# EXPLORE | DIGITAL SKILLS

Introduction to Git & Github

Learning objectives

In this train, we will learn how to:

- Define the characteristics of a Distributed Version Control System (DVCS).
- Describe version control and its applications.
- Install Git and set up a GitHub account.
- Understand GitHub and its functionality.
- Execute Git commands using a terminal/Git Bash.
- Connect Github to Visual Studio Code.

EXPLORE|DIGITAL SKILLS
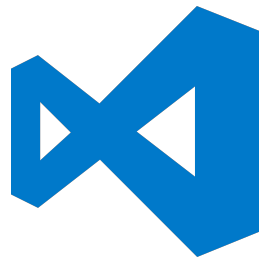
Outline

In this train, we will cover the following:

- The concepts and installation of Git on Windows.
- Bash commands for Git workflow
- Installation of Visual Studio Code.
- Connecting GitHub to Visual Studio Code.
- Running bash commands from Visual Studio Code.
- GitHub nomenclature and definitions.

EXPLORE | DIGITAL SKILLS
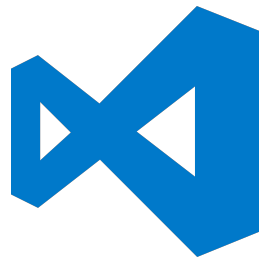
Introduction: What is version control and why is it used?

Version control can be defined as the process of integrating, recording and monitoring changes to code and other files from multiple collaborators. Essentially, version control **tracks the state of files as they change over time.**

Version control is used for the following reasons:

- **Traceability** - allows traceability of code, especially if there are multiple collaborators.
- **Identifiability** - links code to decisions, contributions, contributors, and time.
- **Clarity** - latest version of code is easily distinguishable.
- **Reduces errors** - mistakes and errors in code can be easily identified and fixed; minimizing disruption to all collaborators.
- **High availability/ Disaster recovery** - If something happens, you can immediately switch over to the earlier versions of code to help fix the mistake, minimise disruptions and allow uninterrupted availability.

Secure
Traceable
Reliable
The Version Control System
Distinguishable
Efficient

EXPLORE ║ DIGITAL SKILLS
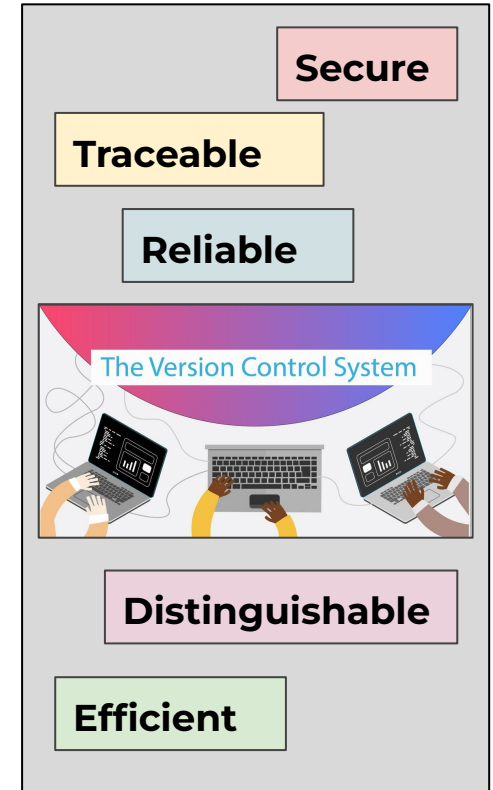
Introduction: What is Git?

Git is a popular **distributed version control system (DVCS)**.

Put simply, it provides a rich set of features which allow many individuals to collaborate *remotely* on a software-based project, with the full power of a version control system.

In a Git-based project, a collaborator will typically work on their own local machine using a full copy of the project known as a **local repository.** As changes are made to the collaborator's version of the project, Git commands are used to store and send these changes to a single **remote repository** which every other collaborator in the project also has access. Git uses a sophisticated process to then **merge** these changes with those of other collaborators, enabling a single reputable and updated version of the project to exist.

