Git & GitHub

Our Coding Safety Net

Why are we here today?

Who is Thinkful

Update with thinkful verbage & styling

Why learn git?

- Ever made a mistake and wish you could take it back?
  - Git lets you do that
- If suddenly broken, allows us to roll back to when our project was working
- Allows for easier collaboration with other developers
- Can be intimidating and complex at first, but a little knowledge will go a long away

How this works

bit.ly/tf-git-start-made-up-link

Join Github

https://github.com/join

Join Cloud 9 With your github account

https://c9.io/login

Create a new workspace from there

What is Git?

- A version-control system that allows us to track changes made to files, and document when / where changes occurred, as well as who made the changes
- Uses a branching system that allows us to modify code without fear of breaking existing functionality

GitHub

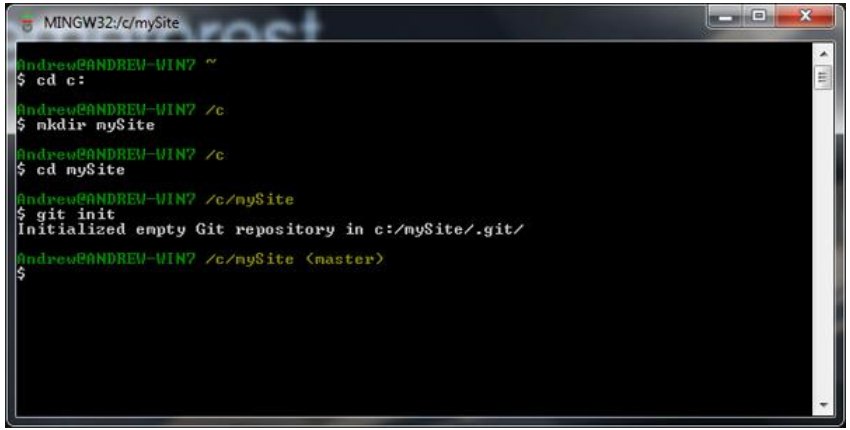
- The most popular site for remotely saving our code
- Allows for easier collaboration and communication of projects involving multiple developers
- GitHub !== Git

Local / Remote Branches

- Local Branch
  - Branch that is stored on your personal machine
  - Can be altered without fear of editing remote branches
- Remote Branch
  - Branch stored in a separate repository, like GitHub, BitBucket, GitLab, etc

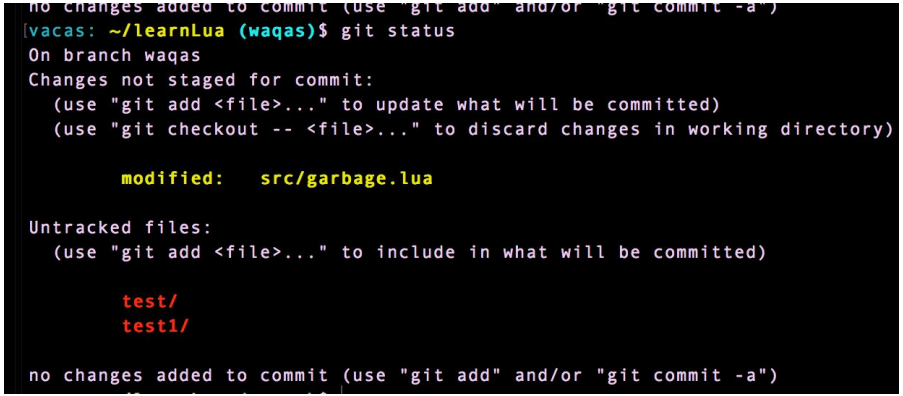
Git Init

- Initializes our current directory as a Git repo
- Allows Git to start tracking changes to your files



Git Status

- One of the most important and helpful Git commands
- Tells us
  - Which files have been altered
  - Which files are currently being tracked
  - Which branch we are currently on
  - Where we are in relation to our remote branch (how many commits ahead / behind)



