git
versions & collaborations

basics

Learning Objectives

- Understand what `git` is and where it is used
- Setup new repo on github account
- Practice local and remote version management

git basics

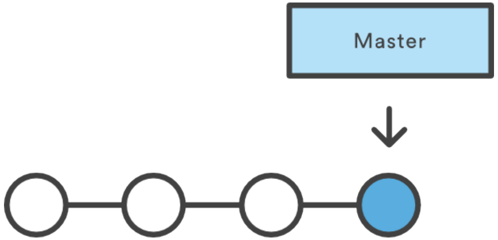
what is git?

version control

- avoids 'indexv1.html, indexv2.html, indexv3FINAL.html' problem
- saves a record of all committed versions of code through its history

collaboration

- allows multiple versions for multiple developers working together
- allows developers to work together worldwide without stepping on each other's toes



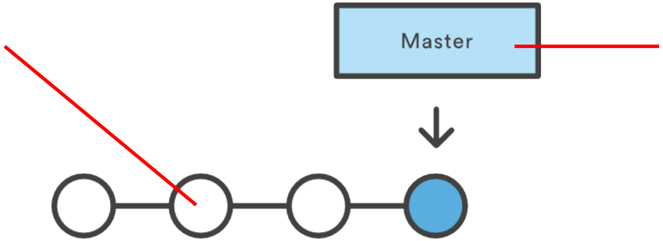
commit history

git basics

what is git?

save vs commit

- save -- overwrites new content onto old content (old content lost)
- commit -- saves current version into database file
- database file maintains record of all commits (all versions)
- commit labels -- identify what work was done and captured by commit



commit

head (points to active commit)

commit history (master branch)

git basics

why learn git?

workflow

- enables changing code and reverting to previous versions as needed
- allows "temporary" or "experimental" changes without disturbing main code base

sharing via github

- code files can be shared with collaborators and general public
- website projects can be hosted for regular web access

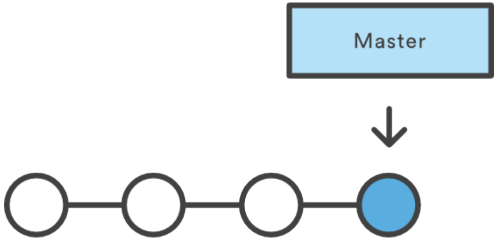
git basics

github

git vs GitHub.com

- git is software; a version control system
- git takes snapshots of your code at certain points in development
- snapshots are stored in a repository (or repo) on your local machine

local repository ("repo")



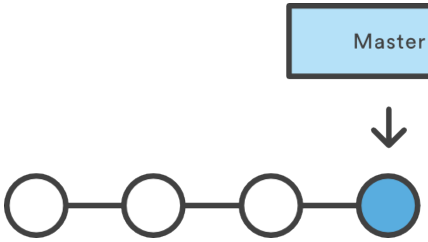
git basics

github

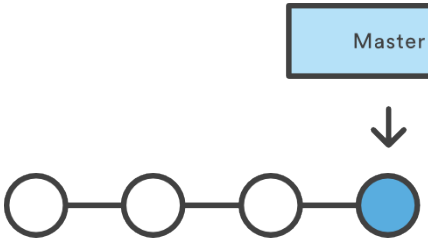
git vs GitHub.com

- git is software; a version control system
- git takes snapshots of your code at certain points in development
- snapshots are stored in a repository (or repo) on your local machine
- GitHub.com is a website that hosts "copies" of your git repositories on a remote server
- GitHub.com can also host your website projects and connect them to the internet

local repo



remote repo (github)



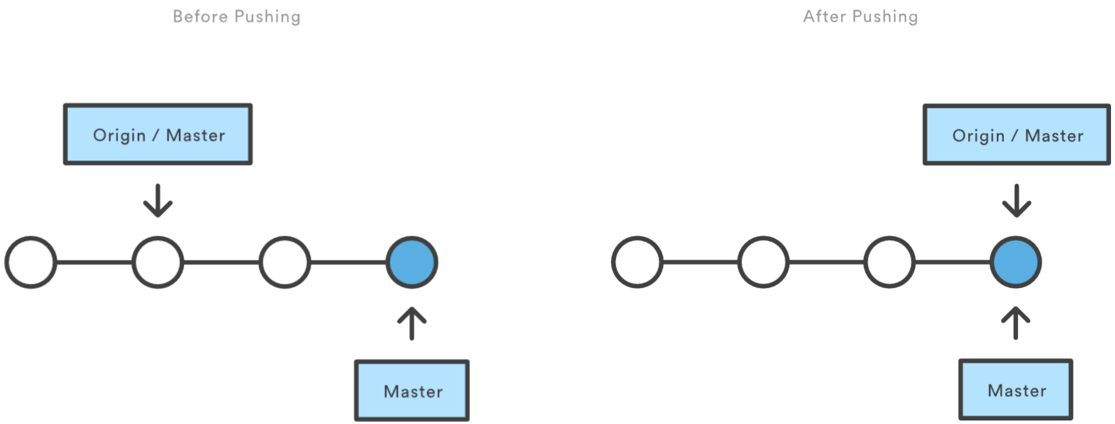
git basics

github

syncing git commits with github

remote

local



git basics

configuring git

```
$ git config --global user.name

$ git config --global user.email
```

