

Introduction to Git and GitHub

(a.k.a. How to take and submit Assessment #1)

Objectives

- Explain Git, and its common commands:
 - `clone`
 - `add`
 - `commit`
 - `push`
- Explain GitHub, and its common commands:
 - `fork`
 - `pull request`
- Take the first assessment (no stress! just a baseline for us) by:
 1. On Github, **forking** the [assessment repo](#) to your own account
 2. **cloning** it locally to your laptop
 3. working on it (there are two parts - a python part and a math part)
 4. **adding** and **committing** the two modified files ([assessment.py](#) and [math_assessment.txt](#))
 5. **pushing** those files up to your GitHub account
 6. issuing a **pull request** so that we instructors can see your answers

Preamble

On your laptop
is Git installed?
how about Python?
you have a text editor?

Do you have a GitHub account?

If you don't, please raise your hand and the instructional staff will help you out.
You need these things for the first assessment.

Git

- Git is open-source, distributed version control software that lets you keep track of file changes in a repository or folder
 - It runs locally on your computer.
 - Checkpoints (commits) keep track of what was changed, and by whom and when.
 - You can load an earlier checkpoint to reverse changes.
 - The distributed aspect of Git allows teams to collaborate on projects.
- Common git commands
 - **init** - initializes a repository as a Git tracked repository (in DSI we don't do this much)
 - **clone** - makes a copy of a Git tracked repository in a new location (in DSI we do this a lot)
 - **add** - adds files & folders to a staging area where their modifications are set to be tracked
 - **commit** - incorporates the modifications in the staging area into a new project checkpoint
- Typical workflow in the DSI:
 - **fork*** -> **clone** -> modify some files -> **add** -> **commit** -> **push***

(* integration with GitHub, will talk about these in a bit)

Git example

```
Terminal File Edit View Search Terminal Help
1 frank@frank-Y700:~/galvanize$ git clone https://github.com/Frank-W-B/dsi-assessment-day1.git
Cloning into 'dsi-assessment-day1'...
2 Username for 'https://github.com': Frank-W-B
Password for 'https://Frank-W-B@github.com': 
remote: Counting objects: 135, done.
remote: Total 135 (delta 0), reused 0 (delta 0), pack-reused 135
Receiving objects: 100% (135/135), 758.71 KiB | 1.17 MiB/s, done.
Resolving deltas: 100% (55/55), done.
Checking connectivity... done.
3 frank@frank-Y700:~/galvanize$ cd dsi-assessment-day1/
✓ ~/galvanize/dsi-assessment-day1 [master|✓]
4 21:44 $ cd src/
✓ ~/galvanize/dsi-assessment-day1/src [master|✓]
5 21:45 $ nano assessment.py
```

- 1 `git clone` my existing git repo (on Github) into my galvanize folder (doesn't have to be galvanize)
- 2 Enter my Github account username and password (note that password doesn't show up as you type)
- 3 Get inside the cloned directory. In my terminal my prompt changes when I'm in a git repo.
- 4 Change into the src directory. I know that [assessment.py](#) is there from looking at file structure earlier.
- 5 I'm going to edit the `assessment.py` file using the **nano** text editor (you could use atom, vim, emacs etc.)

Git example

```
Terminal File Edit View Search Terminal Help
GNU nano 2.5.3      File: assessment.py      Modified

# fake question below
def return_the_square_if_even(lst):
    '''
    INPUT: list of integers
    OUTPUT: list of integers

    Return a list with the square of each element in the list if the
    element is an even number. If the input list is empty return None.
    '''
    pass
```

... do some work ...

```
Terminal File Edit View Search Terminal Help
GNU nano 2.5.3      File: assessment.py      Modified

# fake question below
def return_the_square_if_even(lst):
    '''
    INPUT: list of integers
    OUTPUT: list of integers

    Return a list with the square of each element in the list if the
    element is an even number. If the input list is empty return None.
    '''
    if lst:
        return [elem**2 for elem in lst if elem % 2 == 0]
    return None
```