Distributed Version Control
Main Server Repository
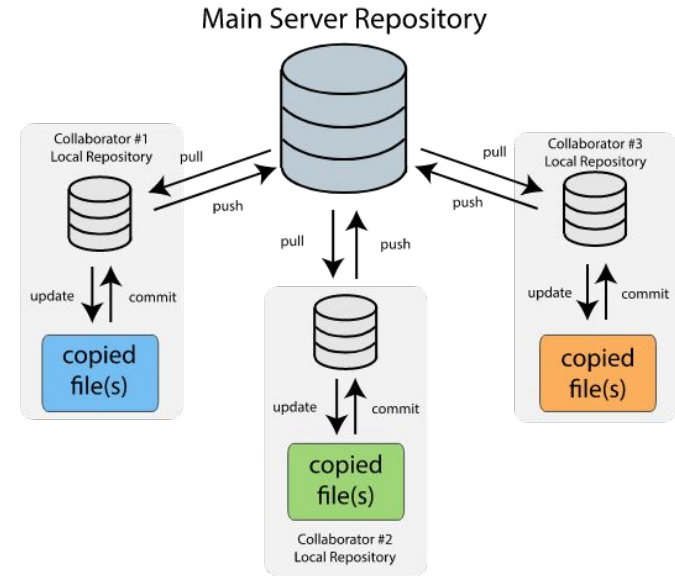
Introduction: What is GitHub?

GitHub is an **online platform providing code hosting management** which is facilitated through Git.
- It provides both public and private remote repository in which code can be stored.
- GitHub provides a visual interface to the often complex process of working with Git.

Here is summary of how different users interact with GitHub through Git:

Git Local Repository   **Git remote Repository**

User 1

User applies Git commands

Git changes added to GitHub

User 2

User applies Git commands

Git changes added to GitHub

User 3

User applies Git commands

Git changes added to GitHub

User 1
User 2
User 3

Getting started with GitHub: Common Terminology

Here are some terms you will come across in GitHub. Try familiarise yourself with them! Note: These are definitions taken directly from the GitHub website .	
Branch	A parallel version of a repository. It is contained within the repository, but does not affect the primary or master branch allowing you to work freely without disrupting the "live" version. When you've made the changes you want to make, you can merge your branch back into the master branch to publish your changes.
Pull Request	Pull requests are proposed changes to a repository submitted by a user and accepted or rejected by a repository's collaborators.
Private repository	Repositories that can only be viewed or contributed to by their creator and collaborators the creator specified.
Remote	This is the version of something that is hosted on a server, most likely GitHub. It can be connected to local clones so that changes can be synced.
Repository	A repository is the most basic element of GitHub. They're easiest to imagine as a project's folder. A repository contains all of the project files (including documentation), and stores each file's revision history. Repositories can have multiple collaborators and can be either public or private.

EXPLORE || DIGITAL SKILLS

Getting started with GitHub: Commands and Functionality

The following are commands used to interact with GitHub. Note: These are definitions taken directly from the GitHub website .	
git pull	Pull refers to when you are fetching in changes and merging them. For instance, if someone has edited the remote file you're both working on, you'll want to pull in those changes to your local copy so that it's up to date.
git push	Pushing refers to sending your committed changes to a remote repository, such as a repository hosted on GitHub. For instance, if you change something locally, you'd want to then push those changes so that others may access them.
git clone	A copy of a repository that lives on your computer instead of on a website's server somewhere, or the act of making that copy. With your clone you can edit the files in your preferred editor and use Git to keep track of your changes without having to be online. It is, however, connected to the remote version so that changes can be synced between the two. You can push your local changes to the remote to keep them synced when you're online.
git commit	A commit, or "revision", is an individual change to a file (or set of files). It's like when you save a file, except with Git, every time you save it creates a unique ID (a.k.a. the "SHA" or "hash") that allows you to keep record of what changes were made when and by who. Commits usually contain a commit message which is a brief description of what changes were made.
git fetch	Fetching refers to getting the latest changes from an online repository without merging them in. Once these changes are fetched you can compare them to your local branches (the code residing on your local machine).
git fork	A personal copy of another user's repository that lives on your account. Forks allow you to freely make changes to a project without affecting the original. Forks remain attached to the original, allowing you to submit a pull request to the original author to update with your changes. You can also keep your fork up to date by pulling in updates from the original.

