

What's the Diff?

Developing with Git and GitHub



First Things First...



git

≠



GitHub

What's Git?

- Distributed Version Control System
- Originally developed by Linus Torvalds
- Allows developers to easily build, share, test, and contribute to software development
- 100% Cross platform - Linux/OS X/Windows
- Highly extensible and customizable
- Easy to learn and free to use!



What's GitHub?

- Service that hosts Git repositories for you
- Extends Git's functionality by introducing a new, social network-like featureset
- GitHub users can easily collaborate, share code, and contribute to Open Source projects
- Many large OS projects hosted here
- ~20 million users, ~57 million repos



Git is not GitHub

- Git is its own thing, you can use Git without GitHub
- Git can run 100% locally, as well
 - *In fact, Git is primarily local*
- GitHub as a platform has substantially extended Git's featureset
- Many other platforms provide Git as a service, along with features similar to GH (see: GitLab, Bitbucket)



Git and GitHub: Like 🥛 + 🍪



- Powerful, decentralized version control
- Extremely flexible and scalable
- Failsafe, high degree of data integrity
- Lightweight and easy to use

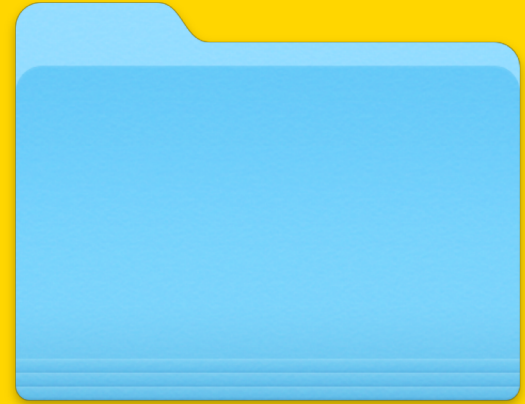


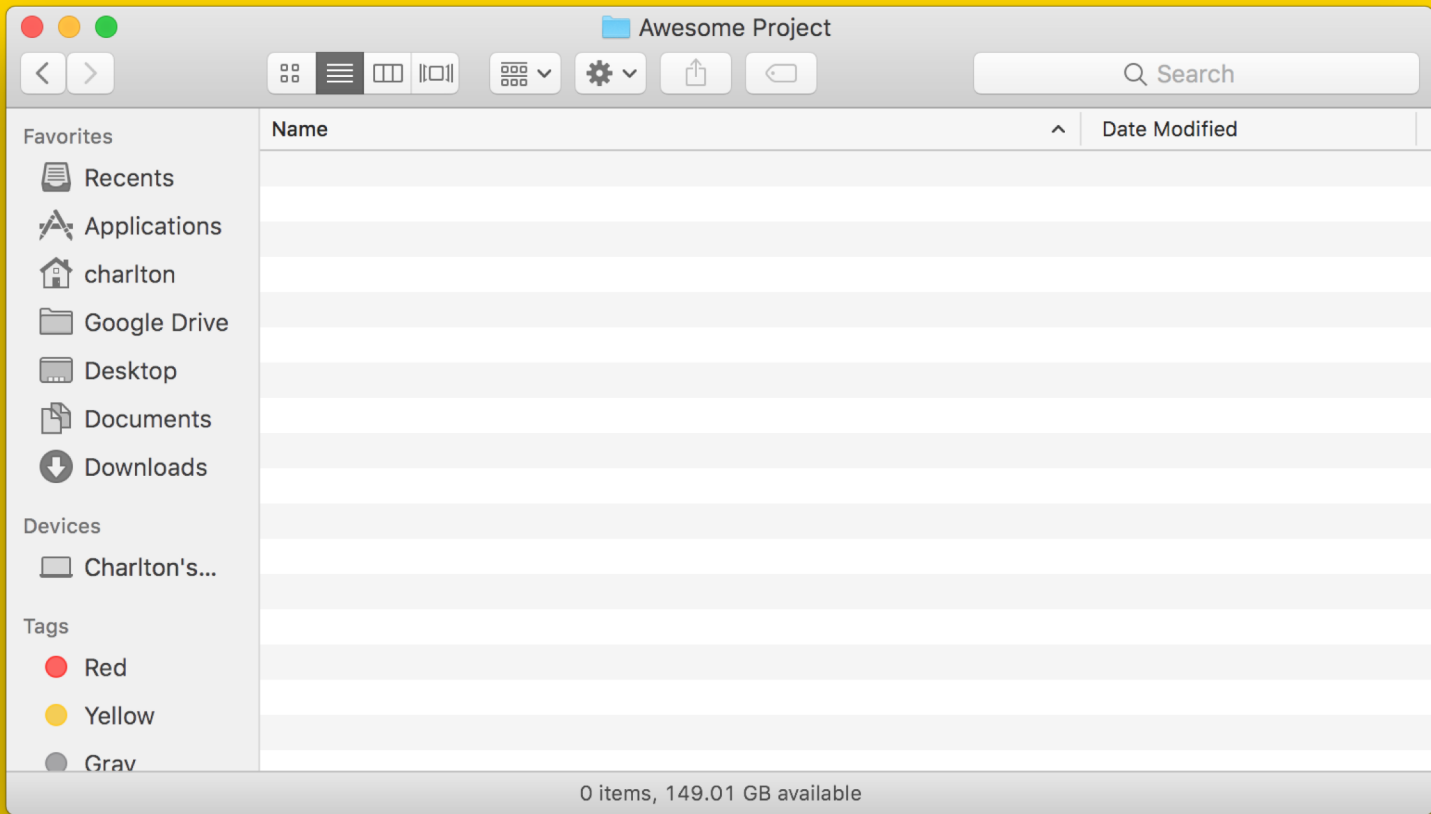
- Easy documentation (READMEs, Wikis)
- Issue tracking and Pull Requests
- Social networking (mentions, threads etc)
- Project analytics and data visualizations
- Third-party service integrations
- Free web hosting with GitHub Pages
- Easy-to-use web UI
- Much, much more!

