**Git**

**Version control**

* Think when you write a document and you forget to save. If the computer dies, you lose your work.
* When you write a program, changes can affect how the whole structure works. One line being changed can break the whole program.
* This is why, when you change something, you have a new **version** of the program. **Version control** is how you manage the changes you make so that you don’t break the program, and can go back when that does happen.
* This helps you keep track of the changes you make, so you can more easily track when something goes wrong.
* This is especially important when you are programming collaboratively. Keeping track and undoing the changes **other people** make.

**Repository (repo)**

* **Working directory/tree:** This is where/what the working project is
* **Index:** Where the files are stored in the version control vault. An INDEX of versions of our program.

**We will create a set of folders to test version control**

* Create a text file (.txt)
* Put any text in it. Will pretend this is our program
* We open a terminal. Rightclick our new folder, choose GIT bash here.
* In the terminal, write **git init** and press enter.
* This creates the index where we will be ablet to put our fake program.
* **git add projectfile.txt**
* This adds our fake program (projectfile.txt) to the index system
* **Git add** \* (adds all files in the folder)
* **Git add** (adds our files to a list that will be in the repository once we commit them)
* **Git commit -m “commit message”**
* This commits all added files to the version control system we have created. “commit message” is essentially a comment we can add when we commit to explain what we are doing to everyone else.

**Add-commit-push functions**

* **Add** puts the changes you have made to your program in a working list
* **Commit** takes the list and stores it as a version on your local computer
* **Push** takes this stored version and puts it in a shared repository (online) for everyone to see.

**Push**

* **git remote add origin** [**https://github.com/saulomeirelles/WomenInPythonShenzhen**](https://github.com/saulomeirelles/WomenInPythonShenzhen)**.**
* This adds saulos github repository as the destination for our push
* **git push origin master**
* This now pushes our committed files to the designated repository (saulos github)