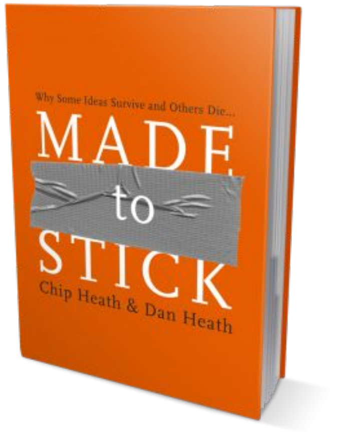
Version control with Git And GitHub

A guide by Patrick Wambua

Intro...

GitHub: https://github.com/patrick-Wambua
LinkedIn:https://www.linkedin.com/in/patrick-wambua/
Twitter:

Find git https://git-scm.org/



1. Overview

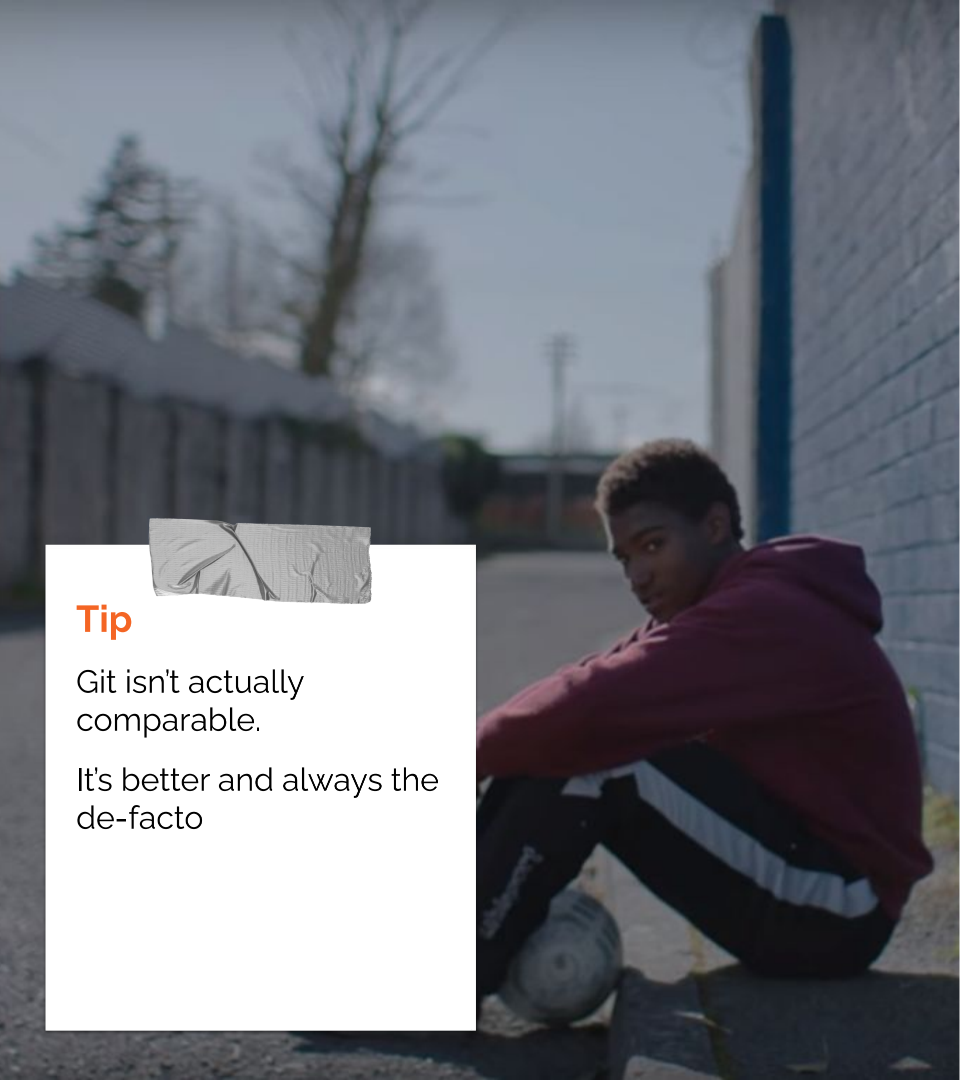
- Git is a powerful and potentially complex tool
- Two goals for the presentation
- 1. Cover core concepts and terminology
- 2. Familiarise with a useful workflow

Git

Git is a version control system: software that manages different versions of files

GIT

- Why is it useful?
- 1. Keep track of changes to files
- 2. Review and revert back to old versions
- 3. Synchronise files between different locations
- 4. Test changes without losing the original copy



Tip

Git isn't actually comparable.

