

Module 1 - Lecture 1

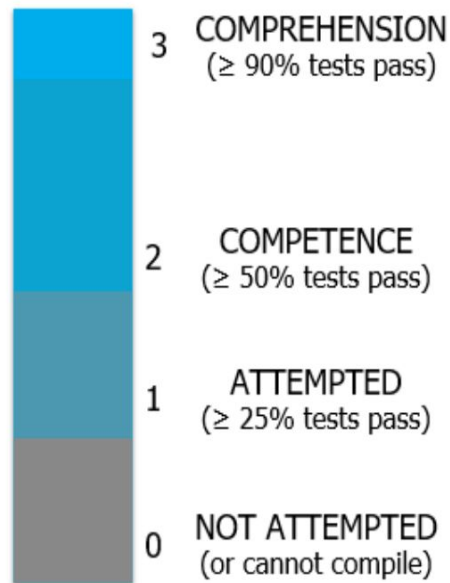
Introduction to Tools



WELCOME!

Mastery and Understanding

- Our exercises focus on **mastery of key concepts**.
- **Feedback will be provided** so you know where you need to improve.
- You must remain **at or above an average of 2.0**.
- **Any work submitted must be your own**. We may ask you to explain your code to us!
- Seek out help from your classmates, academic fellows, and instructors!

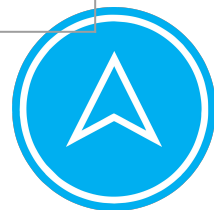


Due Dates

You will receive exercises daily (almost). You submit your work by pushing to GitLab.

Exercise Given	Exercise Due
Monday	Wednesday 8:00AM
Tuesday	Thursday 8:00AM
Wednesday	Friday 8:00AM
Thursday	Monday 8:00AM
Friday	Tuesday 8:00AM

Late exercises receive a 0. You may submit late, but the highest possible score is a 2.



DAILY CADENCE



SUPPORTING RESOURCES



Your schedule

- 8:55 AM (or earlier) - Be in class
- 9:00 AM - Class starts
- A little review
- Learn new material
- Attend Pathway event
- Complete homework
- Prepare for tomorrow
 - Read the book on BootcampOS (LMS)
 - Complete tutorials, if available
 - Take quiz on BootcampOS (LMS)
 - REST!



Spot the difference?



FILES!

What is a File System?

- Controls how data is stored and retrieved
- There are many different types
 - Operating systems support specific file systems
 - What are some differences between file systems on MacOS (APFS) and Windows (NTFS)?



What is a File System?

A collection of:

- Filenames
- Directories
- Metadata



Working with the File System

I need to do:

- Navigate
 - Read
 - Write
-
- This can be done using a Graphical User Interface (GUI) like Finder or File Explorer. However, developers often use a Command Line Interface (CLI).



SHELLS!

What is a Shell?

A **shell** provides a text-based interface to interact with a computer's operating system.

- Within a shell, you write code that the computer understands. This tells the computer what to do.
- Many tasks in programming are done on the command line because it is more flexible than most GUI interfaces and can be scripted.
- We will be using a very popular shell called bash. For the most part, we will use the shell to work with the file system and Git.



Common bash commands

- **Changing Directories**

`cd <directory_path>`

- **Print Working Directory**

`pwd`

- **List Directory Contents**

`ls, ls -al`

- **Moving Files and Directories (also used for renaming)**

`mv <old_path> <new_path>`



Common bash commands cont....

- **Remove Directory**

`rmdir <directory_path>` ** will only work if directory is empty
`rm -rf <directory_path>`

- **Remove File**

`rm <file_path>`

- **Copying Files and Directories**

`cp <old_path> <new_path>`

- **Making Directories**

`mkdir <directory_path>`



Common bash commands cont....

- **Making empty file**

```
touch <file_path>
```

- **View contents of a file**

```
cat <file_path>  ** displays entire file contents in shell
```