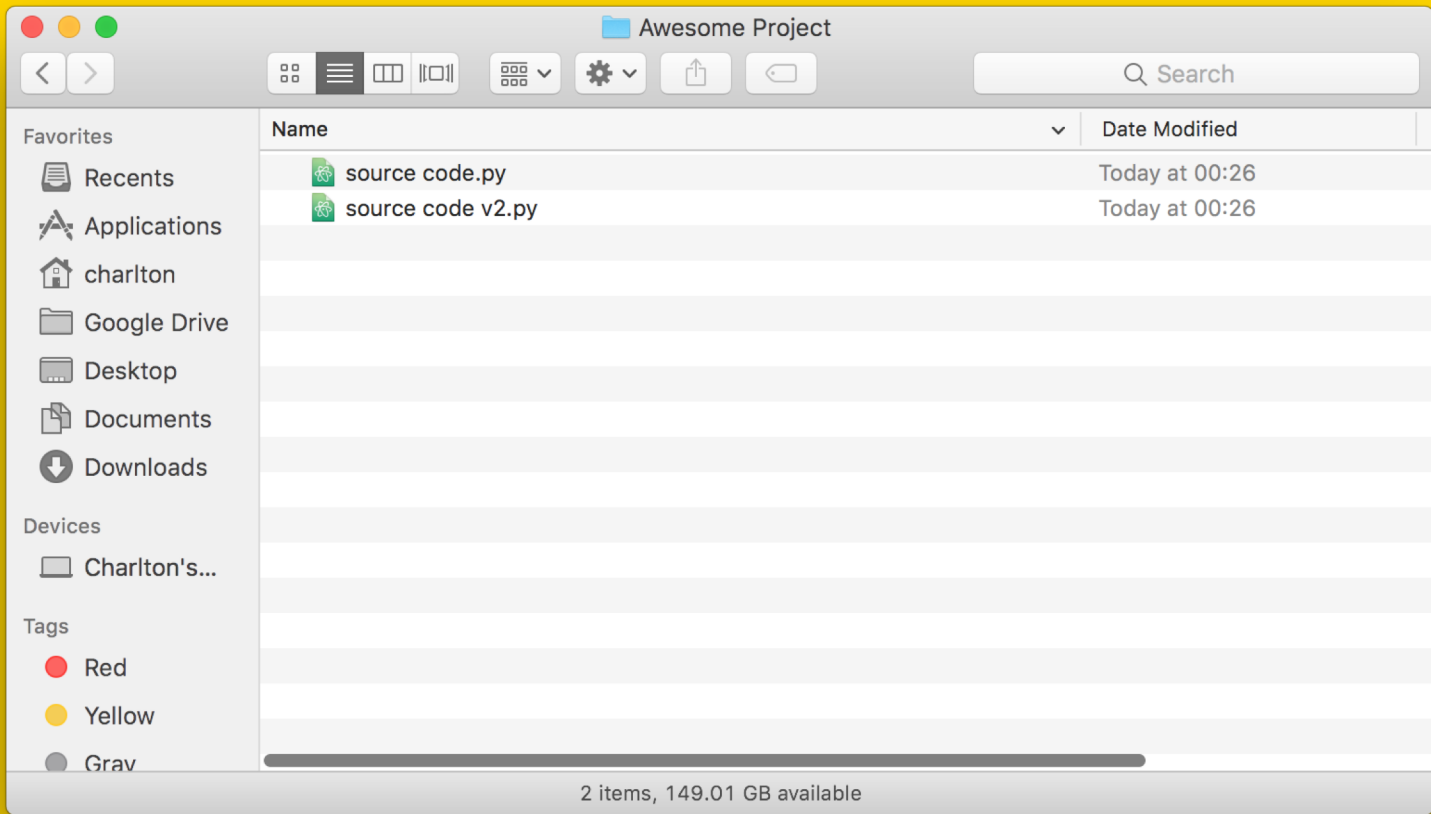
Git Basics



Creating Repositories







Awesome Project

Search

Favorites

- Recents
- Applications
- charlton
- Google Drive
- Desktop
- Documents
- Downloads

Devices

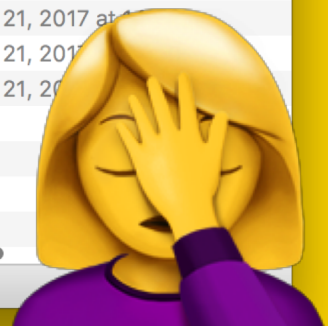
- Charlton's...