EXPLORE ‖ DIGITAL SKILLS

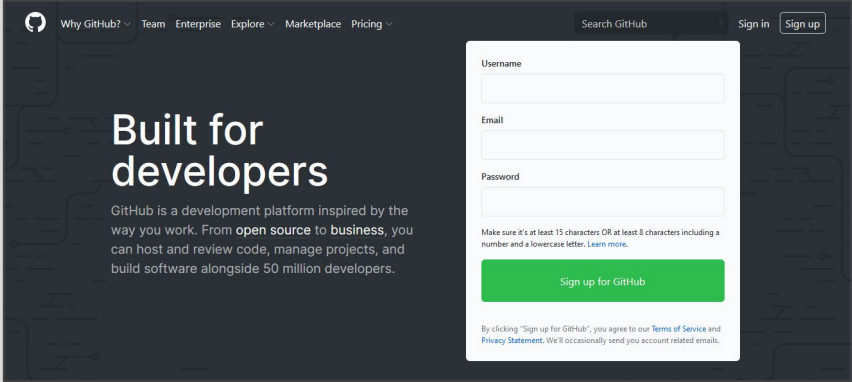# Getting started with GitHub: Creating GitHub Account

First things first, let's go ahead and set up a GitHub account.

Step 1:

Click on this link to setup a GitHub account.

Step 2:

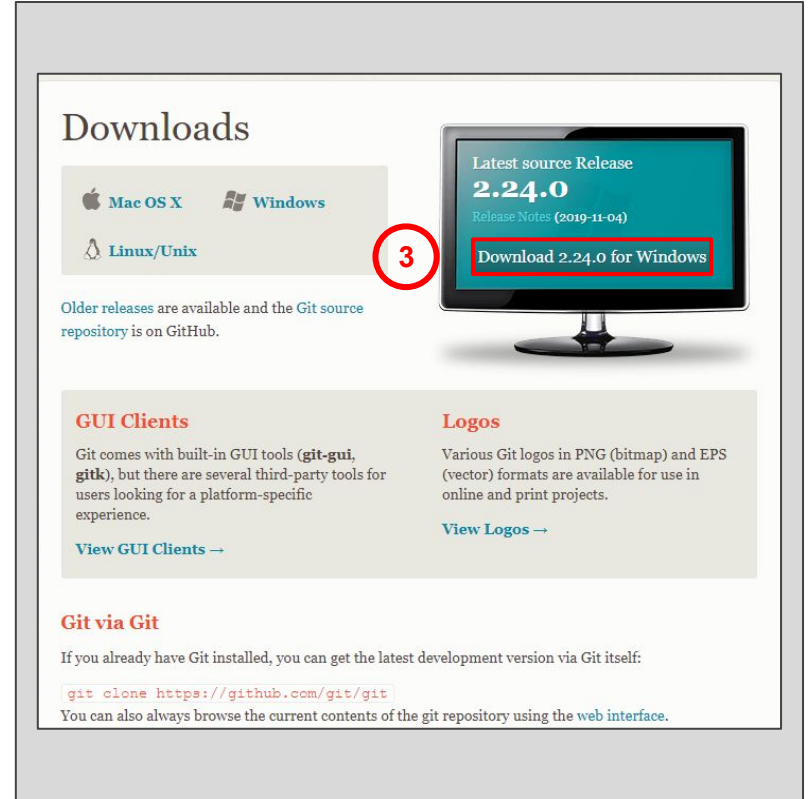Enter in your details and proceed with the account verification process (choose free plan).

Setting up Git on Windows: Installing Git

For macOS <u>head here</u>

Step 3:

