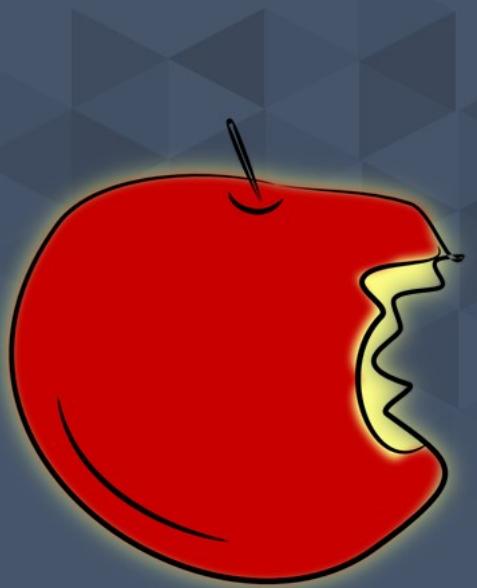
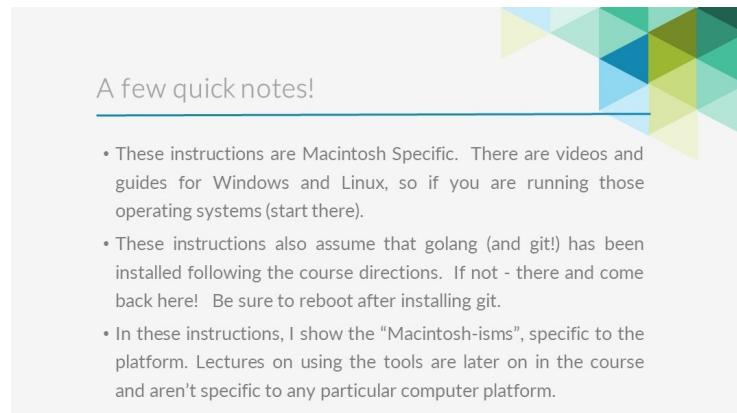




*The Complete Google
Go Programming
Course For Beginners*

OSX
Installation
Visual Studio
Code



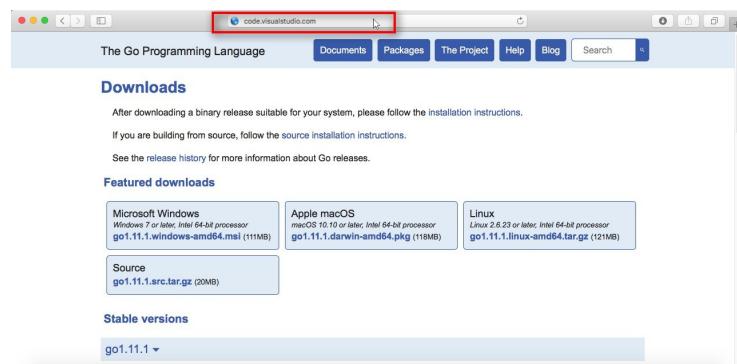


A few quick notes!

- These instructions are Macintosh Specific. There are videos and guides for Windows and Linux, so if you are running those operating systems (start there).
- These instructions also assume that golang (and git!) has been installed following the course directions. If not - there and come back here! Be sure to reboot after installing git.
- In these instructions, I show the "Macintosh-isms", specific to the platform. Lectures on using the tools are later on in the course and aren't specific to any particular computer platform.

Visit code.visualstudio.com

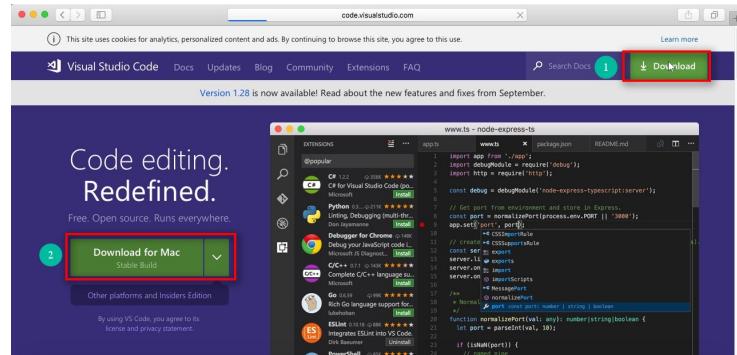
Launch your favorite web browser and visit code.visualstudio.com



Download VS Code

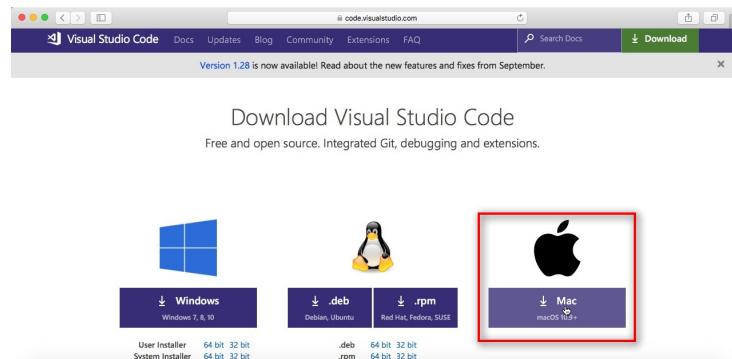
Click download at (1) to see different versions and choose one to download.

Alternatively, you can select "Download for Mac" in (2). The instructions are for download (1) in the top right corner

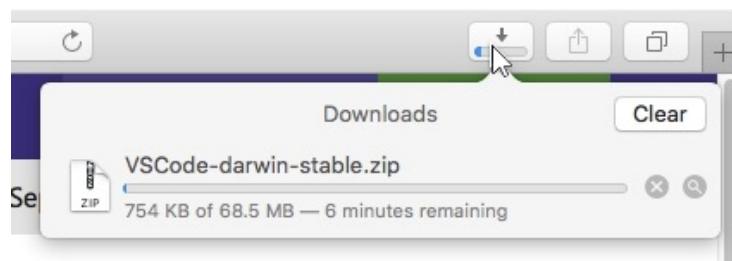


Select the Mac Version

Select the Mac version to begin the download.

