Git/GitHub Workshop

Pitt Computer Science Club
Alex Zharichenko, Business Manager

What are we doing?

1. What & Why
2. Local
3. Remote
4. AND BEYOND!

What is version control?

- It is a system for recording changes to a file or set of files over time
- It allows abilities such as reverting back files/project to a previous state, figure out who to blame for changes and more
- There are many different version control software out there but the main one is git

SUBVERSION

mercurial

Why should we use it?

- Keeping track of changes
- Ability to revert back changes when needed
- Allows easier collaboration between developers
- Allows figuring out who introduced bugs or issues into the code
- Being able to branch off the code and work a part of it and merging it back is nice
- Incredibly fast, secure, and flexible

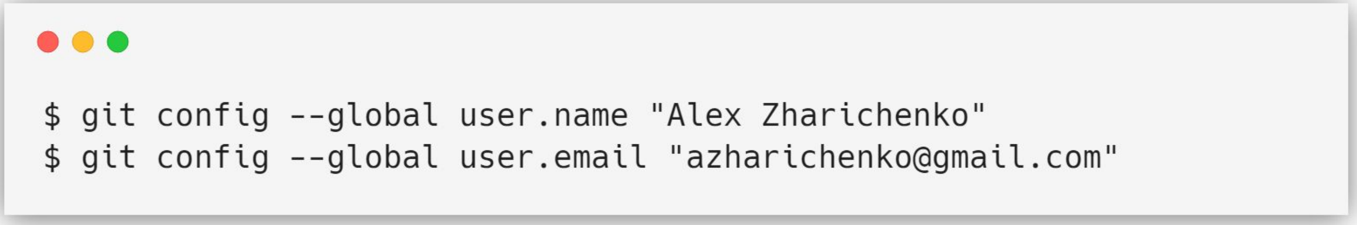
Git

- Is the de facto standard
  - Broadly adopted by many organizations and used frequently
- Was originally developed by Linus Torvalds for the development of the Linux Kernel
- Git is flexible to various development workflows
- Is fast, secure, and flexible
- It has a distributed architecture so that every developer has a working copy of the code with all changes

Let’s do it!

Setting up Git

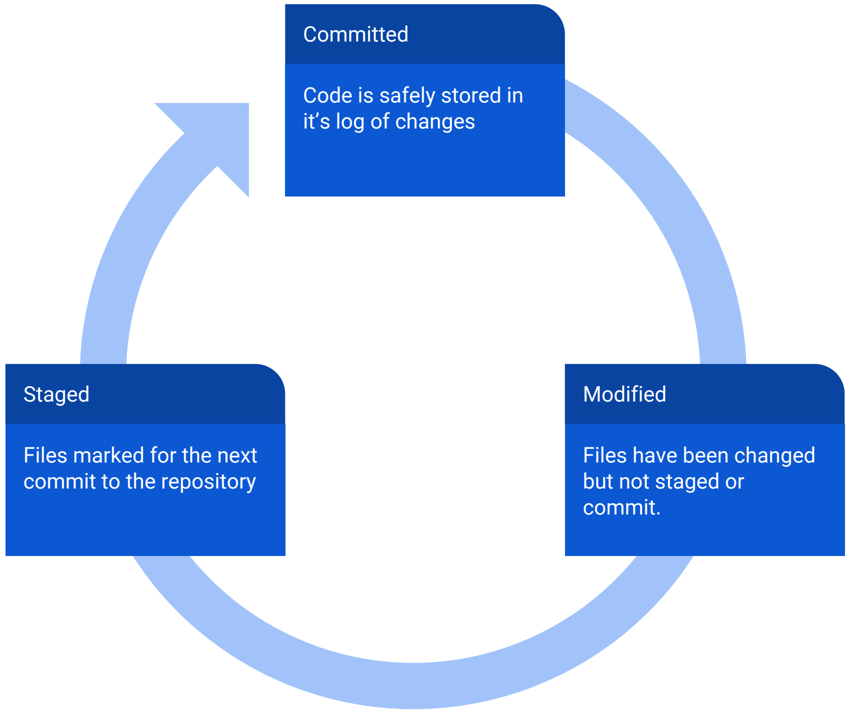
- The basic required configuration that need to be set are name and email
- But there are many other configuration that can be set such as one for signing commits with a PGP key



```
$ git config --global user.name "Alex Zharichenko"
$ git config --global user.email "azharichenko@gmail.com"
```