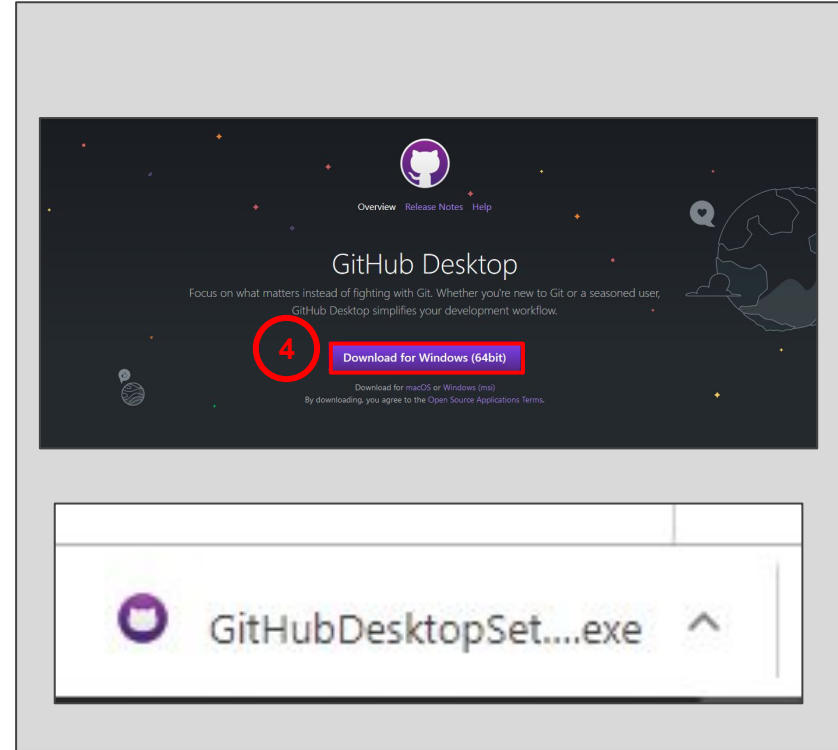Go to this <u>link</u> and download the version for Windows.

Setting up Git on Windows: Downloading GitHub for Desktop

Step 4:

Go to following link and download GitHub for Desktop.

Step 5:

Open and run the .exe file to install the desktop application.

Setting up Git on Windows: Downloading GitHub for Desktop

Step 6:

Sign into Github with same email address you used to create your account.

Step 7:

You can go ahead and create a tutorial repository to learn more about GitHub.

EXPLORE | DIGITAL SKILLS
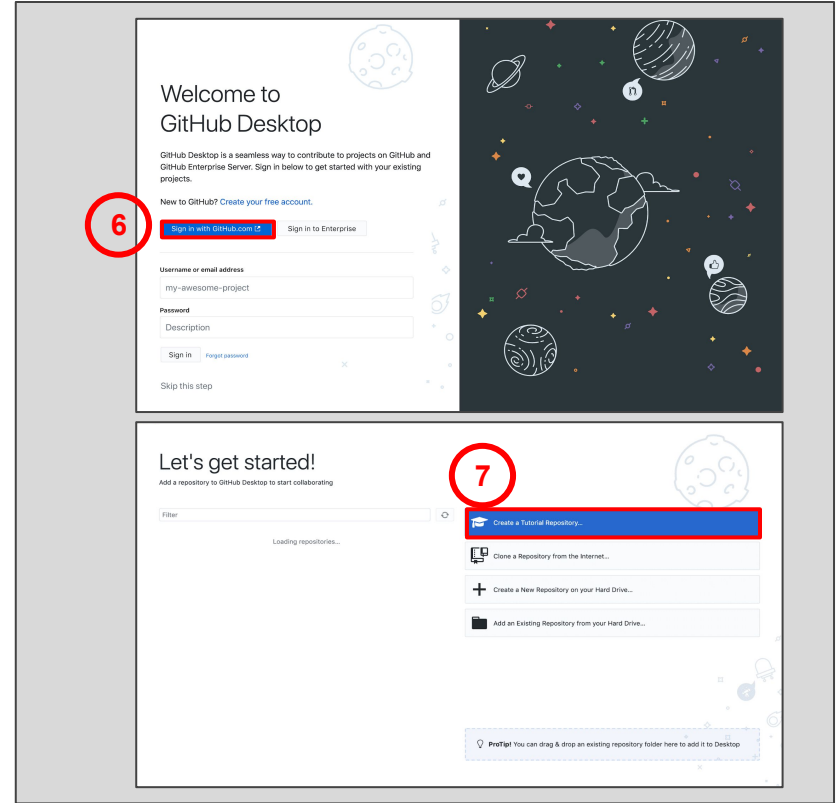
Setting up Git and GitHub: macOS

For Windows [head here](#)

Step 1:

To download Git you can use the following [link](#) , there are several options for installing Git on macOS. You can choose the option of your choice

Step 2:

Next you can use this [link](#) and this [link](#) to set up Git and GitHub on your macOS.