check name and email

- check your user name
- check your email

```
$ git config --global user.name "Tom Beach"

$ git config --global user.email "teb@gmail.com"
```

set name or email

- enter your email
- enter your user name

- git config lets you configure your repository from the command line
- defines the author to be used for all commits in the current repository

git basics

first git project

set up new project

- Finder -- duplicate starters folder
- Finder -- move to EXERCISES and rename folder to git_project1
- Terminal -- navigate to git_project1
- run the git init command to initialize git

```
$ git init
```

- run git init only once at the start of every new project
- run git status to make sure repo was initialized

```
$ git status
```

git basics

first git project

staging files

- run the git add command to select files for staging (prepared for commit)

```
$ git add <filename>
```

- use a period to add ALL files for commit

```
$ git add .
```

git basics

first git project

commiting files

- run the git commit command

```
$ git commit -m "first commit"
```

- the first commit should always be labeled "first commit"
- commits should ALWAYS be labeled to indicate what was done before committing
- git commit takes a "snapshot" of the current state of all staged files and saves state to database
- commit message (label) should be concise and descriptive

```
$ git commit -m "added new items to menubar"
$ git commit -m "Added 'about' to the navigation bar and a page for it."
$ git commit -m "Closes #15 by adding a blue background on hover."
```

git basics

first git project

git status command

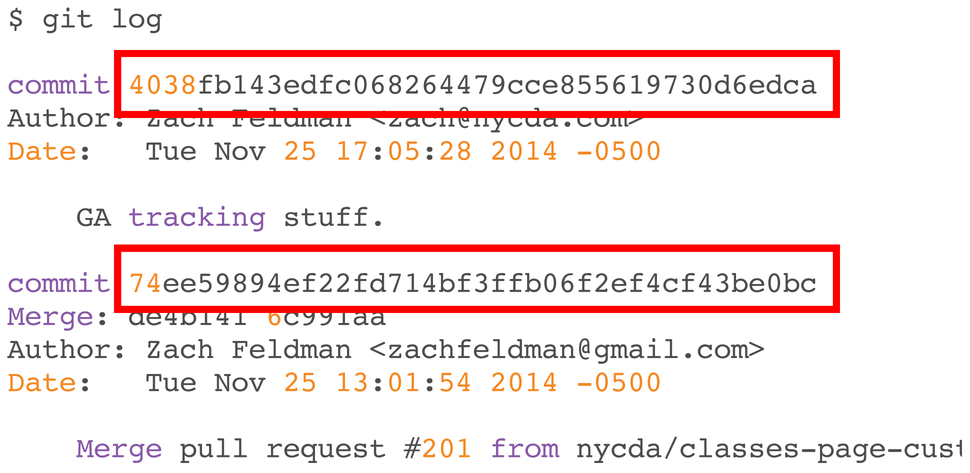
- you can run git status "for free" any time to check repository state
  - see which files have been changed since the last commit
  - check that git is initialized for this project
  - check what files are staged

git basics

first git project

git log command

- provides a list of all commits in project
- allows reverting to previous commits by identifying commit ids



```
$ git log

commit 4038fb143edfc068264479cce855619730d6edca
Author: Zach Feldman <zach@nycda.com>
Date:   Tue Nov 25 17:05:28 2014 -0500

    GA tracking stuff.

commit 74ee59894ef22fd714bf3ffb06f2ef4cf43be0bc
Merge: de4b141 6c991aa
Author: Zach Feldman <zachfeldman@gmail.com>
Date:   Tue Nov 25 13:01:54 2014 -0500

    Merge pull request #201 from nycda/classes-page-cust
```

git basics

first git project

make some changes!

commit your changes

```
$ git status
$ git add .
$ git status
$ git commit -m "what I did"
```

***convention**: always commit immediately after a "YES!" moment*

git basics

Resources

- Code Academy: https://www.codecademy.com/en/courses/learn-git
- TeamTreeHouse: https://teamtreehouse.com/library/git-basics
- Roger Dudler: http://rogerdudler.github.io/git-guide/