Git Commands

Init : cloning existing directory

Add: moves changes from the working directory to stagin area

Status: see ther state of your working directoy and the staged snapshot of your changes

Commit: takes your stages snapshot of changes and commits them to the project

Reset: undoes changes that you’ve made to the files in your working directory

Log : enables you the prose previous changes

Branch: create isolated environment within your repository

Checkout: see and change existing branches

Merge: put everything back together

Tutorials

Creating a repository

1, in the upper right corner use the + drop down menu to select repository

2, fill in name, description and choose visibility

3, select initialize this repository with a README

4, Click Create Repository

Committing changes

1, preview changes

2, describe the change yyou made to the file

3, choose wither existing branch or new branch

4, Click propose file changes

Creating new files

1. On github navigate to the main page of the repository
2. Brose to the folder where you want to create a file
3. Above filkes list click the add file drop down then create new file
4. In the file name type name and use / for directory separator
5. On edit new file tab add content to the file
6. Click preview
7. Describe changes you made
8. Click email address in ther dropdown menue
9. Choose a branch
10. Click propose new file

Moving a file

1. Browse to the file you want to move
2. In upper right corner of the file view click the pencil to edit
3. In the filename firel change the name of the file using these guidelines

To move the file **into a subfolder**, type the name of the folder you want, followed by /. Your new folder name becomes a new item in the navigation breadcrumbs.

To move the file into a directory **above the file's current location**, place your cursor at the beginning of the filename field, then either type ../ to jump up one full directory level, or type the backspace key to edit the parent folder's name.

1. Add a commit message that describes the changes you made
2. Choose branch and create pull request
3. Propose changes to the file

Move a file using command line

1 move the file to a new location within the directory that was created locally on your comp when you cloned the repository

2.open git bash

3 use git status to check the old and new file locstions

4. stage the file for commit to your local repository this will git rm the file from old locationa dn add or git add the file to the new locaitojn

$ git add .

1. # Adds the file to your local repository and stages it for commit.
2. # To unstage a file, use 'git reset HEAD YOUR-FILE'.

5.Use git status to check the changes staged for commit $ git status

> # On branch *your-branch*

> # Changes to be committed:

> # (use "git reset HEAD ..." to unstage)

> #

> # renamed: /old-folder/image.png -> /new-folder/image.png

# Displays the changes staged for commit

6. commit the file that you’ve staged in your local repository $ git commit -m "Move file to new directory"

# Commits the tracked changes and prepares them to be pushed to a remote repository.

# To remove this commit and modify the file, use 'git reset --soft HEAD~1' and commit and add the file again.

7. push the changes in your local reoisitory ($ git push origin *your-branch*

# Pushes the changes in your local repository up to the remote repository you specified as the origin0)

Forking a Repository

1. Open github and navigate to the repository
2. In top right corner clock fork

To clone forked repository

1. Navigate to the fork from the spoon-knife repository
2. Above the list of file click the code button
3. To clone the repository using HTTPS, under "Clone with HTTPS", click . To clone the repository using an SSH key, including a certificate issued by your organization's SSH certificate authority, click **Use SSH**, then click . To clone a repository using GitHub CLI, click **Use GitHub CLI**, then click .
4. Open git bash
5. Change current working directory to the location where you want the clones directory
6. Type git clone and then paste the url you copied earlier EX($ git clone https://github.com/your-username/spoon-knife) and press enter

Configure git to sync your fork with the original repository

1. On github navigate to the cotocat/Spoon-Knife repository
2. a bove the list of files, click  **Code**.
3. To clone the repository using HTTPS, under "Clone with HTTPS", click . To clone the repository using an SSH key, including a certificate issued by your organization's SSH certificate authority, click **Use SSH**, then click . To clone a repository using GitHub CLI, click **Use GitHub CLI**, then click .
4. Open Git Bash.
5. Change directories to the location of the fork you cloned.
   1. To go to your home directory, type just cd with no other text.
   2. To list the files and folders in your current directory, type ls.
   3. To go into one of your listed directories, type cd your\_listed\_directory.
   4. To go up one directory, type cd ...
6. Type git remote -v and press **Enter**. You will see the current configured remote repository for your fork.

$ git remote -v

> origin https://github.com/*YOUR\_USERNAME*/*YOUR\_FORK*.git (fetch)

> origin https://github.com/*YOUR\_USERNAME*/*YOUR\_FORK*.git (push)

1. Type git remote add upstream, and then paste the URL you copied in Step 3 and press **Enter**. It will look like this:

$ git remote add upstream https://github.com/octocat/Spoon-Knife.git

1. To To verify the new upstream repository you have specified for your fork, type git remote -v again. You should see the URL for your fork as origin, and the URL for the original repository as upstream.