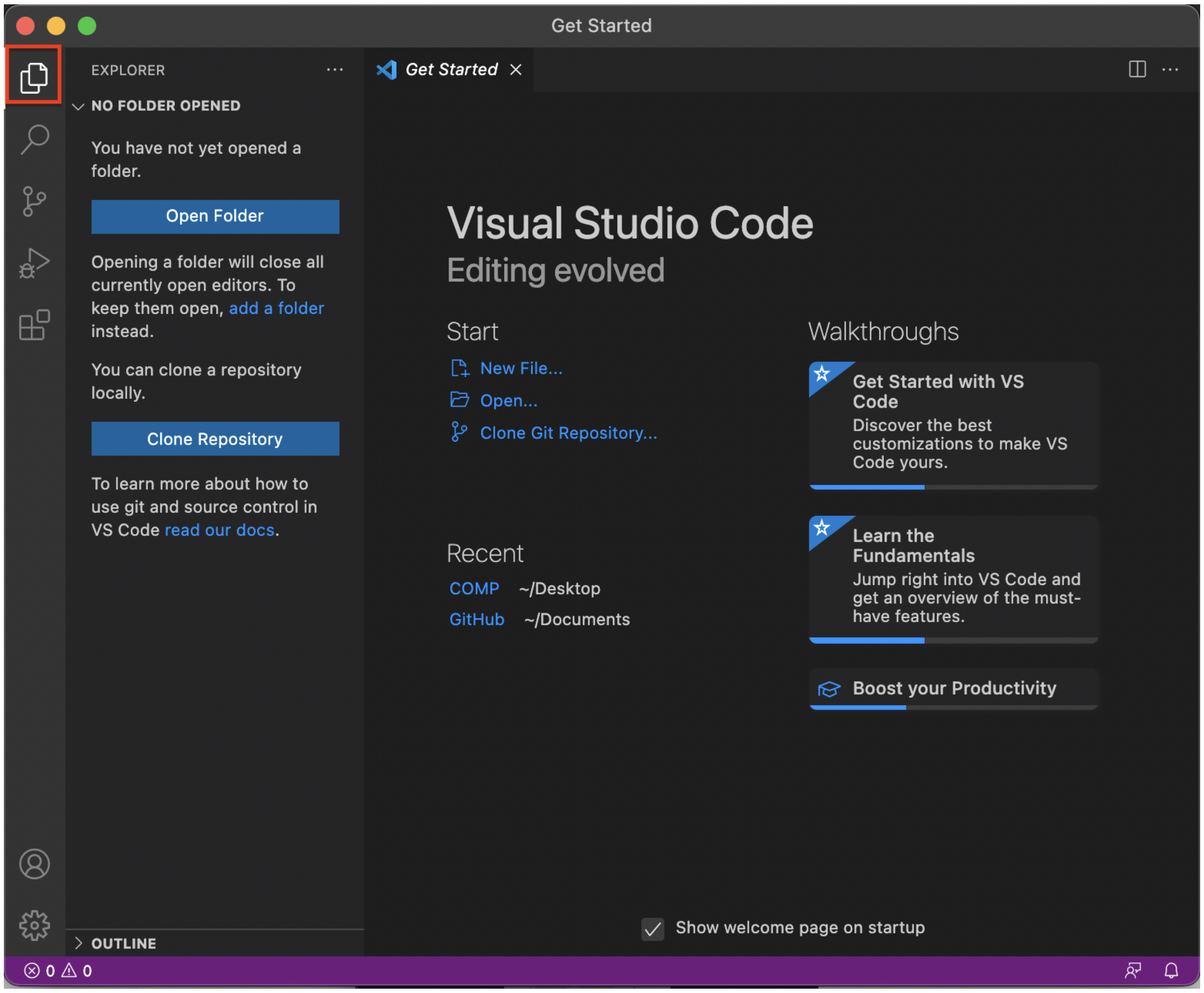

Extension: C/C++
EXTENSIONS: MARK...
C
C/C++ 27.9M ★ 3.5
C/C++ IntelliSense, debug...
Microsoft Install
C/C++ The... 4.2M ★ 3.5
UI Themes for C/C++ exte...
Microsoft Install
C/C++ Exte... 3.9M ★ 3.5
Popular extensions for C+...
Microsoft Install
C/C++ Com... 1.6M ★ 4.5
Compile & Run single c/c+...
danielpinto8zz6 Install
C/C++ Clan... 1.3M ★ 4.5
Completion and Diagnosti...
Yasuaki MITANI Install
Kite AutoCo... 3.2M ★ 3
AI code completions for all...
Kite Install
C/C++ Snipp... 669K ★ 4
Code snippets for C/C++
Extension: C/C++
C/C++ v1.8.4
Microsoft 27,930,450 (461)
C/C++ IntelliSense, debugging, and code browsing.
Install
Details Feature Contributions Changelog
C/C++ for Visual Studio Code
Repository | Issues | Documentation | Code Samples
Live Share enabled
The C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.
Categories
Programming Languages
Snippets Linters
Debuggers
Formatters
Resources
Marketplace
Repository
0 0

6. When installed, you will see a cog on the left list and the options to “Disable” or “Uninstall”



The C/C++ extension adds language support for C/C++ to Visual Studio Code, including features such as IntelliSense and debugging.

7. Go the “Explorer” tab in the top left.



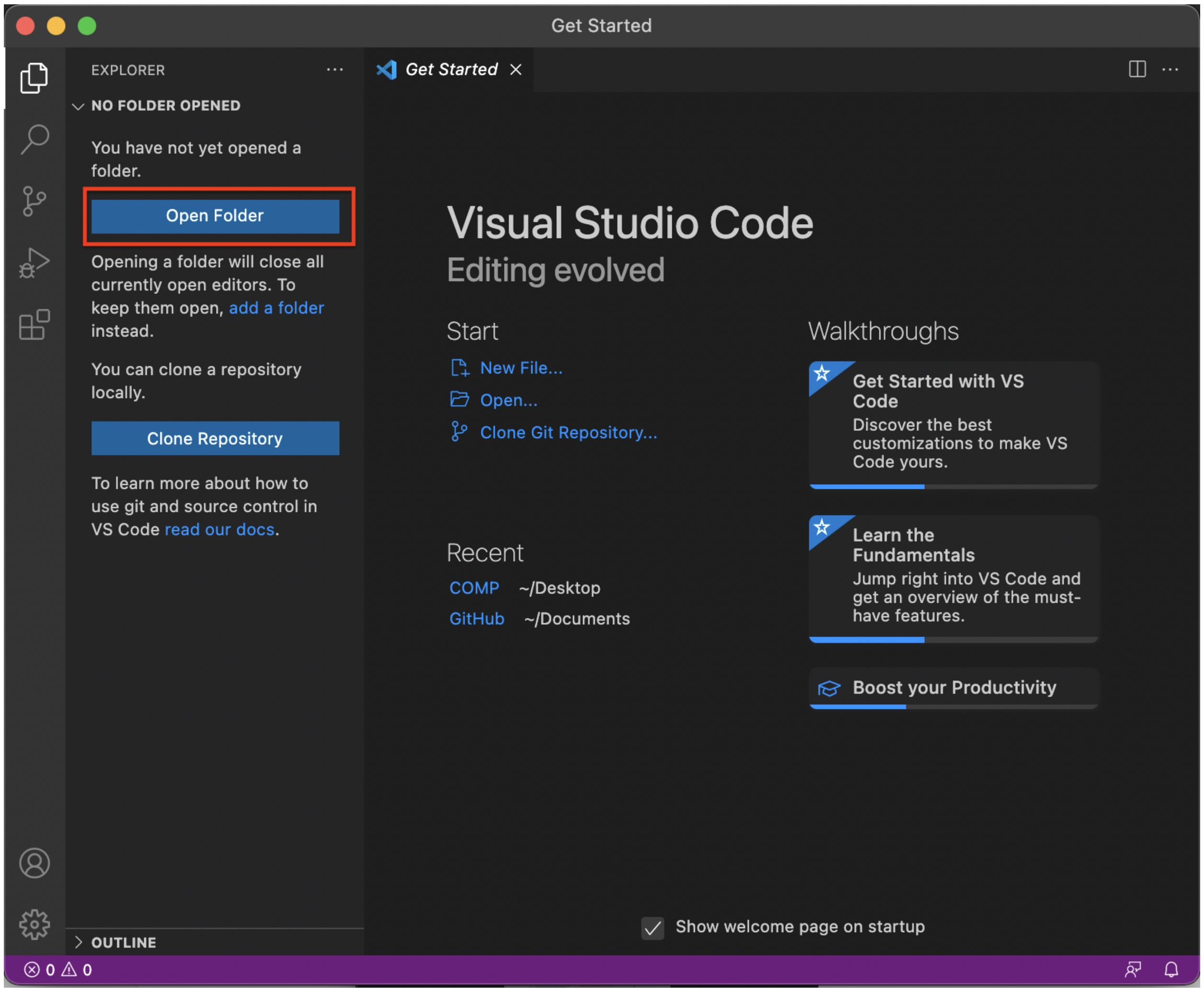
You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