Tags

- Red
- Yellow
- Gray

Name	Date Modified
source code.py	Sep 21, 2017 at 14:50
source code working wtf.py	Sep 21, 2017 at 14:50
source code windows_test2.py	Sep 21, 2017 at 14:50
source code windows_test1.py	Sep 21, 2017 at 14:50
source code test1.py	Sep 21, 2017 at 14:50
source code OLD 8.py	Sep 21, 2017 at 14:50
source code CURRENT.py	Sep 21, 2017 at 14:50
source code copy 11.py	Sep 21, 2017 at 14:50
source code copy 10.py	Sep 21, 2017 at 14:50
source code copy 9.py	Sep 21, 2017 at 14:50
source code copy 6.py	Sep 21, 2017 at 14:50
source code copy 3.py	Sep 21, 2017 at 14:50
source code broken_idkwhy.py	Sep 21, 2017 at 14:50

13 items, 149.12 GB available



Creating Repositories

(For Realsies This Time)

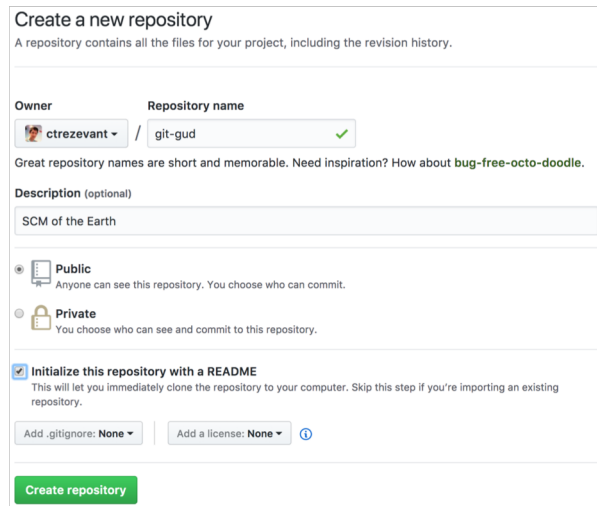


Creating a Repository

Each Git repository has its own directory, you can't start using Git until you initialize one.

There are two ways to make a Git repository:

- Remotely, you can create a Git repo on your service of choice (i.e. GitHub)
- Locally, you can simply run `git init`
 - *If you create a local repo without a remote (e.g. setting an origin), those changes will stay entirely on your machine until `pushed`*



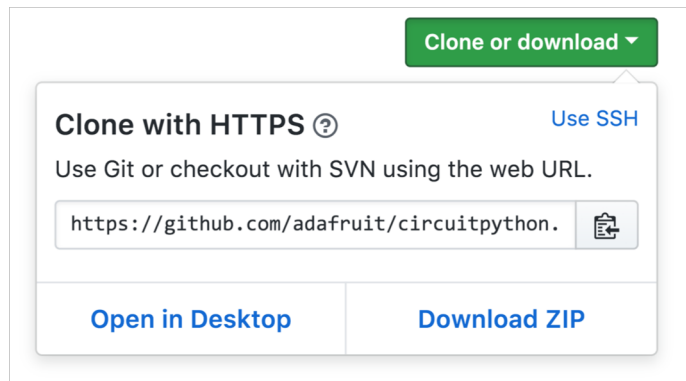
The screenshot shows the GitHub 'Create a new repository' form. At the top, it says 'Create a new repository' and 'A repository contains all the files for your project, including the revision history.' Below this, there are two main sections: 'Owner' and 'Repository name'. The 'Owner' is set to 'ctrezevant' and the 'Repository name' is 'git-gud'. A note below says 'Great repository names are short and memorable. Need inspiration? How about bug-free-octo-doodle.' The 'Description (optional)' field contains 'SCM of the Earth'. There are two radio button options for visibility: 'Public' (selected) and 'Private'. Below these is a checkbox for 'Initialize this repository with a README', which is also selected. At the bottom, there are two dropdown menus for 'Add .gitignore: None' and 'Add a license: None', and a green 'Create repository' button.

```
$ git init
```



Cloning Repositories

- Git repositories can be copied very easily
 - This is called “cloning”
- Creates a local copy of a remote Git repo, which you can immediately start making changes to
- By default, a clone will include all historical versions of every file in the repo
 - This behavior can be altered (for example, with shallow clones)



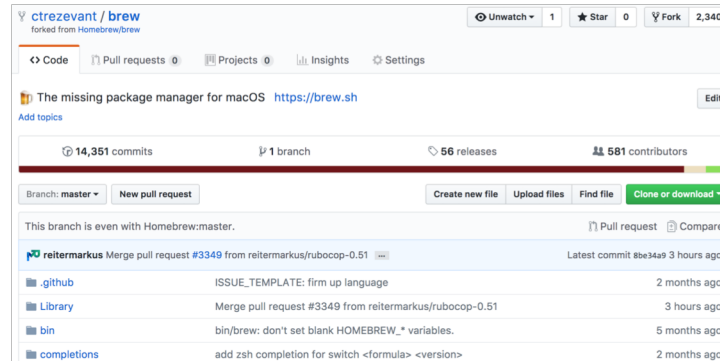
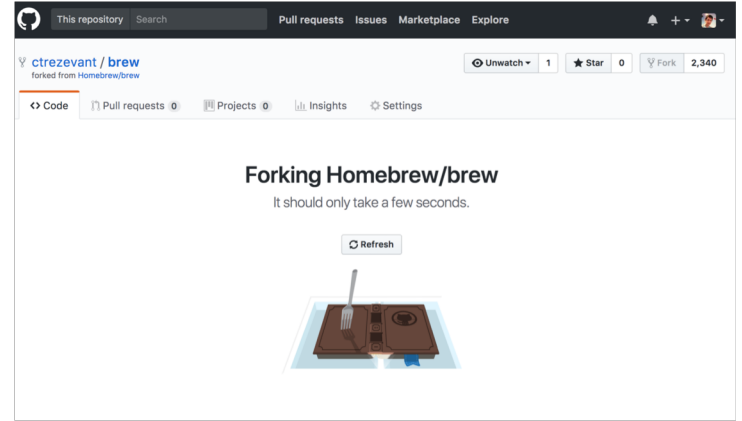
```
$ git clone https://github.com/adafruit/circuitpython.git
```