Stages of Git

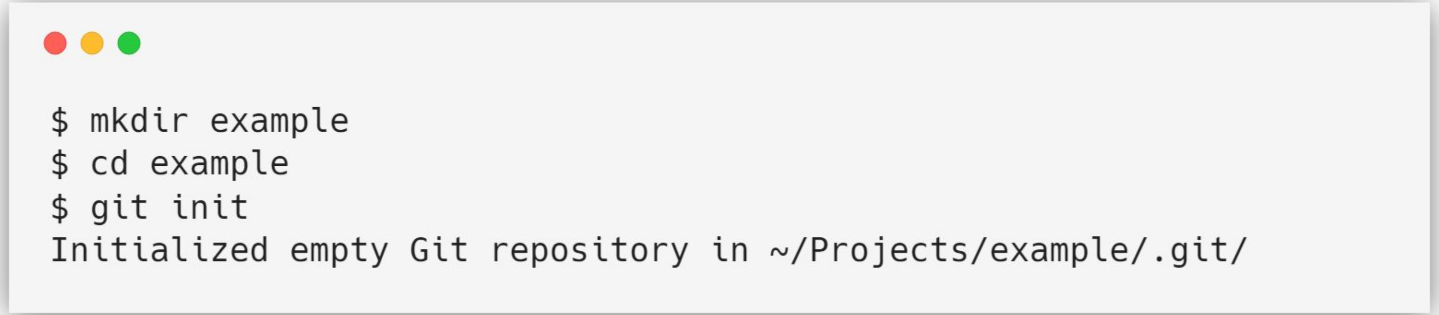
- The process of using git can be split up into three distinct stages


Committed
Code is safely stored in it's log of changes
Staged
Files marked for the next commit to the repository
Modified
Files have been changed but not staged or commit.

git init

git init

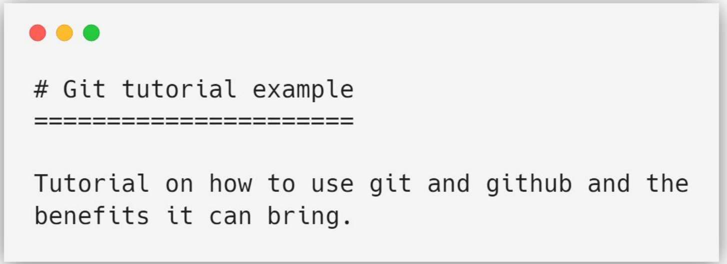
Starts up version control for directory



```
$ mkdir example
$ cd example
$ git init
Initialized empty Git repository in ~/Projects/example/.git/
```

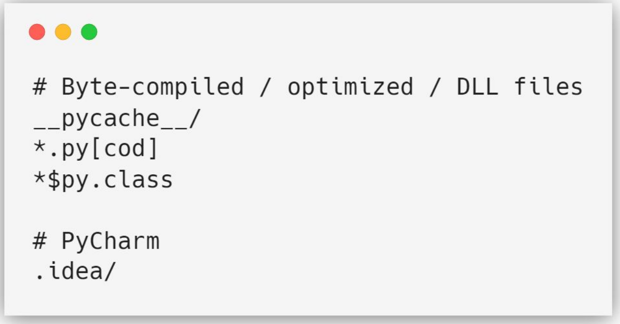
Creating/Adding files into repository

README.md



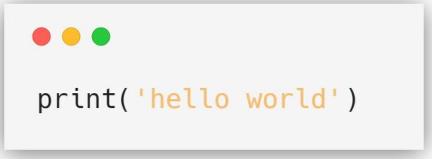
```
# Git tutorial example
======================

Tutorial on how to use git and github and the
benefits it can bring.
```

