



Workshop 101 - Git, Github & Version Control





What is VS Code:

- Visual Studio Code is a code editor created by Microsoft.
 - It has a wide range of support for multiple programming languages such as: Java, Python, SQL, C, C++, JavaScript, and many more
 - Built in Code Completion and Debugging
 - Extensions to enhance your Programming Experience
 - Built in Version Control such as Git. (We'll explain this shortly)



Installation for VS CODE

[Visual Studio Code - Code Editing. Redefined](#)



Installation for Python

What is python:

Python is a high level programming language
language

Used in:

Web Development(Backend)

ML/AI applications

Automation

Data science

Scientific Applications

- Check if you have python installed
 - Go to terminal and do `py --version`
- Download python from python.org
- Download VSCode
 - Add the python extension to view outputs in VSCode



Installation for Java

What is Java:

A high level Object Oriented Programming Language

Used in:

Web Applications(Backend)

Mobile Applications

Big Data Applications

Business Applications

How to Install On your Computer and Configure for VSCode:

1. Go to the following website

[Java Downloads | Oracle](#)

2. After download, confirm the installation by opening command prompt/terminal and typing the following command: `java -version`
3. Open vscode, go to extensions, and download the following: [Extension Pack for Java - Visual Studio Marketplace](#)



Installation for C++

What is C++ ?

A high-level, object-oriented programming language with support for both procedural and object-oriented programming paradigms.

Used in:

- System Software
- Game Development
- Embedded Systems
- High-Performance Applications

Article to help install C++ and setup vsc with C++ in Windows

<https://medium.com/@ahmetekiz/how-to-compile-c-in-vs-code-on-windows-11-11518b471c68>

Windows

Download the c++ from msys2

Open msys2 mingw64 from terminal

Install g++ // command

```
pacman -S --needed base-devel  
mingw-w64-x86_64-toolchain
```

Copy the path, add the path to
environment variables

Check if its properly installed

```
gcc --version  
g++ --version  
gdb --version
```

Mac

Check if c++ is pre installed through terminal

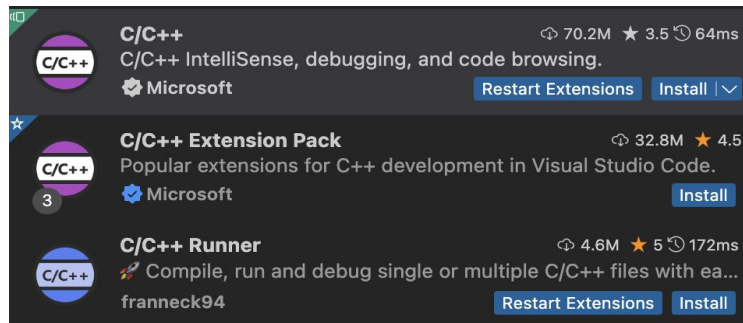
Clang -- version

If not : xcode-select --install

Go to vsc extension

Create c++ project and try to run using f5

Once .vscode is created on default wait for next
steps



Windows

Configure task in VSC in the terminal option in VSC

Select the g++

Debug / Run code

MAC

Check if it is similar to this setup for the json package -> next page shows the setup