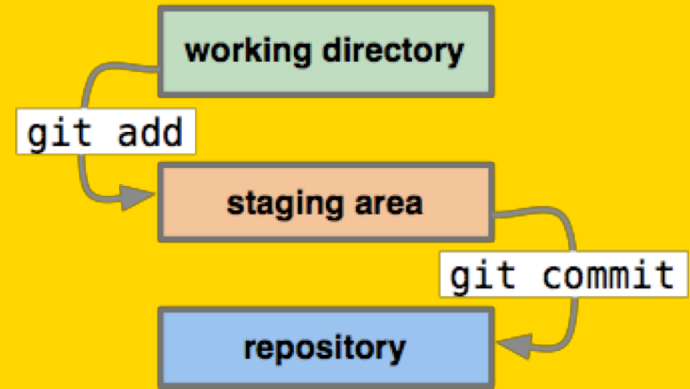
Forking Repositories

- GitHub Specific
- Makes a complete copy of another user's repository, cloning it to your account
- After forking, you can develop independently on your own, or make changes and open a pull request on the source repository (more on that later)

- *smash that Fork button*

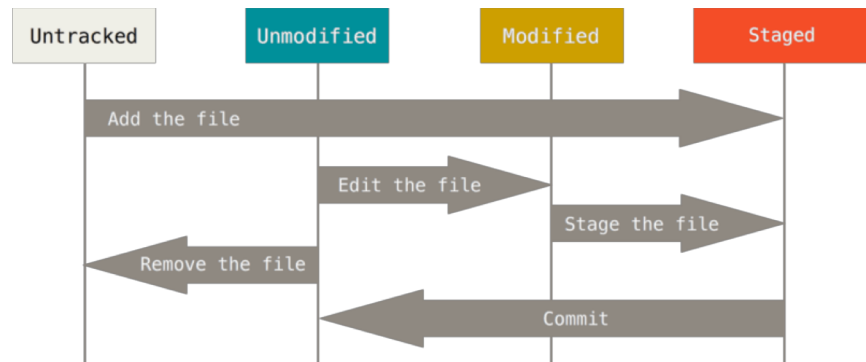


Add, Commit, Push, and Pull



Add

- There are two types of files in Git:
 - “Tracked” files are files being managed by git
 - “Untracked” files are not
- So, `git add` is the first step in a commit
 - Because `add` stages your changes
- Git won't track changes to files you haven't asked it to (i.e. staged) yet
 - There are some exceptions to this (more on that later)



```
$ git add my_code.js  
or  
$ git add .
```



Commit

- Now that you've added your changes, it's time to stage them with a `commit`
- Commits are used to track changes to files
 - As you make changes to one or more files, you “check in” those changes by committing them
- File versions can be compared across commits
 - You can also roll back changes made to a file between commits

Commits on May 11, 2017

- Remove sass.** ...
ctrezevant committed on May 11

Commits on May 8, 2017

- Update README.md**
ctrezevant committed on May 8
- Update README.md** ...
ctrezevant committed on May 8

Commits on May 6, 2017

- Merge pull request #5 from davidbgeek/config-file** ...
ctrezevant committed on May 6

Commits on May 5, 2017

- Fix type matching in sendDirectMessage POST**
humanoid committed on May 5

Commits on May 4, 2017

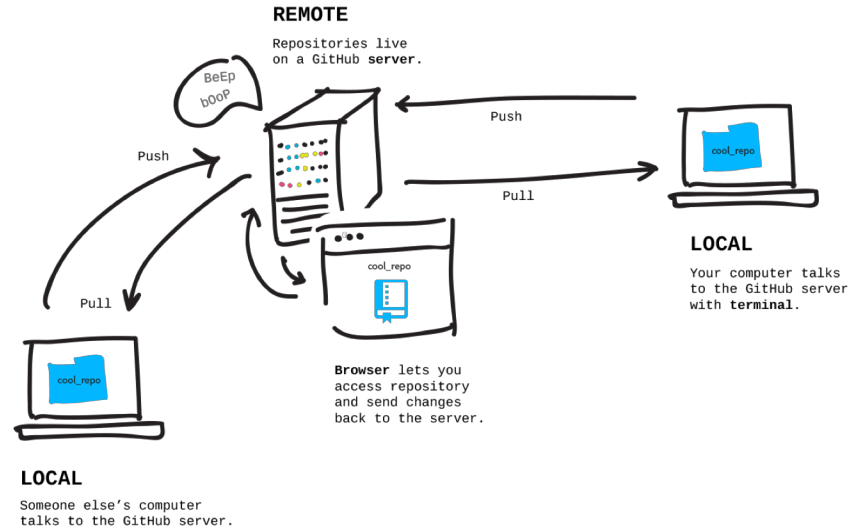
- Python3 Compatibility + Fix sendDirectMessage()**
humanoid committed on May 4
- Create config for auth** ...
humanoid committed on May 4

```
$ git commit -m "My Fantastic Commit"
```



Pushing and Remotes

- `git push` will send your local changes (commits) to a destination repository known as a “remote”
 - Think of it like syncing up your local and remote repositories
 - This is what actually publishes your changes on GitHub
- If no remote is specified, you can add one like so:
 - `git remote add origin <repo url>`