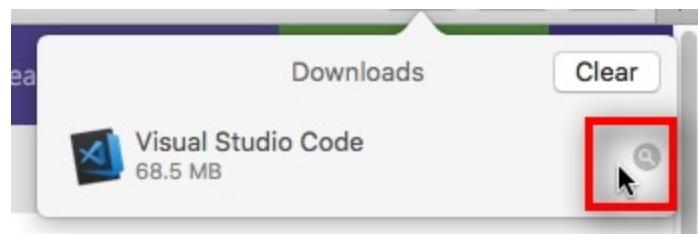


Download will begin



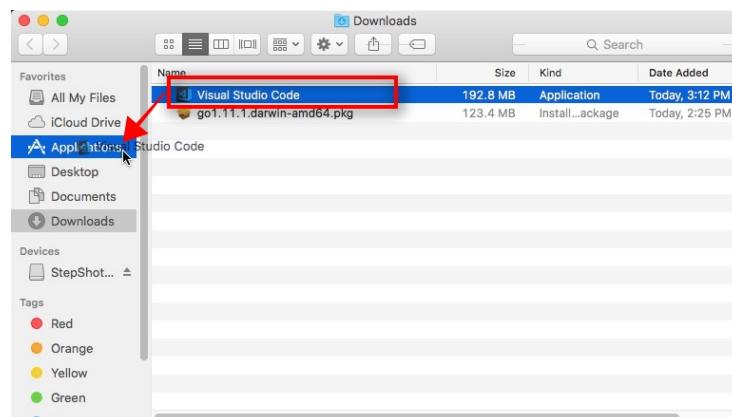
Click on the finder icon to open.

Click on the finder icon to open up the download folder.



Drag Visual Studio Code to the Application Folder

Click and hold on "Visual Studio Code", and hover on the application folder.



Release Visual Studio Code

Release Visual Studio Code in the Applications folder



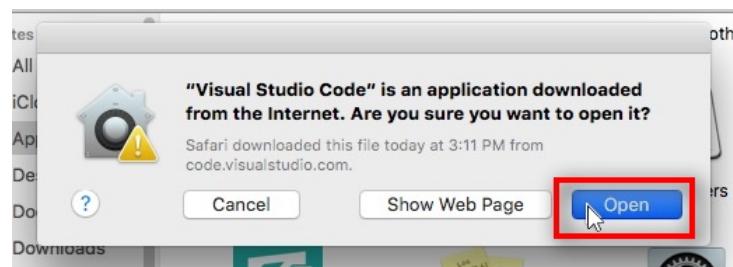
Right Click / Select Open

Right click on VS Code and select open.



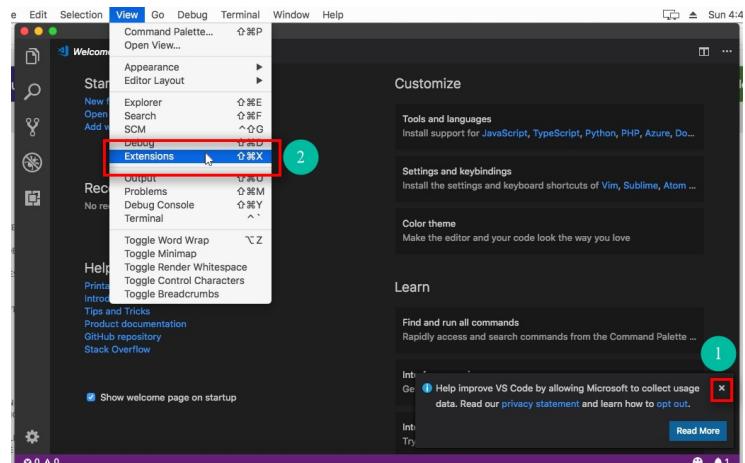
Select Open

Select "Open" to continue opening VS Code. Later, Mac OS will remember your selection and you can launch it from the Launchpad directly.



Close Dialog & Open Extensions

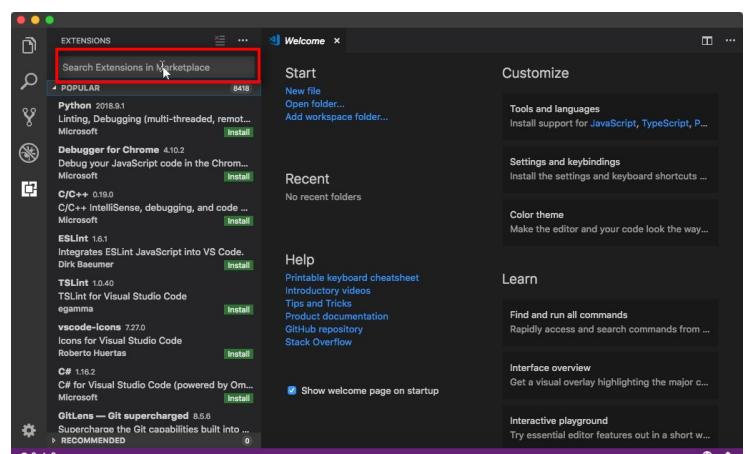
(1) Select close on the dialog box for usage data (you can review this later).



(2) Select View / Extensions to open the extensions window.