.gitignore



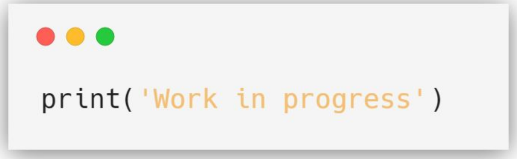
```
# Byte-compiled / optimized / DLL files
__pycache__/
*.py[cod]
*$py.class

# PyCharm
.idea/
```

ready.py



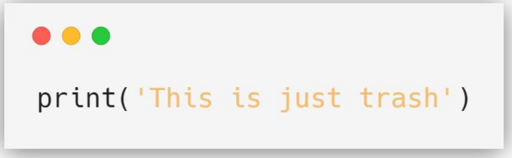
```
print('hello world')
```

work.py



```
print('Work in progress')
```

trash.py

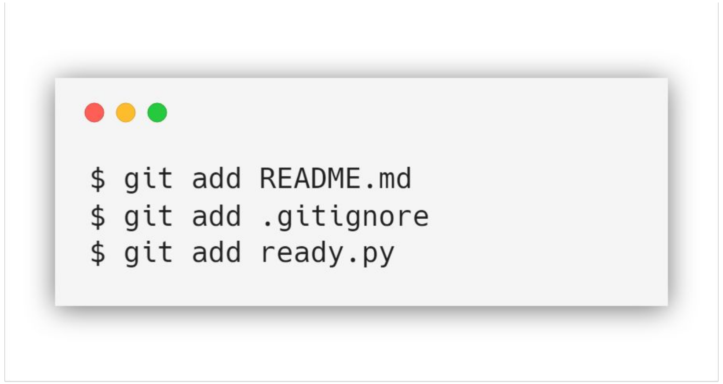


```
print('This is just trash')
```

git add

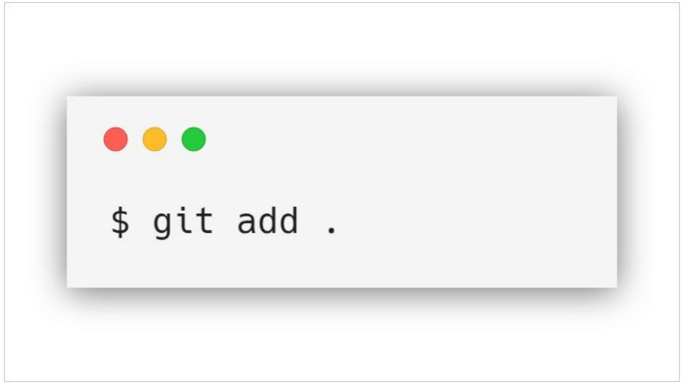
git add [<pathspec>]

Stages file to be committed next



```
$ git add README.md
$ git add .gitignore
$ git add ready.py
```

