# Visual Studio Code Tutorial
# Getting Started With VS Code

## VS Code For Beginners - Programming For Beginners - Learn To Code Using Visual Studio Code

Todd Mcleod

---

## Table of Contents

# Getting started

## Welcome

Welcome to the course. You are taking a great step by enrolling in this course. Better skills create a better life. ***You are on your way to a better life.*** As you learn new skills, you are building a better life. ***I commend you for your efforts to improve your life.*** As you improve your life, you are improving the world --- one person at a time. You are making the world better, and you are making your life better. This is a win-win for everybody. Great work! Also, ***this is your course. Use it in the way which is best for you.*** If you want to skip ahead, skip ahead. As your teacher, my job is to help you succeed. The content here is designed to help you succeed both with visual studio code, and also as a student and in life.
Video file: 01-welcome

## Course resources

The course outline can be downloaded from Udemy.

- github
    - https://github.com/GoesToEleven/vscode-tutorial
    - https://github.com/GoesToEleven/cit90fa2019

Video file: 03-resources

# Learning tools

## Exercises

***The exercises are for your benefit.*** The more you do something, the better you get at it. The exercises ***are not required.***

Sometimes my courses are used in classes at colleges, universities, high schools, and other schools. If this is the case and this course here is being used in a course you are taking at a school, your teacher there might require that you do the assignments. If that is your case, check with your teacher.

Video file: 09-exercises

## Acceleration

**You can change the speed of videos: either slow them down, or speed them up.** Adjust the speed as necessary for your learning style. Fast-forward if you want to skim over material. Slow me down if you need me to go slower.

Video file: 05-accelerate-learning

# Coding fundamentals

## Introduction

This section will show you how to set up your development environment. If you are experienced with programming, you can skip this section and any other sections you want. This is ***your course*** after all. I am here to help you. As there are a variety of students who will take this course, I believe in providing a ***clear continuity of curriculum*** so that even those who are just getting started have a ***pathway to success.***

video: 11-intro

## The terminal

- terminology
    - GUI = graphical user interface

        - CLI = command line interface - **command line**
            - **terminal** = text input/output environment; console = physical terminal
                - unix / linux / mac
                    - **shell / bash / terminal**
                - windows
                    - **command prompt / windows command / cmd / dos prompt**

video: 12-terminal

## Bash on Windows

- https://git-scm.com/
- linux on Windows
    - developer features
        - Linux subsystem for Windows
    - bash
    - article with steps here and another article

video: 13 bash on windows

# Bash fundamentals

## Working with directories

- shell / bash commands
    - **pwd**
    - **ls**
    - **cd <directory>**
    - **cd ../**
    - **cd** or **cd ~**
    - **mkdir <directory>**
    - **clear** or **command + k**

video: 14 Working With Directories

## Viewing directory details

- shell / bash commands
    - **ls -la**

owner group bytes last modification hidden & name

```
total 72
drwxr-xr-x+ 20 spockmcleod  staff    680 Jul 31 15:44 .
drwxr-xr-x   5 root         admin    170 Dec 30  2016 ..
-r--------   1 spockmcleod  staff      7 Dec 30  2016 .CFUserTextEncoding
-rw-r--r--@  1 spockmcleod  staff  14340 Aug  1 06:52 .DS_Store
drwx------  19 spockmcleod  staff    646 Aug  1 07:07 .Trash
drwxr-xr-x  14 spockmcleod  staff    476 Jul 26 13:56 .atom
-rw-------   1 spockmcleod  staff  10321 Aug  1 07:16 .bash_history
drwx------  41 spockmcleod  staff   1394 Aug  1 07:16 .bash_sessions
drwx------   3 spockmcleod  staff    102 Aug 15  2016 .cups
-rw-------   1 spockmcleod  staff   1024 Dec 30  2016 .rnd
drwx------   4 spockmcleod  staff    136 Feb 15 14:48 Applications
drwx------+  4 spockmcleod  staff    136 Jul 26 13:16 Desktop
drwx------+  9 spockmcleod  staff    306 Jul 31 16:43 Documents
drwx------+  4 spockmcleod  staff    136 Jul 31 10:12 Downloads
```