Search for the Go extension

In the dialog box, type in Go

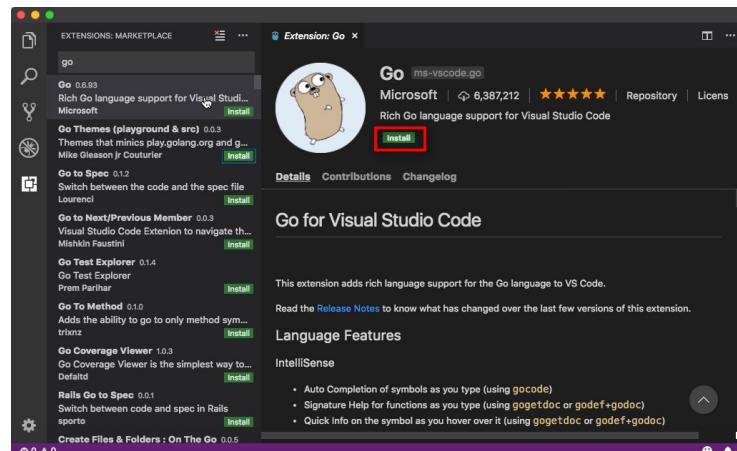


Go for Visual Studio Code

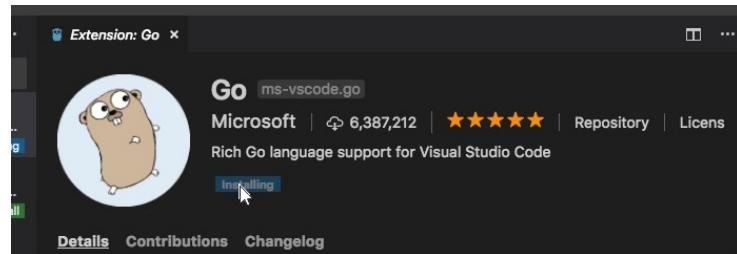
Go for Visual Studio Code, by Microsoft is the extension desired.

It will likely be the first one in the list when searched.

Click install to begin the installation process.

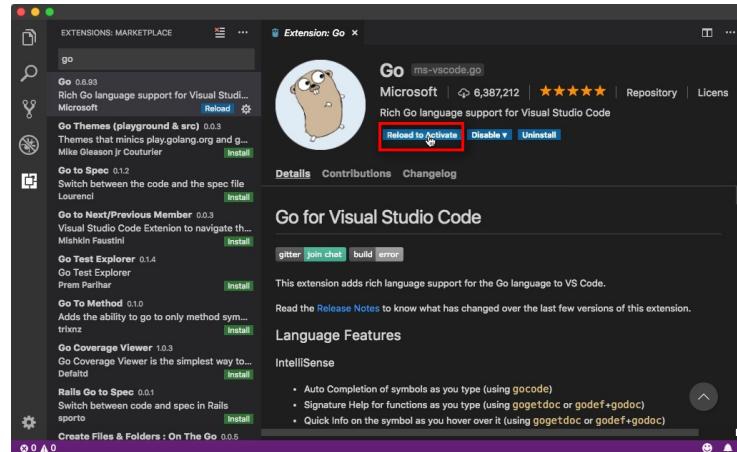


Installation will begin



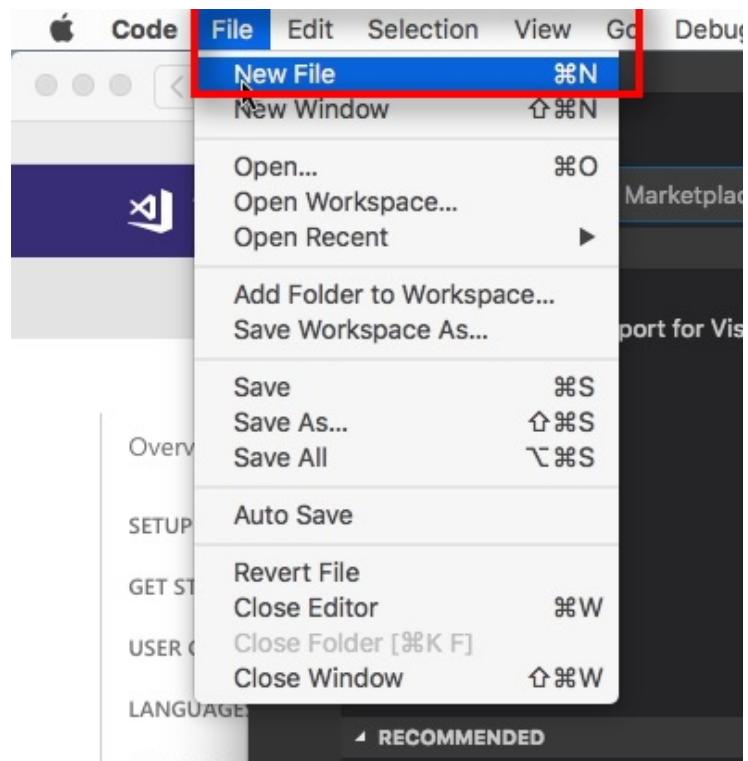
Reload to Activate

Once installation is complete, select "Reload to Activate", the extension in VS Code.