- **Clear shell of all text**

```
clear
```



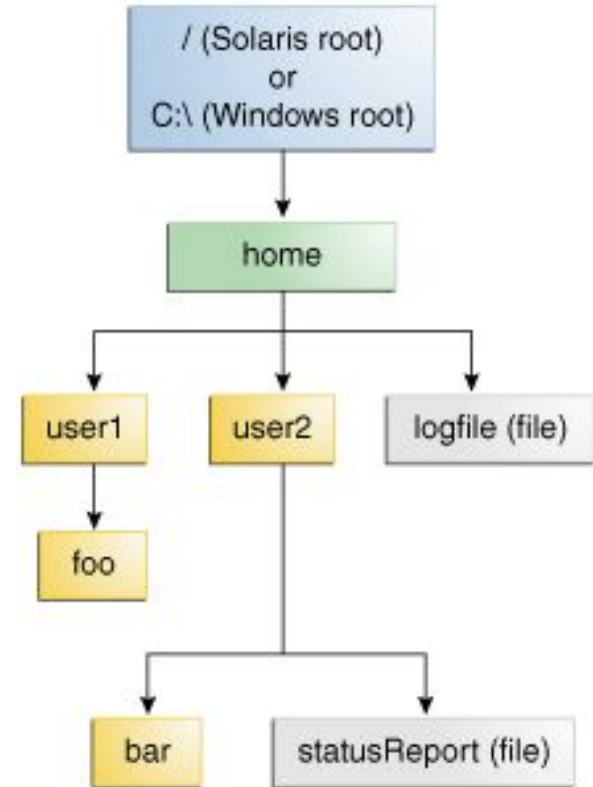
Absolute vs. Relative File Paths

A file path defines where a file or directory is. File paths can be **absolute** or **relative**.

An **absolute** path begins at the root of the file system.

A **relative** path begins where you are.

Paths are relative by default unless you provide an absolute path (begin with the root)



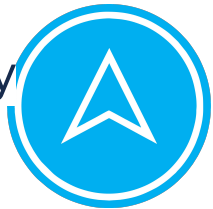
Miscellaneous shell identifiers

`/` represents the root directory when at the beginning of a path. Anywhere else, `/` is a directory separator.

`~` is an alias for the absolute path to your home directory (the Student folder) e.g. `~` is equivalent to `/c/Users/Student`

`.` is a reference to the current working directory

`..` is a reference to the parent of the current working directory



Let's Code!

Breakout

There is a file located in ~/dir1 named techelevator.txt

Your starting location is ~/start and your home directory is /c/Users/Student

Your task:

- Create a directory named end in your home directory.
- Move the file into the end directory.
- Remove the directory that the file was in initially.
- How did you solve it?

Git!