Second option

Download the c/ c++ runner

And configuration option chose c/c++ runner

Run code after you give permission

EXPLORER

main.cpp

c_cpp_properties.json

Settings

OPEN EDITORS

main.cpp

c_cpp_properties.json

Settings

C++ EXAMPLE

.vscode

c_cpp_properties.json

launch.json

settings.json

main

main.cpp

c++ example

c_cpp_properties.json

Settings

.vscode > {} c_cpp_properties.json > ...

```
1  {
2    "configurations": [
3      {
4        "name": "macos-clang-arm64",
5        "includePath": [
6          "${workspaceFolder}/**"
7        ],
8        "compilerPath": "/usr/bin/clang",
9        "cStandard": "${default}",
10       "cppStandard": "${default}",
11       "intelliSenseMode": "macos-clang-arm64",
12       "compilerArgs": [
13         ""
14       ]
15     }
16   ],
17   "version": 4
18 }
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Code

[Done] exited with code=0 in 0.763 seconds

[Running] cd "/Users/natashapiedrabuena/Desktop/c++ example/" && g++ main.cpp -o main && "/Users/natashapiedrabuena/Desktop/c++ example/"main
Hello World
thank you for testing

[Done] exited with code=0 in 0.672 seconds

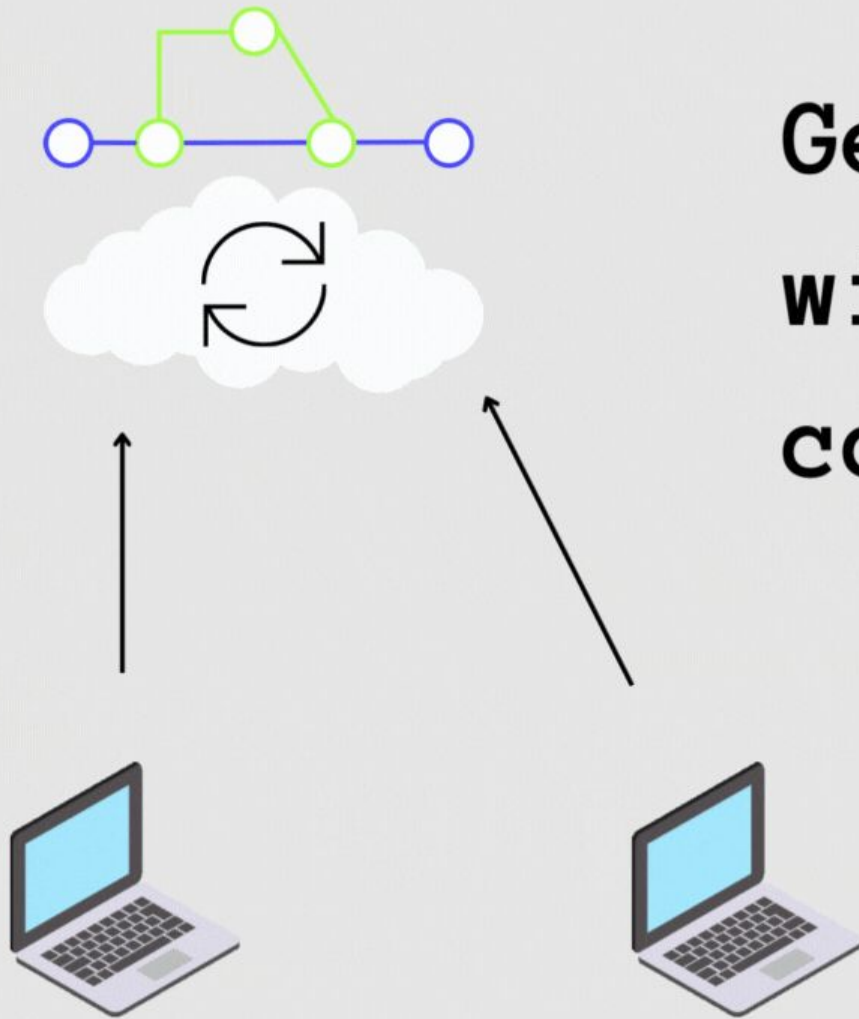
[Running] cd "/Users/natashapiedrabuena/Desktop/c++ example/" && g++ main.cpp -o main && "/Users/natashapiedrabuena/Desktop/c++ example/"main
Hello World
thank you for testing

[Done] exited with code=0 in 0.681 seconds

> OUTLINE

> TIMELINE

main* 0 0 c++ example Debug C/C++ Runner: Debug Session (c++ example) Connect Colorize: 0 variables Colorize



**Get started
with version
control**

What is Version Control



- Version control is a way for developers to save changes to their work/task/project overtime without overwriting previous versions of said work/task/project.
- This practice allows you to save changes to create new saves on a project, and allows you to refer back to previous version of said project if need.
- Why is version control important? Without it, it will be extremely difficult) to track changes in a task. Additionally, you also run the risk of losing your work.





Benefits of Version Control

Easier and smoother collaboration amongst developers which leads to much better project/code quality.

- The ability to review, comment, and improve code(whether it's the code of another developer or even your own)
- You have the ability to diagnose problems, prevent future problems, and make changes at a much faster rate(This comes heavily into play when you are working on project/task with a deadline).

Now..... if we know version control...

What is Git?

- Git is one of the most popular version control system used by developers on a global scale.
- What makes Git unique is that it enables developers to have multiple copies of a project codebase, which happen to be independent of each other
- Additionally, it has the ability to store the history of a project, locally. Which allows users to work on said project on their own machine(personal computer). This prevents any crashes/network issues, git allows for backups to still exist, and saves/changes to even occur in a offline environment.
- Overall, git is a fast, reliable, and collaborative version control system



Git flow

Modified/working dictionary

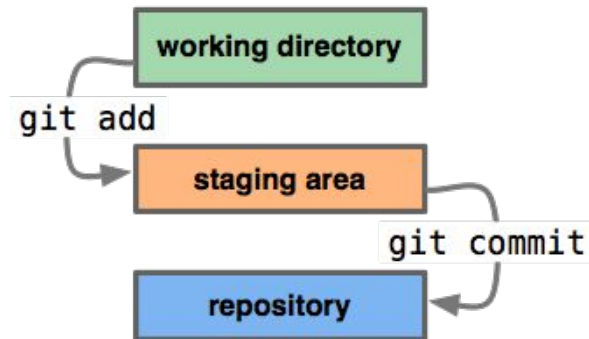
File has been changed. But has not been saved.

Staged

Modified file is ready to go for the next commit(save). File is on standby

Committed

File has been saved onto a database.



Key words to know

 **Repository-** Another word for database.

Branch- Separate/new versions of the main repository. Allows you to work on different parts of the project, without affecting the main repository/project. When work is done, can be merged onto main project.

Analogy- The project/main repository is the tree. The branches are still part of the tree.

Local Repositories- Databases hosted on a individual/personal computer

Remote Repositories- Databases hosted on a server that is accessible to everyone.(internet or some type of network)

Common Git Terms/Commands to Know

- **Git Init** - used to start a project