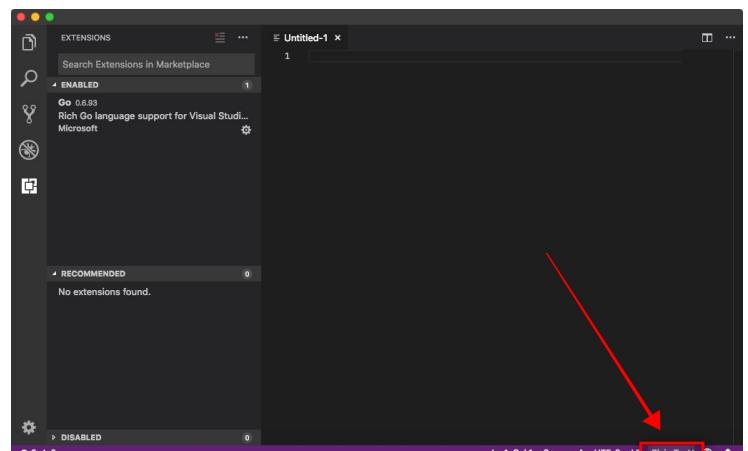
Open a New File

Select File / New File. We'll use this to continue the installation process



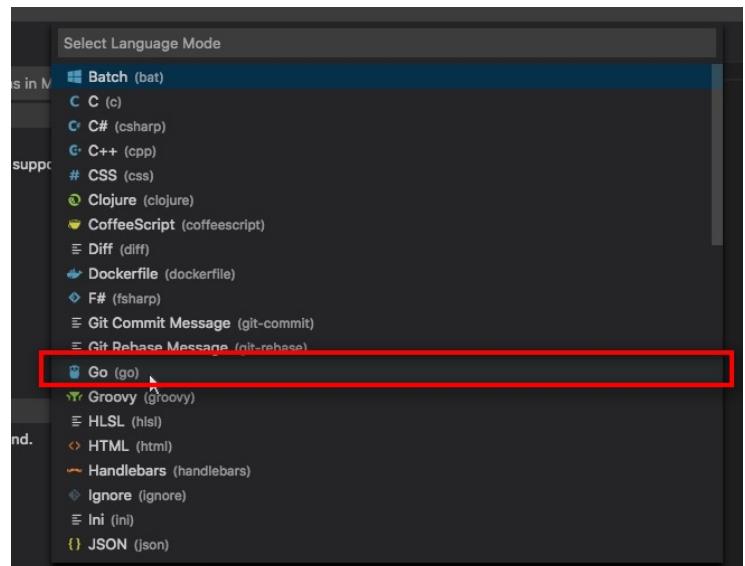
Change "Plain Text"

Select Plain Text in the bottom right hand corner



Select Go

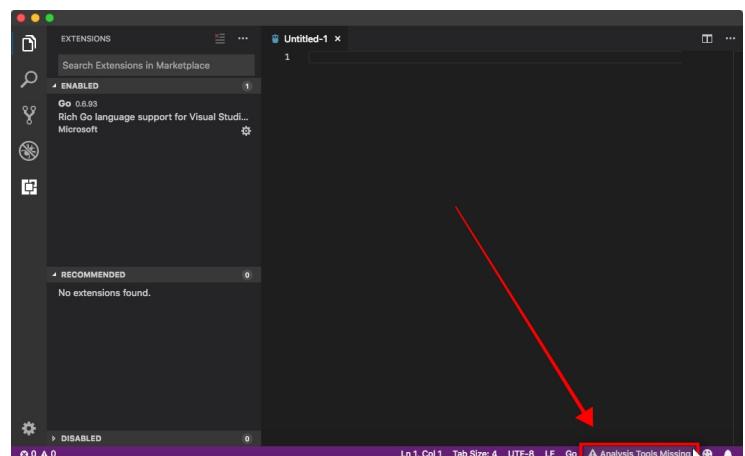
Select Go from the dropdown.
This will force the type to
change to a go source code
file in VS Code.



Analysis Tools Missing

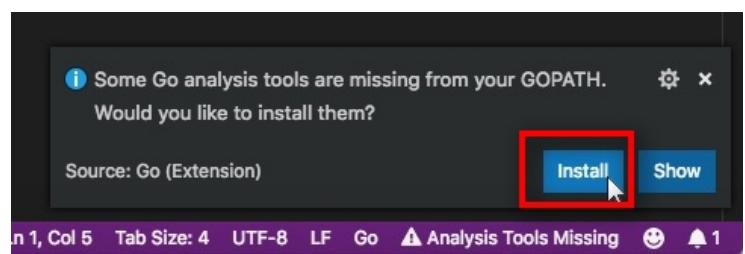
Once VS Code realizes the analysis tools for go are missing, it will put a message on the bottom right hand corner.

Click on it.



Analysis tools

Select Install to install the analysis tools into VS code for Go



Installation will begin

The output console will show you the progress of the installation. A total of 12 tools will be installed, so this will take some time. Don't rush it.

A screenshot of the Visual Studio Code interface. The left sidebar shows the Extensions view with one extension named 'Go' listed under 'ENABLED'. The main workspace shows an untitled file with a single character '1'. Below the workspace is the 'OUTPUT' tab of the Command Palette, which displays the following text:

```
Installing 12 tools at /Users/dave/go/bin
gocode
gopkgs
go-outline
go-symbols
guru
gorename
dlv
gocode-gomod
godef
godef-gomod
goreturns
golint
```

Installation is Complete

1) Installation is complete once you get the message "All tools successfully installed. You're ready to Go :)."

A screenshot of the Visual Studio Code interface. The left sidebar shows the Extensions view with one extension named 'Go' listed under 'ENABLED'. The main workspace shows an untitled file with a single character '1'. Below the workspace is the 'OUTPUT' tab of the Command Palette, which displays the following text:

```
goreturns
golint
```

Below this, there is a list of successful installations:

```
Installing github.com/mdempsky/gocode SUCCEEDED
Installing github.com/uudashr/gopkgs/cmd/gopkgs SUCCEEDED
Installing github.com/ramya-rao-a/go-outline SUCCEEDED
Installing github.com/acroca/go-symbols SUCCEEDED
Installing golang.org/x/tools/cmd/guru SUCCEEDED
Installing golang.org/x/tools/cmd/gorename SUCCEEDED
Installing github.com/derekparker/delve/cmd/dlv SUCCEEDED 2
Installing github.com/stamblerre/gocode SUCCEEDED
Installing github.com/rogpeppe/godef SUCCEEDED
Installing github.com/ianthehat/godef SUCCEEDED
Installing github.com/sqs/goreturns SUCCEEDED
Installing golang.org/x/lint/golint SUCCEEDED
```