Accessing Git Bash

In many cases we issue Git commands through a terminal or a Git Bash interface. The following steps detail how to open up a Git Bash in Windows:

Step 1:

If you right click in any file or on your desktop, you should now be able to see two new options:
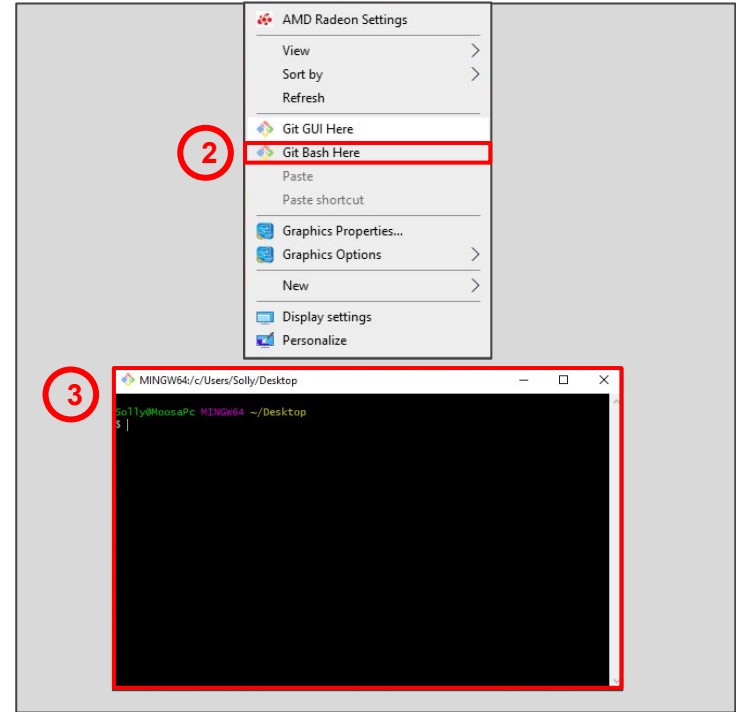
      i.   "Git GUI Here",
    ii.   "Git Bash Here".

Step 2:

Click on "Git Bash Here".

Step 3:

You should get a command line style interface. This is the *Git Bash* interface.

Bash commands for Git workflow: Exercise

To test what we've learnt so far, let's complete the following exercise.

Note: You can use the Git bash interface you were introduced to in the previous slide to help you by running each of the requested Bash commands and observing the output. Good luck!

Git Exercise

Type git help and press enter. Then try answer the following questions:

Define the following:
rm
mv
pull
push
commit
ls-files

What is the difference between clone and init?
Type in git help -a. How would you exit that command? Hint: it's Quite simple.
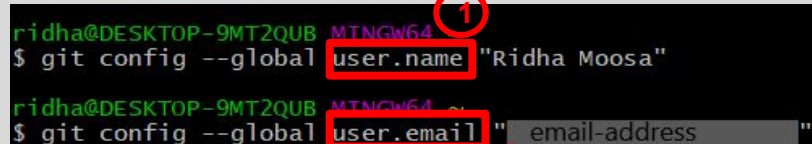
EXPLORE | DIGITAL SKILLS

Bash commands for Git workflow: Setting up Git Bash

One of the first things you should do is configure your git authorship.
This can be done with the git config --global command.

Look at the example below:

1. Type in "user.name" + your preferred user name in quotation marks
2. Type in "user.email" + your email address in quotation marks

EXPLORE || DIGITAL SKILLS

Bash commands for Git workflow: Common commands

Here is a list of some useful bash commands that you can use on your local machine

NOTE: some of the commands may vary based on your operating system.

cd	Change directory/file path (e.g., "cd .." will take you to your home path).	**cp**	Copy a filename to a new filename (e.g. "cp old_file.text new_file.txt" will rename the "old_file" text document to "new_file" text document; destination directory can also be specified when copying at the end of the script.
pwd	Displays name of working directory.	**mv**	Rename a specified file (e.g. "mv file1 file2" will change name "file1" to file2.
ls	Lists all the items in that directory (DIR is the CMD equivalent).	**rm**	Removes a specified file. Use with caution as it is permanently deleted!
mkdir	Makes a new directory (e.g. "mkdir ya" makes a directory called "ya".	**rmdir**	Removes a specified directory.
sudo	Activates user admin rights ; you will need to input your username and password.	**find**	Searches for a file/directory within a directory ; can use a wild card.