8. You can now open a folder for all your code to be easily accessible. Alternatively, you can create a folder for your COMP subject and sub-folders for Assignments, Lectures and Workshops and drag it in.


Get Started

EXPLORER

NO FOLDER OPENED

You have not yet opened a folder.

Open Folder

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

Clone Repository

To learn more about how to use git and source control in VS Code read our docs.

OUTLINE

Get Started

Visual Studio Code

Editing evolved

Start

New File...

Open...

Clone Git Repository...

Recent

COMP ~/Desktop

GitHub ~/Documents

Walkthroughs

Get Started with VS Code
Discover the best customizations to make VS Code yours.

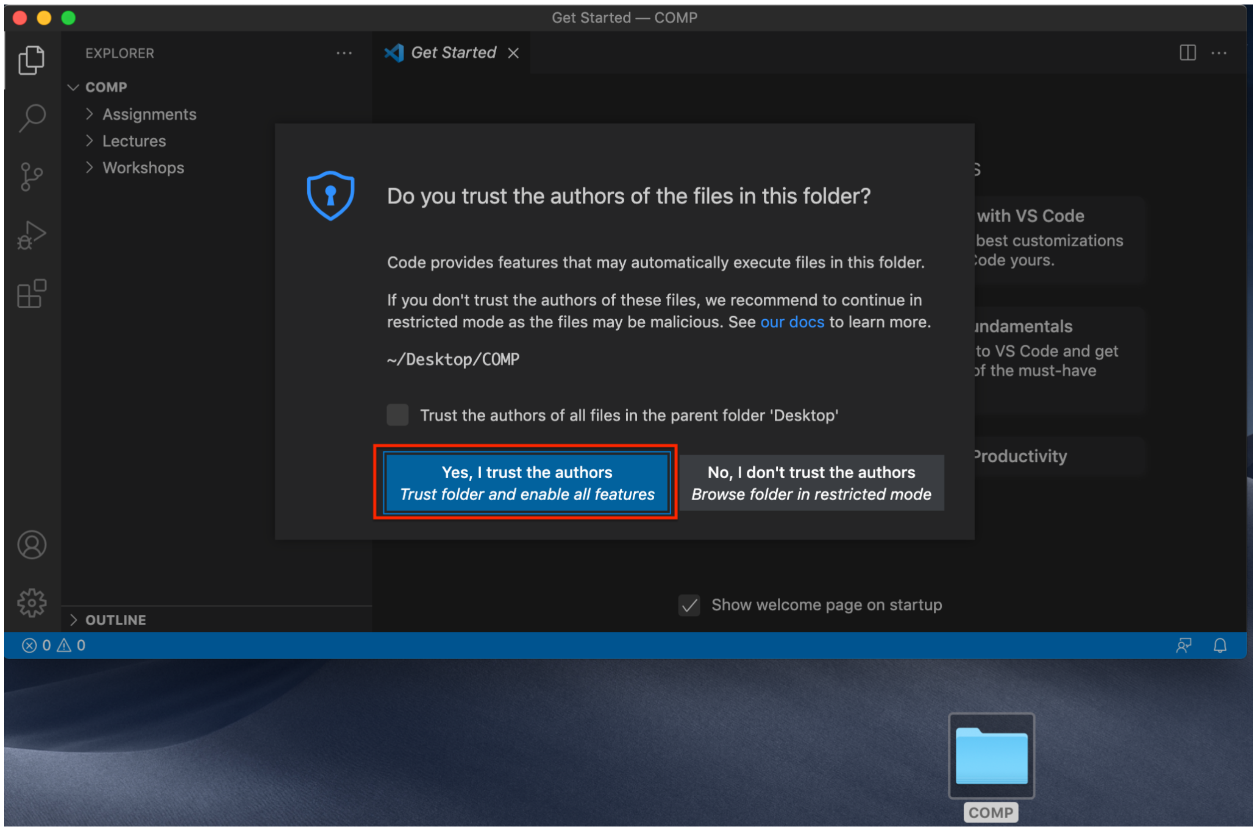
Learn the Fundamentals
Jump right into VS Code and get an overview of the must-have features.

Boost your Productivity

Show welcome page on startup

0 0

9. You can allow access of your own files in the folders you open.


Get Started — COMP
EXPLORER
COMP
Assignments
Lectures
Workshops
Get Started
OUTLINE
0 0
COMP

Do you trust the authors of the files in this folder?

Code provides features that may automatically execute files in this folder.

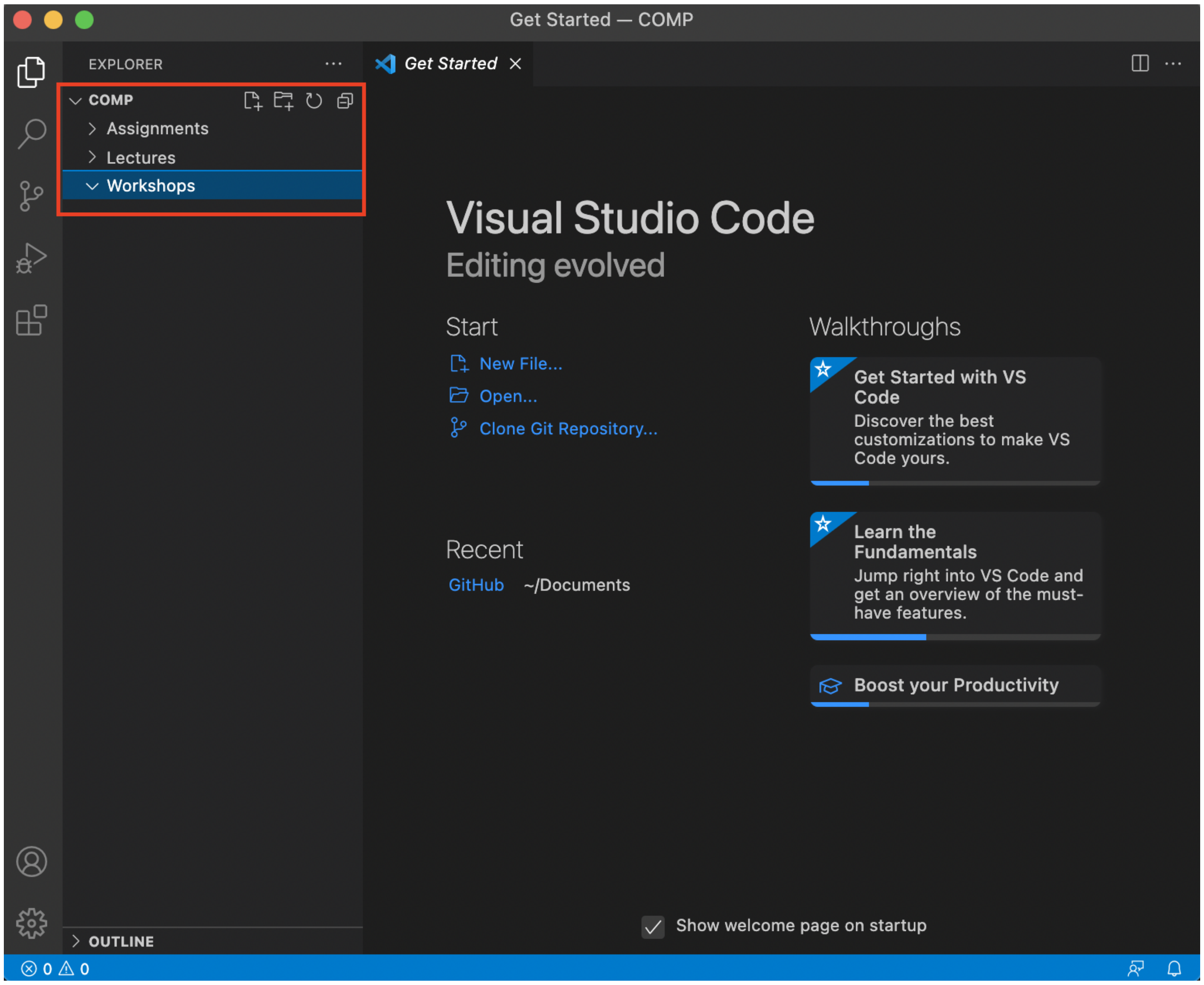
If you don't trust the authors of these files, we recommend to continue in restricted mode as the files may be malicious. See our docs to learn more.

