FTDS // GIT & GITHUB

Hacktiv8 DS Curriculum Team

Phase 0 Day 2 AM 2021

Phase 0

2021

Contents

WEEK 1
GIT & GITHUB

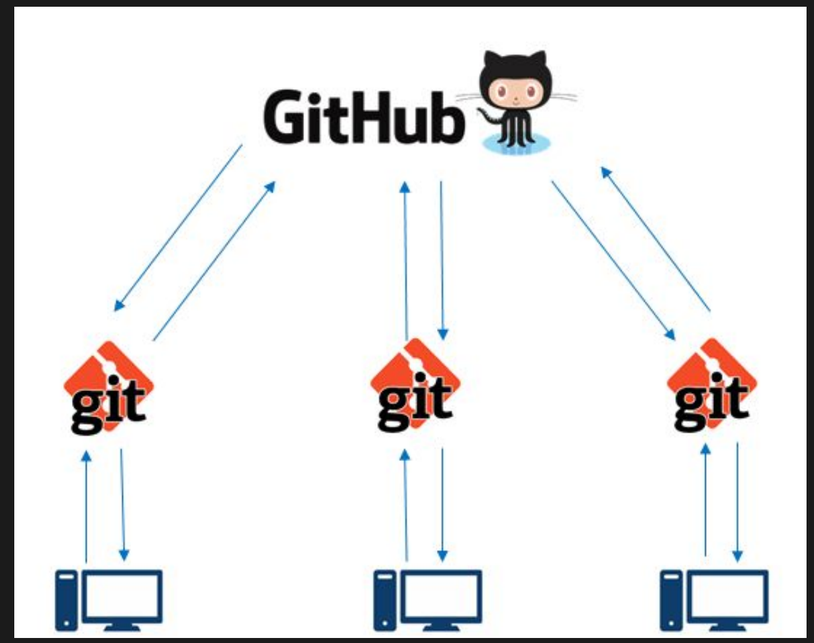
What is Git?

- Git is a Version Control System (VCS) designed to make it easier to have multiple versions of a code base, sometimes across multiple developers or teams.
- It allows you to see changes you make to your code and easily revert them.
- It is NOT GITHUB!

WEEK 1
GIT & GITHUB

What is Github?

- Github.com is a website that hosts git repositories on a remote server.
- Hosting repositories on Github facilitates the sharing of codebases among teams by providing a GUI to easily fork or clone repos to a local machine.
- By pushing your repositories to Github, you will pretty much automatically create your own developer portfolio as well!



WEEK 1

GIT & GITHUB

Bash Shell Basics

Shell is the command interpretor in an operating system such as Unix or GNU/Linux, it is a program that executes other programs.

Bash is the shell, or command language interpreter, for the GNU operating system.

The name is an acronym for the ' Bourne-Again SHell ', a pun on Stephen Bourne, the author of the direct ancestor of the current Unix shell sh , which appeared in the Seventh Edition Bell Labs Research version of Unix.

WEEK 1
GIT & GITHUB

Bash Shell Basics

Basic Commands:

- **echo is a command to write a text or string, example: echo "Hello hacktiv8" . If you want to add text to the file : echo "Hello, my name is sardi" >> name.txt .**
- **pwd (print working directory) is a command to find the path of the directory (folder) you are currently using.**
- **cd to call a directory or path. example: cd /home/username/sardi/hacktiv8 . If you want to move to the root directory, you can type the command: cd ~. If you want to go to the parent directory of hacktiv8 by typing: cd .. , then you will be in the path /home/username/sardi/.**
- **ls is the basic command used to view the contents of a directory.**
- **mkdir is the command that works to create a new directory. example: mkdir name_directory**

WEEK 1
GIT & GITHUB

Configuring git

Confirm that you have git:

- Open your terminal and run ‘git’ .
- If you see a ‘command not recognized’ error, you probably haven’t installed git yet.