It's better and always the de-facto

Git

Why Git and not some other software?

1. Performance
2. Flexibility
3. Popularity

GitHub

https://github.com/

GitHub...

GitHub is a hosting site for Git repositories

It was founded in 2008 and acts as a container where developers and communities can store code base.

(with lots of extra features)

Tip

Github has other alternatives such as

- ★ **GitLab**
- ★ **BitBucket**
- ★ **Host _your_own**

.

Create an Account with GitHub…

www.github.com

GitHub

Why is GitHub useful?

1. Free hosting for open source projects
2. Interface for browsing and editing code
3. Workflows for collaborating with others
4. API for doing other fancy things

Tip

Why GitHub and not any other:

★ **Popularity**

Let's get started...

Using Git's primary workflow.

Installation of Git and GitHub…

Install and setup...

- http://git-scm.com/download (try this first)
- **Linux:** apt-get install git-core
- **Mac:** http://code.google.com/p/git-osx-installer/
- **Windows:** http://msysgit.github.com/
- Github GUI

Setup...

git config --global user.name “patrick-Wambua”

git config --global user.email “pwnthiwa@gmail.com”

//

This email should be registered in your Github

Line breaks (\r\n in Windows vs. \n in Mac/Linux)

Mac/Linux: git config --global core.autocrlf input

Windows: git config --global core.autocrlf truegit config --global user.name “Charles Liu”

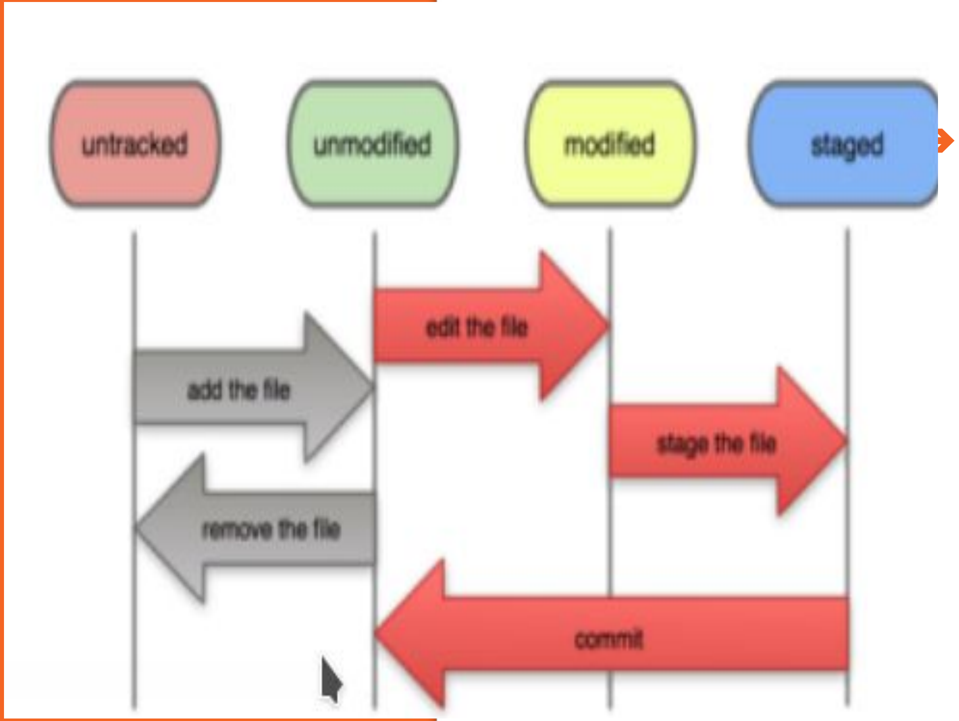
¨ git config --global user.email “cliu2014@mit.edu”

Snapshots-Essentially records what all your files look like at a given point in time
Commit-The act of creating a snapshot.*>>git add>>git commit*
Repo-A collection of all the files and the history of those files
The act of copying a repository from a remote server is called **cloning.**
The process of downloading commits that don't exist on your machine from a remote repository is called **pulling changes**
3 possible states for a file
¤ Changed¤ Staged¤ Committed