EXPLORE || DIGITAL SKILLS

Git Workflow Summary

Here is a overview of the general git workflow.

Git clone - Create a working copy of a local repository.

Git pull - Fetch and merge changes on the remote server and update the local repository to match that content.
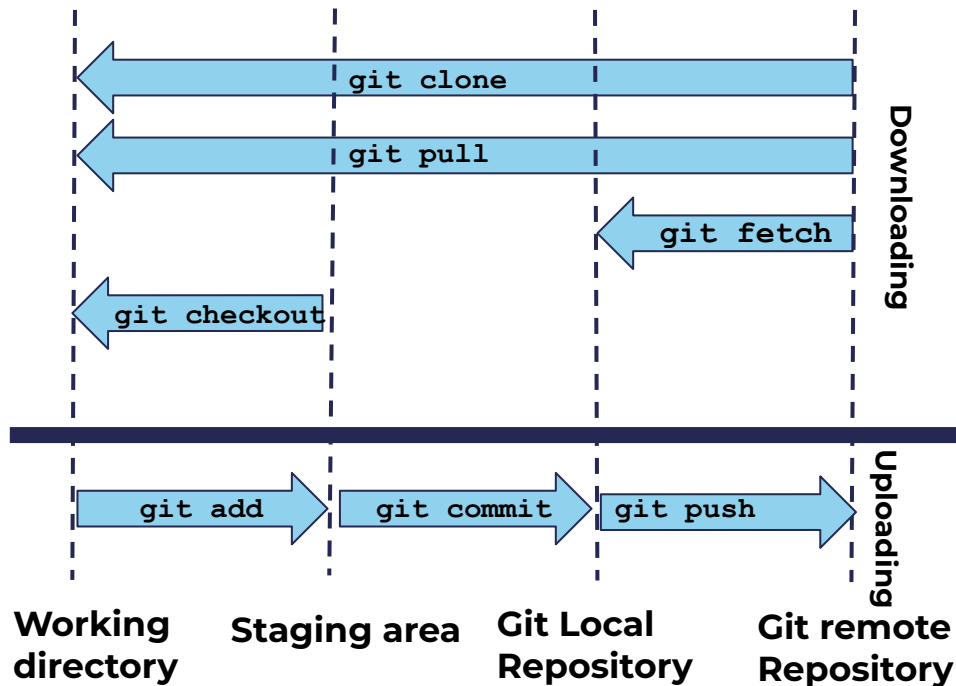
Git checkout - switch to another branch and updates files in the working directory, to match the version stored in the branch.

Git add - adds a change in the working directory to the staging area.

Git fetch - download content from remote repository.

Git commit - Commit previously staged changes.

Git push - upload local repository content to a remote repository.

`git clone`

`git pull`

`git fetch`

Downloading

`git checkout`

Uploading

`git add`　`git commit`　`git push`

Working directory　**Staging area**　**Git Local Repository**　**Git remote Repository**

EXPLORE | DIGITAL SKILLS

Visual Studio Code

Visual studio code is a streamlined code editor with support for not only software development tasks like debugging, and code completion, but version control too!

Important features of Visual Studio Code for Version Control:

- Visual Studio Code **supports Git by default**.

- Visual Studio Code **enables better understanding of Git commands** as it provides a git output window and shows you commands that it is using.

- **Provides several extensions to expand the functionality of Git** as seen in the image on the right.

Let's go ahead and install Visual Studio Code!

