GIT GUIDE

This guide is a work-in-progress, and may change in the future – medatech@medasf.org

Mission Economic Development Agency | Instructor Eduardo Garcia

WHAT IS GIT?

The open-source software Git was created by Linus Torvald (Yes! The same guy that made the Linux kernel). Git was created so that files can be “saved” and then kept somewhere secret so the user doesn’t have to worry about organizing versions of their work. Git is one of the solutions for *software version control*, software designed to manage versions of projects and other software. Other popular software packages are Subversion (now under Apache) and Mercurial.

HOW DOES GIT WORK?

Unlike pressing Save in your code or text editor, Git saves multiple files at once (and you have control over which files too), but with one extra feature, Git saves all history of each “save”, which allows the user to go back in time to a specific save.

COMPARISON TO WINDOW’S SHADOWCOPY AND APPLE’S TIME MACHINE

Git is OS platform independent, this means that Git can be installed on Mac, Windows, and Linux/Unix machines. Because of this independence, developers can share Git-saved projects with each other without any compatibility issues. This compatibility has lead to popular online services that allow developers to share their work with others using the Git software. Two great examples of these online services is GitHub and BitBucket, websites that are the “Facebook” for developers where they share code instead of images and videos.

GIT SLANG

- ➔ **commit** – The act of “saving” a git project and adding the changes to the history of changes. *Please commit your git project before leaving class!*
- ➔ **(git) repository** – A git project folder. *My git repository was destroyed!*
- ➔ **push** – Sending the latest copy of your git repository to another computer that has an outdated version of the same git repository. *I need the latest version of the project, can you push your repository?*

➔ **clone** – The act of making a copy of an existing git repository. *You can just clone that repository to grab a copy of the source code.*

➔ **branch** – A git branch is an alternate copy of the existing repository so you can try something out to see if it works. You can imagine this as a parallel dimension where the original dimension is unaffected. The "original" branch is called the master branch. *I'm going to branch this repository so I can test an idea that I want to implement.*

➔ **merge** – The act of merging two branches into one, generally used to combine working code into a single branch. *My idea worked, so I will be merging my code into the master branch.*

CREATING A GIT REPOSITORY

(1) Open a terminal window and point the terminal to the folder you want to convert into a git repository.

(2) Type the command and press enter: **git init**

(3) Follow the instructions presented to you in the terminal.

CLONING AN EXISTING REPOSITORY

(1) Open a terminal window and point the terminal to the folder that you want the git repository folder to be saved in.

(2) Find the URL for the repository you want to clone, you will be using this URL in the next step.

(3) Type the command and press enter: **git clone *url address from step 2***

COMMITTING A GIT REPOSITORY

(1) Open a terminal window and point the terminal to the folder that is a git repository

(2) If you want to save all changed files type the command and press enter:

git add -A

(3) If you want to save a specific file, type the command and press enter:

git add *filename*

(4) (Optional) You can confirm the changes you want to make by typing the command and pressing enter: **git status**

(5) To submit the commit type the following (Note: “message” should be a description about the save itself and not the project, it should be surrounded by double quotes): **git commit -m “*message*”**

(6) Follow step 4 to check if the commit was successful. A message “Working tree clean” means that the current status of the repository is *no different* than the last commit.

PUSHING A REPOSITORY

(1) Open a terminal window and point the terminal to the folder that is a git repository

(2) Type the command and press enter: **git push**

NOTES