```
Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git init
Initialized empty Git repository in C:/Users/Akash Jha/Desktop/Git/.git/

Akash Jha@LAPTOP-LJJ1U61G MINGW64 ~/Desktop/Git (master)
$ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
```

Git Clone- Copy of a code source from a remote repository.(such as github)

```
# Syntax
$ git clone <repository>

# Example
$ git clone https://github.com/username/myproject.git
```




Git Add

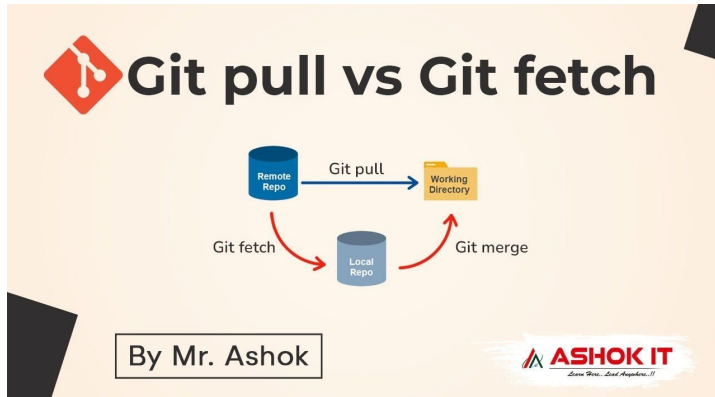
- Adds new or changed files with local repository
- These new files/file changes would basically be set in the stage area, which prepare said file to be used in the next commit.

Git Commit

- **One of the most important git commands.**
- Allows you to save changes. (acts as a checkpoint for a project/task)
- Also, when making a commit, you **MUST** write a message stating what you did.
- Also, commits list the times and date said change was made.

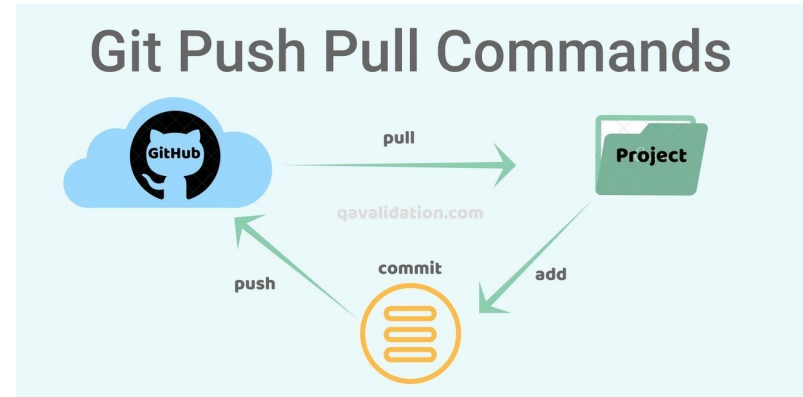
Git Pull

- Used to acquire/update information from the remote repository to the local repository.
- (acts as a updater)



Git Push

- Used to send modified information from local repository to remote repository
- (acts as an uploader)



Git Remote:

- Establishes a connection between the local repository to the remote repository server.

Git Status

- States all the necessary information about a current

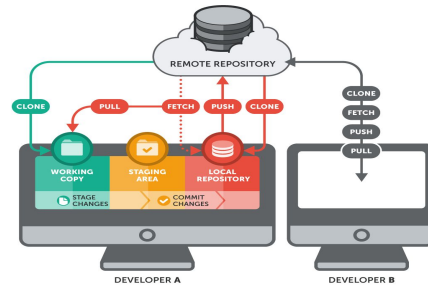
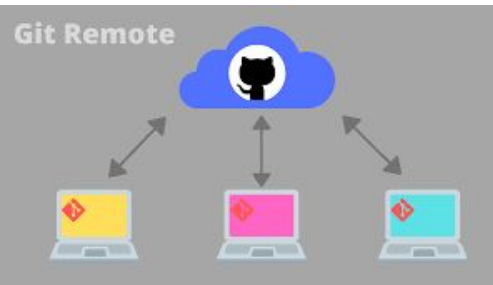
```
@Harish MINGW64 /e/ToolsQA/First Project (master)
$ touch ABC.txt