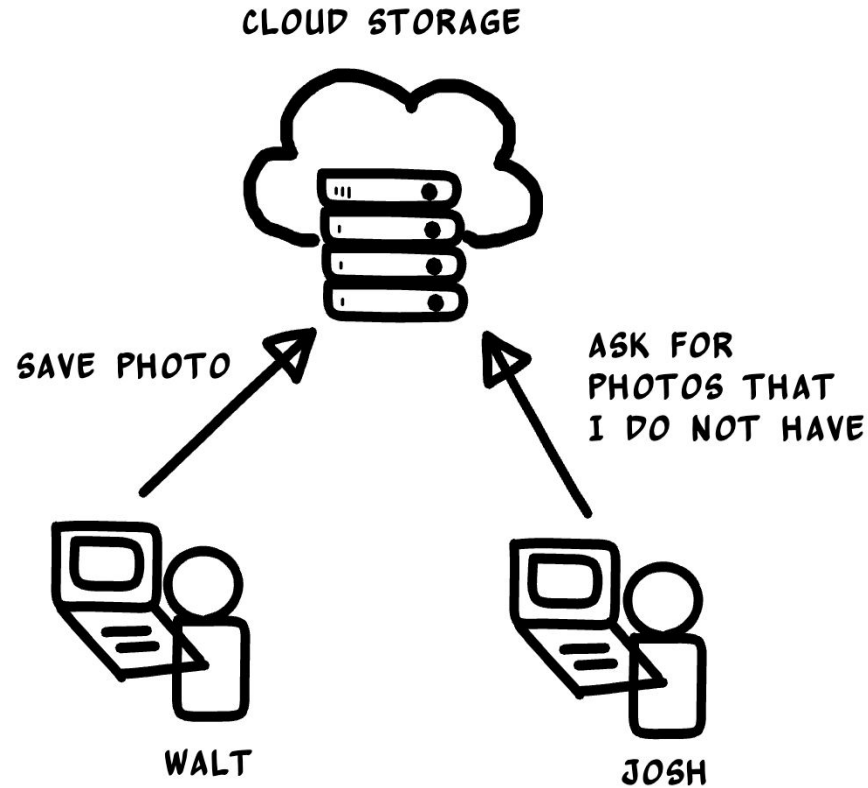
Dropbox

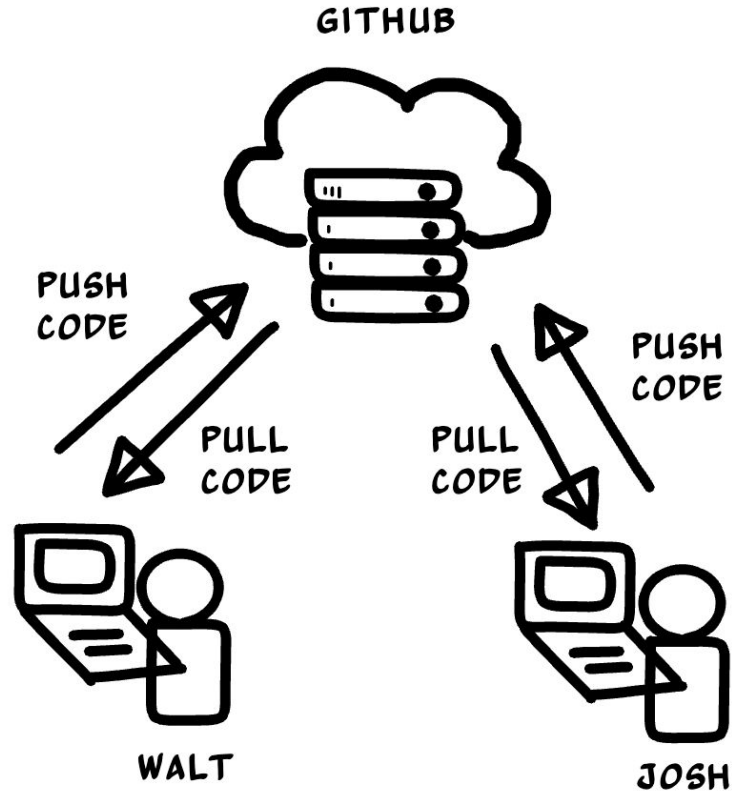


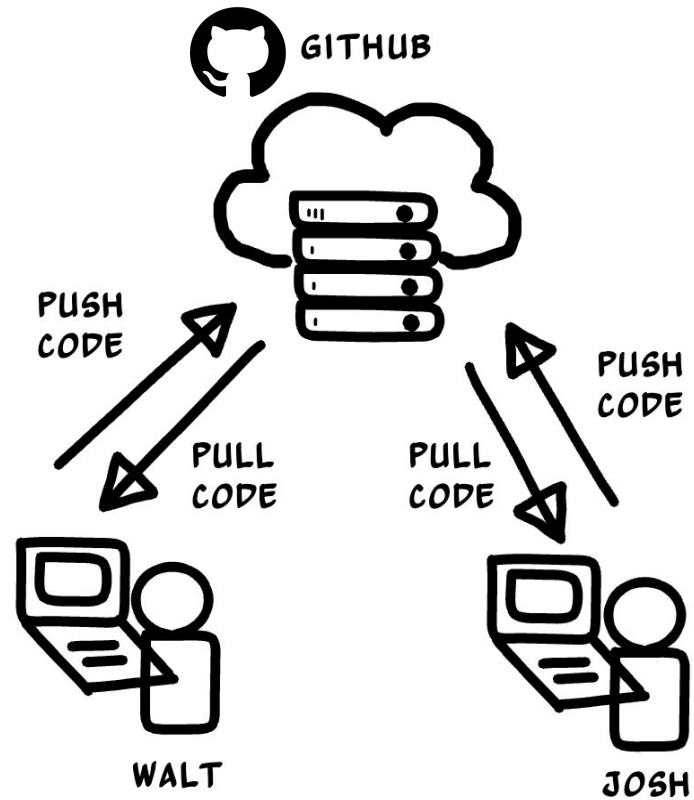
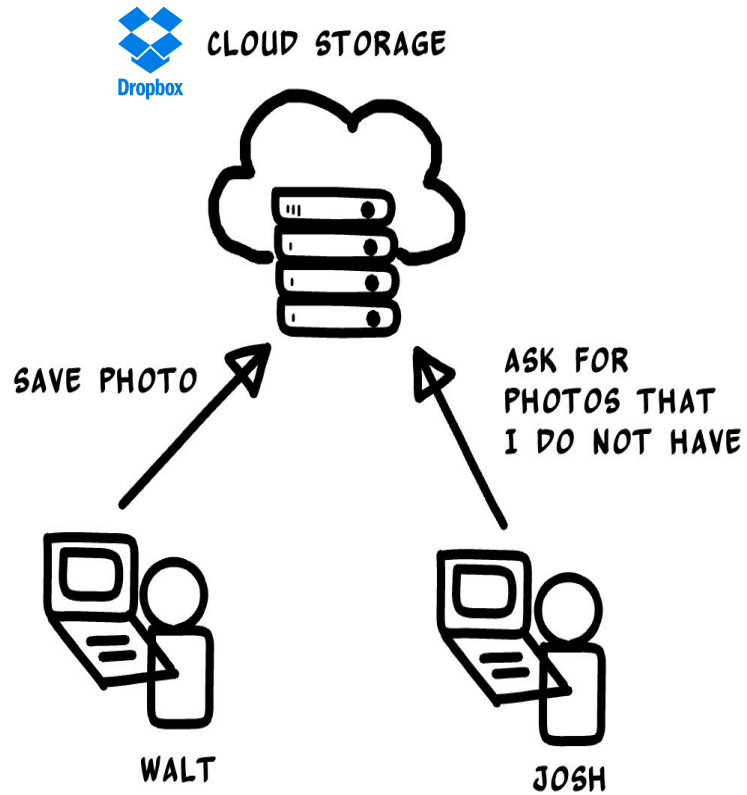
iCloud



