Nathan Strom
HWRS 501

GitHub 101 Cheat Sheet

GitHub

⇒ Website / web-based platform that allows for collaboration and storage of code / coding projects.
⇒ Service for the sharing, collaborative editing, and updating of code.
⇒ Hosts **Repos** (Repositories)

GitHub utilizes **Git.**

⇒ **Git:** The **Version Control** system that **GitHub** utilizes
⇒ Utilizes **Version Control** to log the history of develop/changes of a code project.
- **Version Control** – a software that logs the progress and changes of a coding project, document, etc.
  - Tracks changes made by various parties to the same document.
  - Allows for **Branches**:
    - changes that have been made to an original document but not integrated back to the original source code.
    - Allows for editing of a copy of the main code w/o implementing the edits.

Git & GitHub… what is the difference???

⇒ Git is the software that performs a Version Control while GitHub is the host service… it allows for the hosting and management of **Git Repos**
- GitHub is the web-based interface that hosts Git Repos

Common GitHub Terms	Definition
REPO (Repository)	⇒ Stores a collection of files and historical edits of these files. ⇒ Logs activity and edits to the files ⇒ Version control uses repos to organize the changes/edit history of files/projects/documents.
COMMIT	⇒ a record of changes, edits, editor, and timestamp information, and attached edit messages that get logged to changes on a repo. ⇒ Shows a breakdown of the editing progress through time.
PUSH	⇒ uploading changes and commits made on a local repo to the remote repo… you are "pushing" your changes forward to the remote.
PULL	⇒ the act of "downloading" or integrating changes into a repo existing on a local machine. ⇒ This will integrate changes made from other parties on the remote repo to your locally stored repo.
ORIGIN	⇒ the mother of all repos of the project. This is often hosted on the GitHub cloud or some type of cloud storage and where final changes and edits get pushed to and pulled from.

Nathan Strom
HWRS 501

LOCAL	⇒ an "off the cloud" area of storage and editing… essentially your own computer
REMOTE	⇒ hosted on the cloud server… often hosted on GitHub… several team members of a project can pull changes from a computer and push their local changes.
CLONE	⇒ a copied repo from the remote server onto your local machine. Contains repo history and previous versions.
BRANCH	⇒ a copy of a repo from a specified time of edit history. It is used to test edits from the original without editing the original, then giving the option whether the coder would like to integrate such edits back to the original. ⇒ Multiple branches can be made and multiple "what-if" type edits can be tested, all without modifying the original code.
FORK	⇒ Making a copy of a repo and storing it on a remote server. Your personal copy is in the cloud. ⇒ Do not need to use local machine push and pull type commands, can instead use PULL REQUESTS.
PULL REQUEST	⇒ process of setting/indicating that repo changes made are being requested to be implemented into the main code. This can sometimes allow for review by other team members before implementation.
FETCH	⇒ The action of downloading/extracting changes made to a remote repo and storing them onto your computer. ⇒ Fetching DOES NOT integrate/merge them into your locally stored repo.
STAGE	⇒ act of choosing which changes you would like to commit.
COMMIT	⇒ the process of saving the stage files onto your local system and prepares them to be pushed forward to the remote repo.

Repo vs. Folder… What's the difference?

⇒ Both a folder and a Repo store files/information on your computer. But Repos have a Version Control — the files edits and changes are embedded w/in the Repo.

Pulling vs. Fetching… What's the difference?

⇒ The process of Fetching is enacted in the act of Pulling. By fetching, you are only retrieving the change history but not automatically integrating them into your local repository.

Local vs. Remote Vs. Origin… What's the difference?

⇒ Local is stored on a local system such as a computer while remote refers to the storage on a cloud or internet-based system — a storage server that is disconnected from the physical device one is working on and others can access from. Origin refers to the source a project or group is extracting their repos and data from.

How does a Repo work? What is its purpose? How does it relate to GitHub?

⇒ Purpose — have a working history of a project — allows for the analysis and ability to revert to previous states.

From Repo edit to submission

⇒ **Imagine a scenario:** There is a buggy code file in the GitHub cloud. It is contained in the Remote Repo. The boss wants you to fix it. Here's the process.

⇒ To Pull or Clone??? What to consider…
- If you already have a copy of this Repo on your remote system and you want to update it with the changes made since the time you copied… then PULL
- If you do not have a previously cloned Repo… then CLONE

⇒ Make the changes/edits to project on local system into local repo. Once these are complete → STAGE the files → COMMIT the changes → PUSH to the Remote Repo
- This process can be done through the computers Terminal, command line, or run through a software such as GitKraken.

⇒ After a successful PUSH, another project member could then PULL from the remote repo and integrate your changes onto their local system. They could run your version of the previous program and run it on their computer, make changes, etc. and repeat the entire cycle.

Fork vs. Clone

⇒ The difference lies in the location of storage. A clone is a repo personal copy to a local machine while a fork is a personal copy stored with a remote server. The storage locations cause slight differences in how changes are submitted and implemented to the main code.

Why Branch?

⇒ Branching is useful because it allows one to split off from the repo at a chosen time of edit history (at the "Head"), make edits and alterations, test these changes, then decide if and what they would like to integrate back to the main repo.