Git example

⑥

```
~/galvanize/dsi-assessment-day1/src [master|+1]
22:57 $ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)
```

```
    modified:   assessment.py
```

```
no changes added to commit (use "git add" and/or "git commit -a")
```

⑦

```
~/galvanize/dsi-assessment-day1/src [master|+1]
22:58 $ git add assessment.py
```

⑧

```
~/galvanize/dsi-assessment-day1/src [master|●1]
22:58 $ git status
```

```
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
```

```
    modified:   assessment.py
```

⑨

```
~/galvanize/dsi-assessment-day1/src [master|●1]
22:58 $ git commit -m "Finished python portion"
```

```
[master 64eeb10] Finished python portion
1 file changed, 17 insertions(+), 231 deletions(-)
rewrite src/assessment.py (99%)
```

⑩

```
~/galvanize/dsi-assessment-day1/src [master ↑1|✓]
```

```
22:59 $ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
```

⑥ Use **git status** to check the status of the repository. Git knows **assessment.py** has been modified, but the changes are not staged for commit.

⑦ **git add** **assessment.py** to the staging area

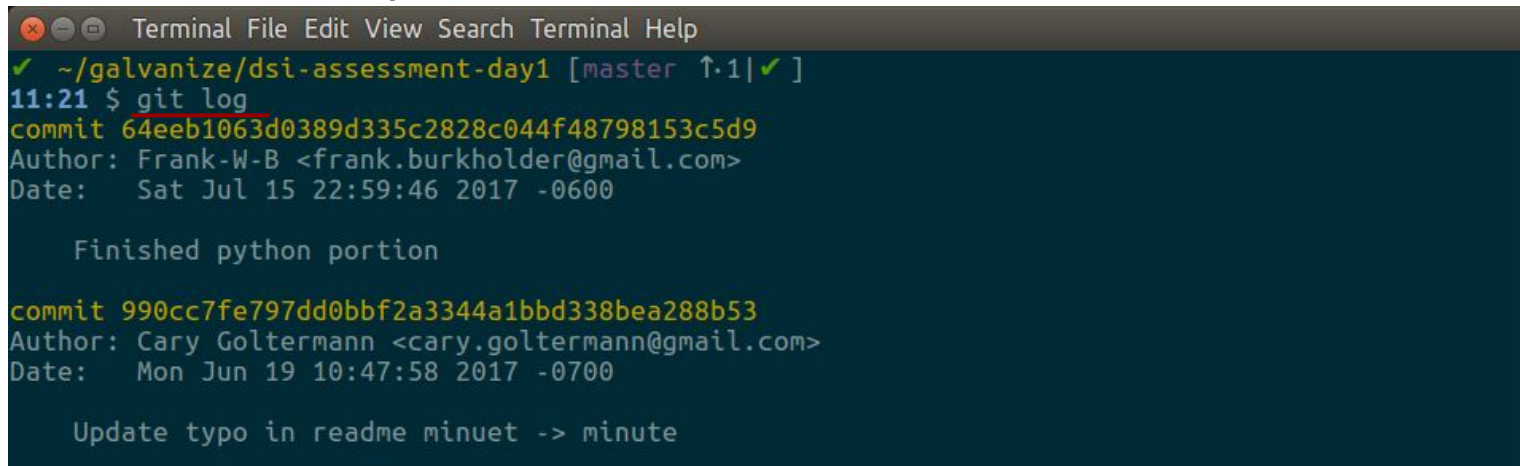
⑧ **git status** now shows there are changes to be committed

⑨ **git commit** the changes and add an explanatory message

⑩ **git status** now shows that there is nothing to commit, but we are ahead of our 'remote' and we should **git push** to fix that....

