University day 2: Version control and GitHub

What is GIT

* A distributed version control system.
* It is the most popular version control system
* It can be used both collaboratively and for your own person projects
* Essential to understand if you want to work in development.

What is version control

* A system that records changes to a file or files over time
* This means you can so recall specific versions later if needed
* In this course we will be using code and text files
* You can version control almost any file
* Using a version control means that if you break code or lose it you can recover it.

What is GitHUB?

* Github is an online git repository store
* Commonly used to store and version control code
* Repositories range from personal to enterprise
* Some are public, some are private
* Alternatives include bitbucket, sourceforge, and AWS Code Commit

What are repositories

Creating a GitHUB account

Cloning repos

Adding files

Commiting files

Git config

Push

Pull

Initialising a local repo

Basic GIT Workflow #1:

* Step 1: Clone an existing remote repository.
* Step 2: Make changes to the code.
* Push the changes to the remote repository
* Pull

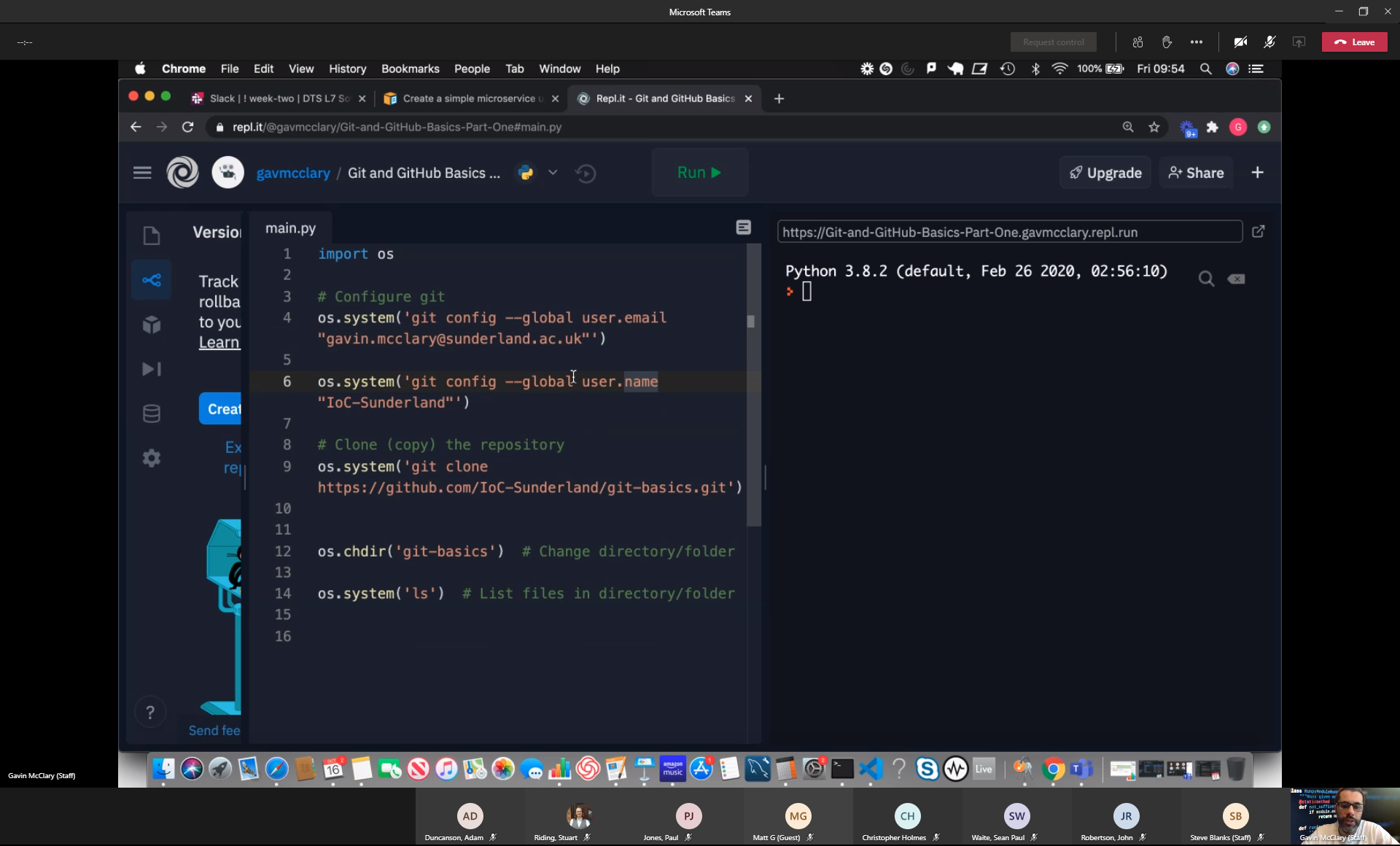
Commands:

* Os.system(‘git status’)
  + Item should be red
* Touch z
  + This creates an empty file\_name
* Os.system(‘git add z)
* Os.system(‘git status’)
  + Item should be green now
* Os.system(‘git commit -m “Added file called z”)
* Os.system(‘git status’)
  + Should not be green or red, and is now in the local machine (github wont show the result yet)
* Os.system(‘git push’)
  + This pushes it to the remote repository
  + Enter username for GitHUB
  + Enter password for GitHUB
* Os.system(‘git pull’)
  + If the remote and local repository are the same, then it’ll say it’s up to date.
  + However, if you make a change it’ll notice.
* Os.system(‘git status’)

Os.system(‘mkdir new-repo’)

Os.chdir(‘new-repo’)

Generic items required to start the command line configuration



# Clone down a repo

* Git clone https://///
* Git status
* Touch z
* Vi z
* Git commit -m “added z file”
* Git status
* Git add .
* Git status
* Git push

# Make a local repo

* Create a folder
* cd into folder
* ​git init .
* ​then create a file
* ​git add .
* ​git commit -m "some message"
* ​then go to git hub
* ​and create a repo with the same name as the folder
* ​git remote add origin [https](https://your) your repo link
* ​then you can git push -u origin master
* git checkout main

# task:

look at github workflows

* readme.md
  + md = mark down
  + # one hashtag is **heading**
  + ## two hashtags is **sub-headings (SMALLER)**
* **Git ignore**
  + **.gitignore (start the file)**
    - List of files you don’t want to push
  + For example if we have 3 files:
    - My\_test
    - My\_password
    - My\_life
  + And we don’t want to push my\_password, we just simply add my\_password into the .gitignore file and push that
* How to create different branches \*\*look into this\*\*

Afternoon Lesson:

Gitlog

* Author/User working on the repo
* User’s email
* Date/time
* SHA’s
  + Simple hashing algorithm
  + Identify each commit uniquely
* Commit history
  + Git log –oneline
  + Git log –stat
  + git

Flags

Git show

Checking out

Branching

* Tree, each branch is slightly different
  + Git branch
    - \* means head
  + Cut a new branch
    - Git checkout -b <branch name>
    - Git branch
  + Push the branch
    - Git push origin new-branch

Example:

* Git checkout -b new-branch
* Git branch
  + New-branch