~/Desktop/COMP

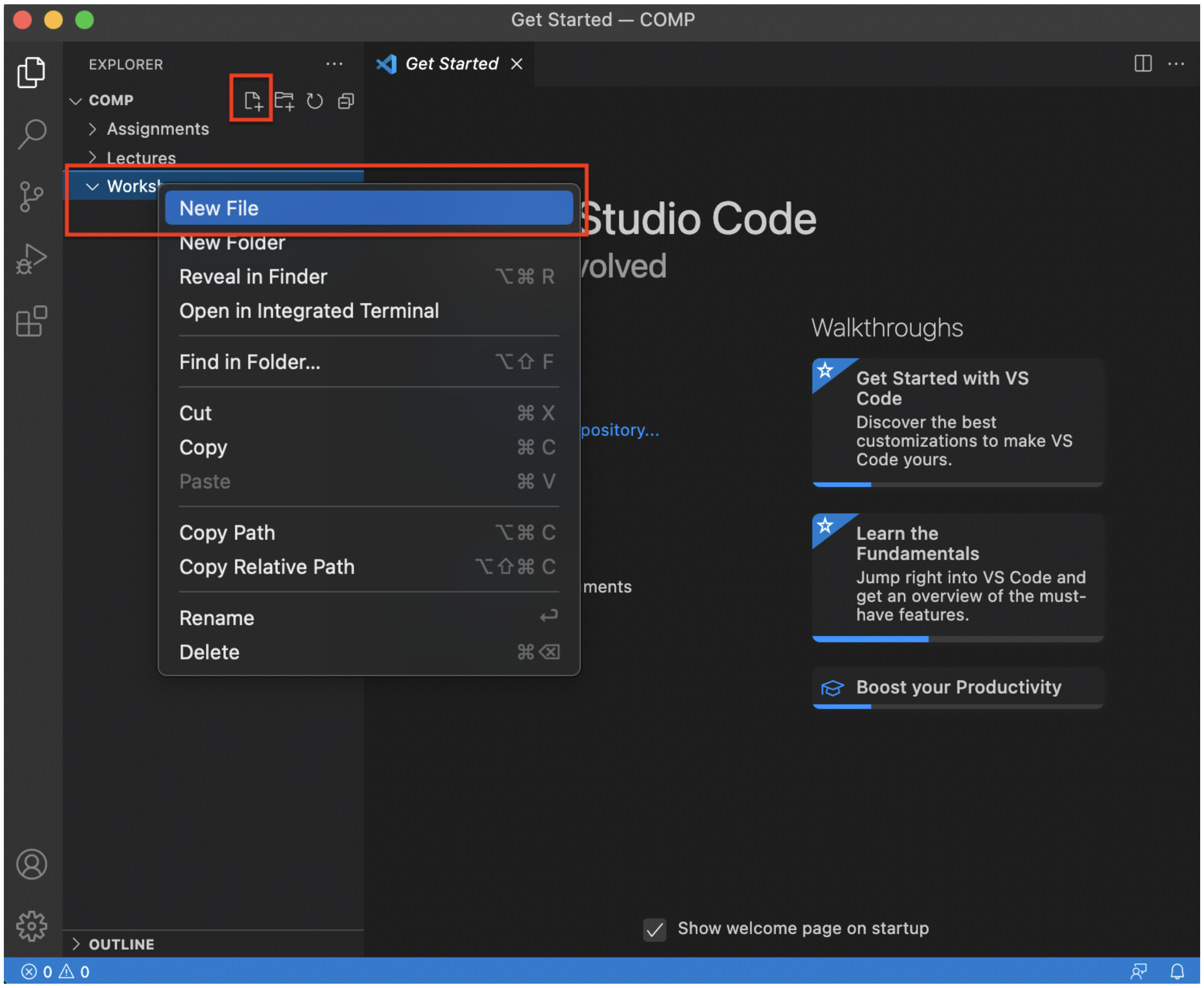
Trust the authors of all files in the parent folder 'Desktop'

Yes, I trust the authors
Trust folder and enable all features
No, I don't trust the authors
Browse folder in restricted mode
Show welcome page on startup

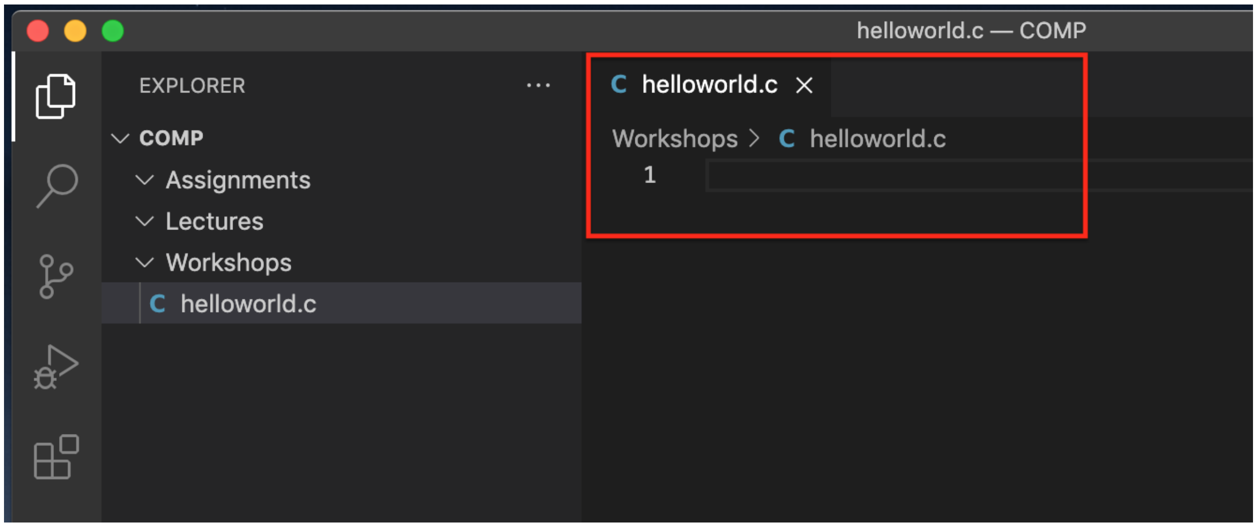
10. You can open your “Workshops” sub folder in the “Explorer”. Note the down arrow for the current folder.



