By Chris Maltais - Email any questions to chris.maltais@queensu.ca

Git Cheatsheet

Initializing your local repository and linking to GitHub

1. Change directory to the where you want the project setup (Documents, QMIND folder, etc)
    a. `cd ~/documents/Queens/QMIND`
2. Create a a new project directory and change into it
    a. `mkdir project-name && cd $_`
3. Initialize a local git instance in this folder
    a. `git init`
4. Connect this local instance to a remote Bitbucket repository
    a. `git remote add origin https://github.com/QMIND-Team/your-project-here.git`
    b. **Tip:** "origin" is the name of the connection you're creating between your local git repository and the GitHub repo
5. Check the connection was created
    a. `git remote -v`
6. Pull the files from the GitHub repo to your local folder
    a. `git pull origin master`
    b. **Tip**: Pulling is just copying the contents of the remote repo to your own local folder

By Chris Maltais - Email any questions to chris.maltais@queensu.ca

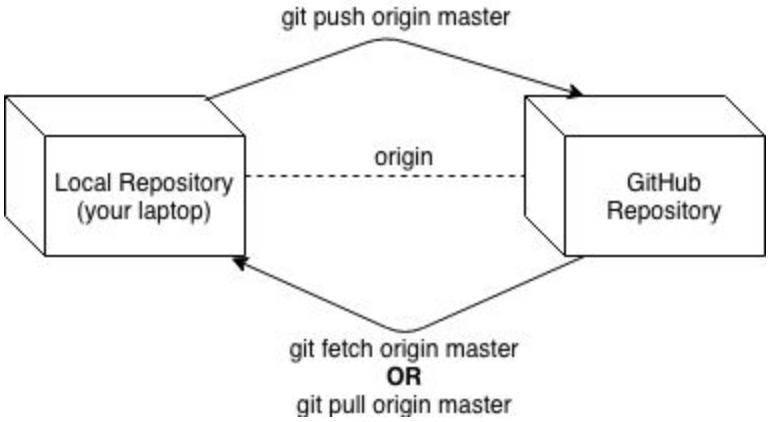
Git workflow using only master branch (good for warm-up projects)

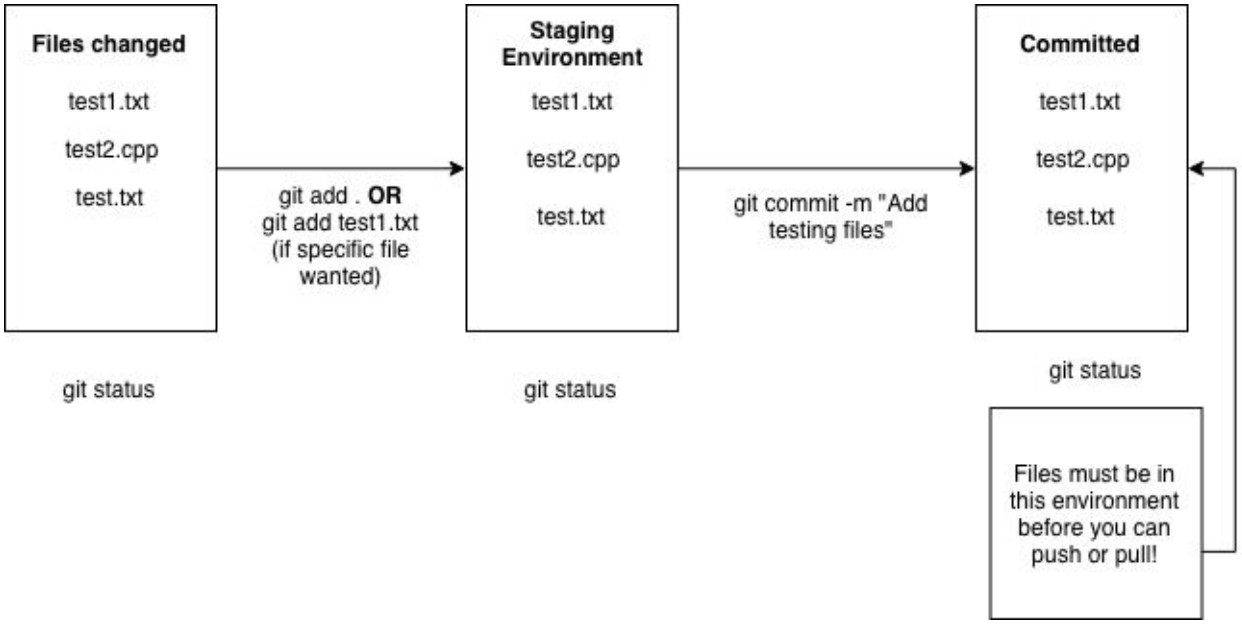
1. Every time you begin development, fetch the latest changes/data from GitHub
    a. `git fetch origin master`
2. Create changes, improvements, features
3. Look at the files you've changed
    a. `git status`
    b. **Tip:** This is one of the most useful commands! Figure out if your files are staged or committed yet!
    c. **Tip:** Can (and should) be used anywhere in this process (repeatedly)
4. When your changes are ready, add them to the staging environment
    a. For all files:
        i. `git add .`
    b. For specific files:
        i. `git add filename.py`
5. When you want to "commit" (i.e. save) the current version you're working on
    a. `git commit -m "Present tense message about what changes I've made here"`
6. When you want to push your code up to the Bitbucket repository for everyone to see
    a. `git push origin master`
    b. **Tip:** You may get errors here! Read error message carefully.
    c. **Tip:** You may need to fetch changes again, someone may have pushed a change in the time you were developing!
    d. **Tip:** You may get a merge conflict here, that means you edited the same line as someone else. No worries! Just open up that file and find the lines with the conflict. You can manually edit the changes, then push the file to GitHub again. It looks something like:

```
i. <<<<< HEAD
ii.   Local repo changes are here
iii.  ==============
iv.   Remote repo changes are here
v.  TAIL >>>>>>>>>>
```

By Chris Maltais - Email any questions to chris.maltais@queensu.ca

Visualizations to help understand:





By Chris Maltais - Email any questions to chris.maltais@queensu.ca