or



```
$ git add .
```

git status

git status

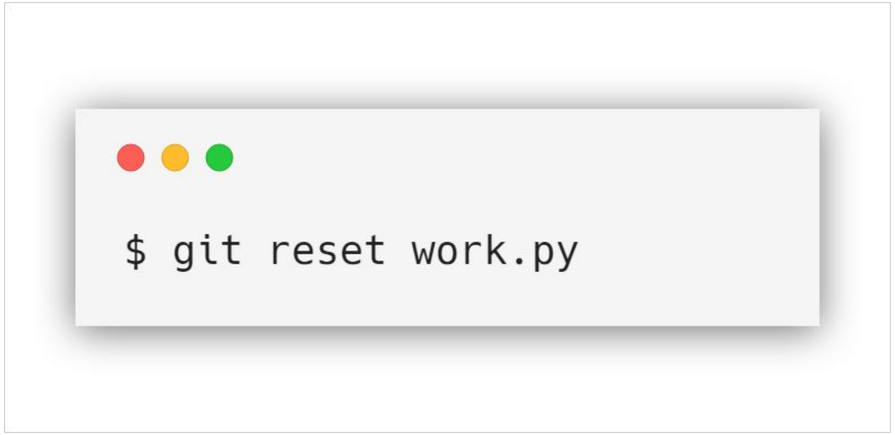
Shows files that have been modified and whether or not they have been staged

```
$ git status
```

git reset

git reset [<paths>]

Unstages file, while keep its contents

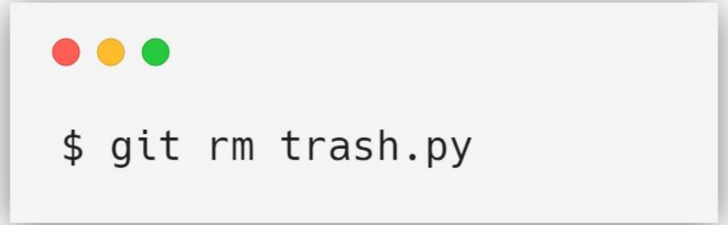


```
$ git reset work.py
```

git rm

git rm [<file>]

Removes file from directory and stage

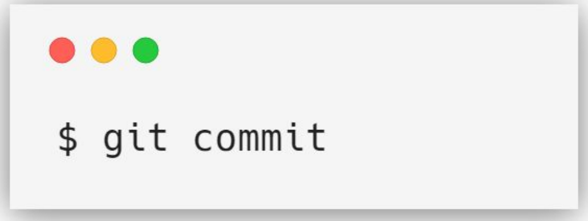


```
$ git rm trash.py
```

git commit

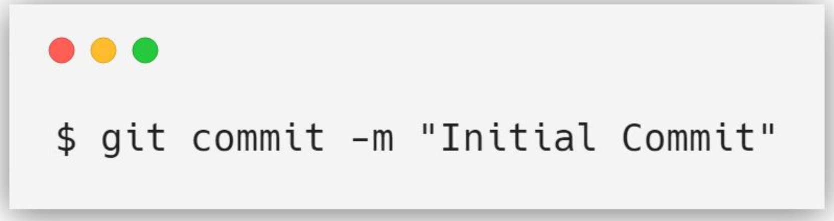
git commit -m [message]

Commits changes to log



```
$ git commit
```

or



```
$ git commit -m "Initial Commit"
```

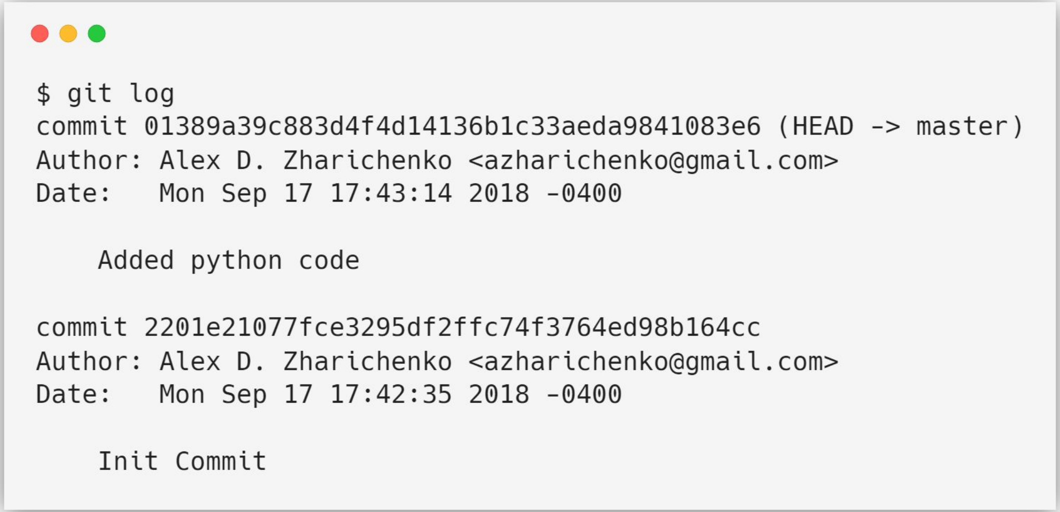
We have made our first commit!