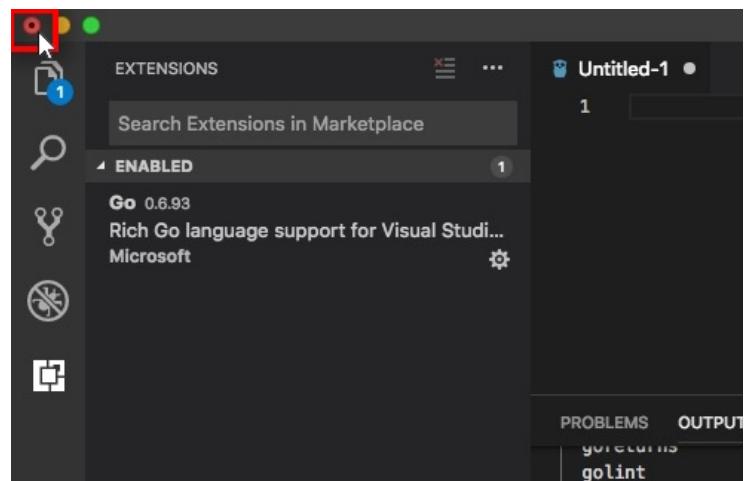
At the bottom of the output window, a message indicates the completion of the process:

All tools successfully installed. You're ready to Go :). 1

2) Note that this is delve - a debugger for the Go programming language

Close VS Code

Close VS Code.



Don't save

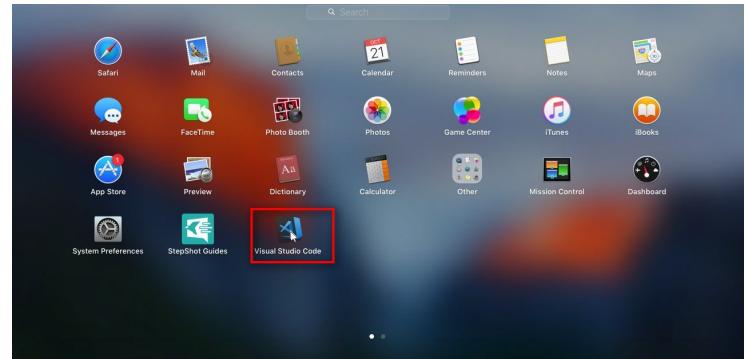
You can ignore the empty program we were creating, and select "Don't Save"



Select Launchpad

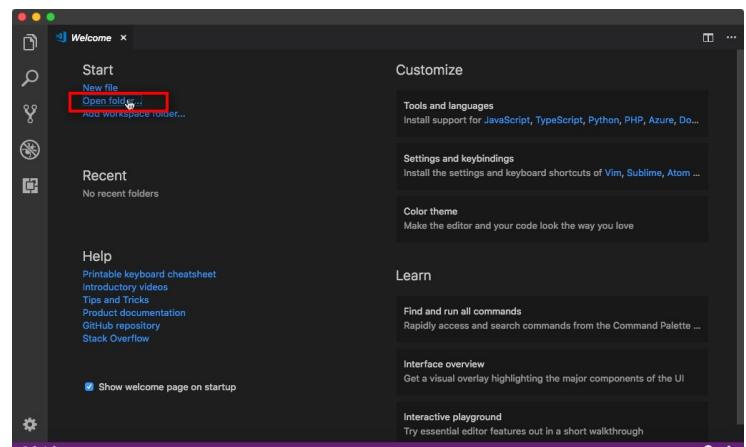


Launch Visual Studio Code



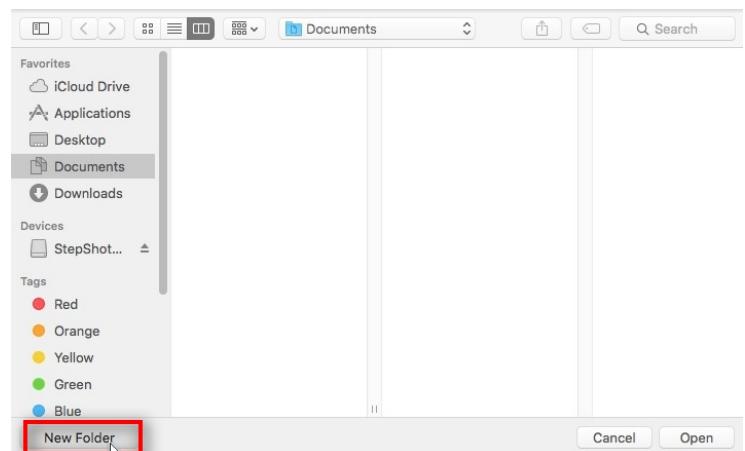
Creating a project folder

Select open folder, to begin creating a project folder where we'll have our Go code experiments.



