Terminal & Git Tutorial

Prepared by the Michigan Hackers Machine Learning Team

What is Git?

Most commonly used version control system for tracking changes in code files during software development.

Designed to coordinate work among programmers. Stores edit history of files.

We will be using Github, basically Google Drive for software, to collaborate on projects and keep track of progress.

Terminal

Provides a command-line interface to the operating system which allows you to DIVE files and applications. Git commands are run on terminal.

For MacOC/Linux, terminal is pre-installed as an application.

For Windows, download it from the Windows App Store.

(Windows is a bit more nit-picky with terminal because it isn’t a Unix OS, but we shouldn’t have any problems with Git on the windows terminal)

Useful Terminal Commands

Directory Commands

```
#pwd                                      // print working directory
#cd <dir>                                 // change directory
#cd ..                                    // go to parent directory
#cd ~                                     // go to home directory
#ls                                       // list contents in directory
#ls -A                                    // list all contents in dir
#mkdir                                    // make new directory
# rmdir <dir>                             // remove dir directory
```

File Commands

```
#touch <filename>                         // create file
#rm <filename>                            // delete file
#mv <filename> <newname>                  // rename file
#mv <filename> <dir>                      // move file to directory
```

Other Commands

```
#code .                         // open current dir in default text editor
#code <filename>                // open file in default text editor
#clear                          // clear terminal
#exit                           // exit terminal
```

Installing Git on MacOS/Linux

Open Terminal.

First check if Git is already installed on your computer by running the command

```
#git --version
```

If git isn’t installed, install HomeBrew first by running command:

```
#/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install.sh)"
```

Then run the command

```
#brew install git
```

Installing Git for Windows

Open Windows Terminal

First check if Git is already installed on your computer by running the command
#git --version

If git isn’t installed, install it from this link

Setting Up Git

Set up Git on the Terminal Command Line using *git config* commands:

```
#git config --global user.name "Rajas Gupta"            // Set name
#git config --global user.email rajasg@umich.edu        // Set GitHub email
#git config --global core.editor VScode                 // Set default editor
```

This step completes initializing git on your computer.

Cloning Repository

On terminal, go to the directory where you want to save the MHML GitHub repository on your computer using command

```
#cd <directory-location>
```

Create a folder called MHML and go to that folder in terminal

```
#mkdir MHML
#cd MHML
```

Copy and paste the following command to clone the machine-learning repo:

```
#git clone https://github.com/michiganhackers/machine-learning.git
```

Tracking files with Git

The following commands are helpful with tracking changes in the repo

#git status	List which files are staged, unstaged, and untracked.
#git log	Display the entire commit history using the default format. For customization see additional options.
#git diff	Show unstaged changes between your index and working directory.

Push and Pull

If changes are made to the main repo, you should **pull**. If you make changes on your local repo and want it saved to the main repo, you should **push**.

A good rule of thumb is to always **pull** before you start any work and **push** after you're done with your work.

To pull from the main repo, type in command:

```
#git pull
```

Disclaimer: this will delete all your unsaved work on the local repo.

To push your work to the main repo, type in commands:

```
#git add .                        // this stages your work
#git commit -m <description>      // this commits your work to your local repo
#git push                         // this pushed your commit to the main repo
```

Useful Git Commands

Setting Up Git

```
#git config --global user.name "<ur name>"       // Set name
#git config --global user.email <emal>           // Set GitHub email
#git config --global core.editor VScode          // Set default editor
```

Tracking Files

```
#git status                             // show unstaged files
#git log                                // show git commit history
#git diff                               // show main vs local
```

Pull

```
#git pull .                  // pulls latest commit on main
#git fetch                   // fetches main (doesn't change local)
#git reset                   // resets local to feteched commit
```

**Use fetch/reset if pull doesn't work and you want to restart your work.*

Push

```
#git add .                      // stages changes
#git commit -m <message>        // commits staged changes
#git push                       // pushes commited changes
```

Thank You!

Michigan Hackers Machine Learning Team
Contact Info:
Rajas Gupta | rajasg@umich.edu
Vijay Shamra | vsharm@umich.edu

Provide Feedback