Let’s make another

git log

git log

Shows commit history

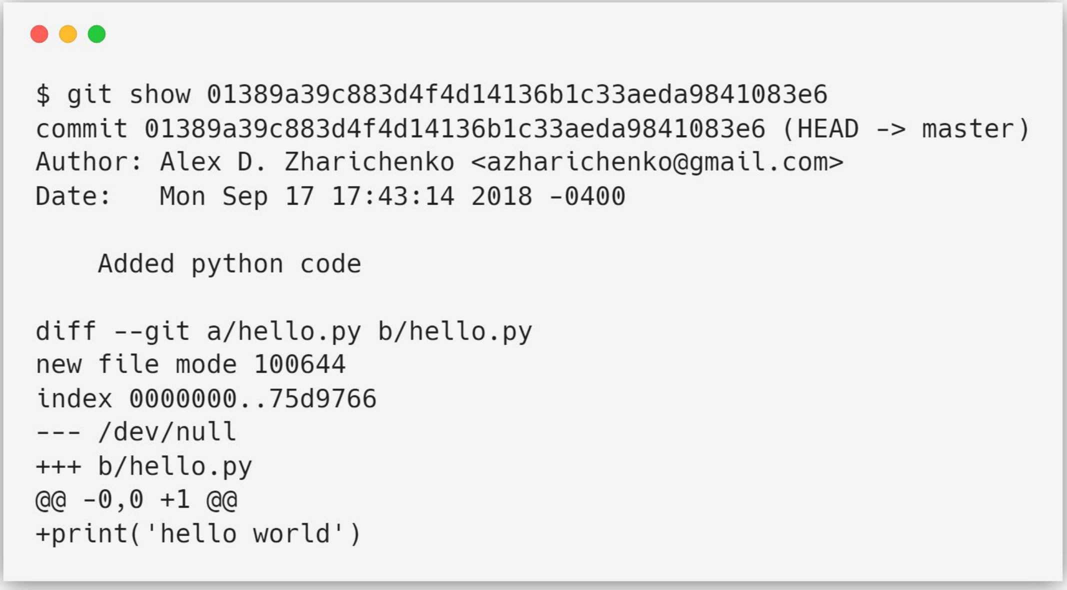


```
$ git log
commit 01389a39c883d4f4d14136b1c33aeda9841083e6 (HEAD -> master)
Author: Alex D. Zharichenko <azharichenko@gmail.com>
Date:   Mon Sep 17 17:43:14 2018 -0400

    Added python code

commit 2201e21077fce3295df2ffc74f3764ed98b164cc
Author: Alex D. Zharichenko <azharichenko@gmail.com>
Date:   Mon Sep 17 17:42:35 2018 -0400

    Init Commit
```

git show

git show [<blob>]