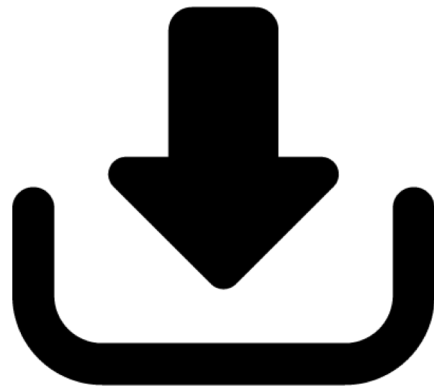


```
$ git push <remote destination> <branch name>
```



Pull

- `git pull` will retrieve the most recently committed changes from the `origin`
 - You can also specify specific origins and branches, if you like
- Good idea to **do this semi-regularly**, so you can stay up to date with everyone else's changes



```
$ git pull <remote origin> <branch name>
```



Diffs

- Show how a file has changed, line by line
- Diffs compare versions of a file between commits, or across branches (more on that later)
- Insertions (green) have been added
- Deletions (red) have been removed
- Won't work on binary files, but they're perfect for source code

```
Fix type matching in sendDirectMessage POST
master (#5)
humanoid committed on May 5
1 parent 8ea5658 commit a1b288a2490b7d1046dd467d9c268d9632e53bb8

Showing 1 changed file with 2 additions and 5 deletions.
Unified Split

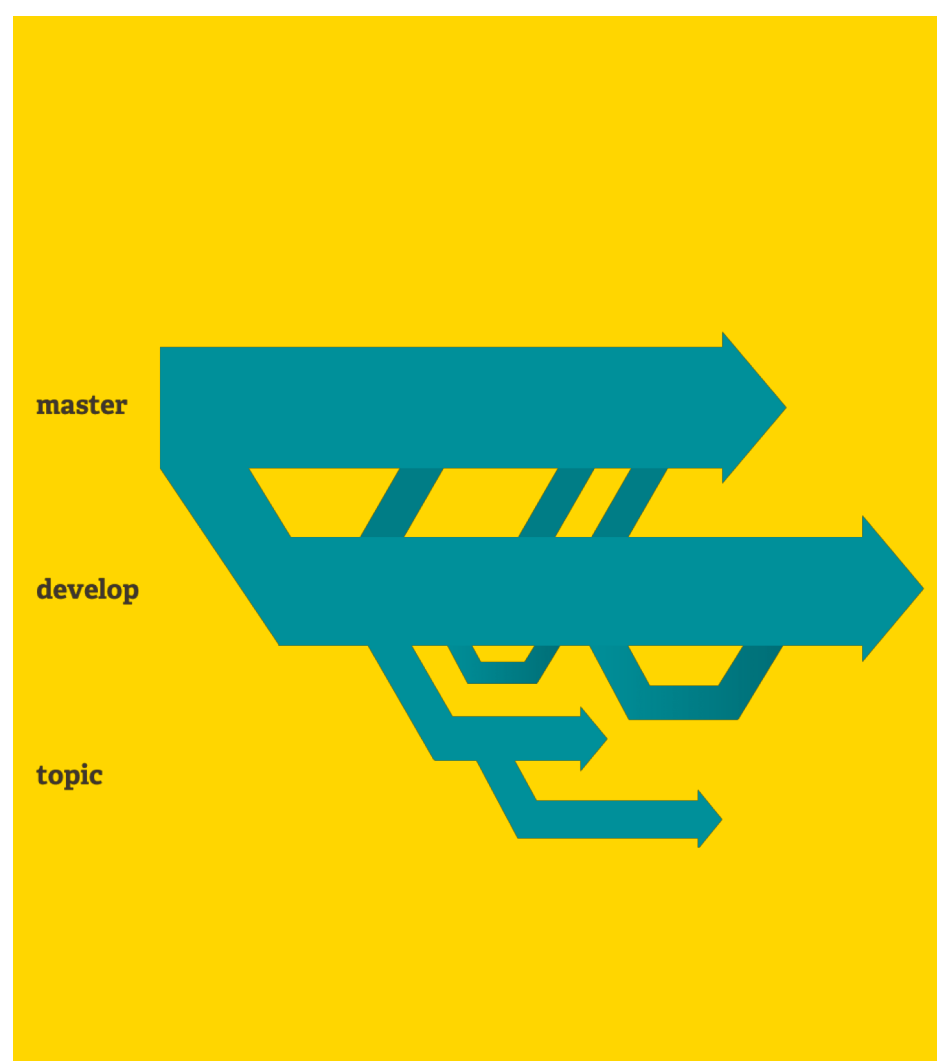
7 edsby.py
View

@@ -680,16 +680,13 @@ def getDirectMessages(self):
680 680     """
681 681     def sendDirectMessage(self, message):
682 682         payload = {
683 -         '_formkey': self.studentData['formkey'],
683 +         '_formkey': self.studentData['formkey'],
684 684         'nodetype': message['nodetype'],
685 685         'body': message['text'],
686 686         'media-fill-include-integrations-integrationfiledata': message['filedata'],
687 687         'media-fill-include-integrations-integrationfiles': message['files']
688 688     }
689 -     return requests.post(
690 -         'https://' + self.edsbyHost +
691 -         '/core/create/' + message['to'] + '?xds=MessagesCompose&permaLinkKey=false&processed=true',
692 -         data=payload, cookies=self.getCookies(), headers=self.getHeaders()).json()
689 +     return requests.post('https://' + self.edsbyHost + '/core/create/' + str(message['to']) + '?xds=MessagesCompose&permaLinkKey=false&processed=true',
693 690         data=payload, cookies=self.getCookies(), headers=self.getHeaders()).json()
694 691     """
695 692     Allows you to search for any higher level user (teacher, administrator)
```