Tip

- ★ Basic workflow
- ★ Adding and committing files
- ★ The git log
- ★ The staging area
- ★ Removing files
- ★ Viewing diffs of files
- ★ The .gitignore file

Basic Workflow



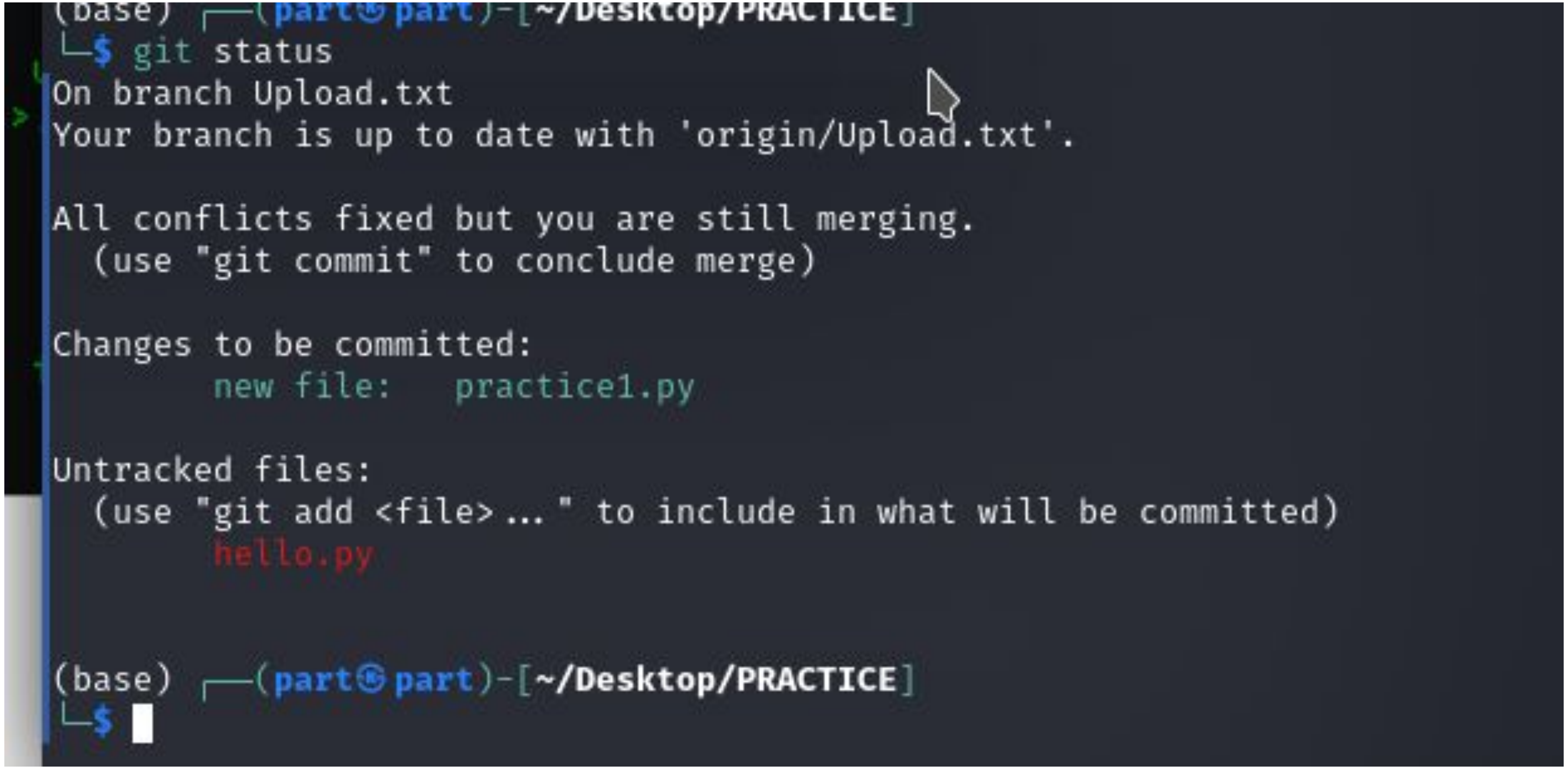
- **git init**
  create git project in existing directory

