Prepared by Joseph Mulingwa Kithome.

Git & GitHub

Definition of terms:

Git is a distribute version control tool that supports non-Linear workflows by providing data assurance.

Version control System

A version control system, or VCS, tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:

1. Which changes were made?
2. Who made the changes?
3. When were the changes made?
4. Why were changes needed?

Distributed version control system

Git is an example of a distributed version control system (DVCS) commonly used for open source and commercial software development. DVCSs allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.

Purpose of Git

Git lets developers see the entire timeline of their changes, decisions, and progression of any project in one place. From the moment they access the history of a project, the developer has all the context they need to understand it and start contributing.

Repository

A repository, or Git project, encompasses the entire collection of files and folders associated with a project, along with each file's revision history. The file history appears as snapshots in time called commits, and the commits exist as a linked-list relationship, and can be organized into multiple lines of development called branches.

Basic Git commands

- **git init** initializes a brand new Git repository and begins tracking an existing directory. It adds a hidden subfolder within the existing directory that houses the internal data structure required for version control.

- **git clone** creates a local copy of a project that already exists remotely. The clone includes all the project’s files, history, and branches.
- **git add** stages a change. Git tracks changes to a developer’s codebase, but it’s necessary to stage and take a snapshot of the changes to include them in the project’s history.
- **git commit** saves the snapshot to the project history and completes the change-tracking process. In short, a commit functions like taking a photo. Anything that’s been staged with git add will become a part of the snapshot with git commit.
- **git status** shows the status of changes as untracked, modified, or staged.
- **git branch** shows the branches being worked on locally.
- **git merge** merges lines of development together. This command is typically used to combine changes made on two distinct branches.
- **git pull** updates the local line of development with updates from its remote counterpart.
- **git push** updates the remote repository with any commits made locally to a branch.

How Github Fits in

GitHub is a Git hosting repository that provides developers with tools to ship better code through command line features, issues (threaded discussions), pull requests, code review, or the use of a collection of free and for-purchase apps in the GitHub Marketplace. With collaboration layers like the GitHub flow, a community of 15 million developers, and an ecosystem with hundreds of integrations, GitHub changes the way software is built.

The GitHub flow

1.Create a branch

2.Add commits

3.open a pull request

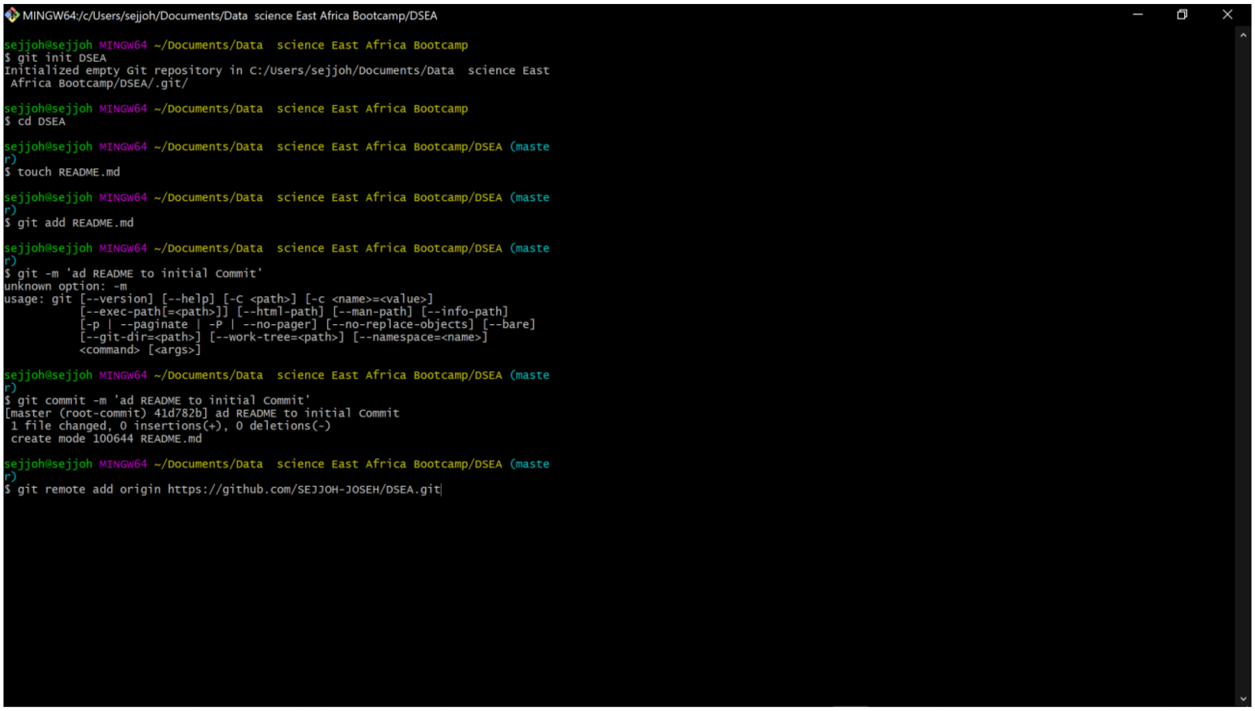
4.Discuss and review the code.

55.Merge

6.Deploy.

My Git cheat Sheet

Start a new repository and publish it to GitHub



MINGW64:/c/Users/sejjoh/Documents/Data science East Africa Bootcamp/DSEA

```
sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp
$ git init DSEA
Initialized empty Git repository in C:/Users/sejjoh/Documents/Data  science East
 Africa Bootcamp/DSEA/.git/

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp
$ cd DSEA

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp/DSEA (maste
r)
$ touch README.md

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp/DSEA (maste
r)
$ git add README.md

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp/DSEA (maste
r)
$ git -m 'ad README to initial Commit'
unknown option: -m
usage: git [--version] [--help] [-C <path>] [-c <name>=<value>]
           [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
           [-p | --paginate | -P | --no-pager] [--no-replace-objects] [--bare]
           [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
           <command> [<args>]

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp/DSEA (maste
r)
$ git commit -m 'ad README to initial Commit'
[master (root-commit) 41d782b] ad README to initial Commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 README.md

sejjoh@sejjoh MINGW64 ~/Documents/Data  science East Africa Bootcamp/DSEA (maste
r)
$ git remote add origin https://github.com/SEJJOH-JOSEH/DSEA.git
```

Then git **push.**

Git Features

It is non- linear- git allows users from all over the world to perform operations on a project remotely.

It is secure-keeps a record of all the commits done by each of the collaborators on the local copy of the developer.

Lightweight- stores all the data from the central repository on the local repository while cloning is done.

Branching-git allows its users to work on a line that runs parallel to the main project files.

Open source-designed to handle everything small or large projects with speed and efficiency.

Reliable

Introduction to python.

Python is an interpreted, high-level, general purpose

programming language.

Created by Guido van Rossum and first

released in 1991,

Uses of Python

- Web development (server-side),
- Software Development,
- Mathematics,
- System Scripting.

Simple python program

```
print("Hello,World!")
```

Python Variables

A variable is created immediately it is instantiated.

```
X=2
```

Variables cannot start numbers.