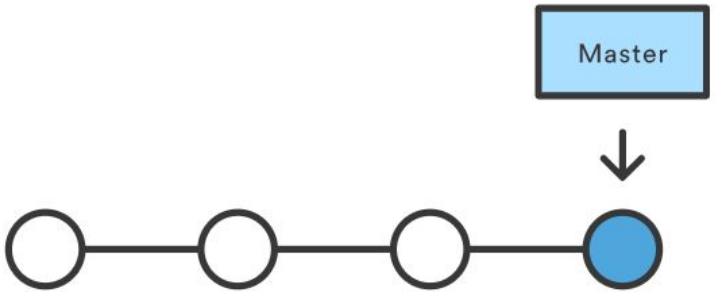
Git Add

- Tells Git which files to add to the staging area, to prepare for commit
- We can add specific files or all files that have been altered

```
~/.emacs.d(master) $> git init
Reinitialized existing Git repository in /home/nick/.emacs.d/.git/

~/.emacs.d(master) $> git add -A

~/.emacs.d(master) $> git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   .gitignore
        modified:   auto-save-list/.saves-8600-nick-ThinkPad-X200~
        modified:   projectile.cache

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)
  (commit or discard the untracked or modified content in submodules)

        modified:   el-get/ace-jump-mode (untracked content)
        modified:   el-get/ace-window (untracked content)
        modified:   el-get/dash (untracked content)
        modified:   el-get/f (untracked content)
        modified:   el-get/flycheck (untracked content)
        modified:   el-get/fuzzy (untracked content)
        modified:   el-get/s (untracked content)


~/.emacs.d(master) $>
```

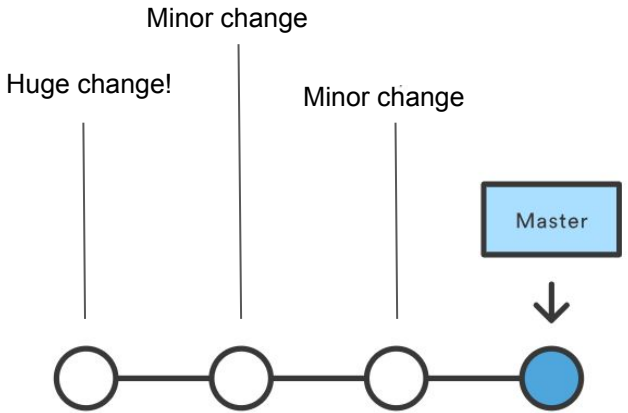
Git Commit

- Each commit is a snapshot of our project at that given time
- Commits our code to our branch



Commit Often!

- Good Git habits will save you from yourself
- If we make monumental changes to our code base between commits, it’s harder to know which code is breaking our project / we may lose code that is actually functional
- Git allows us to roll back to any of our previous commits