Shows commit

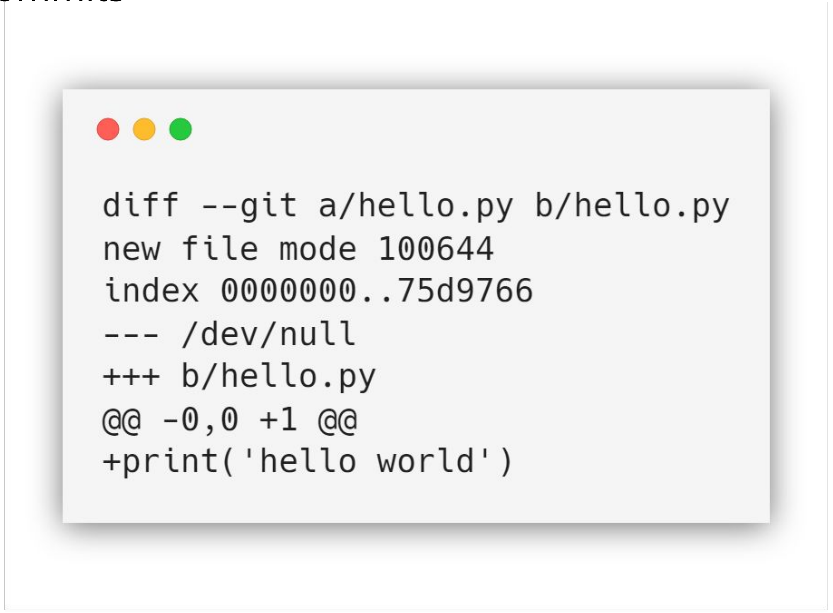


```
$ git show 01389a39c883d4f4d14136b1c33aeda9841083e6
commit 01389a39c883d4f4d14136b1c33aeda9841083e6 (HEAD -> master)
Author: Alex D. Zharichenko <azharichenko@gmail.com>
Date:   Mon Sep 17 17:43:14 2018 -0400

    Added python code

diff --git a/hello.py b/hello.py
new file mode 100644
index 0000000..75d9766
--- /dev/null
+++ b/hello.py
@@ -0,0 +1 @@
+print('hello world')
```

git diff

git diff [<blob>] [<blob>]

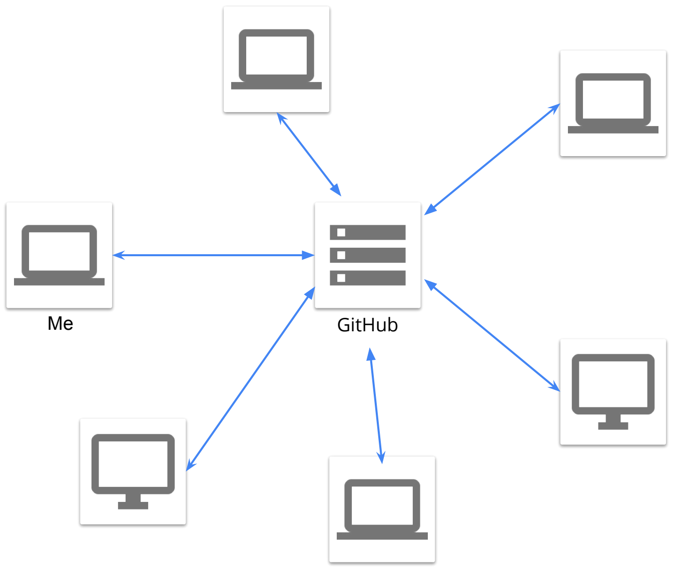
Shows changes between commits



```
diff --git a/hello.py b/hello.py
new file mode 100644
index 0000000..75d9766
--- /dev/null
+++ b/hello.py
@@ -0,0 +1 @@
+print('hello world')
```

How do I collaborate with others?