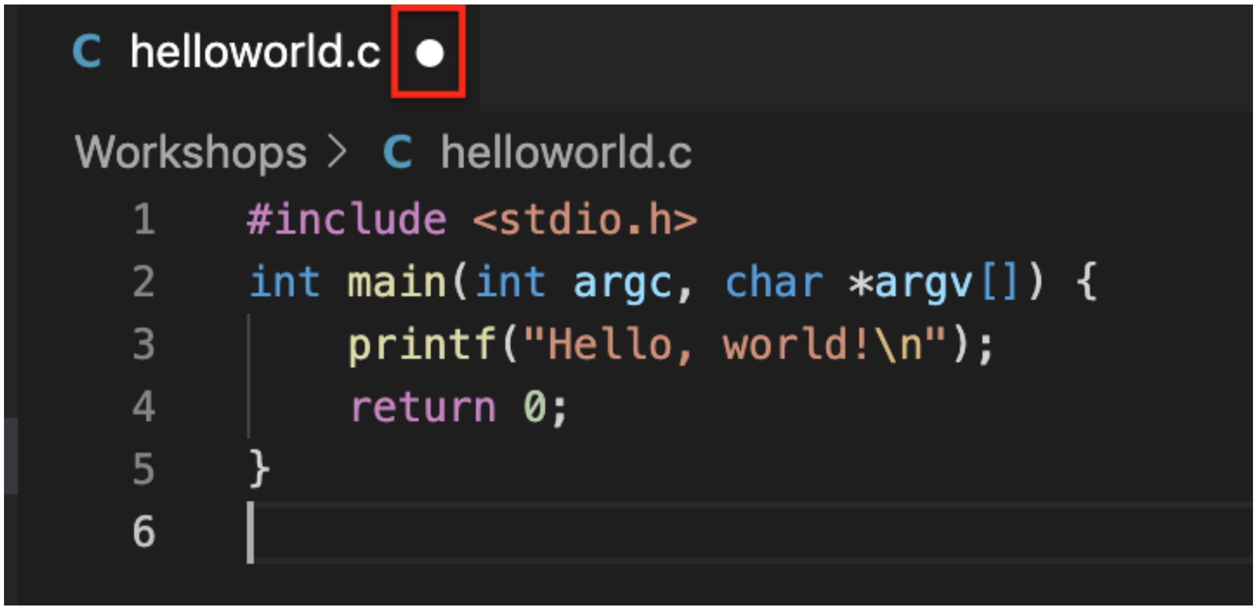
11. Create a new file by secondary click and “New File”. Alternatively, the document with a plus icon also creates a new file.



12. Name the file “helloworld.c” and you can type code in the section on the right that opens.

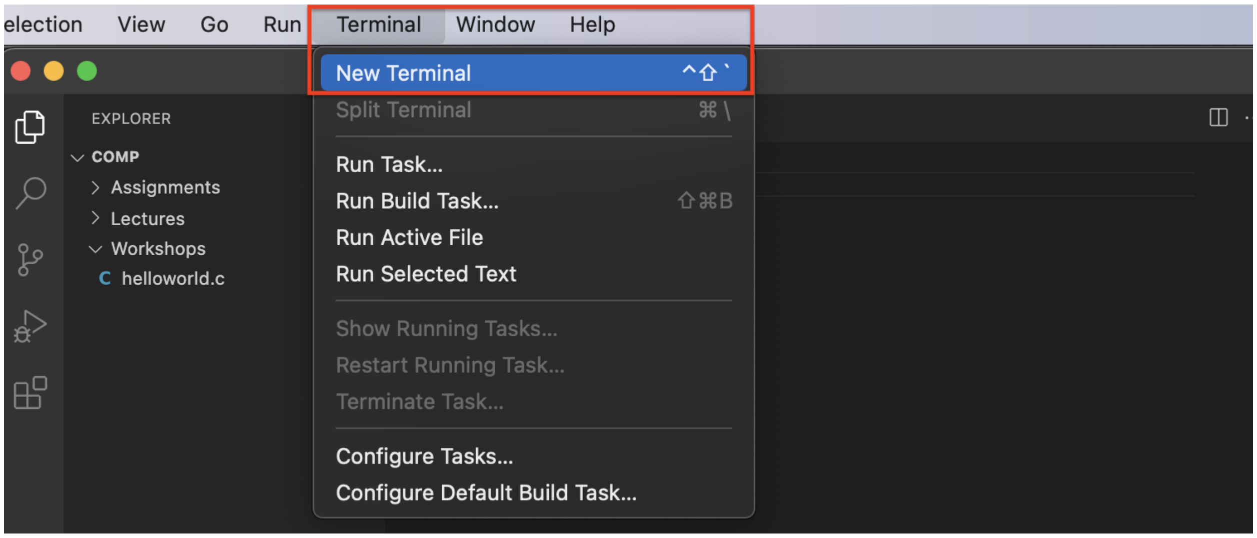


13. Note the circle next the file name, this is because it has not been saved since it's last changes. Save with a shortcut, "ctrl+s" or "command+s" depending on your OS. Alternatively use the menu, and select "File" then "Save".

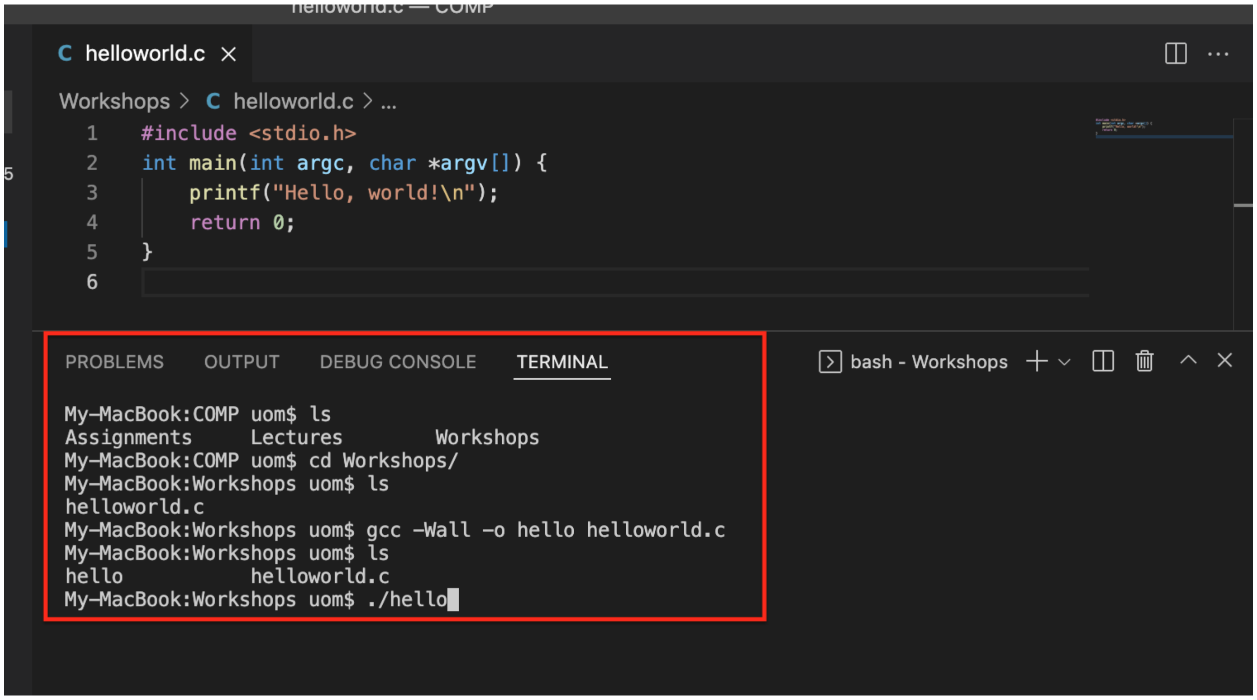


```c
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

14. You can now compile your program to run it with a Terminal window. Note, if you have not installed GCC please follow those steps first.



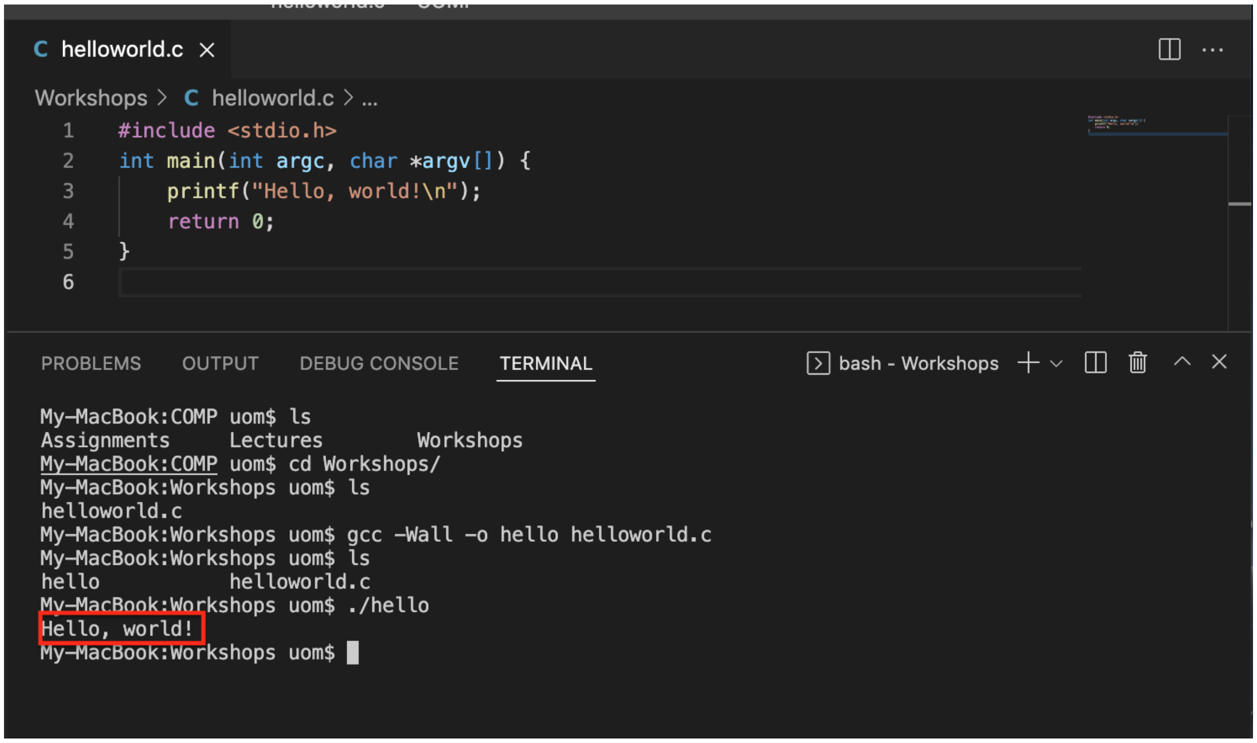
15. You can type commands to see your current accessible files "ls" in the Terminal, to navigate to your folder "cd <folder_name>", to compile your program "gcc -Wall -o <executable_name> <C_program_name>" and finally, to execute it "./<executable_name>". Note, these commands are further described after step 16.