Git Clone / Forking

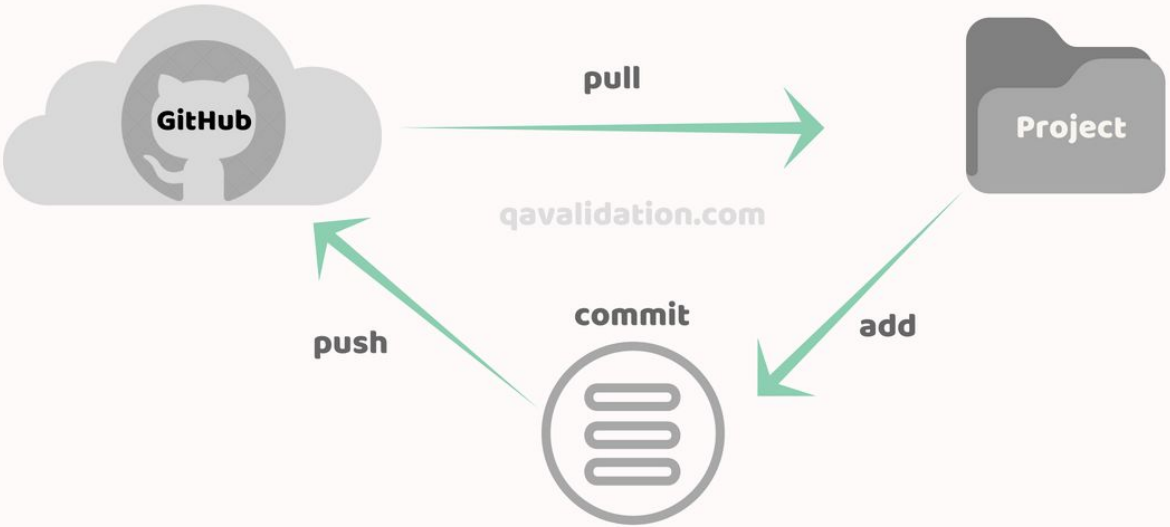
Git Clone

- Copying a project from a remote repo and downloading it to your local machine
- Will need access to push your changes to the remote

Forking

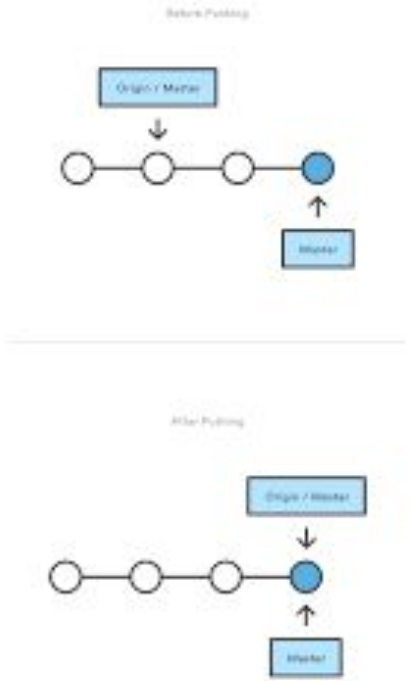
- Similar to clone, but creates your own branch of the project
- Allows you to hack away without fear of altering repo you forked from
- Your changes will need to go through a pull request process to be merged (to be discussed next)