d = directory  r = read  w = write  x = execute

rwxrwxrwx = owner, group, world

video: 15 Viewing Directory Details

## Working with files

- shell / bash commands
  - **touch <file name>**
    - example: touch temp.txt
  - **nano <file name>**
  - **cat <file name>**

video: 16 Working With Files

## Removing items

- shell / bash commands
  - **rm  <file name>**
  - **rm -rf <file or folder name>**

video: 17 Removing Items

## Working with permissions

- shell / bash commands
  - **chmod <owner group world> <file or folder>**
    - example: chmod 764 temp.txt
  - **sudo**
    - super user do
- permissions
  - owner, group, world

    - **r, w, x, no permission**
    - **4, 2, 1, 0**

r = read  w = write  x = execute
**rwxrwxrwx = owner, group, world**
video: 18 Working With Permissions

# Environment variables

## Review
video: 19 Review

## Introduction
- Environment variables are a set of ***variables on a computer.*** They are part of the environment in which a process runs.
- shell / bash commands
    - **env**
- "path" environment variable
    - PATH is an environment variable on Unix-like operating systems, DOS, OS/2, and Microsoft Windows, specifying a set of directories where executable programs are located. In general, each executing process or user session has its own PATH setting.
    - often executable files will be in a "bin" folder which stands for *binary.*

video: 20 Introduction

## Setting environment variables - mac, linux
- .bash_profile & .bashrc
    - **.bash_profile** is executed for login shells
        - When you login (type username and password) via console, either sitting at the machine, or remotely via ssh, .bash_profile is executed to configure your shell before the initial command prompt.
    - **.bashrc** is executed for interactive non-login shells
- shell / bash commands
    - **export <environment variable name>="<path>"**
    - **$<environment variable name>**

examples of setting environment variables:
- export GOPATH="/Users/toddmcleod/go"
- export PATH="$PATH:/Users/toddmcleod/go/bin"

video: 21 Setting Env Var - Mac & Linux

## Setting environment variables - windows

Search for “environment variables”
video: 22 env windows

# Hands-on exercises - ninja level 1

## Hands-on exercise #1

For this hands-on exercise, at the terminal:

- navigate to “Documents”
- create a folder “HappyDog”
- navigate to that folder
- look to see if there are any files or folders in that folder
- navigate back to documents

video: 23 Hands On 1

## Hands-on exercise #2

For this hands-on exercise, at the terminal:

- navigate to the folder “HappyDog”
- create a file “be-happy.txt”
- edit this file to include items that make you happy
- show the contents of this file at the terminal
- navigate to “Documents”

video: 24 Hands On 2

## Hands-on exercise #3

For this hands-on exercise:

- Navigate to “HappyDog”
- change the permissions on the file “be-happy.txt”
    - user: read, write
    - group: read
    - world: nothing
- confirm that the permissions are set that way
- navigate to “Documents”

video: 25 Hands On 3

## Hands-on exercise #4

For this hands-on exercise:

- Navigate to “HappyDog”
- Delete “be-happy.txt”

video: 26 Hands On 4

## Hands-on exercise #5

For this hands-on exercise:

- Navigate to “Documents”
- Delete “HappyDog”
- confirm this directory has been deleted
- navigate to your user’s folder

video: 27 Hands On 5

# Getting started

## Why VS Code?

There are ***many reasons to use vs code:***

- Free
- Open source
- Extensible
- Cross platform
    - Files go back-and-forth between platforms seamlessly
- Powerful
- Popular
- Light weight
- Microsoft
- Supports many languages
- Great with Go
- More benefits:
    - https://code.visualstudio.com/
        - intelliSense code completion
        - debugging
        - built-in git
        - extensions
    - https://code.visualstudio.com/docs/?dv=osx
        - first steps
            - intro videos
            - settings
        - keyboard shortcuts
    - https://code.visualstudio.com/docs/editor/whyvscode
        - why did we build vs code

