Module 1-1

Command Line Shell & Version Control

What is a Command Line Shell?

- A shell is the means by which the user interacts with the computer.
  - Shells can be in the form of a graphical user interface (i.e. Windows, MacOS)
  - Shells can be in the form of a command line, where users type in commands.
- Information Technology professionals should be familiar with **command line shells**.
- In this class we will be using GitBash, which allows for UNIX commands from a windows workstation.

Command Line Commands: Moving Around

- Data in your workstation are organized into files and folders.
- The main command to move around folder is *cd.* There are several variations of these:
  - **cd ~** : Returns you to your home directory.
  - **cd <directory name>** : Takes you to a specified directory i.e. cd workspace takes you to a folder called workspace
  - **cd ..** : Takes you one level up.
- You can always see what directory you're in by typing pwd.
- The **ls** command lists all the files in the current directory.

Let’s Try this!

Moving Around: Absolute Path

- When you used the pwd command, the output would have looked something like this:

  ```
  Andy Chong Sam@DELL-JAVA MINGW64 ~/workspace
  $ pwd
  /c/Users/Andy Chong Sam/workspace
  ```

  Recall that pwd displays the current directory. Note that the response from this command is an absolute path since it starts with a slash (/).

Moving Around: Relative Path

- A relative path is differentiated from the absolute path by the absence of the initial slash:
  - **cd /c/Users/Andy Chong Sam/workspace** uses an absolute path to get me to the workspace folder.
  - Alternatively, if I were already in my respective user folder (Andy Chong Sam), typing **cd workspace** uses a relative path to get me to the workspace folder.

Moving Around: The Tilde (~)

- The tilde (~) is a special symbol used to denote the home directory. For all of your workstations this has been set to: /c/users/<Your username>

```
Andy Chong Sam@DELL-JAVA MINGW64 ~/workspace
$ cd ~/workspace
```

Therefore, the above command will take you to: */c/Users/Andy Chong Sam/workspace/*

Moving Around: Making Directories

- To create a directory we use the **mkdir <filename>** command.

Command Line Commands: Copying

- To copy a file from 1 directory to another: cp <source> <destination>

  ```
  Andy Chong Sam@DELL-JAVA MINGW64 ~
  $ cp ~/testdir/file.txt ~/othertestdir
  ```

- To move a file from 1 directory to another: mv <source> <destination>

  ```
  Andy Chong Sam@DELL-JAVA MINGW64 ~
  $ mv ~/othertestdir/file.txt ~/testdir/
  ```

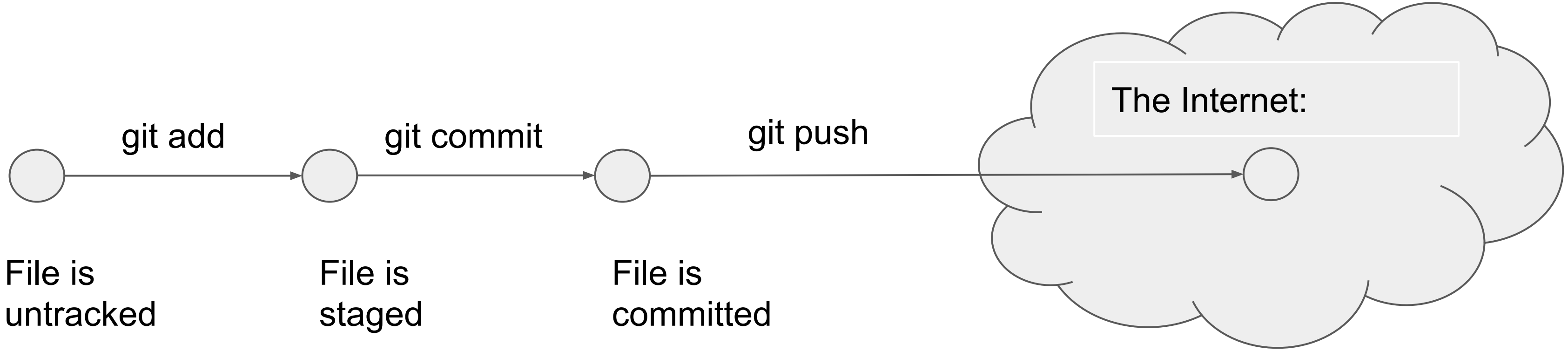
- Copy and Move differ in that the latter will remove the file from the source. With copy, the source retains a copy (pun intended) of the file.

Source Control : What it is

- Source control software allows developers to save and version their code.
- In this class, we will be using git / bitbucket.
- Git is an example of a distributed source control system, where a repository exists locally on your own workstation and on a central network location.

Source Control : Git Flow (Checking In Changes)

- **git status**: See the current status of your files.
- **git add -A**: Stage any files you have changed.
- **git commit -m “Commit message”**: Commit files to your local repository
- **git push origin master**: Push committed changes to network repository.



Source Control : Git Flow (Pulling Changes)

- **git pull upstream master:** Pulls latest from the remote repository.

- In this class we make a distinction between “upstream master” and “origin master”. Always pull from upstream master and push to origin master! There are some circumstances where this will change - the instructor will let you know.

(And refer to your handy cheat sheet).

Setup!

1. Clone your repository:
    - By now you should have received access to your BitBucket repo. You should be able to go to the following URL on your browser: **https://bitbucket.org/te-det-cohort-1/<*yourname*>/src/master/**
    - At the top of the page there is a clone command, copy this, it should look something like this: **git clone https://achongsam_TE@bitbucket.org/te-det-cohort-1/andychongsam.git**
    - Open Git Bash and navigate to your workspace folder. Paste the command you copied in and press enter. The shortcut for pasting things into Git Bash is **Shift+Ins**.

Setup!

2. Run the setup script:

Hey, check out what happened! You have a folder now inside your workspace with your name on it. Go into that folder by typing: **cd <yourname>**. i.e. cd johnsmith

There is 1 file sitting inside the folder, it is a shell executable that we need to run. Go ahead and run it by typing: **sh setup.sh**

Follow the prompt instructions.

This is the only time in the class you need to run this setup script!

Setup!

3. Let’s do our first pull.

- Make sure you’re in your name directory. Again, we can check with the **pwd** command. The output should be something like: **/c/Users/John Smith/workspace/cohort-1/johnsmith**
- Go ahead and type: **git pull upstream master**

Final Notes

- You want to pull often:
  - Pull when your instructors ask you to.
  - Pull first thing in the morning when you get to class.
  - Pull when you get back from lunch
  - Pull before you plan to push an assignment.
- Instructors will only grade what has been pushed to the BitBucket git repository. You can always check the web version of the repository to do a spot check to make sure what you pushed is actually there: **https://bitbucket.org/te-det-cohort-1/andychongsam/src/master/**