Git PUSH PULL



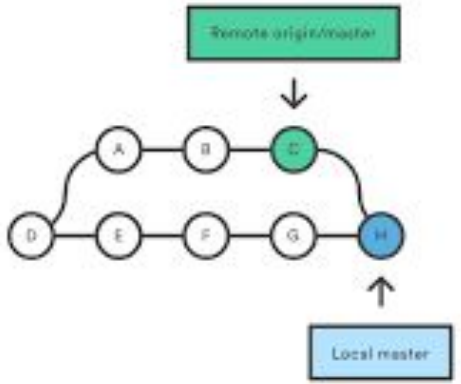
Git Push

- Pushes our local branch up to our remote branch



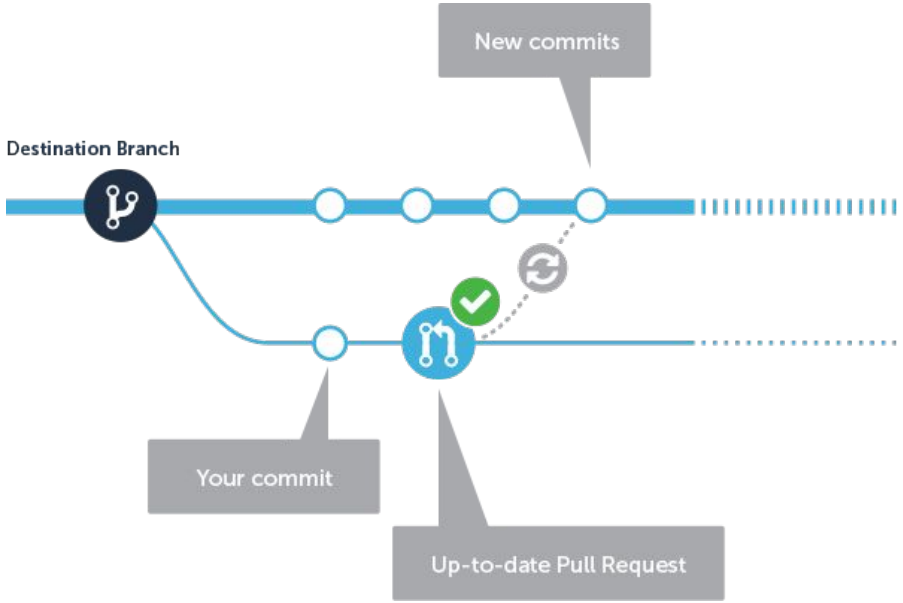
Git Pull

- Pulls files and changes down from our remote branch



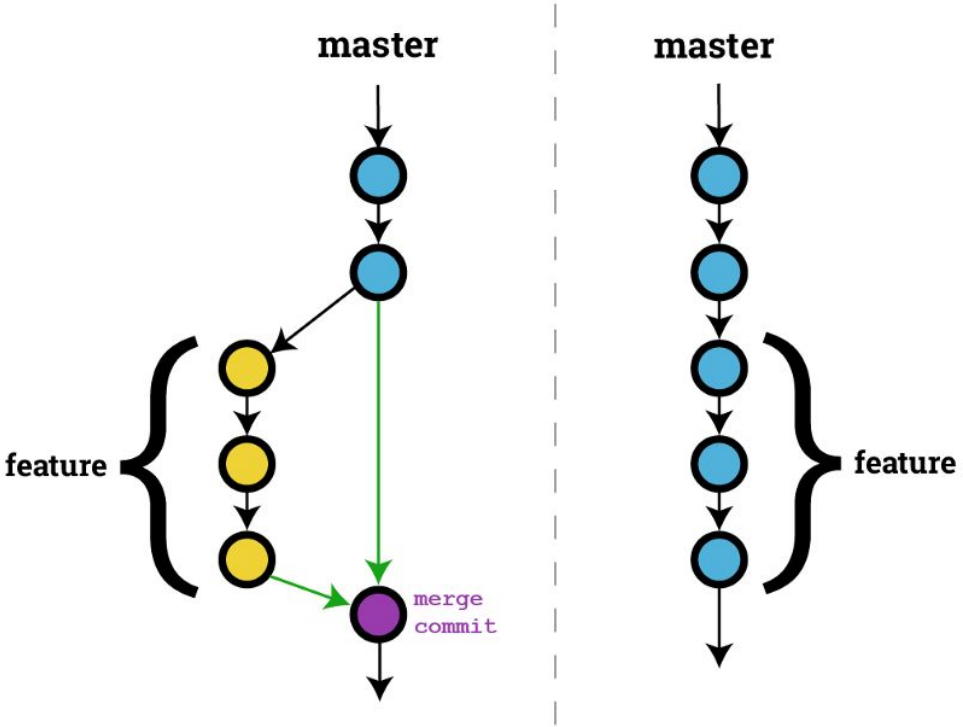
Pull Requests

- Your code will typically be reviewed by one or more parties, and then, if accepted, merged into the branch



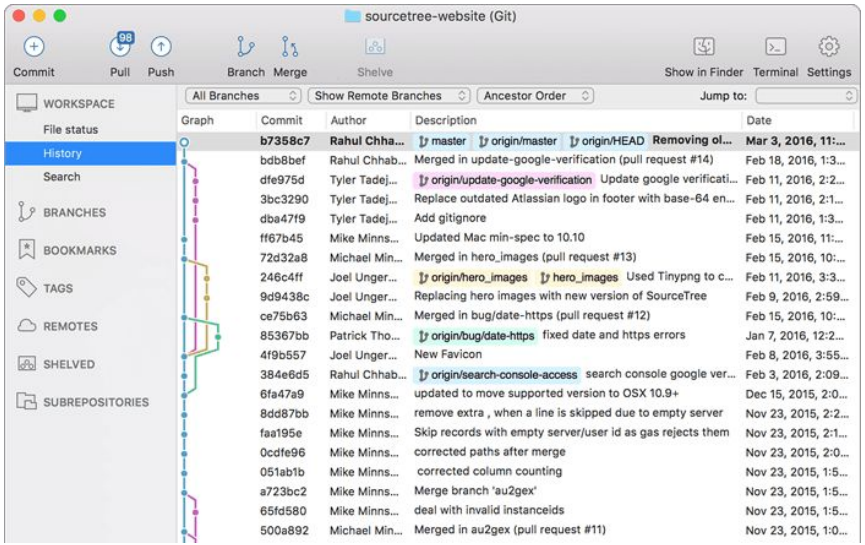
Git Merge

- Similar to pull requests, but typically performed on branches on your local machine
- Good practice: every new feature you add, create a new branch, and then merge
    - Prevents breaking functionality on your master