\$ git diff [file]

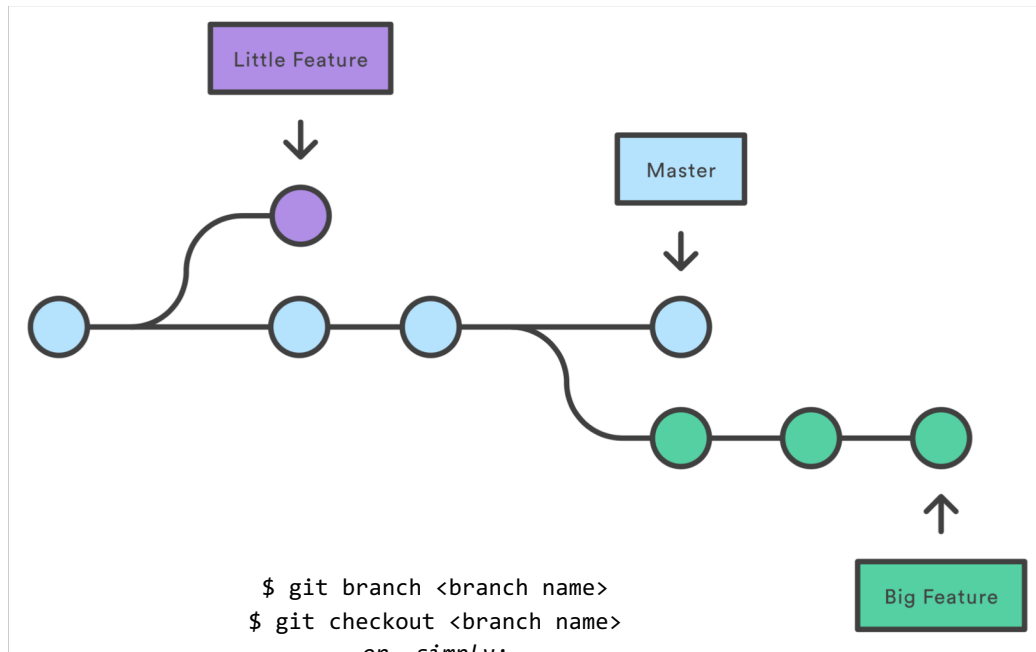


Branching and Merging



Branches

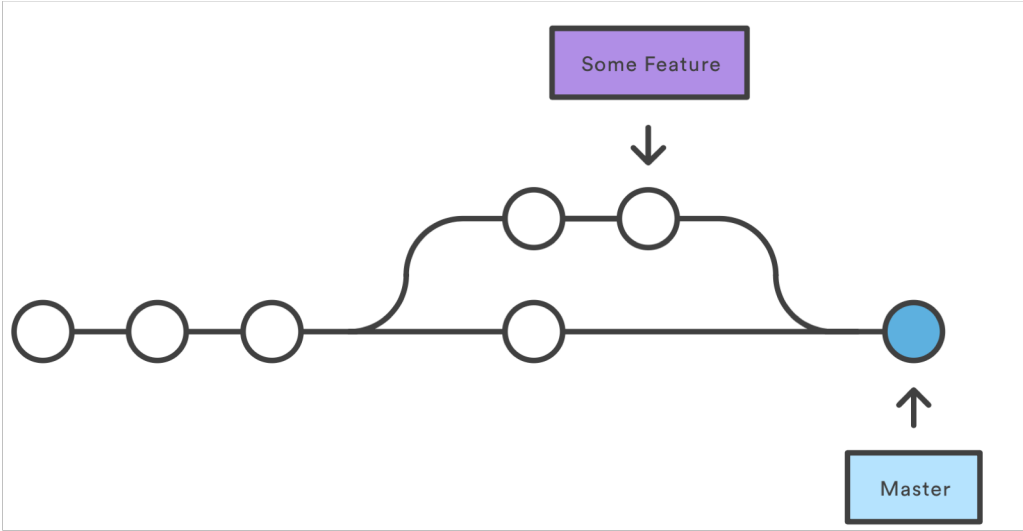
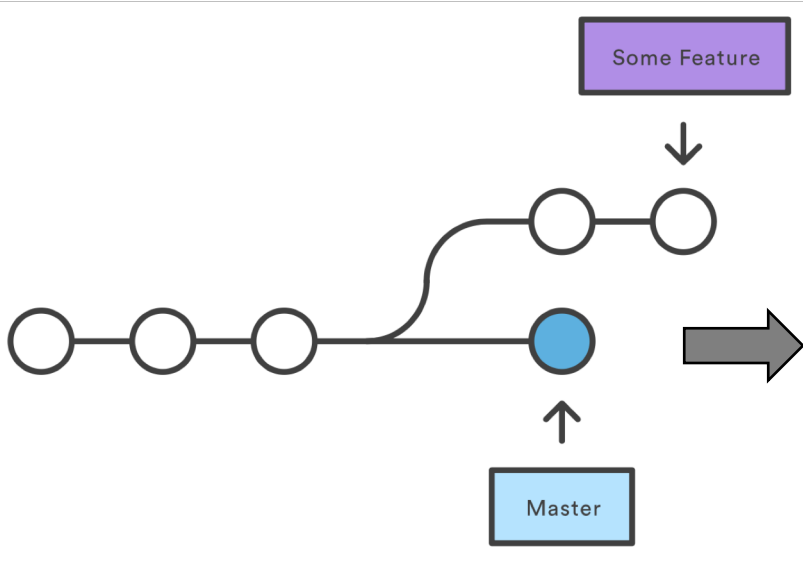
- Disposable copies of your code at a point in time
- Allows features to be developed independently
- Prevents stepping on others' toes
- Enables you to bring your changes into the main project through “merging”



```
$ git branch <branch name>  
$ git checkout <branch name>  
or, simply:  
$ git checkout -b <branch name>  
to delete:  
$ git branch -d <branch name>
```