```c
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

```
My-MacBook:COMP uom$ ls
Assignments     Lectures        Workshops
My-MacBook:COMP uom$ cd Workshops/
My-MacBook:Workshops uom$ ls
helloworld.c
My-MacBook:Workshops uom$ gcc -Wall -o hello helloworld.c
My-MacBook:Workshops uom$ ls
hello           helloworld.c
My-MacBook:Workshops uom$ ./hello
```

16. Your program will execute in the Terminal window



```c
#include <stdio.h>
int main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

```
My-MacBook:COMP uom$ ls
Assignments     Lectures        Workshops
My-MacBook:COMP uom$ cd Workshops/
My-MacBook:Workshops uom$ ls
helloworld.c
My-MacBook:Workshops uom$ gcc -Wall -o hello helloworld.c
My-MacBook:Workshops uom$ ls
hello           helloworld.c
My-MacBook:Workshops uom$ ./hello
Hello, world!
My-MacBook:Workshops uom$
```

Running Programs via Interactive Shells

For non-trivial programs, it is much more convenient to run them directly from an interactive shell (command line) than via VSCode's debugger or Grok's Run button, especially if input is to be taken from a file, or output collected into a file. Here are the commands you need. As using interactive shell (i.e., cmd/DOS in Windows) is required in this subject, you should familiarise yourself with these commands.

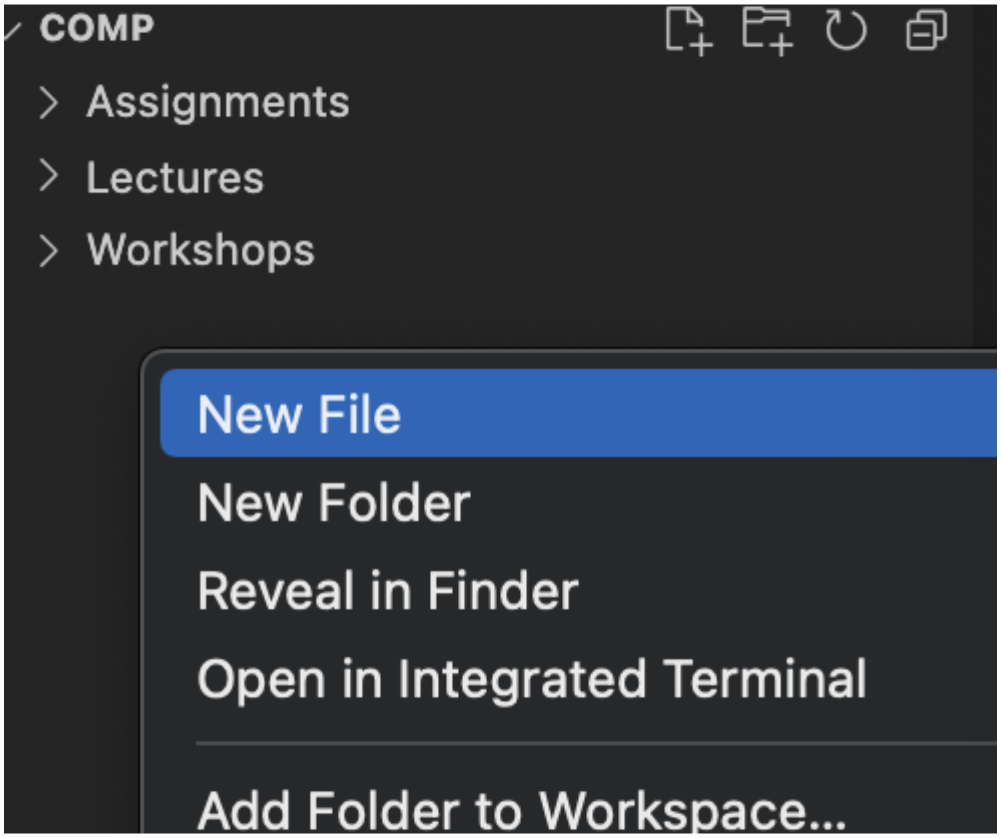
Operation	Command for Mac Terminal or WSL terminal
Start a window to execute a shell	On MacOS, find "Applications", then "Utilities", then double click "Terminal" (might also be available directly from your dock, and if it isn't you can put it there). Open "Powershell" and enter "wsl"
See where you are	pwd
Generate a directory listing	ls
Change to a subdirectory	cd *directory* Note that typing tab may do filename completion, and directory .. is the parent.
Move up one directory	cd ..
Delete a file or directory	**rm** *filename* **rmdir** *directoryname*
Execute a program	*./project*
View a text file	more *filename*

Use of "<" does input file redirection, and ">" does output file redirection.

MacOS provides the ability to open a terminal at any directory with a secondary click.

Additional Setup for VSCode

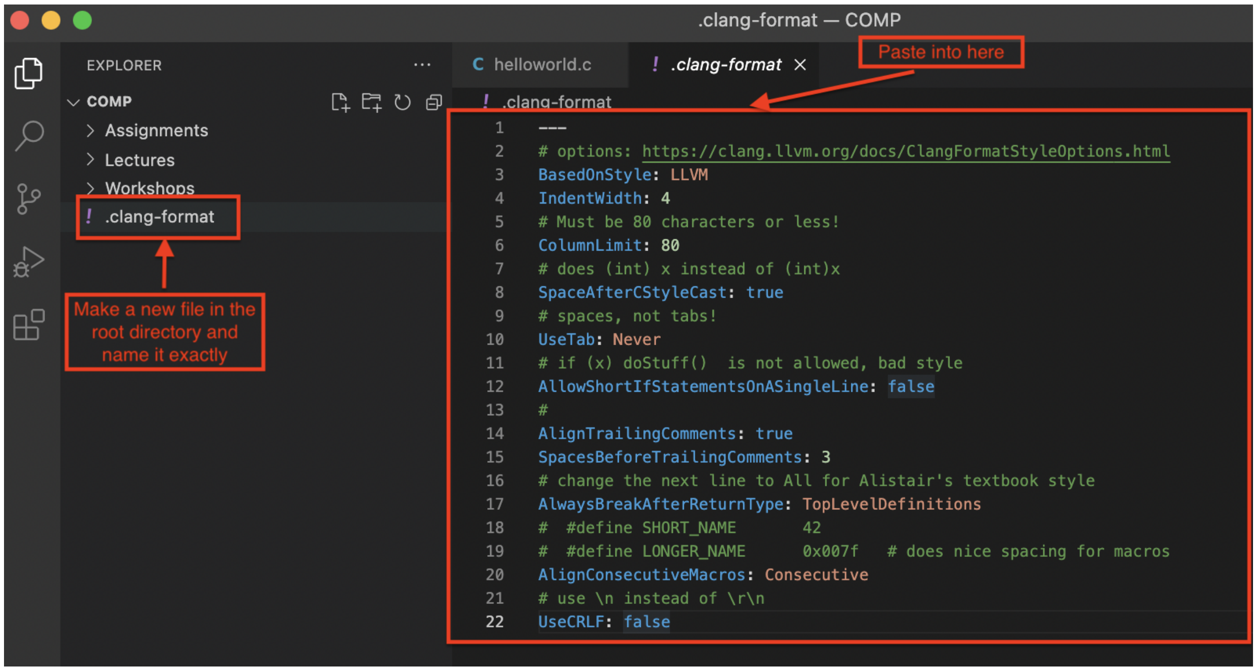
17. Create a new file in the root directory, in this case the “COMP” folder.



18. Name it “.clang-format” and paste the following in to the file (sourced from https://hackmd.io/@exradr/clang-format):

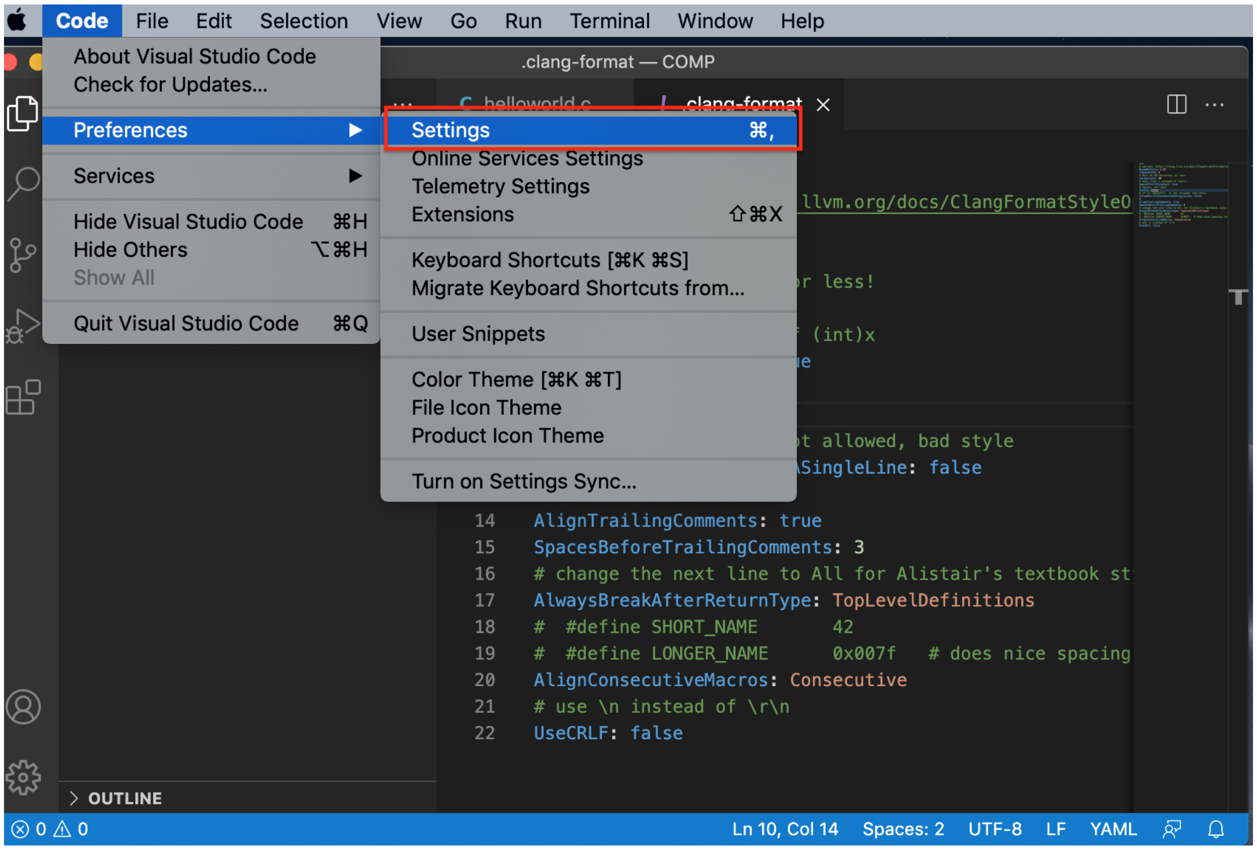
```
---
# options: https://clang.llvm.org/docs/ClangFormatStyleOptions.html
BasedOnStyle: LLVM
IndentWidth: 4
# Must be 80 characters or less!
ColumnLimit: 80
# does (int) x instead of (int)x
SpaceAfterCStyleCast: true
# spaces for indentation, not tabs!
UseTab: Never
# if (x) doStuff()  is not allowed, bad style
AllowShortIfStatementsOnASingleLine: false
AlignTrailingComments: true
SpacesBeforeTrailingComments: 3
```