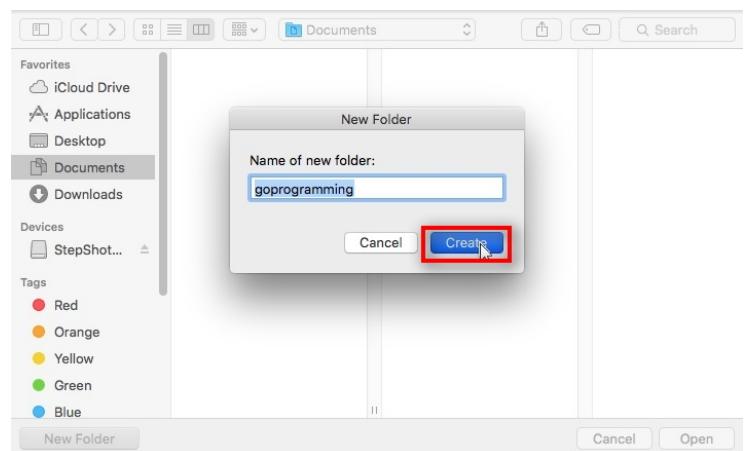
Create new folder

From documents, select on the bottom left "New folder", to create a new folder



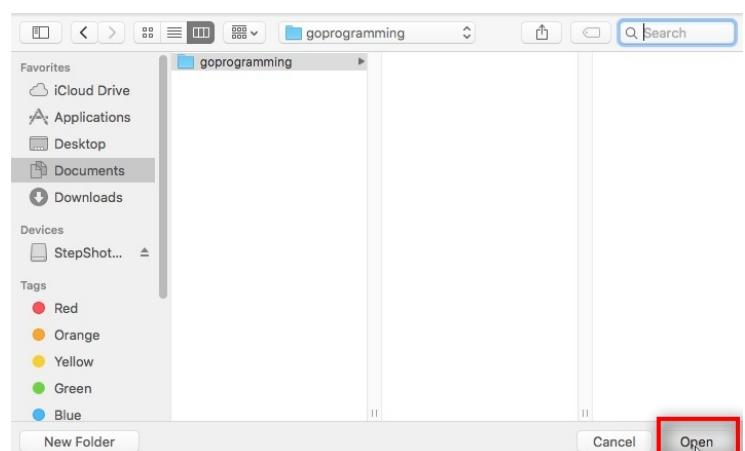
Create goprogramming

Name the folder goprogramming, and select create.



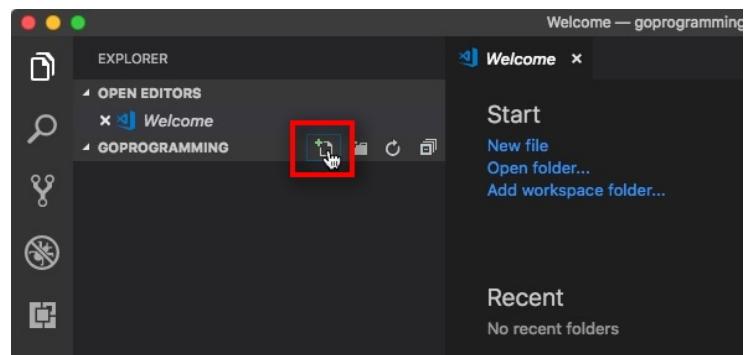
Open

Once created, simply select open



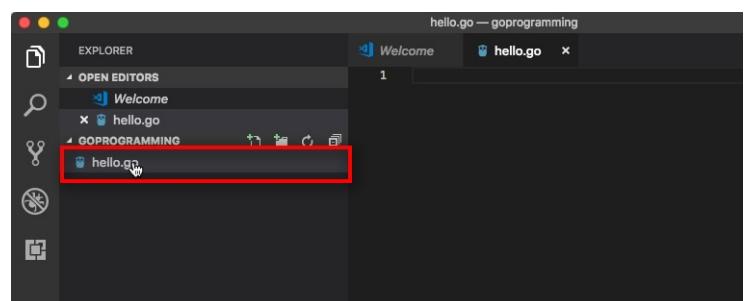
Create a new file

From goprogramming, select the new file icon.



hello.go

Enter hello.go and select it to open the hello.go source code file into the editor.



Enter in the first golang program into the window.

```
// hello.go  
package main
```

A screenshot of the VS Code interface showing the content of the 'hello.go' file. The code is:

```
1 // hello.go  
2 package main  
3  
4 import "fmt"  
5  
6 func main() {  
7     fmt.Println("Hello")  
8 }
```

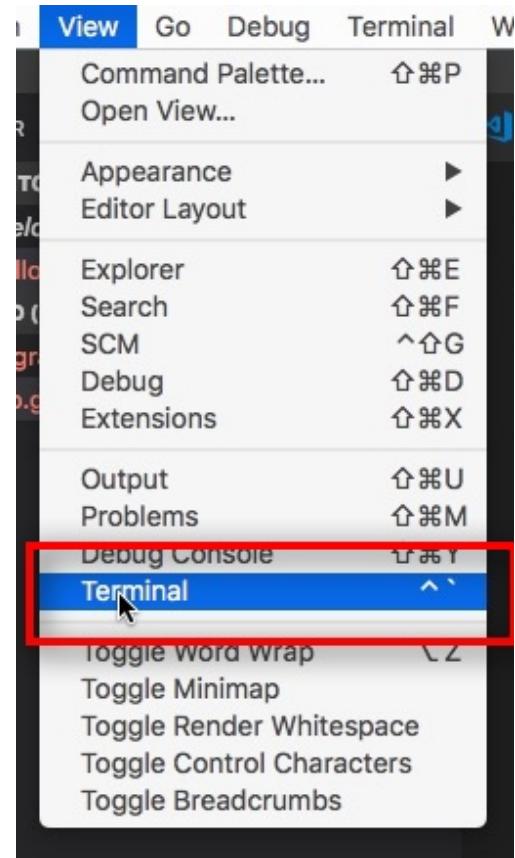
The code editor has syntax highlighting, with 'fmt' and 'Println' in orange.

```
import "fmt"
```

```
func main() {  
    fmt.Println("Hello")  
}
```

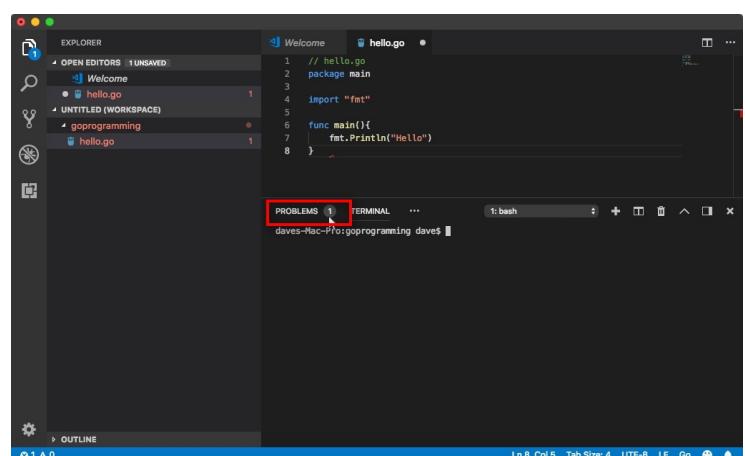
Open the terminal view

Select view/terminal to launch the terminal.