Google Photos







Common Git commands

- **Create a repository on your local computer**

```
git init
```

- **Copy and track a repository from another location e.g. GitLab**

```
git clone <location_of_remote_repository>
```



Common Git commands

- **Check status of the repository**

`git status`

Files or Directories will be in one of the following statuses

- **Untracked** - a file or directory that is brand new to the repository.
- **Modified** - a file or directory that is tracked by Git and has been modified, but not yet staged for commit.
- **Staged** - a file that has been staged for commit.
- **Committed**



Common Git commands

- **Stage files**

```
git add <file_path_1> ... <file_path_n>
```

```
git add <directory_path_1> ... <directory_path_n>
```

```
git add .
```

```
git add -A
```

- **Commit staged files**

```
git commit -m <message>
```



Common Git commands

- **Pull and integrate (merge) changes from remote location**

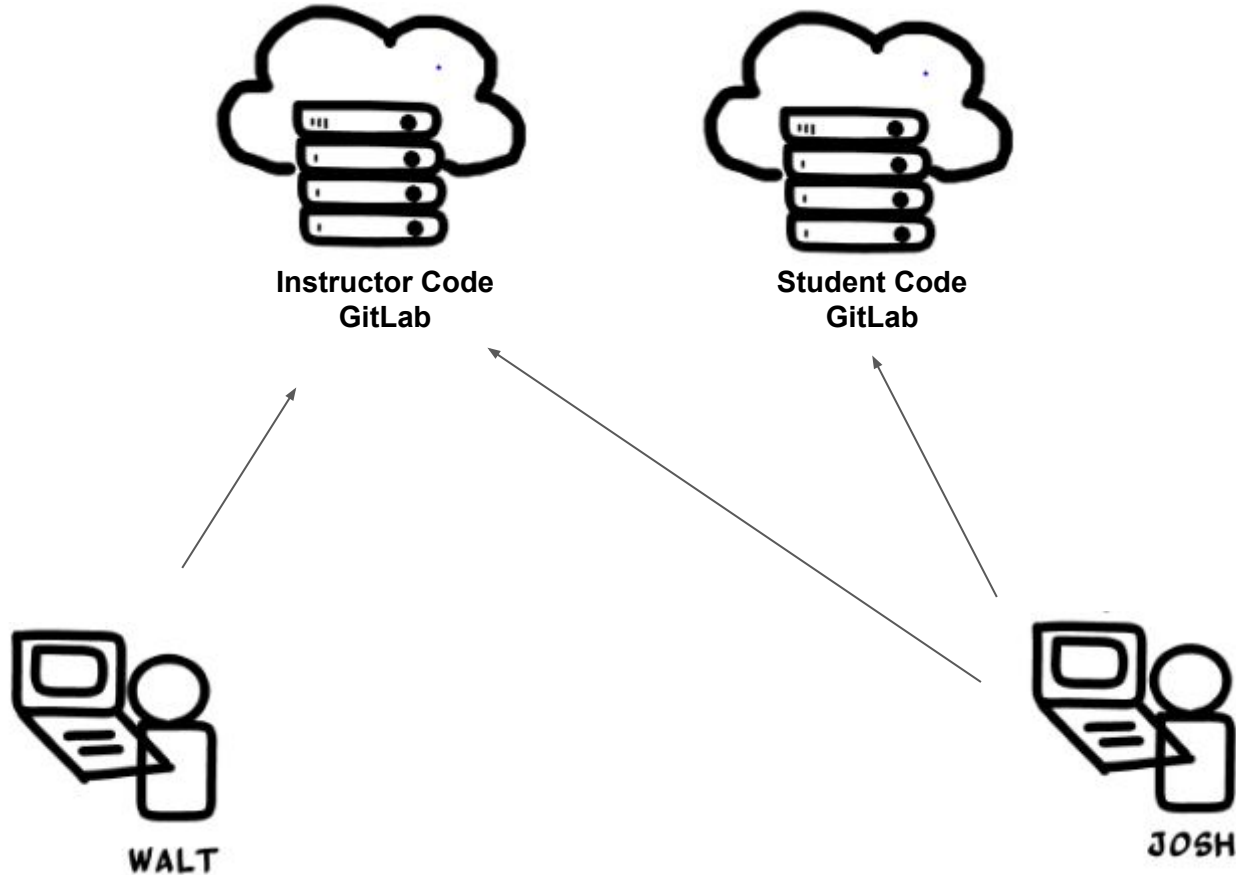
```
git pull <remote_alias> <branch_name>
```