  ¤ Make Git start to “watch” for changes in the directory

Basic WorkFlow

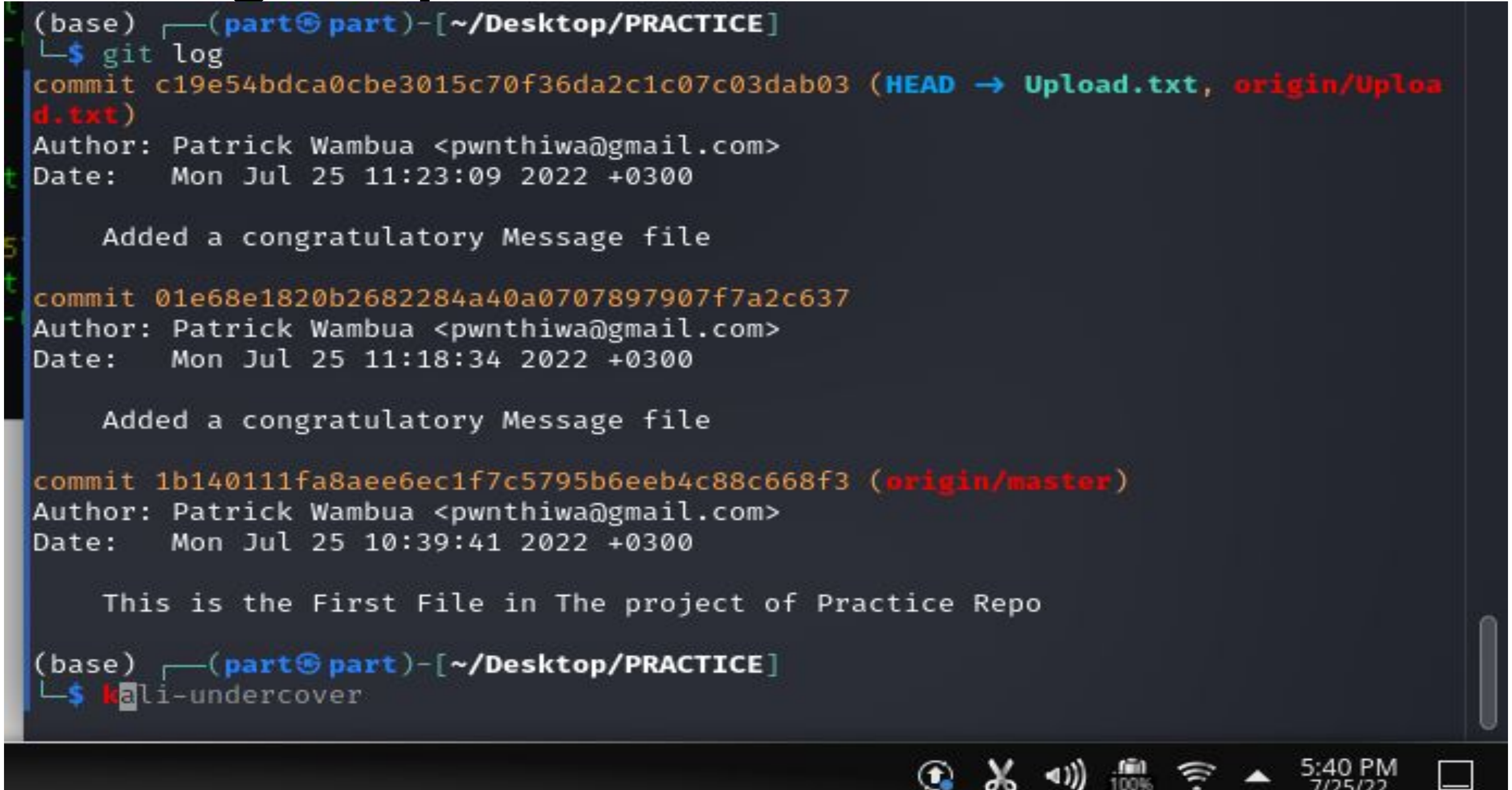
➔ Add files to be committed with git add <filename>
  ¤ Puts the file in the “staging area”
- Create a commit (a “snapshot”) of added files with
- git commit, followed by a commit message
- Use git status to see the current status of your working tree