Git example

To view the checkpoints use `git log`



```
Terminal File Edit View Search Terminal Help
~/galvanize/dsi-assessment-day1 [master ↑1|✓]
11:21 $ git log
commit 64eeb1063d0389d335c2828c044f48798153c5d9
Author: Frank-W-B <frank.burkholder@gmail.com>
Date: Sat Jul 15 22:59:46 2017 -0600

    Finished python portion

commit 990cc7fe797dd0bbf2a3344a1bbd338bea288b53
Author: Cary Goltermann <cary.goltermann@gmail.com>
Date: Mon Jun 19 10:47:58 2017 -0700

    Update typo in readme minuet -> minute
```

- ⑪ The most recent commit - note the commit identifier **64eeb10...** and the commit message “Finished..”
- ⑫ The penultimate commit by Cary where he fixes a typo in the README. Good commit message.

This is enough Git for now, on to its interaction with GitHub...

GitHub

- GitHub is a web hosting service for Git repositories (repos).
 - Github's mascot: the Octocat
 - It's "in the cloud."
 - Git exists independently of GitHub, while the converse is not true.
 - There are other websites that serve as Git hosting services, e.g. Bit Bucket.
 - As of April 2017, GitHub reports having almost 20 million users and 57 million repositories, making it the largest host of source code in the world, according to Wikipedia.
 - Github repos often serve as **remotes**:
 - A remote is a shared Git repository that allows multiple collaborators to work on the same Git project from different locations.
 - Galvanize's gSchool (<https://github.com/gSchool>) is where you will find DSI repos.
 - Public repos are free, but private repos you have to pay for.
 - Any repo that you **fork** (more in a bit) from gSchool will remain a private repo.
 - **Your GitHub account is the public face of your code and documentation and is instrumental to finding employment as a Data Scientist.**



GitHub

- What you will be doing mostly on GitHub:
 - **fork** (button): Creates your own copy (in your GitHub account) of another GitHub repository.

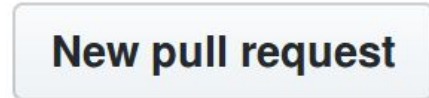


- **clone** (button): Copies the URL of your GitHub repository to the clipboard so that you can paste it into Terminal and `git clone` it with Git. **Clone, don't download.**



- **pull request** (button): **ONLY FOR ASSESSMENTS, when you are finished.** Asking permission of the *upstream* repository (where you forked your GH repository from, i.e. gSchool) to merge changes that exist in your GH repository.

1)



2)



Git & GitHub Interaction

- **clone:**

- After pressing the  button on GitHub, you `git clone` in Terminal:

```
Terminal File Edit View Search Terminal Help
frank@frank-Y700:~/galvanize$ git clone https://github.com/Frank-W-B/dsi-assessment-day1.git
```

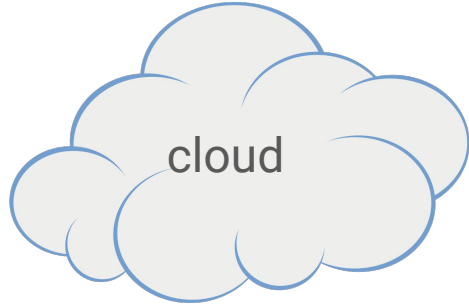
- **push:**

- You work on your files locally. After `git add` adding them to the staging area, then `git commit` committing them to a checkpoint, you `git push` to your remote *origin* (your GitHub repo) your *master* branch*. (Use `git remote -v` to check the *origin* of your local repo.)

```
Terminal File Edit View Search Terminal Help
✓ ~/galvanize/dsi-assessment-day1 [master|+2]
13:29 $ git remote -v
origin https://github.com/Frank-W-B/dsi-assessment-day1.git (fetch)
origin https://github.com/Frank-W-B/dsi-assessment-day1.git (push)
✓ ~/galvanize/dsi-assessment-day1 [master|+2]
13:29 $ git add math/math_assessment.txt src/assessment.py
✓ ~/galvanize/dsi-assessment-day1 [master|●2]
13:29 $ git commit -m "Completed math and python portions"
[master 00a3744] Completed math and python portions
 2 files changed, 2 insertions(+), 2 deletions(-)
✓ ~/galvanize/dsi-assessment-day1 [master ↑1|✓]
13:30 $ git push origin master
```

*master is the default branch, not going to talk about branches right now.