video: 28 Why VS Code

## Downloading and installing - mac

***Downloading and installing vs code*** is a straight-forward process:

- https://code.visualstudio.com/
- MAC
    - drag to applications directory
- Opening vs code
    - click it in applications
    - spotlight search
        - cmd + spacebar
            - type in “vs code”

video: 29 VS Setup - Mac

## Downloading and installing - windows

***Downloading and installing vs code*** is a straight-forward process:

- https://code.visualstudio.com/

video: 30 VS Setup - Windows

# Exploring vs code

## The welcome screen

The welcome screen has a lot of useful features. **Instead of dismissing the welcome screen, take a moment to look at it.** This video covers the features found on the welcome screen. To bring the welcome screen back up, use the menu option “HELP / WELCOME”
video: 31 welcome screen

## Interface overview

The interface overview introduces the different areas of vs code. A lot of great shortcuts are shown in the interface overview:

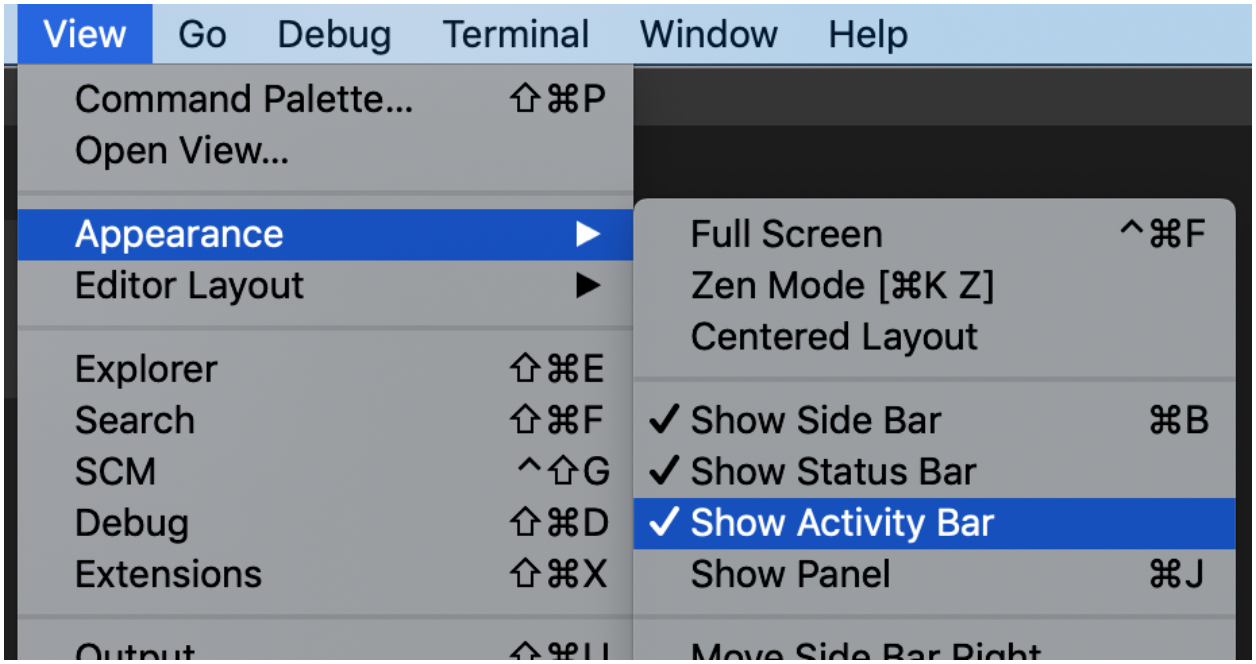
- file explorer
    - command + shift +E
- find / search across files
    - command + shift + F
- source code control
    - control + shift + G
- debug

    - command + shift + D
- extensions
    - command + shift + E
- command palette
    - command + shift + P
- errors and warnings
    - command + shift + M
- terminal
    - control + back-tick

video: 32 interface overview

## View menu

If at any time your vs code does not look like my vs code, go to the “view” dropdown menu and play with the options.