GUIs

- Graphical user interfaces can make visualizing our git history much simpler



Sourcetree

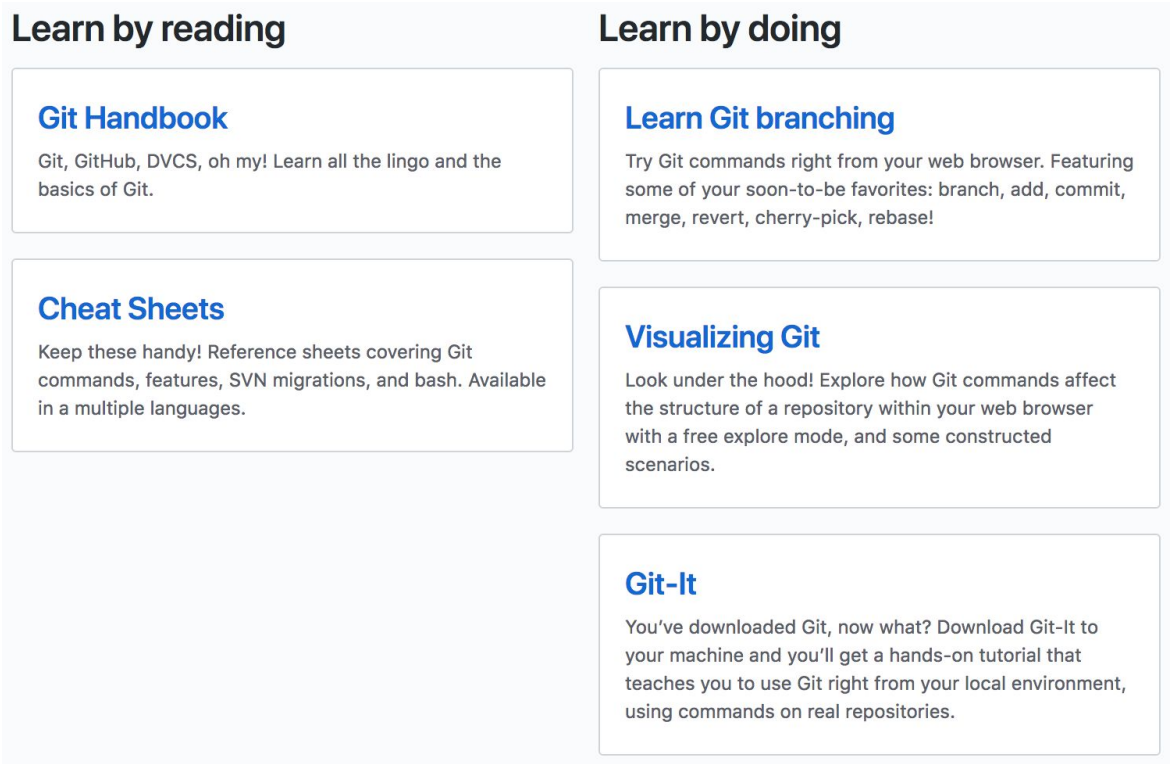
GitHub Desktop

Summary

- Local & Remote
- Init
- Status
- Add
- Commit
- Clone
- Forking
- Push
- Pull
- Pull Requests
- Merge
- GUI

Assignments for tonight

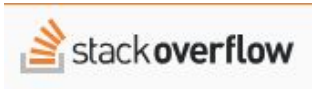
bit.ly/tf-git-classroom



jquery changing background color

jquery changing **css**
jquery changing **text**
jquery changing **background color**
jquery changing **class**

Ways to learn to code



Two Week Free Trial

bit.ly/tf-free-ff-trail-replacewithrealone

- Free trial of Full Stack Flex online program
- Start with HTML, CSS & JS
- Personal Program Manager
- Unlimited Q&A Sessions
- Student Slack Community