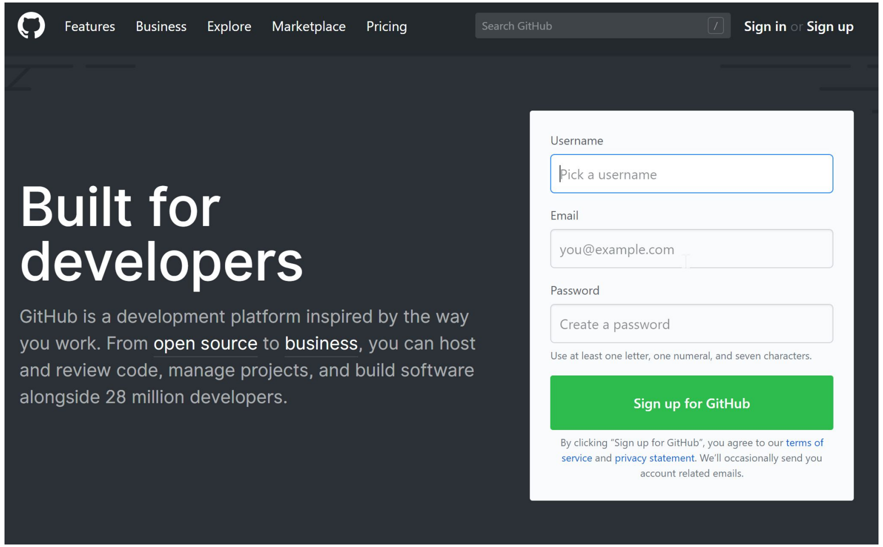
But what happens if my computer breaks?



How can I do this all for free?

To GitHub!

How to get on GitHub



Built for developers

GitHub is a development platform inspired by the way you work. From open source to business, you can host and review code, manage projects, and build software alongside 28 million developers.

Username: Pick a username

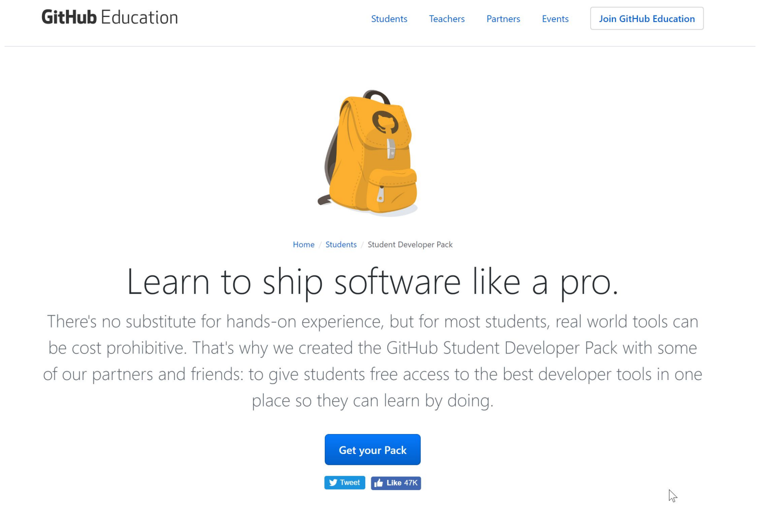
Email: you@example.com

Password: Create a password

Use at least one letter, one numeral, and seven characters.

By clicking "Sign up for GitHub", you agree to our terms of service and privacy statement. We'll occasionally send you account related emails.

github.com



Learn to ship software like a pro.

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

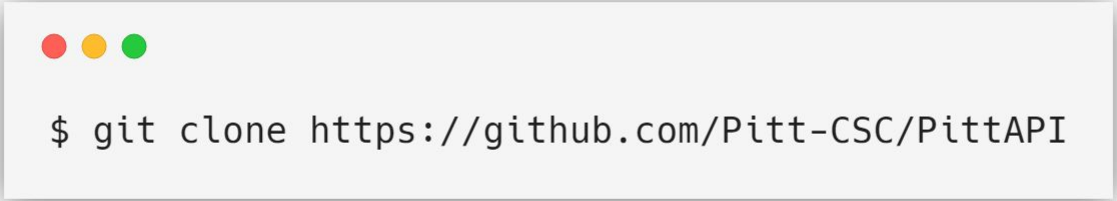
education.github.com/pack

Time for a demonstration

git clone

git clone [url]

