

# Introduction to Git and Github

# What is Git?

A free and open source distributed version control system.

Records changes to a file or set of files over time so that you can recall specific versions later.



# Installing Git

You may already have git installed: from the terminal (or Anaconda Prompt on Windows), try typing

```
$ git --version
```

If this doesn't work, browse to <https://git-scm.com/downloads> and download the appropriate version.

# Creating a Git Repository

Two common ways:

1) Convert an existing folder into a git repository:

```
$ git init
```

2) Cloning existing repository (say, from github):

```
$ git clone https://github.com/...
```

# Three States

Tracked files can be in one of three states:

- Modified: changed since the last commit and not yet committed



`$ git add <file(s)>`

- Staged: marked to go into your next commit snapshot



`$ git commit`

- Committed: safely stored in your local database.

# Checking the Status

To see the status of your git repository, use

```
$ git status
```

If you have any modified files, you can see the changes using:

```
$ git diff
```

To view a history of commits, use

```
$ git log
```

# Remotes

Git can function completely locally, but it is very powerful for collaboration when you use remote repositories (We will be using GitHub).

Remotes are versions of your project that are hosted on the Internet or network somewhere.



# Pushing to Your Remotes

When you have your project at a point that you want to share, you have to push it upstream.

```
$ git push <remote> <branch>
```

Most commonly, this will look like

```
$ git push origin master
```

If you are pushing to GitHub, you will be prompted to input your username and password at this point.



# Typical Workflow for Solo Project

**(1) Work on files locally. When done, run**

```
$ git add <files>
```

**(2) Commit changes:**

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

**(3) Push to GitHub:**

```
$ git push origin master
```