video: 33 view menu

## Code menu

- Code
    - about
    - check for updates
    - preferences
    - quit

video 34 code menu

# Files, folders, and workspaces

## File menu

- new files
- auto save
    - saves changes as you work

video: 35 file menu

## Getting started with git

- creating a github account
- creating a github repo
- the “git” story and Linus, The Patron Saint of Programming
- Nelson Mandela and Invictus

video 36 Getting Started With Git

## Using SSH for authentication

- setting up SSH key using shell / bash / terminal
    - works on Mac, Linux, and Windows
    - ssh-keygen -t rsa
- Putting the SSH pub key on github
- Asymmetric public - private key encryption
- Symmetric key encryption

video 37 Using SSH

## Cloning a repo using ssh

- Connecting a folder on our computer with github

video 38 Cloning Repo

## Folders and workspaces

- We’re actually going to start writing code.
- changing theme
- folder
    - open a folder
- workspaces
    - save as a workspace

    - open a workspace

video: 39  Folders and Workspaces

## Installing go

- https://golang.org
- adding the “go” extension to vs code
    - windows potential snafu: if you didn’t allow git-scm to use third-party software, then vs code will not be able to install the “go” extension. To solve this, reinstall git-scm and allow third-party software to use git

video 40 Installing Go

## Git in vs code

- stage changes
- commit changes
- push changes

video 41 Git In VSCode

# Hands-on exercises - ninja level 2

## Hands-on exercise #1

For this hands-on exercise:

- change vs code’s theme to “high contrast” (or something else)
- enable auto save every 1000 ms
- check for updates for vs code

video: 42 Hands On 1

## Hands-on exercise #2

This hands on exercise has been deleted.

video: 43 Hands On 2

## Hands-on exercise #3

For this hands-on exercise:

- generate an SSH key at the terminal / bash / shell
- add the public key (.pub) to github

video: 44 Hands On 3

## Hands-on exercise #4

For this hands-on exercise:

- clone a github repo using ssh

video: 45 Hands On 4

## Hands-on exercise #5

For this hands-on exercise:

- in vs code, open the folder of the repo you cloned in “hands-on exercise #4”
- create a file
- save as a workspace
- close the workspace
- open the workspace

video: 46 Hands On 5

## Hands-on exercise #6

For this hands-on exercise:

- in vs code, open the workspace you created in “hands-on exercise #5”
- in vs code, stage changes, commit changes, and push changes
- verify the file you created is on github

video: 47 Hands On 6

# Editing code

## Preview

Video:48 Preview

## Find and replace

You can use find and replace in a single file or in multiple files
video: 49 Find and Replace

## Toggle comments

You can toggle comments in vs code with these shortcuts:

- command + /
- alt + shift + A

video: 50 Toggle Comments

## Emmet

Emmet allows you to use short-hand notation for HTML code.
video: 51 Emmet

## Duplicating and moving lines

Use these shortcuts to duplicate and move lines:

- duplicate
    - alt + shift + up arrow
    - alt + shift + down arrow
- move
    - alt + up arrow
    - alt + down arrow
- delete line
    - cmd + delete
    - cmd + shift + k

video: 52 Dup & Move Lines

## Keyboard shortcuts

You can find a keyboard shortcut map by either ....

- code / preferences / keyboard shortcuts
    - in here you can search for things
    - you can also add key bindings

… or by …

- going to **“help / keyboard shortcuts”**

… or by …

- searching google for “vs code keyboard shortcut reference” or by going here:
- https://code.visualstudio.com/docs/getstarted/keybindings

The command palette also shows you keyboard shortcuts for some things, so you can search that for what you want to do, then look at the shortcut there. To get to the command palette:

- cmd + shift + P

video: 53 Key Shortcuts

## Multiple cursors

You can select multiple items at once. Also, notice that vs code highlights curlies.