EXPLORE | DIGITAL SKILLS
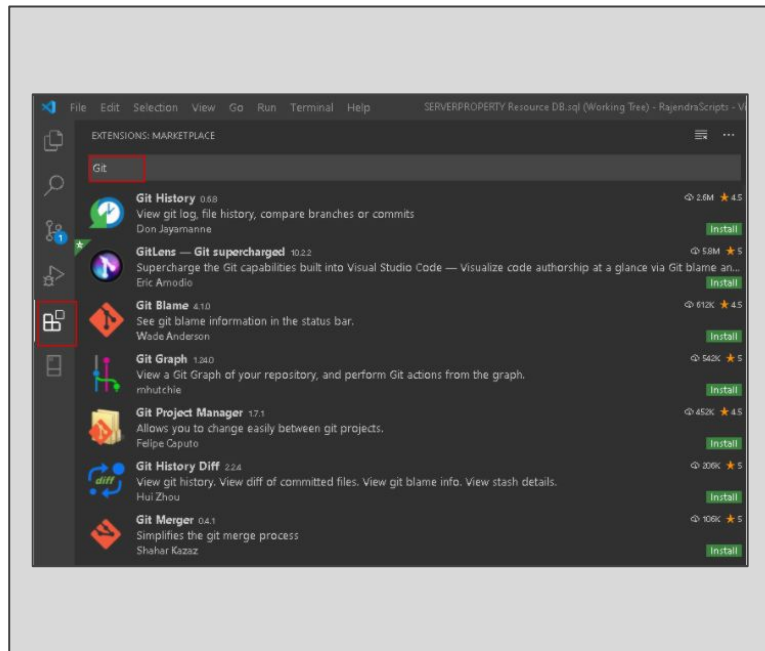
Installing Visual Studio Code on Windows
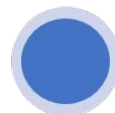
Step 1
Install the Windows version of VS code

Step 2
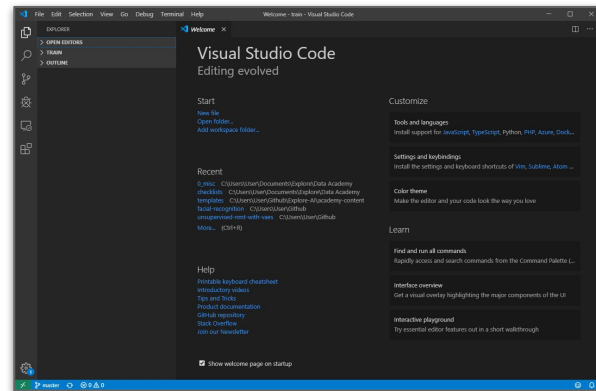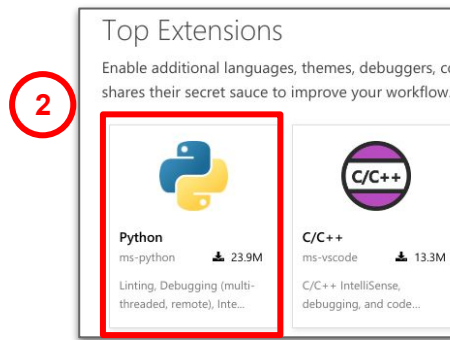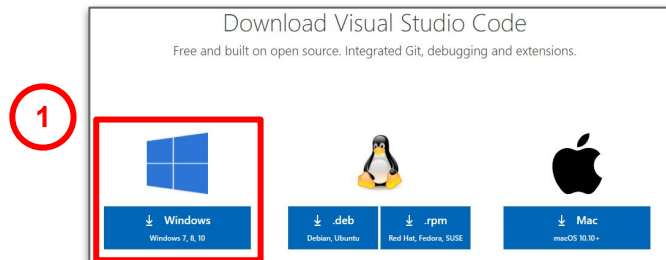Download and install the Python extension for VS code

Step 3
Open VS Code. Click 'New file' or add a workspace you already use.

Step 4
Start coding!

EXPLORE | DIGITAL SKILLS

Installing Visual Studio Code on macOS

Step 1

Step 2

Step 3

Step 4

Install the macOS version of VS code

Locate downloaded archive file.
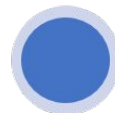
Move the visual studio code.app to the application folder

Download and install the Python extension for VS code.
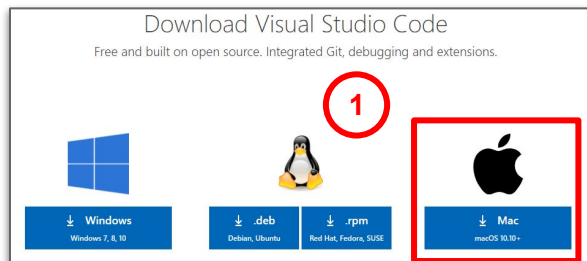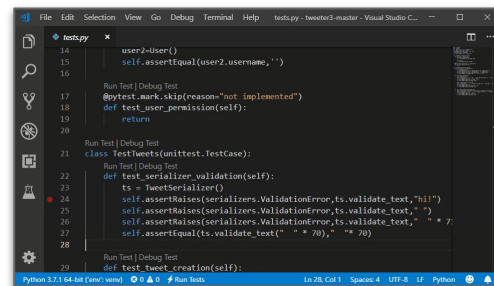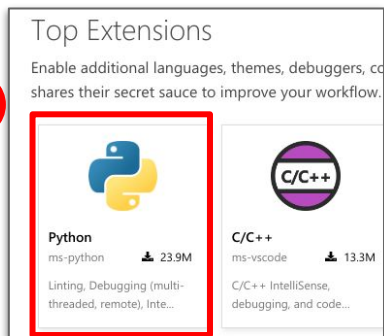
Be sure to take note of the Python requirements!