```
# change the next line to All for Alistair's textbook style
AlwaysBreakAfterReturnType: TopLevelDefinitions
#  #define SHORT_NAME       42
#  #define LONGER_NAME      0x007f   # does nice spacing for macros
AlignConsecutiveMacros: Consecutive
# use \n instead of \r\n - prevents issues on Windows
UseCRLF: false
```



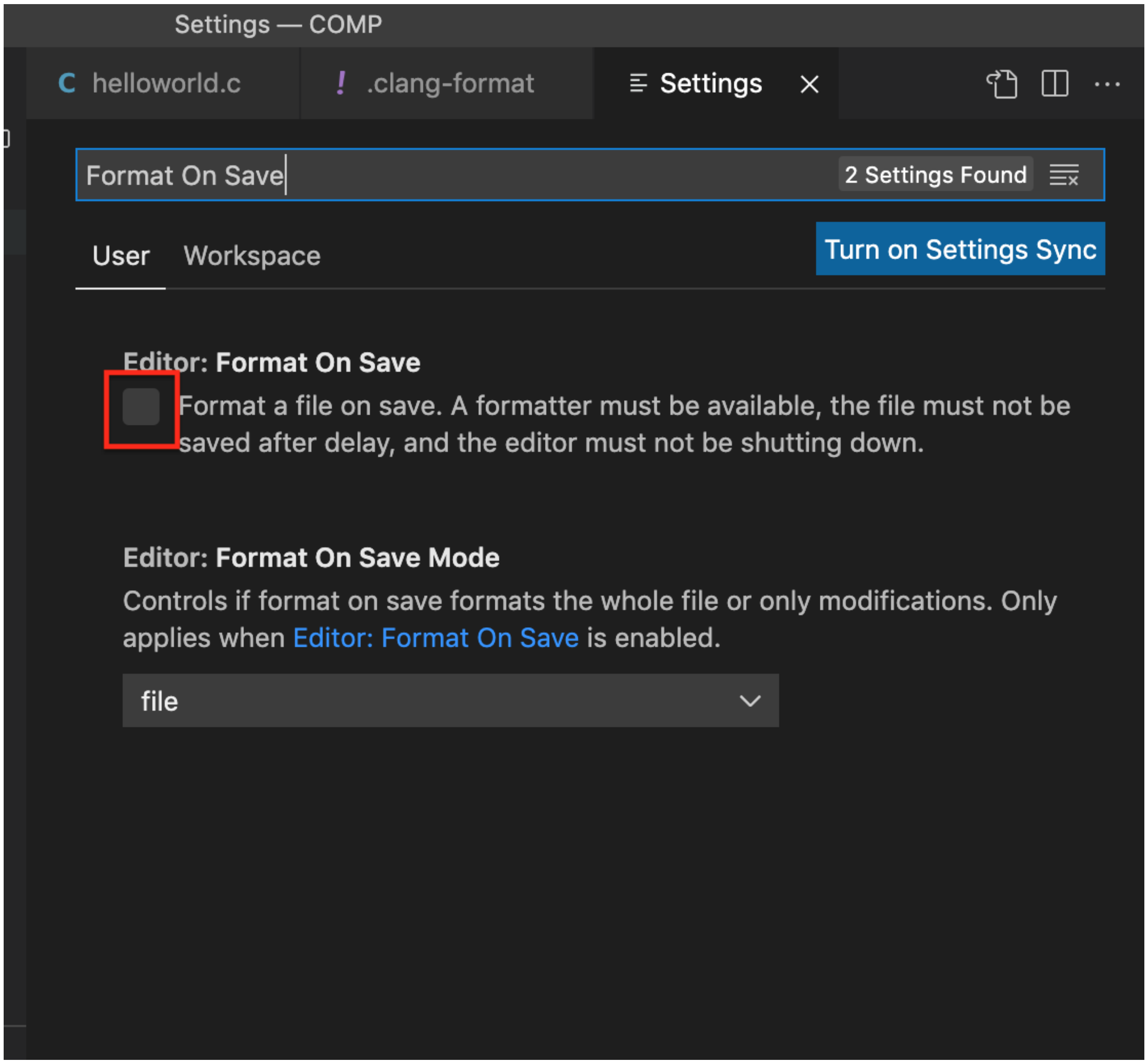
```
---
# options: https://clang.llvm.org/docs/ClangFormatStyleOptions.html
BasedOnStyle: LLVM
IndentWidth: 4
# Must be 80 characters or less!
ColumnLimit: 80
# does (int) x instead of (int)x
SpaceAfterCStyleCast: true
# spaces, not tabs!
UseTab: Never
# if (x) doStuff()  is not allowed, bad style
AllowShortIfStatementsOnASingleLine: false
#
AlignTrailingComments: true
SpacesBeforeTrailingComments: 3
# change the next line to All for Alistair's textbook style
AlwaysBreakAfterReturnType: TopLevelDefinitions
#  #define SHORT_NAME       42
#  #define LONGER_NAME      0x007f   # does nice spacing for macros
AlignConsecutiveMacros: Consecutive
# use \n instead of \r\n
UseCRLF: false
```

19. Go to "Settings". You can use the shortcut, "command + ," or "ctrl + ," depending on your OS.