Git Status output



```
(base) ┌──(part㉿part)-[~/Desktop/PRACTICE]
└─$ git status
On branch Upload.txt
Your branch is up to date with 'origin/Upload.txt'.

All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)

Changes to be committed:
        new file:   practice1.py

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        hello.py


(base) ┌──(part㉿part)-[~/Desktop/PRACTICE]
└─$ 
```

Git log Output



```
(base) ┌──(part㉿part)-[~/Desktop/PRACTICE]
└─$ git log
commit c19e54bdca0cbe3015c70f36da2c1c07c03dab03 (HEAD → Upload.txt, origin/Uploa
d.txt)
Author: Patrick Wambua <pwnthiwa@gmail.com>
Date:   Mon Jul 25 11:23:09 2022 +0300

    Added a congratulatory Message file

commit 01e68e1820b2682284a40a0707897907f7a2c637
Author: Patrick Wambua <pwnthiwa@gmail.com>
Date:   Mon Jul 25 11:18:34 2022 +0300

    Added a congratulatory Message file

commit 1b140111fa8aee6ec1f7c5795b6eeb4c88c668f3 (origin/master)
Author: Patrick Wambua <pwnthiwa@gmail.com>
Date:   Mon Jul 25 10:39:41 2022 +0300

    This is the First File in The project of Practice Repo

(base) ┌──(part㉿part)-[~/Desktop/PRACTICE]
└─$ kali-undercover
```

The Typical workflow...