Open VS Code.

Click 'New file' or add a workspace you already use.

Start coding!

Download Visual Studio Code
Free and built on open source. Integrated Git, debugging and extensions.

Windows
Windows 7, 8, 10

.deb
Debian, Ubuntu

.rpm
Red Hat, Fedora, SUSE

Mac
macOS 10.10+

①

Top Extensions
Enable additional languages, themes, debuggers, co shares their secret sauce to improve your workflow.

③

Python
ms-python   ⤓ 23.9M
Linting, Debugging (multi-threaded, remote), Inte...

C/C++
ms-vscode   ⤓ 13.3M
C/C++ IntelliSense, debugging, and code...

EXPLORE DIGITAL SKILLS

Running Bash Commands from Visual Studio Code

It is possible to run bash commands from within Visual Studio Code as well! Follow the steps below to test it out:

1) Open up Visual Studio Code.
2) On the top tab, click on "Terminal" and select "New Terminal".
3) Great job! Just make sure that your script is set to "bash" in the drop down menu.
   If not, open up the drop down and select "Select Default Shell".

EXPLORE DIGITAL SKILLS
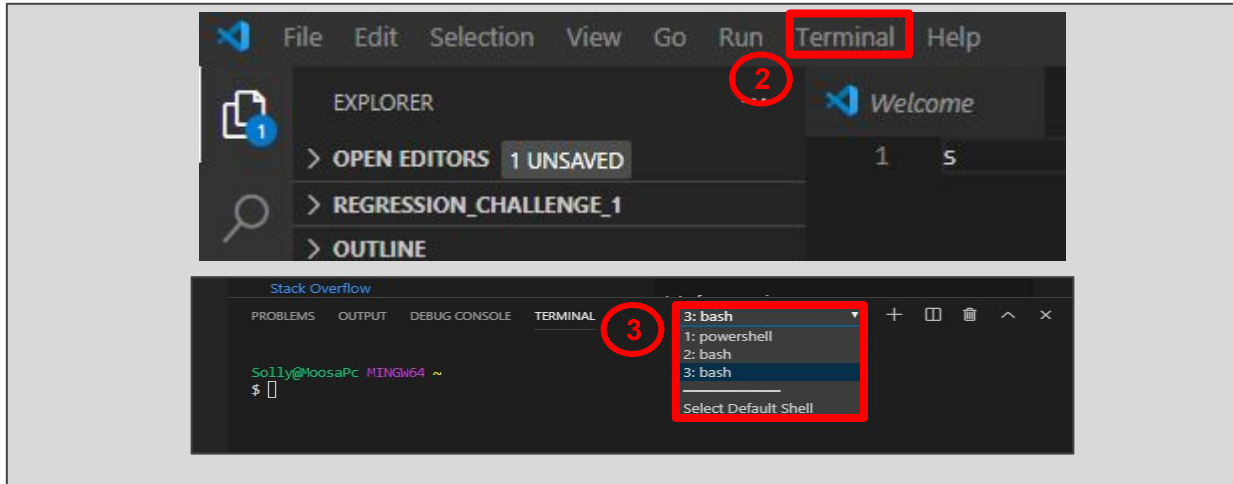
Conclusion

In this train we learnt about version control and it's application. We learnt how to install Git, set up a GitHub account and execute Git commands from a terminal.

We covered a range of useful bash commands, Git nomenclature and definitions that you can reference to give you full control of your code versioning!

EXPLORE || DIGITAL SKILLS

Appendix

- <u>Common Git commands</u>
- <u>Git tutorial</u>
- <u>An introduction to Git</u>
- <u>Getting started with GitHub</u>
- <u>Git basics: beginners guide</u>