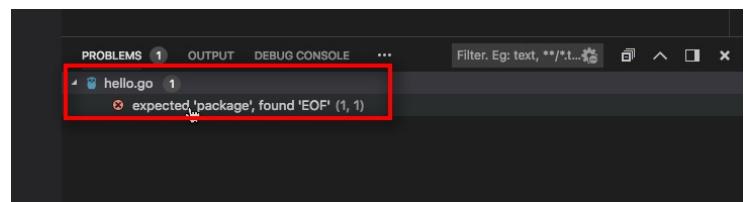
Notice there are problems reported.

Select the problems tab to see the problems reported.



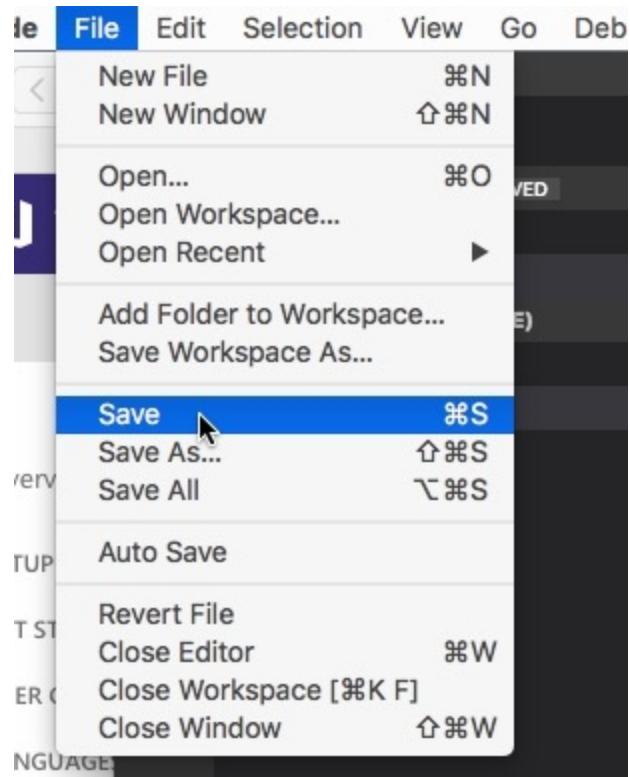
Expected 'package' error

Expected 'package', found 'EOF', is a common error message. It's solved by saving our program.



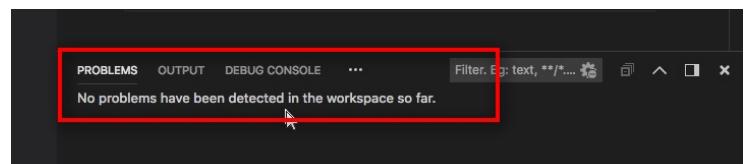
Save the program

Select File/Save to save the program.



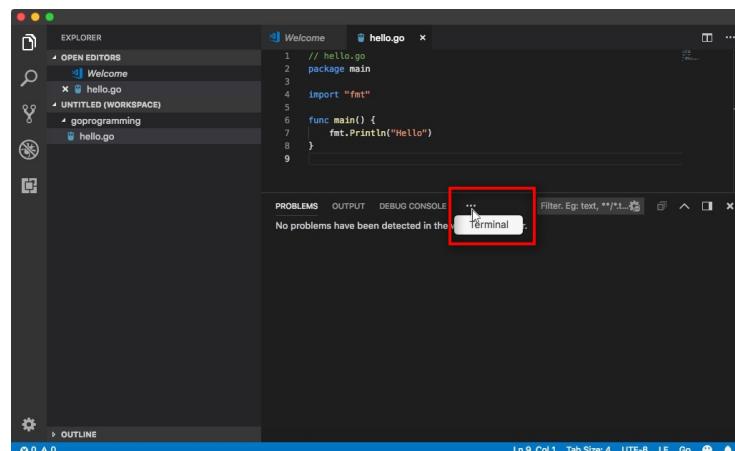
Problem solved

Notice the problem is no longer present once the file is saved.



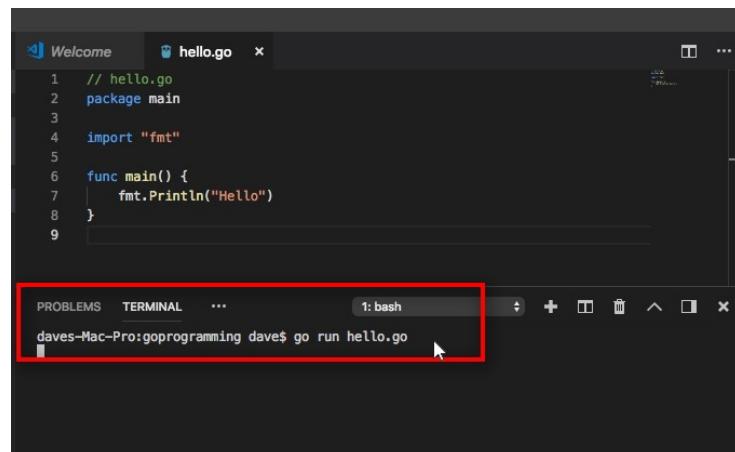
Re-open terminal

Click on the three dots if not visible, and then select terminal.