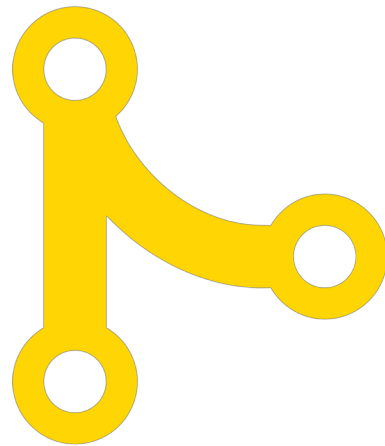


Merging: Conceptual Overview



Merging

- Merging will take the changes in one branch and integrate them into another one
 - This is performed using `git merge`
- Merging will merge all changes, commit histories, etc from the source branch into the destination branch



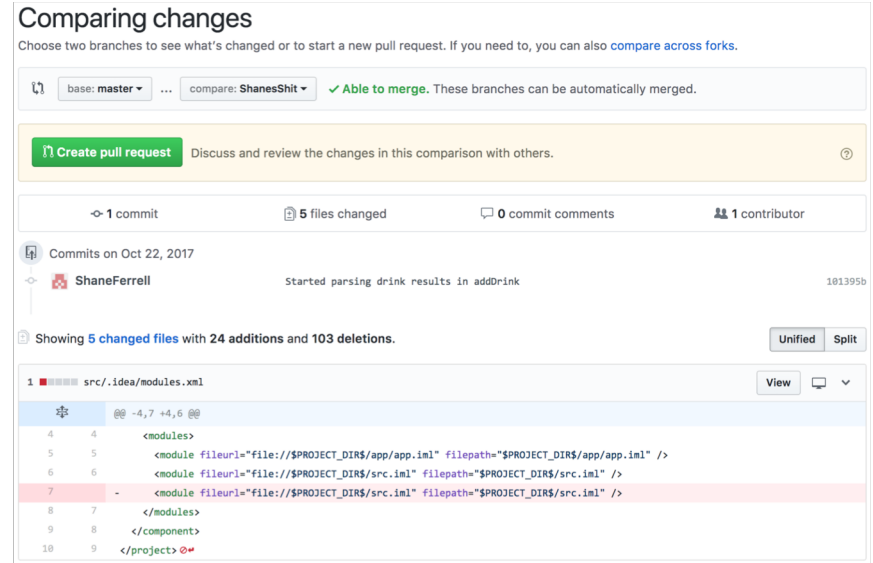
```
$ git checkout <branch you want to  
merge into>
```

```
$ git merge <branch you want to  
merge changes from>
```



Pull Requests

- GitHub-specific
- Effectively a wrapper around Git's already existing merge command, but adds social features into the mix
- Allows repository owners to easily manage, view, and comment on new features that contributors want to bring in



The screenshot displays the GitHub 'Comparing changes' interface. At the top, it shows the comparison between 'base: master' and 'compare: ShanesHit', with a green checkmark indicating 'Able to merge'. Below this, there is a green 'Create pull request' button and a yellow bar with the text 'Discuss and review the changes in this comparison with others.' The interface also shows '1 commit', '5 files changed', '0 commit comments', and '1 contributor'. A commit by 'ShaneFerrell' is listed with the message 'Started parsing drink results in addDrink'. The main part of the screenshot shows a diff for the file 'src/.idea/modules.xml', with a red line indicating a deletion of a module entry.

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: master ... compare: ShanesHit ✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit 5 files changed 0 commit comments 1 contributor

Commits on Oct 22, 2017

ShaneFerrell Started parsing drink results in addDrink 101395b

Showing 5 changed files with 24 additions and 103 deletions. Unified Split

```
1 src/.idea/modules.xml View
@@ -4,7 +4,6 @@
4 <modules>
5 <module fileurl="file://$PROJECT_DIRS/app/app.iml" filepath="$PROJECT_DIRS/app/app.iml" />
6 <module fileurl="file://$PROJECT_DIRS/src.iml" filepath="$PROJECT_DIRS/src.iml" />
7 - <module fileurl="file://$PROJECT_DIRS/src.iml" filepath="$PROJECT_DIRS/src.iml" />
8 </modules>
9 </component>
10 </project>
```



Let's practice!

github.com/ctrezevant/git-workshop

Un-F%&\$ing Things

Practical application of Git against Murphy's Law



Git log: Your Source of Truth

- Running `git log` will show you the full list of commits in your repository, for the current branch and location of HEAD

```
(Charltons-MacBook-Pro:gittest charlton$ git log
commit 71f55dcd210d9213108d452b48d1d239d7cc1c4f (HEAD -> master)
Author: Charlton Trezevant <ct@ctis.me>
Date:   Tue Oct 24 18:01:56 2017 -0400

    Added my code.
```

```
$ git log
```