Git and GitHub schematic

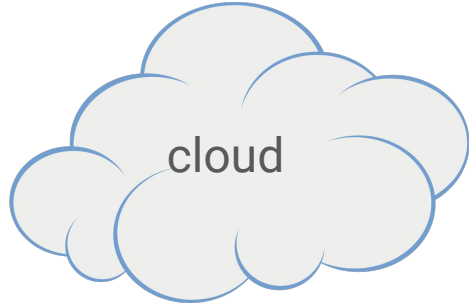


local



your laptop

Git and GitHub schematic



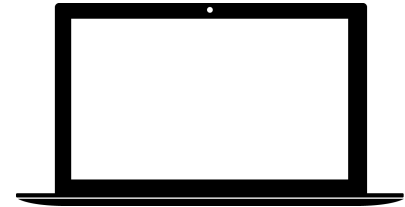
GitHub

gSchool

GitHub

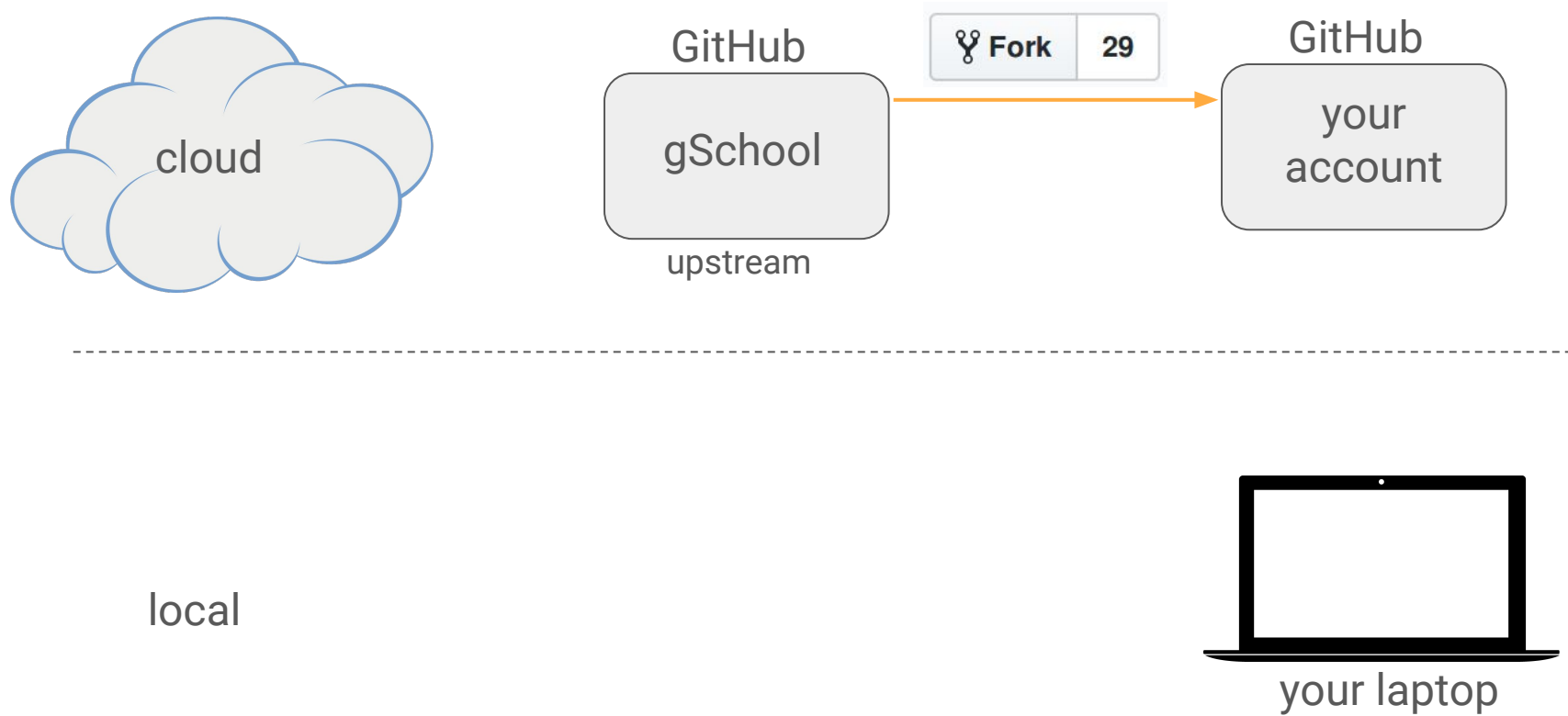
your
account

local

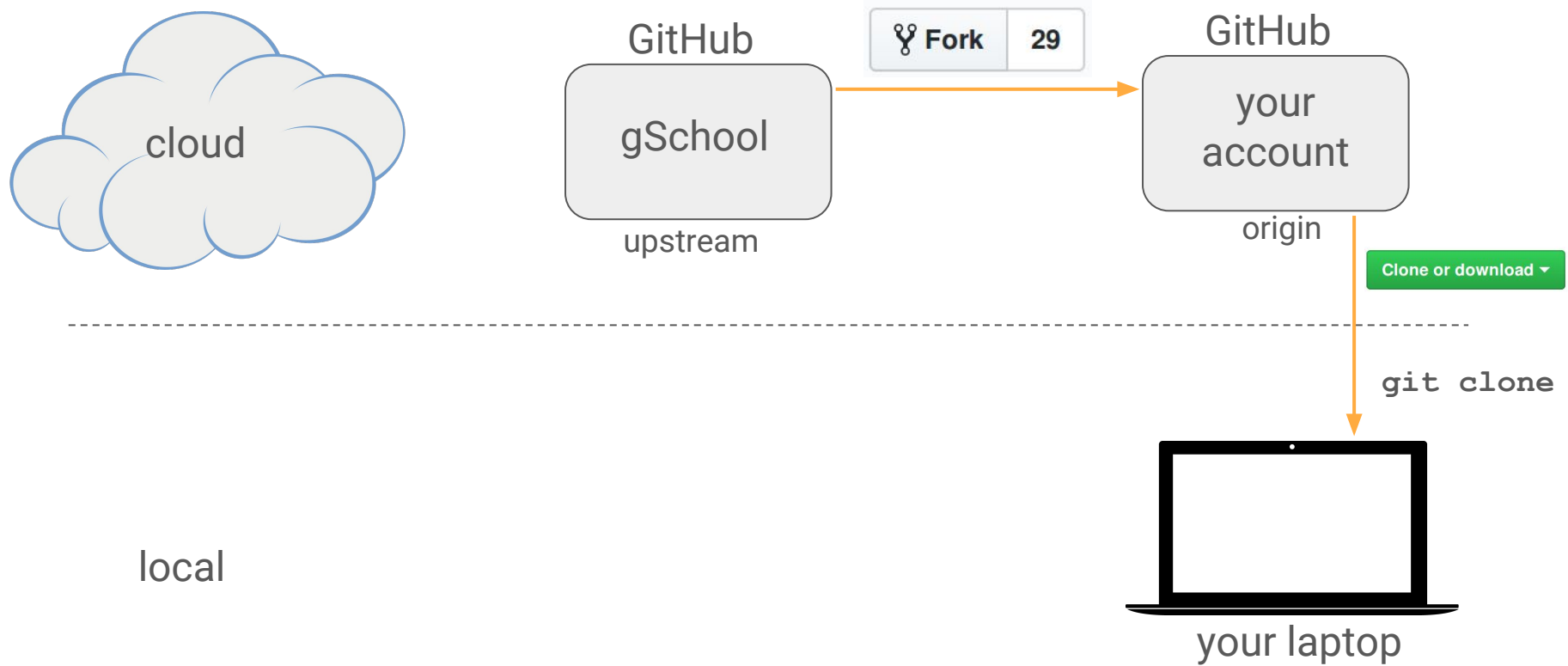


your laptop

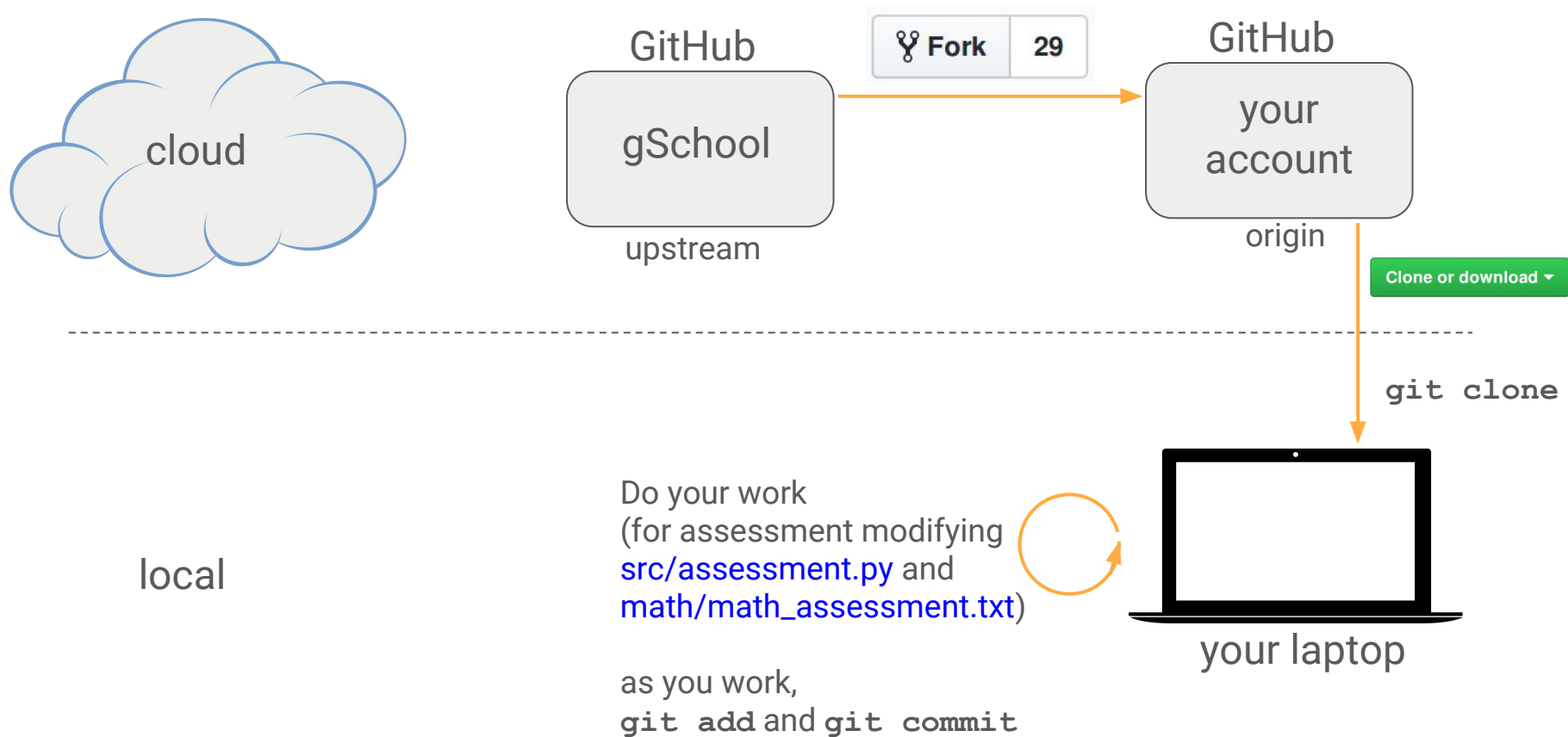
Git and GitHub schematic