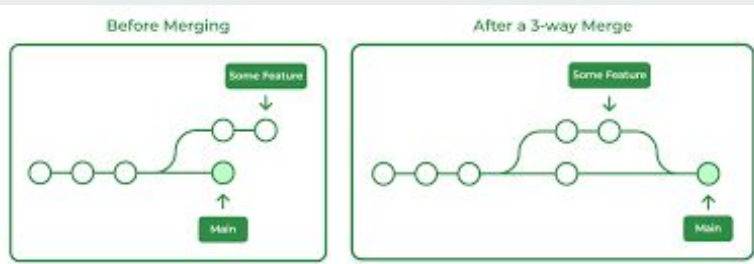
@Harish MINGW64 /e/ToolsQA/First Project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        ABC.txt

nothing added to commit but untracked files present (use "git add" to track)

@Harish MINGW64 /e/ToolsQA/First Project (master)
$ |
```

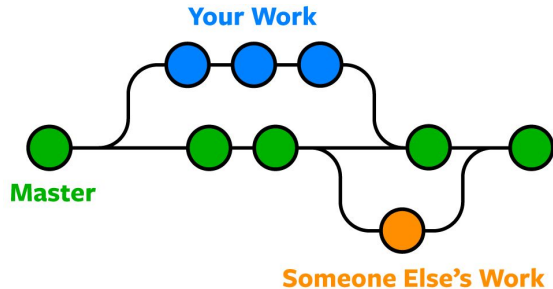




Git Merge:

Integrating a feature branch into a main/master branch

Analogy: merging onto another lane on a highway.



Git log:

Showing history of commits

Git Pull request:

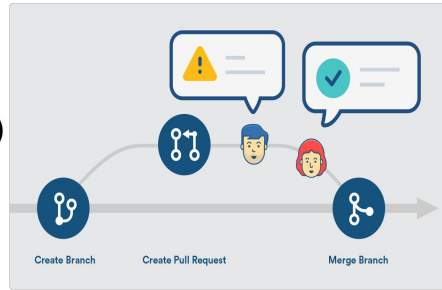
Used to make proposals to merge/add changes from feature branch to main branch.

Mainly used for code review

Examples:

-Feature Additions: New features is made. Pull request is created to merge feature onto main branch after team approval.

Bug fixes: Bug is founded, pull request is created for team to review said bug.





Merge Conflicts

What causes merge conflicts?

- Merge conflicts occur in Git when multiple contributors make changes to the same part of a file

How to Resolve Merge Conflicts?

- When a conflict occurs, Git stops the merge process and gives you the opportunity to resolve the conflict manually.

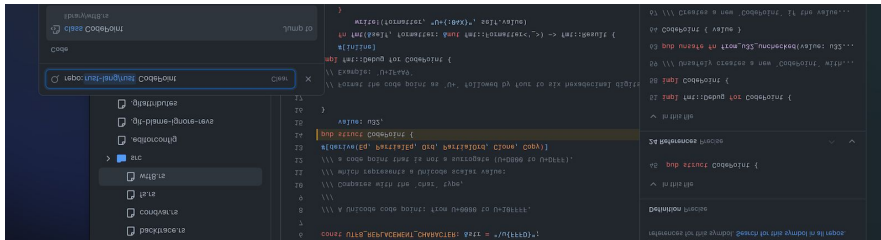


Steps: To resolve

1. **Identify the Conflict**
 - **"This branch has conflicts that must be resolved."**
2. Pushing Changes and Creating a Pull Request
 - `git fetch origin feature-branch`
 - `git checkout feature-branch`
3. Merge the Target Branch into Your PR Branch
4. **Stage the Resolved Files**
5. Commit the Changes
6. Push the Changes

What is Github?

- Github is online platform(remote repository), which allows for collaboration on software projects.
- Github analogy- If git is the videos, than github is the video platform(YT)
- Allows Developers to host, review, manage, and collaborate on software.
- Github is also a remote repository that is accessible to EVERYONE(not only just teammates, but other people on **ALL of** github)



Why is github important?



- Github allows for enhanced collaboration.
- Not only can you view your own work/projects on github, but you can also view the work/projects of other developers.
- Additionally, you can also contribute open source projects
 - Also applicable to team development
 - Used within public repositories.
- Within interviews, you can showcase your projects.

Two git terminal commands to know

Git -help

-List of all the git commands(use if you forget a certain command)