- This is just useful, in general



Git reset

- Everyone makes mistakes
- Sometimes these mistakes mean we have to nuke some changes
 - `git reset` is here to help
- Resetting will revert the repository's state to how it was at the commit you specify

```
Charltons-MacBook-Pro:gittest charlton$ git reset --hard 6ee545c76a0ab8c71472c536986c170e71beb5f5  
HEAD is now at 6ee545c Updated man
```

```
$ git reset <hash of commit you want to revert to>
```



Git blame

- `git blame` teaches us to be accountable for our actions
- Blame will show you:
 - A list of commits on that file
 - Who's responsible for those changes
 - The actual changes made in those commits

```
Charltons-MacBook-Pro:gittest charlton$ git blame my_file
71f55dcd (Charlton Trezevant 2017-10-24 18:01:56 -0400 1) this is my code!
b614a9a0 (Charlton Trezevant 2017-10-24 18:13:09 -0400 2)
b614a9a0 (Charlton Trezevant 2017-10-24 18:13:09 -0400 3) whoops haha i broke ur code brah
```

```
$ git blame <path to file>
```



Git status

- `git status` will show you the current branch you're on
- It'll also show you whether there are unstaged changes or untracked files in your repository

```
Charltons-MacBook-Pro:gittest charlton$ git status  
On branch master  
nothing to commit, working tree clean
```

```
$ git status
```



Merge conflicts

- Merge conflicts happen when people step on each other's toes
- `git status` and `git pull` will both tell you whether conflicting changes are incoming
- Conflicting changes will be placed in a file for manual resolution
- Best way to avoid: pull changes often, communication is key!

```
$ git status
# On branch branch-b
# You have unmerged paths.
#   (fix conflicts and run "git commit")
#
# Unmerged paths:
#   (use "git add ..." to mark resolution)
#
# both modified:   styleguide.md
#
no changes added to commit (use "git add" and/or "git commit -a")
```

```
If you have questions, please
<<<<<<< HEAD
open an issue
=====
ask your question in IRC.
>>>>>> branch-a
```



More Cool Stuff



GitHub Student Pack

Many tech organizations offer products and services to ~~young and impressionable minds~~ students through GitHub, for free or reduced pricing.

Notable perks include:

- Free, unlimited private GitHub repos
- \$110 AWS Credit
- \$50 credit for new DigitalOcean users
- Free domain name + SSL cert
- Stripe platform credit
- Free private builds on Travis CI
- Unreal Engine

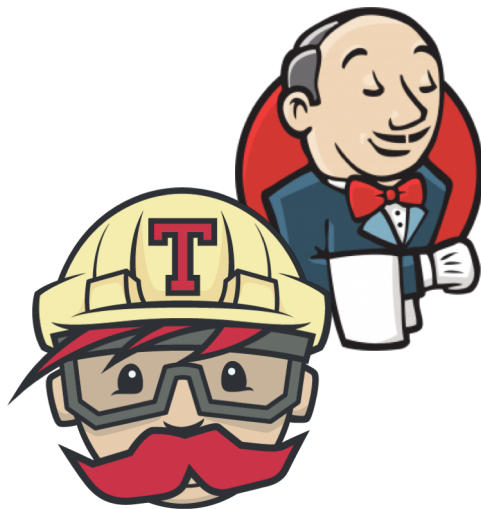


education.github.com/pack



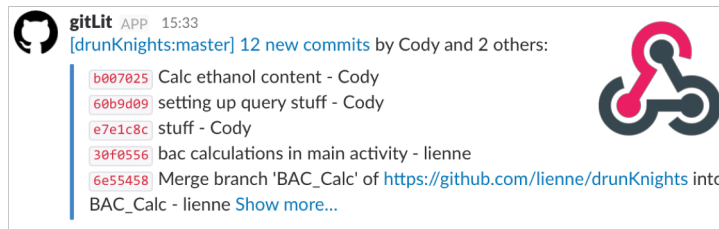
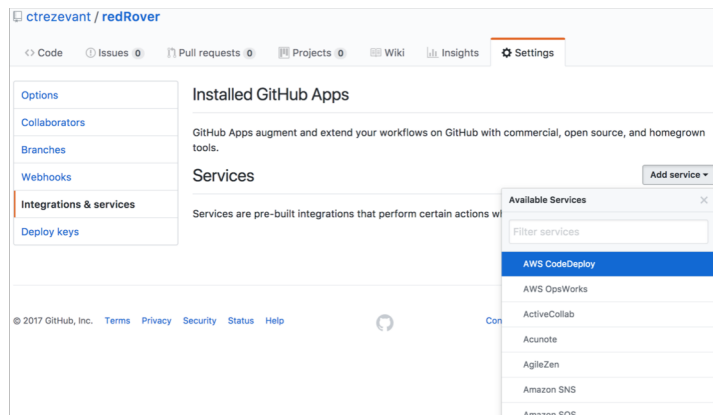
Continuous Integration

- Automatically builds, tests, and deploys your project as you develop
- Will report the status of those tests back to you
- Completely transparent, runs automatically and behind the scenes
- Great way to ensure you're shipping working code



Other Integrations

- GitHub features a wide variety of services you can deploy with a click
- Many of these exist to help make software development easier
 - Kind of like IFTTT, but with Git as a trigger
- GitHub also has an open and well documented API, that you can use to build your own!



github.com/marketplace



Congratulations

You can now do Git things!



Fork Me on GitHub



@charlton on Slack

github.com/ctrezevant

www.ctis.me

Extra links, docs, etc:

ctis.me/s/git