Remotes in GitHub

1. on GitHub: create a GitHub repository **planets**

queenqueen89 / planets

click “Add a README file”

click “Add .gitignore” (if needed)

copy/paste the git link: https://github.com/queenqueen89/planets

2. on terminal: push the files from computer to GitHub

git remote add origin https://github.com/queenqueen89/planets.git

check remote status

git remote -v

it prints out:

origin https://github.com/queenqueen89/planets.git (fetch)

origin https://github.com/queenqueen89/planets.git (push)

now, push the file to GitHub:

git push origin master

it prints out:

remote:

To https://github.com/queenqueen89/planets.git

Check whatever on GitHub is same on computer:

\* it’s good to use “pull” before you “push” something to GitHub (important for collaboration)

git pull origin master

It prints out:

Already up to date.

Quiz

**Question 1**: Browse to your planets repository on GitHub. Under the Code tab, find and click on the text that says “XX commits” (where “XX” is some number). Hover over, and click on, the three buttons to the right of each commit. What information can you gather/explore from these buttons? How would you get that same information in the shell?

**Answer**:

The first button gives the full ID of the commit.

In the shell, type git log to find the same info.

The second button gives the changes that were made in the commit.

In the shell, type git diff to find the same info.

The third button shows you all files in the repository at the time of the commit.

In the shell, type git checkout ID (the identifier of the commit).

**Question 2**: In this lesson, we introduced the “git push” command. How is “git push” different from “git commit”?

**Answer**: git commit only saves changes to your local computer, while git push saves all changes to GitHub.

Move back to master branch:

git checkout master