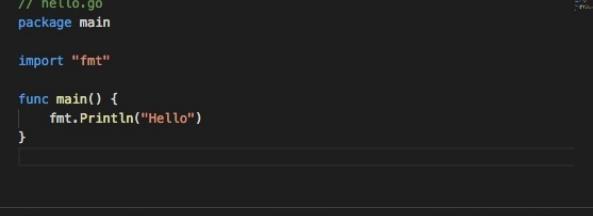
Method 1 of running programs [Command Line]

*To run the program successfully, you can start go and simply type in the terminal window:
go run hello.go
and press enter.*



The program runs

The program successfully runs, and you see Hello appear in there terminal.



```
// hello.go
package main

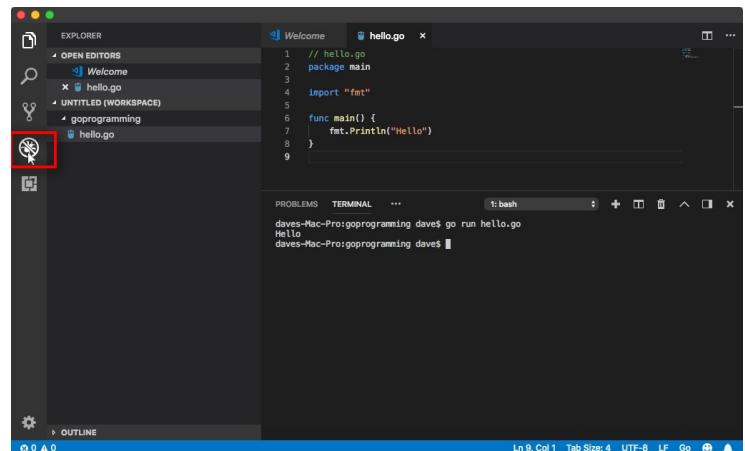
import "fmt"

func main() {
    fmt.Println("Hello")
}

PROBLEMS TERMINAL ...
1: bash + - ×
daves-Mac-Pro:goprogramming dave$ go run hello.go
Hello
daves-Mac-Pro:goprogramming dave$
```

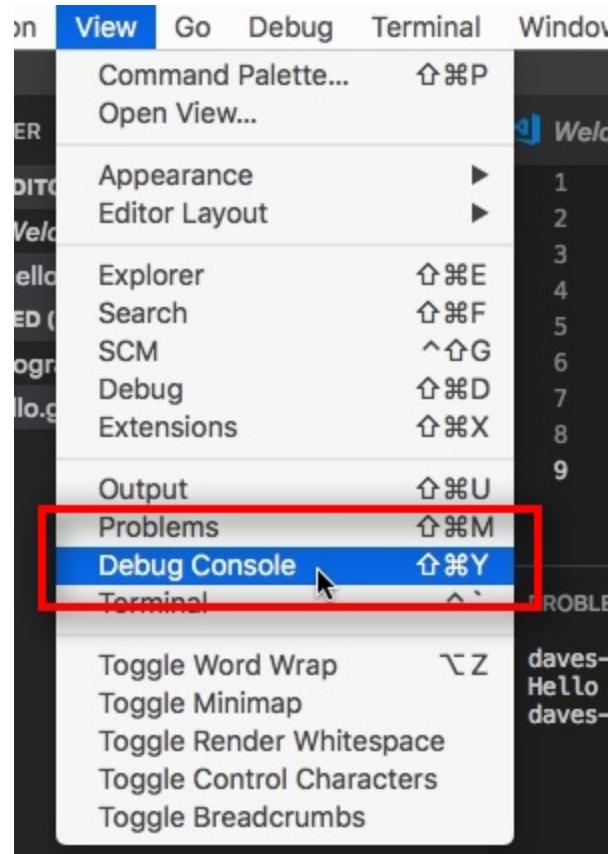
Method 2 of running programs [Debugger]

Start the debugger by selecting the "Debug option" on the left.



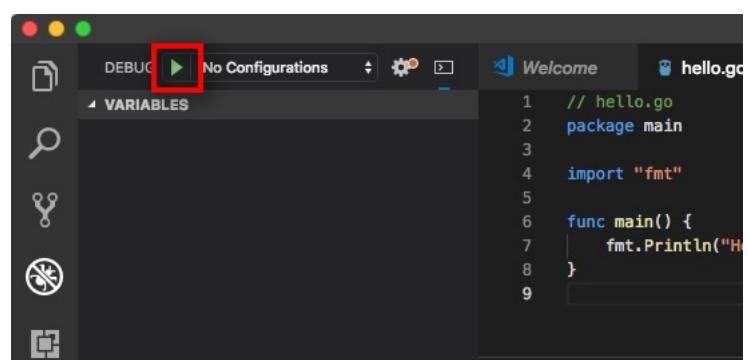
Also open the debug console

Select View / Debug Console to see the output of the program being run.



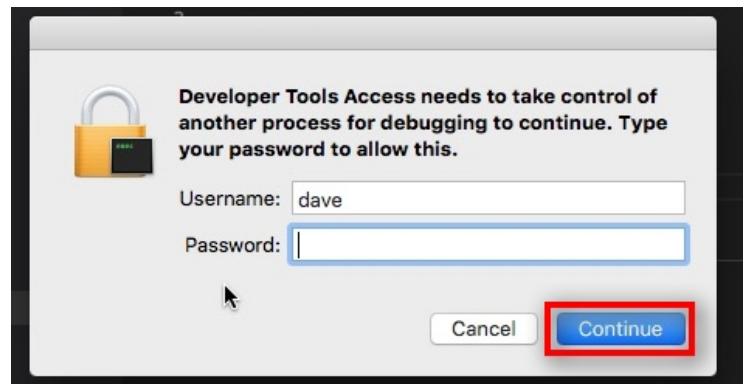
Run the program

Run the program by selecting the green icon on the top left.



Password prompt

You'll be prompted the first time you run the debugger to allow the debug process to continue. Enter in your password, and then select continue.



The program runs

Finally, the debug console will show the output of the program running.