Clones repository from remote



```
$ git clone https://github.com/Pitt-CSC/PittAPI
```

git pull

git pull

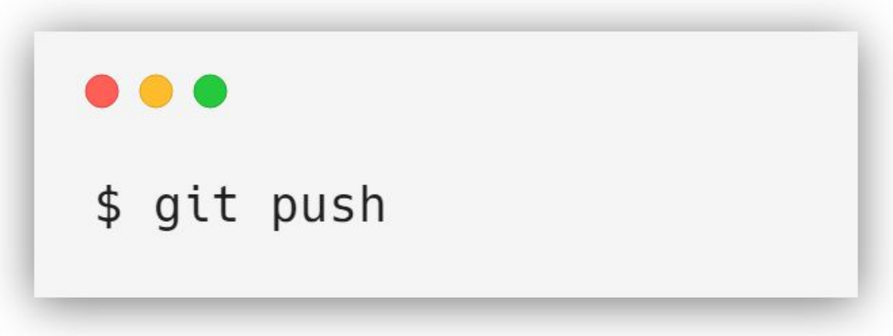
Pull in new commits from remote

```
$ git pull
```

git push

git push

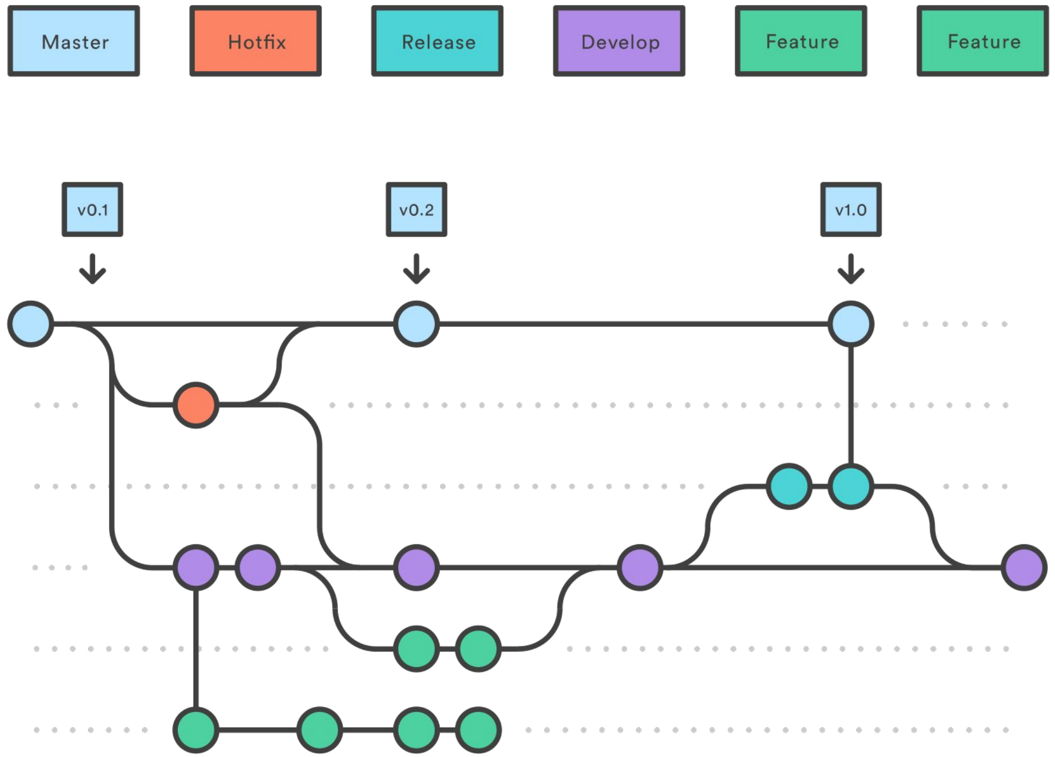
Push your commits to the remote



```
$ git push
```

To Beyond!

Branching



Questions?