```
These are common Git commands used in various situations:

start a working area (see also: git help tutorial)
  clone      Clone a repository into a new directory
  init       Create an empty Git repository or reinitialize an existing one

work on the current change (see also: git help everyday)
  add        Add file contents to the index
  mv         Move or rename a file, a directory, or a symlink
  restore    Restore working tree files
  rm         Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
  bisect     Use binary search to find the commit that introduced a bug
  diff       Show changes between commits, commit and working tree, etc
  grep       Print lines matching a pattern
  log        Show commit logs
  show       Show various types of objects
  status     Show the working tree status

grow, mark and tweak your common history
  branch     List, create, or delete branches
  commit     Record changes to the repository
  merge      Join two or more development histories together
  rebase     Reapply commits on top of another base tip
  reset      Reset current HEAD to the specified state
  switch     Switch branches
  tag        Create, list, delete or verify a tag object signed with GPG
```

Git -version

To see if Git is already installed on your computer(more applicable to those who have macs).

Use this to check if git is installed on your device, after you download it.

```
PS C:\Users\phili> git --version
git version 2.41.0.windows.3
```

Installing Git on Windows OS,

For Windows OS Users:

Go to search- Type Command Prompt or Terminal

Once there, type the following command:

```
git -version
```

Go to this link:

<https://gitforwindows.org>

Follow the installers instructions

Once installation is complete, go back to command prompt/terminal and enter the command: `git -version` (verify if the installation actually occurred)



Signing up for github

<https://github.com>

Installing Git for MacOS: Homebrew:

Go to search- Type Terminal

Once there, type the following command:

`git -version`

Go to this link:

<https://brew.sh>

The process will take some time, Script explains what it will do. Than does it.

Than one homebrew is installed, you can type either one of the following:

`brew --version` or `which brew`

Once it is verified that it is installed, type the following command on ~~terminal~~ `brew install git`

Once installation is complete, go back to terminal and enter the command: `git --version` (verify if the installation actually occurred)

Xcode

Open terminal

Enter in the following command:

```
xcode-select --install
```

Follow the prompts

To check if it is installed:

```
xcode-select -p  
git --version
```



Linux Installation

Run with package management tool that comes with distribution

If you're on fedora: enter the following command:

```
sudo dnf install git-all
```

Ubuntu/Debian:

```
sudo apt install git-all
```





EXERCISES/MORE DEMONSTRATIONS!!!





CALCULATOR!!!!

Math template

add function(1st)

subtraction function(2nd)

multiplication function(3rd)

division function(4th)

actual calculator function

make it with a while/do while loop, until user says
exit^