- Git status

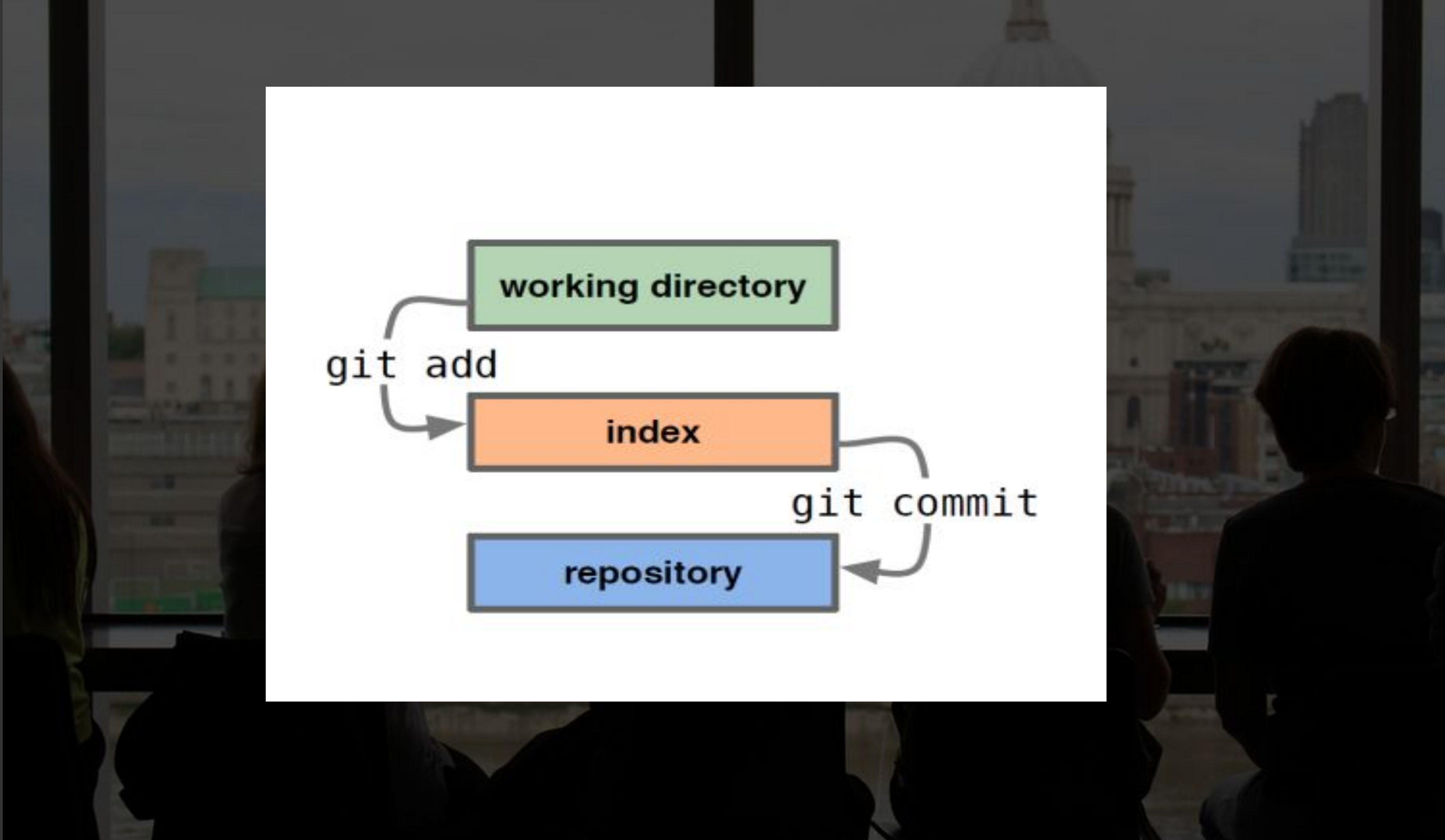
See what Git thinks is going on

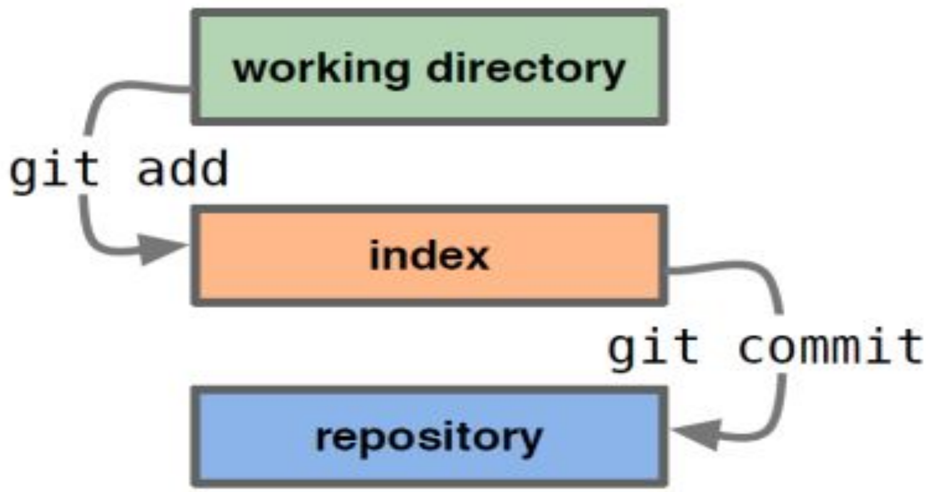
Use this frequently!

- Work on your files

Git add “your editfiles”

- git commit –m “What I did”




working directory
git add
index
git commit
repository

Branching...

- ★ List all branches in the project – ***git branch***
- ★ Create a new branch – ***git branch <branchname>***
- ★ Switch to a branch – ***git checkout <branchname>***
- ★ Create and immediately switch – ***git checkout –b <branchname>***
- ★ Delete a branch – ***git branch –d <branchname>***
- ★ ***The main branch in a project is called the master branch***

GitHub for collaboration

- ★ Creating a repo on Github
- ★ Remotes
- ★ Remote-tracking branches
- ★ Push, fetch, and pull
- ★ The git clone command

Remotes

A target computer that has Git repos that you can access

¤ Via http(s), ssh, or git protocols

git remote add <remotename> <remoteaddress>

git remote –v (view remotes)

git remote rm <remotename>

¨ Often, with one remote, we name it “origin”

Authenticating on GitHub…

Need to generate a keypair:
Basically a personal access Token from GitHub Settings

Pushing and fetching...

- git push <remotename> <branchname> sends your
code in the branch up to the remote
- Often just git push: depends on settings but often
equivalent to git push origin master
- git fetch <remotename>

Additional Resources:

- https://git-scm.com/

Official git site and tutorial:

GitHub guides:

- https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf

Interactive git tutorial:

- https://guides.github.com/

Command cheatsheet:

- https://git-scm.com/
- https://try.github.io/levels/1/challenges/1

Visual/interactive cheatsheet:

- http://ndpsoftware.com/git-cheatsheet.html