```
llvm.org/docs/ClangFormatStyleO
r less!
(int)x
e
t allowed, bad style
SingleLine: false
AlignTrailingComments: true
SpacesBeforeTrailingComments: 3
# change the next line to All for Alistair's textbook st
AlwaysBreakAfterReturnType: TopLevelDefinitions
#  #define SHORT_NAME       42
#  #define LONGER_NAME      0x007f   # does nice spacing
AlignConsecutiveMacros: Consecutive
# use \n instead of \r\n
UseCRLF: false
```

20. Do a search in the search bar for “Format On Save” and tick the box.



Editor: Format On Save

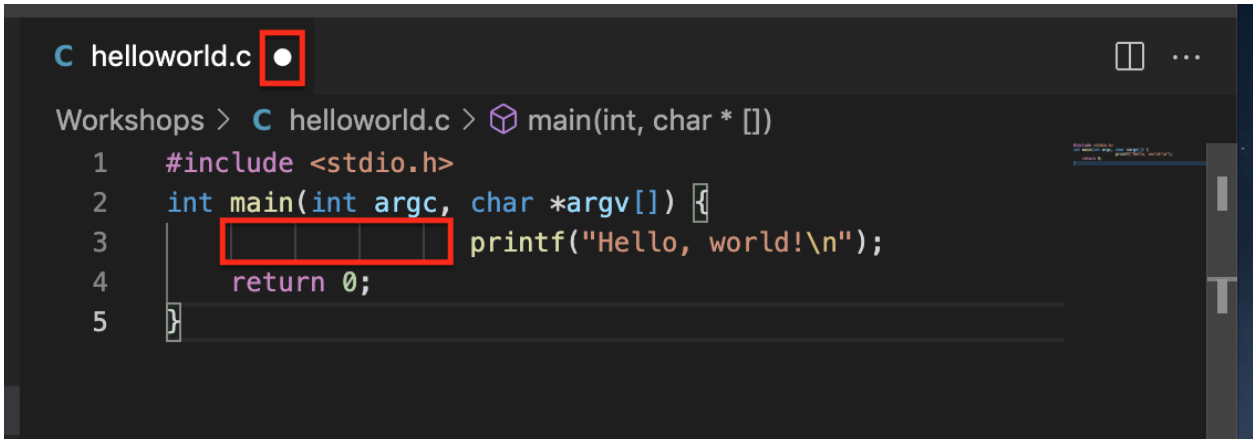
Format a file on save. A formatter must be available, the file must not be saved after delay, and the editor must not be shutting down.

Editor: Format On Save Mode

Controls if format on save formats the whole file or only modifications. Only applies when Editor: Format On Save is enabled.

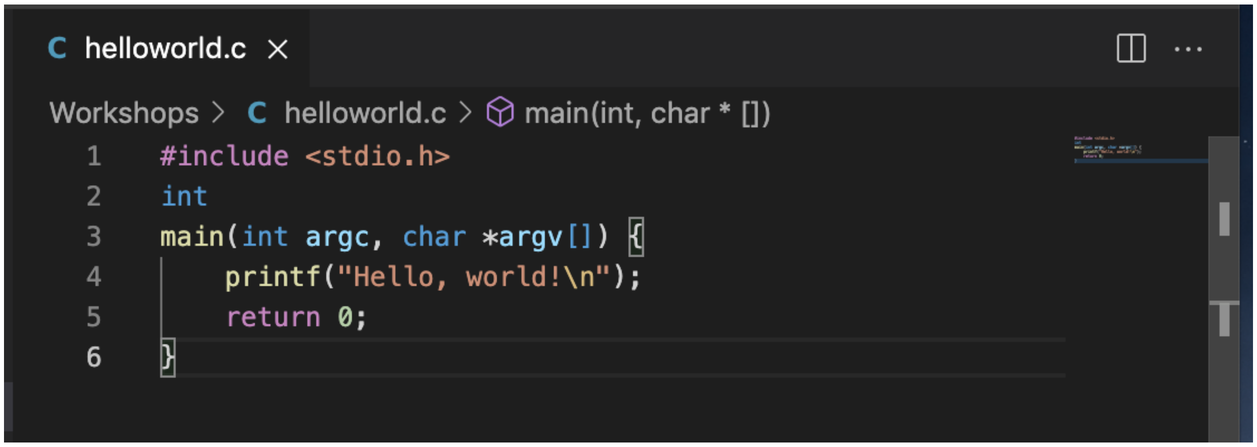
file

21. You can test the autoformatting by editing the “helloworld.c” program from earlier. Add in some inconsistent indentation and notice the circle showing changes.



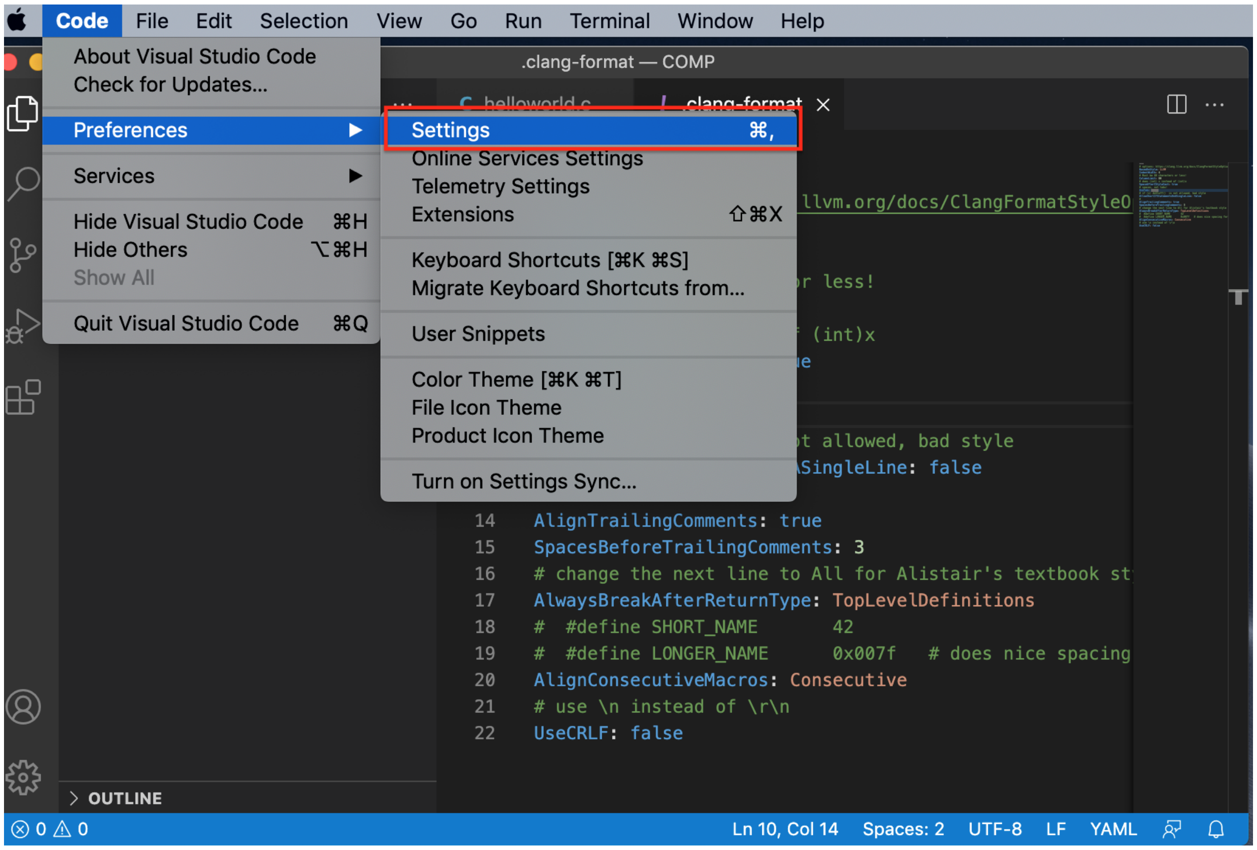
```c
#include <stdio.h>
int main(int argc, char *argv[]) {
                printf("Hello, world!\n");
    return 0;
}
```

22. Save the file and the program will fix the formatting issues.



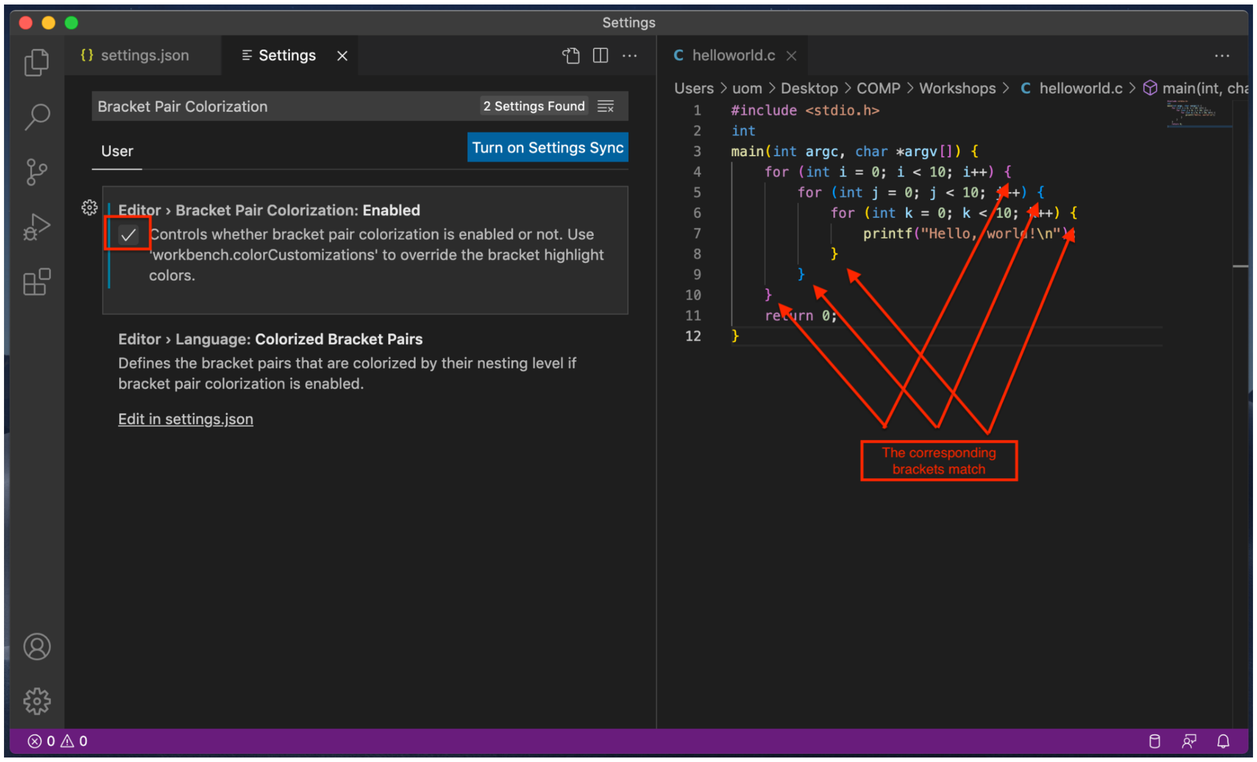
```c
#include <stdio.h>
int
main(int argc, char *argv[]) {
    printf("Hello, world!\n");
    return 0;