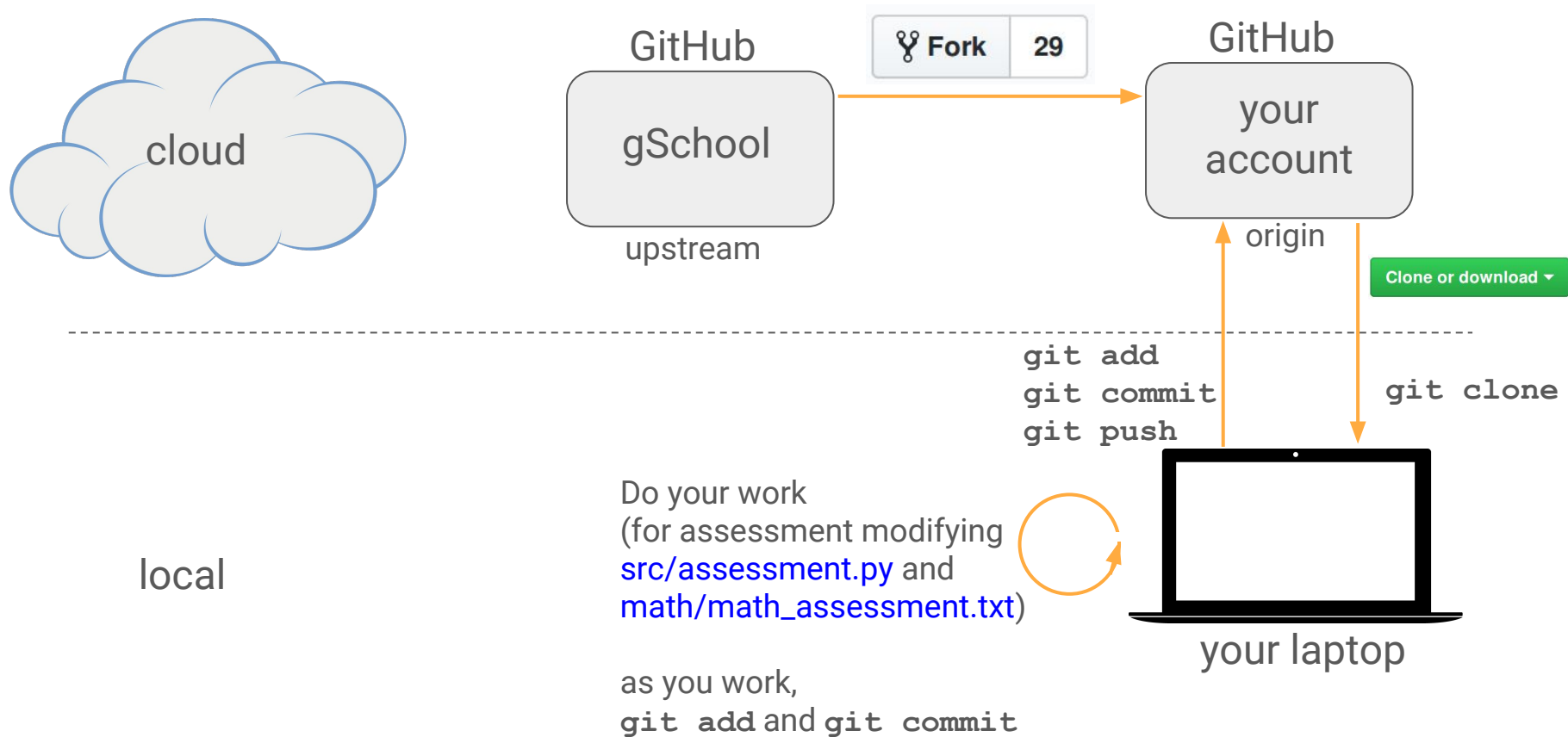
Git and GitHub schematic



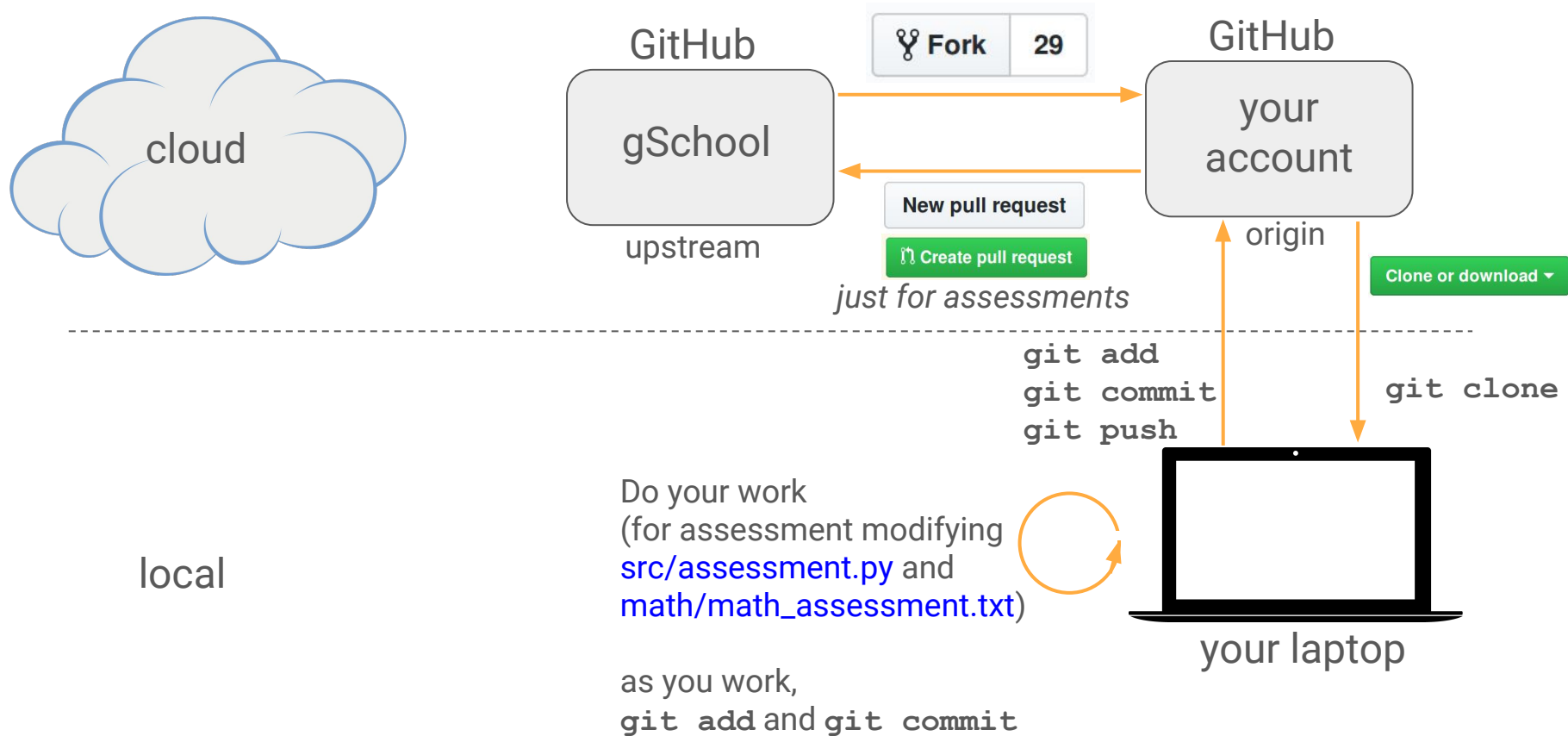
Git and GitHub schematic







Git and GitHub schematic



Git and GitHub schematic



Now go take Assessment #1!

1. Go to <https://github.com/gSchool/dsi-assessment-day1>
2.  Fork 29 to your GitHub account
3.  then `git clone` it locally, check with `git remote -v`
4. work on [assessment.py](#) and [math_assessment.txt](#)
5. `git add`, `git commit`, and then `git push` when done
6.  then  so that we can see your answers
7. Relax