**Github** – website that displays files in remote repos in the cloud…where you push to and pull from

**Git** – version control software on your local machine

**Repo** – this is a file folder “on steroids”; it keeps track of everything—all changes to things within it

**Local vs. Remote vs. Origin**

* **Local** refers to items on your local machine (also working directory vs. local repo--different)
* **Remote** is anything on the cloud/Github
* **Origin** is the nickname for the default remote repo that your local repo is linked to

**How to Get Your Own Version of a Repo**

* **Clone**: this makes an exact copy of a remote repo on your local machine, create a local repo
* **Fork**: this makes a copy of a remote repo in your space on the cloud—creates a remote copy

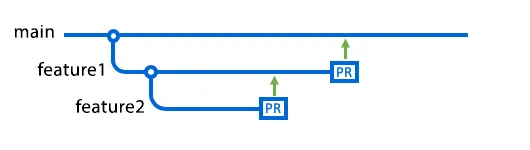
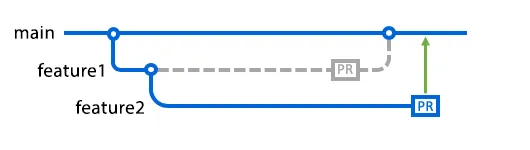
**Making Changes to a Repo**

* **Stage**: mark files that you have changed to go into the next update that gets saved locally
* **Commit**: saves changes to your local repo; this is a point to which you can go back
* **Push**: sends all of your changes that have been committed locally to the remote repo

**Retrieving Changes to Remote Repo**

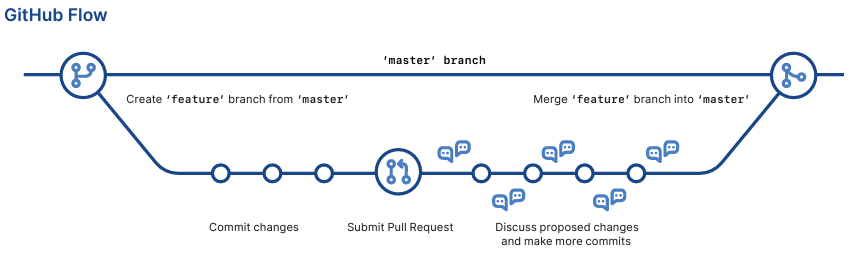
* **Fetch**: like half of a pull; updates your local repo with latest from remote, but doesn’t make any changes to the local working directory (won’t see in local folder)—it doesn’t overwrite anything
* **Merge**: takes files that have been fetched to your local machine and puts them into local folder
  + *Changes/updates files that are there from updates from fetch—or adds new files, etc.*
* **Pull**: grabs everything that’s new in the remote repo and merges it with/updates the local repo
  + *A combination of a Fetch and a Merge*

**Branches**

* It’s a parallel version of a repo—still in the repo, but doesn’t affect the main branch
* Creates an “easy bailout option” for you—can do whatever to the branch and doesn’t affect main
* If you create a branch, main is the base; create one from another branch, it is the base; if first branch is merged back to main before second branch, then main becomes base for second branch

**Pull Request (PR)**

* NOT the same as a Pull; this notifies others that you’ve pushed changes to a branch in a repo
* Allows discussion and review of changes prior to merging into the main/base branch
  + *Others can review proposed changes, add comments, and contribute/add commits*
* Can push commits to existing pull request; appear in chronological order



*Image from* [*https://training.github.com/downloads/github-git-cheat-sheet.pdf*](https://training.github.com/downloads/github-git-cheat-sheet.pdf)

**Git Commands for Command Line**

git fetch

* get all of the branches/updates from remote repo, doesn’t change working directory (pull part 1)

git merge

* takes all changes from remote repo and updates your working directory (pull part 2)

git push

* take local changes and sends to/updates remote repo

git pull

* fetches and merges all changes on remote repo to local repo and working directory

git status

* shows modified files in the working directory that have been staged for next commit

git add

* stages all changes to get ready for a commit

git commit -m “[description of what you did]”

* saves all changes from staged files into a point to which you can go back to; a checkpoint

*GIT FLOW DIAGRAM*

*REMOTE CLONE REPO*

***INBOUND FLOW from remote to local***

***OUTBOUND FLOW from local to remote***

*\*Note: The local repos are the same, just duplicated to make space*

*Fork*

*Commit*

.git index

*Stage*

\*LOCAL REPO (.git)

*Push*

*Merge*

*Pull*

*Fetch*

WORKING DIRECTORY (FOLDER)

*Clone*

*REMOTE*

*REPO*

\*LOCAL REPO (.git)