- **Push committed changes to remote location**

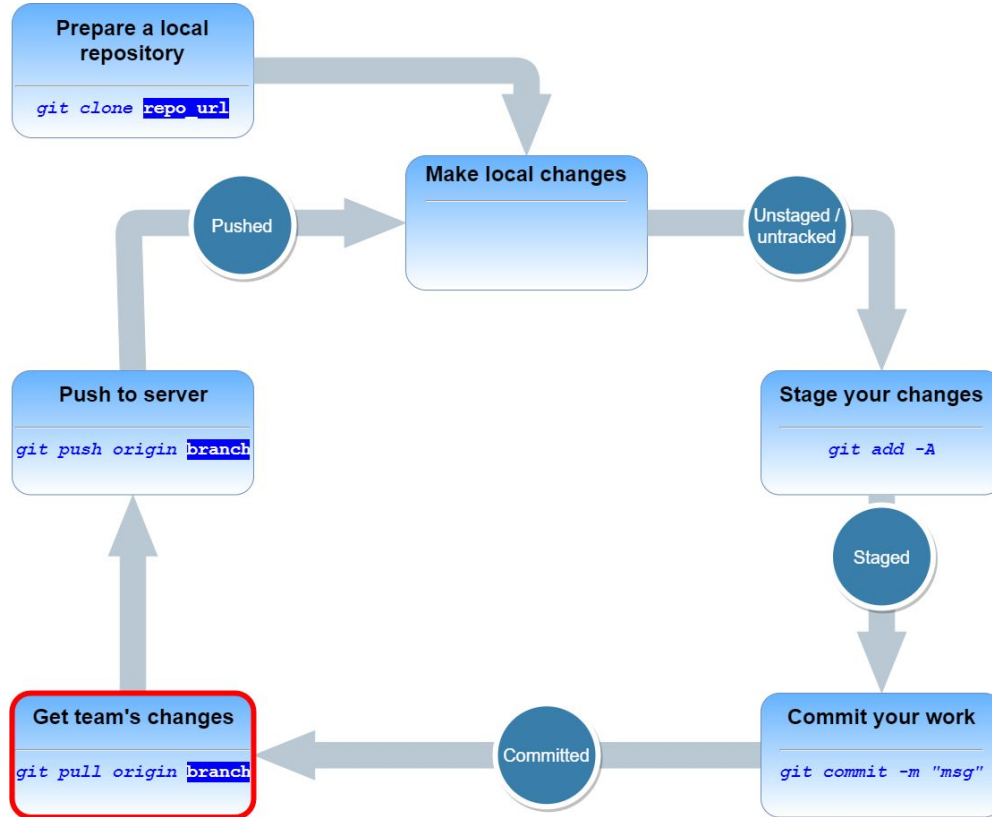
```
git push <remote_alias> <branch_name>
```



What is a remote?



Git Workflow



Let's Code!

QUESTIONS?