- ***select all of the same thing***
    - ***cmd + shift + L***
- ***put your cursor in different places manually***
    - ***alt + click***

    - ***alt + click***
    - ***alt + click***
    - ***alt + click***
- ***select next occurrence of word under cursor or of the current selection***
    - ***cmd + d***
- ***undo last cursor action***
    - ***cmd + u***

***hit escape to exit***
video file: 54 Multi Cursor

## Easily seeing documentation

You can use “command + click” to see the documentation for different items in your code. This feature is a little buggy in the go programming language.
video: 55

# Ninja level skills

## The command palette

The command palette is ***where you enter commands in vs code.*** You will want to command vs code to do things for you.

- ⌘P will let you navigate to any file or symbol by typing its name
    - Type ? into the input field to get a list of available commands you can execute
- ⇧⌘O will let you navigate to a specific symbol in a file
- ^G will let you navigate to a specific line in a file
- ^⇧Tab will cycle you through the last set of files opened
- ⇧⌘P will bring you directly to the editor commands

Video file 56 Command Pallete

## Setting a path variable

***Path variables are environment variables.*** You can set up variables that your computer can then use. This will make your life easier. Here is the first one we will set up:

- ***In the command palette:***
    - ***Shell command: install ‘code’ command in PATH***

Test it out

- In terminal:
    - navigate to dir where you want to work

- code .

Investigating

```
highlanders-MacBook-Pro:bin highlander$ echo $PATH
/Users/highlander/Documents/libwebp-1.0.2-rc1-mac-10.14/bin:/Users/highlander/go/bin:/Users/highlander/google-cloud-sdk/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/go/bin
highlanders-MacBook-Pro:bin highlander$ pwd
/usr/local/bin
highlanders-MacBook-Pro:bin highlander$ ls -la
total 78024
drwxrwxrwx   9 root        wheel       288 Aug  9 13:53 .
drwxr-xr-x  16 root        wheel       512 Mar  4 11:05 ..
lrwxr-xr-x   1 highlander  wheel        28 Mar  4 11:05 brew -> /usr/local/Homebrew/bin/brew
lrwxr-xr-x   1 highlander  wheel        68 Aug  9 13:53 code -> /Applications/Visual Studio Code.app/Contents/Resources/app/bin/code
lrwxr-xr-x   1 highlander  wheel        35 Mar  4 11:06 git-lfs -> ../Cellar/git-lfs/2.7.1/bin/git-lfs
lrwxr-xr-x   1 highlander  wheel        36 Mar  4 11:12 gulp -> ../lib/node_modules/gulp/bin/gulp.js
-rwxr-xr-x   1 root        wheel  39945984 Feb 28 03:53 node
lrwxr-xr-x   1 root        wheel        38 Mar  4 10:54 npm -> ../lib/node_modules/npm/bin/npm-cli.js
lrwxr-xr-x   1 root        wheel        38 Mar  4 10:54 npx -> ../lib/node_modules/npm/bin/npx-cli.js
```

Video file: 57 Setting a Path Variable

## Mini-map and breadcrumbs

A small version of your open file; allows you to see where you're at in the file.
Video file: 58 Minimap & Breadcrumbs

# Customizing vs code

## Adjusting preferences / settings

- ***Code / preferences  / settings***
- user settings vs workspace settings
    - workspace - vs code out of the box
    - user - your settings that override workspace settings
- searching
    - you can search for the item you want
- some suggestions
    - enable telemetry **UNCHECK**
    - **autosave: afterDelay**
        - **auto save delay: 1000 ms**
    - **format on save**

```
{
 "telemetry.enableTelemetry": false,
 "editor.fontLigatures": true,
 "editor.fontFamily": "'Fira Code', Consolas, 'Courier New', monospace",
 "terminal.integrated.shell.windows": "C:\\WINDOWS\\System32\\cmd.exe",
 "files.insertFinalNewline": true,
 "editor.fontSize": 12,
```

```
 "window.restoreFullscreen": true,
 "files.trimTrailingWhitespace": true,
 "editor.formatOnPaste": true,
 "editor.multiCursorModifier": "alt",
 "editor.snippetSuggestions": "top",
 "editor.formatOnSave": true,
 "files.eol": "\n",
 "explorer.autoReveal": false,
 "editor.minimap.enabled": false
}
```

video: 59 Adjust Prefs & Settings

## Snippets

Learn about snippets

- https://code.visualstudio.com/docs/editor/userdefinedsnippets

Find snippets in marketplace to install:

- https://code.visualstudio.com/docs/editor/extension-gallery#_browse-and-install-extensions
    - @category:snippets

To create a snippet:

- command palette
    - snippets
    - follow the suggestions

To use a snippet:

- type in the keyword for the snippet, make sure the snippet is highlighted, then hit tab

Example Go snippets

```
        "func": {
            "prefix": "func",
            "body": [
                "func ${1:identifier}(${2:params}) {",
                "${0:code}",
                "}"
            ],
            "description": "general function block"
        },
```

```
    "func return": {
        "prefix": "func return",
        "body": [
            "func ${1:identifier}(${2:params}) ${3:return} {",
            "${0:code}",
            "}"
        ],
        "description": "general function block with return"
    },

    "func returns": {
        "prefix": "func returns",
        "body": [
            "func ${1:identifier}(${2:params}) (${3:returns}) {",
            "${0:code}",
            "}"
        ],
        "description": "general function block with returns"
    },
```

video file: 60 Snippets

# Coding essentials

## Errors

Errors are underlined. Code suggestions come up.
video file: 61 Errors

## Debugging code

Click the bug icon in the tabs area. Here are the basics of debugging:

- set a breakpoint(s)
- start
    - continue
    - step over
    - step into
    - step out
    - restart

- stop
- watch
    - variable tracking by name
- call stack
- variables

**Troubleshooting:** if you are on mac and you get this error “developer tools access needs to take control of another process for debugging to continue” then enter this code at the terminal:

- sudo /usr/sbin/DevToolsSecurity --enable
- source: stackoverflow

video file: 62 Debugging

## Markdown

Markdown is a lightweight markup language with plain text formatting syntax. Its design allows it to be converted to many output formats, but the original tool by the same name only supports HTML. Markdown is often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor. (source: wikipedia)

- https://help.github.com/en/articles/basic-writing-and-formatting-syntax
- https://guides.github.com/features/mastering-markdown/

video: 63 Markdown

## Extensions

Here are some extensions:

- vs liveshare
- auto close tag and auto rename tag
- prettier for JS
- path intellisense

Video file: 64 Extensions

# Hands-on exercises - ninja level 3

## Hands-on exercise #1

Clone the codebase “cit90fa2019” from github. Find the file with the comment “I found this file”
Video file: 65 Hands On 1

## Hands-on exercise #2

Open the file “main.go” in the folder “017-lotta-vars” and perform the following operations:

- using multiple cursors, change the variable “z” to “y”
- using “move line up” or “move line down” move the call to bar() above the call to foo().
- comment out the last line in main “Program about to end!” using the keyboard shortcut

Video file: 66 Hands On 2

## Hands-on exercise #3

Open the file “index.html” in the folder “016-html” and add the following html using emmet:

**<h1 title="item1">Header 1</h1>**
**<h2 title="item2">Header 2</h2>**
**<h3 title="item3">Header 3</h3>**

Video file: 67 Hands On 3

## Hands-on exercise #4

Using the command palette

- Open the file “zep.go”
- show all of the symbols / identifiers for funcs and variables
- move the call to bar() above the call to foo().

As a reminder, here are some of the command palette’s commands:

- ⌘P will let you navigate to any file or symbol by typing its name
    - Type ? into the input field to get a list of available commands you can execute
- ⇧⌘O will let you navigate to a specific symbol in a file
- ⇧⌘P will bring you directly to the editor commands

Video: 68 Hands On 4

## Hands-on exercise #5

Using the command palette

- Open the file “bugs.go”

Debug the code using debugging
video: 69 Hands On 5

# Farewell

## Congratulations!

- Great job completing the course. I am so proud of you.

Video file: 70 Congratulations

## Bonus lecture