Your commits will have your name and email attached to them. To confirm that this information is correct, run the following commands:

- git config --global user.name
  - Example: git config --global user.name “Jon Rosado”
- git config --global user.email
  - Example: git config --global user.email “jon.rodado42@gmail.com"

WEEK 1
GIT & GITHUB

Connecting git with Github

- In order to prevent having to enter your password each time you push up to Github, you must configure git and Github to recognize Secured Shell (SSH) keys that you generate.
- To check and see if you have any recognized SSH keys active on Github, go to https://github.com/settings/keys
- If using Mac OS X, you can also configure your keychain to automatically enter your password via HTTPS.

WEEK 1
GIT & GITHUB

Connecting git with Github

- From your project directory, run `git init` to initialize a git repository.
- Go to Github, and create a new repository with the name of your project.
- Follow the instructions on Github to connect your initialized git repository to the remote server on Github.
- Before we can upload all the revisions in the local repository, we have to add the remote first ,

```
git remote add origin https://github.com/sardiirfan27/belajar_git.git
```

- There are two remote URL options that we can provide:
  - Via HTTPS: https://github.com/sardiirfan27/belajar-git.git
  - and via SSH: git@github.com:sardiirfan27/belajar-git.git

*Please Note: you must have files in your project directory to commit in order to push anything to your remote server

WEEK 1
GIT & GITHUB

Connecting git with Github

- If there are no more changes, enter the command :
  - git commit -m "Initial Commit add Session 1" then enter.
- Run the command git push origin master to push the committed files to the remote repository. You may be asked to enter a password for the first push repository.

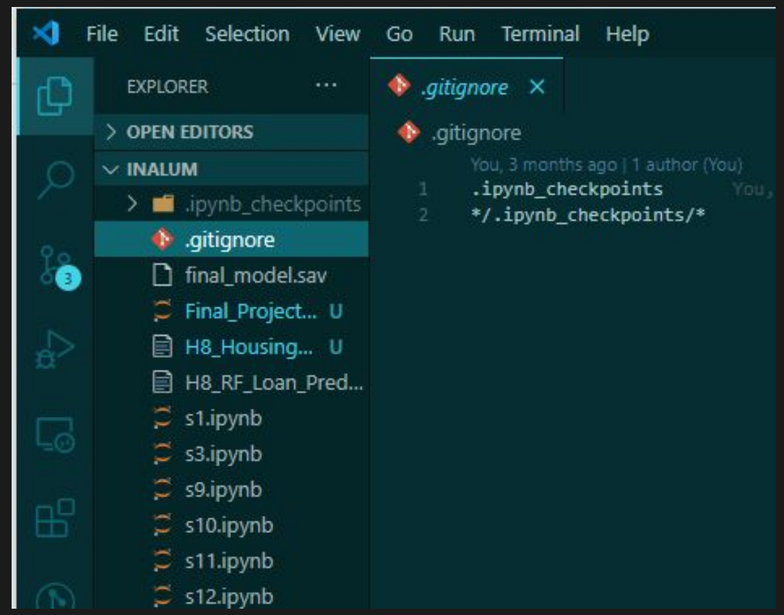
WEEK 1

GIT & GITHUB

Git Ignore

.gitignore is one of the default Git software ignorers. With gitignore, files generated by jupyter notebook such as .ipynb_checkpoints will not enter the stagging area when we run the git add command.

Create a .gitignore file in the Notebook's root directory in Visual Studio Code or your preferred editor. In the gitignore file, enter the name of the file/directory you want to ignore. As an example:



WEEK 1

GIT & GITHUB

Readme

A README is a file that contains information about other files in a computer software directory or archive. In this case, a README is created to annotate your github repository.

Create a README.md file in the Notebook's root directory in Visual Studio Code or your preferred editor. Enter your program name, batch, and identity in the README file that you have created. Include any additional information that you think is necessary.

WEEK 1
GIT & GITHUB

Branch

A branch is essentially is a unique set of code changes with a unique name. Each repository can have one or more branches, so you can create a new branch in your repository and push to it. For example:

1. **clone your repository: git clone https://github.com/latihan3/wqw.git**
2. **move to the directory of the cloned repository: cd wqw**
3. **Create a new branch: git checkout –b my_branch**
4. **Edit, add and commit your files.**
5. **Push your branch to the remote repository: git push -u origin my_branch**

Hacktiv8 DS
Curriculum
Team

External References

Colab Link

Visit Here