}
```

23. Go back to “Settings”



```
AlignTrailingComments: true
SpacesBeforeTrailingComments: 3
# change the next line to All for Alistair's textbook st
AlwaysBreakAfterReturnType: TopLevelDefinitions
#  #define SHORT_NAME       42
#  #define LONGER_NAME      0x007f   # does nice spacing
AlignConsecutiveMacros: Consecutive
# use \n instead of \r\n
UseCRLF: false
```

24. Search for “Bracket Pair Colorization” and tick the box. Note that each set of brackets in the nested code has a unique colouring for easier identification.



Settings
settings.json
Settings
helloworld.c
Users > uom > Desktop > COMP > Workshops > helloworld.c > main(int, cha
Bracket Pair Colorization
2 Settings Found
User
Turn on Settings Sync
0 0
The corresponding brackets match

Editor › Bracket Pair Colorization: Enabled

✓ Controls whether bracket pair colorization is enabled or not. Use 'workbench.colorCustomizations' to override the bracket highlight colors.

Editor › Language: Colorized Bracket Pairs

Defines the bracket pairs that are colorized by their nesting level if bracket pair colorization is enabled.

Edit in settings.json

```
#include <stdio.h>
int
main(int argc, char *argv[]) {
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++) {
            for (int k = 0; k < 10; k++) {
                printf("Hello, world!\n");
            }
        }
    }
    return 0;
}
```

Prepared by Thomas Minuzzo and Liam Saliba, February 2022