

All links can be found on:

<https://github.com/UWB-ACM/git-gud-workshop-spring-2019>

1. Install Git (command line)

<https://git-scm.com/downloads>

2. Create a GitHub account if you don't have one already.

Use a private email (not your school email)

<https://github.com/>

3. Open this URL for later:

<https://bit.ly/30YX7o2>

```
$ git gud  
git: 'gud' is not a git command. See 'git --help'.
```

The most similar command is
gui

Presenters: Lizzy Presland, Brady Pascoe

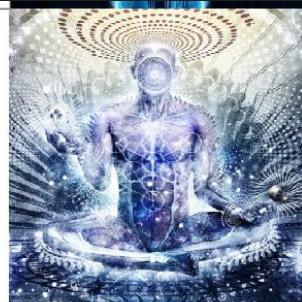
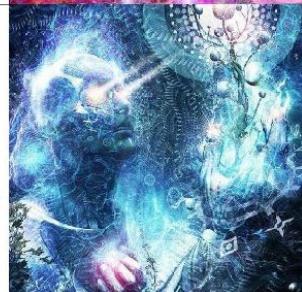
Git Gud Workshop

By your friends at

UWB ACM

Agenda

- Introduction to Git
- Commands Overview
- Workshop #1: First Commit
- Workshop #2: Sync Changes
- Best Practices
- Workshop #3: Making a Pull Request
- Flow, GUIs, and Further Topics
- Additional Resources

Git		Using a single file on a shared computer	
Google Drive		Placing a video camera recording the keyboard	
Uploading the code to an external hard drive via USB		Keeping the IDE open permanently and using ctrl-z to go back to previous versions	
Mailing the code		Wiping the hard drive and re-writing the program	
Printing the code			

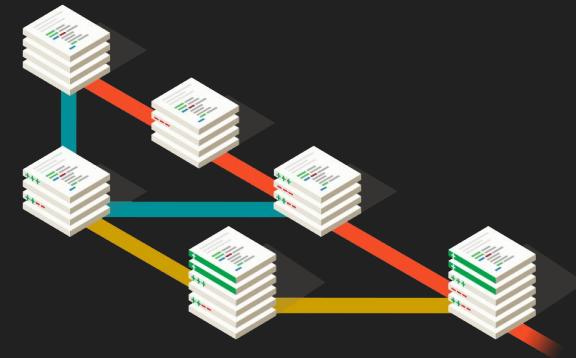
Introduction to Git

What version control is + why it matters

What's Version Control?

(If you've had CSS 360, this may ring a bell)

- Tracks revision history
 - See which user(s) made changes to document(s)
- Allows users to switch between versions
 - “Roll-back” to old versions (aka “revert”)
- Implemented in cloud drive platforms
 - e.g. Google Drive



What's Git?

What is Git?

- Version Control Software!
- Tool for tracking revisions of files
- Enables synchronization for distributed teams

Who uses Git?

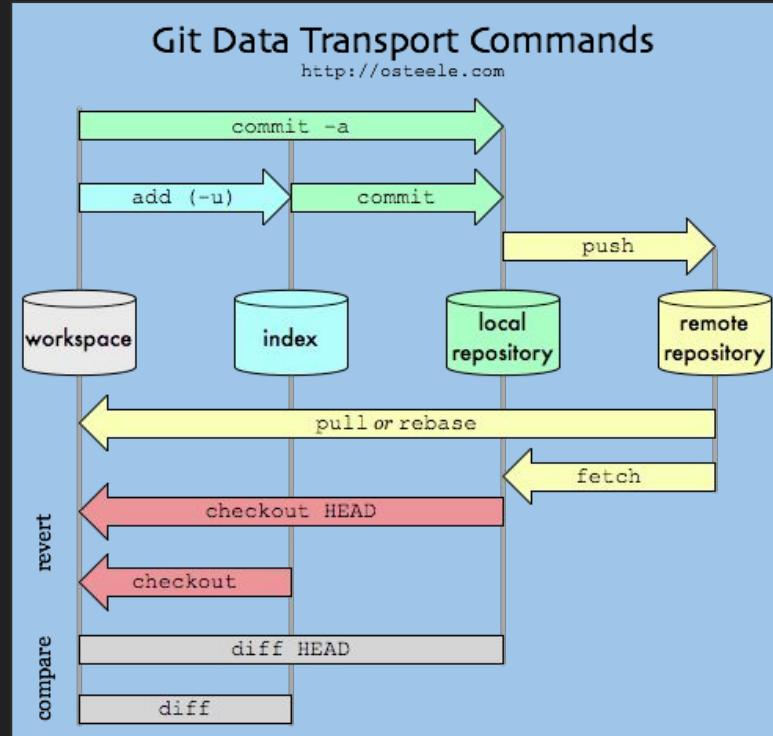
- Amazon, Microsoft, Facebook, Netflix, Twitter, Linux, Android, Eclipse, Starbucks, Google, Hashicorp, eBay...
- Over 31 million users on GitHub. (Git != GitHub)
- Your classmates! (Solo, or for group projects)



Why Git?

- Keep files in sync!
 - Among workstations
 - Among teammates
- Backup source code
 - Hard drive crashes
- Distributed collaboration
 - Work remote!
- Manage complex workflows

It's great for code!



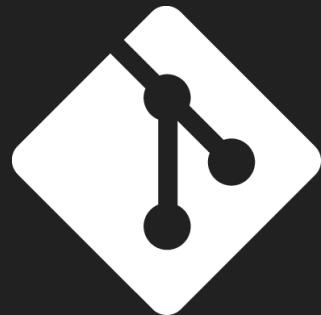
What *isn't* Git?

Git != GitHub

GitHub *uses* Git*

Git != Google Drive

(Git doesn't sync in real-time)



≠



≠



Git != GitHub

Git (Launched April 2005)

- Version Control Software
- Independent of any online services
 - Doesn't even have to connect online



GitHub (Launched April 2008)

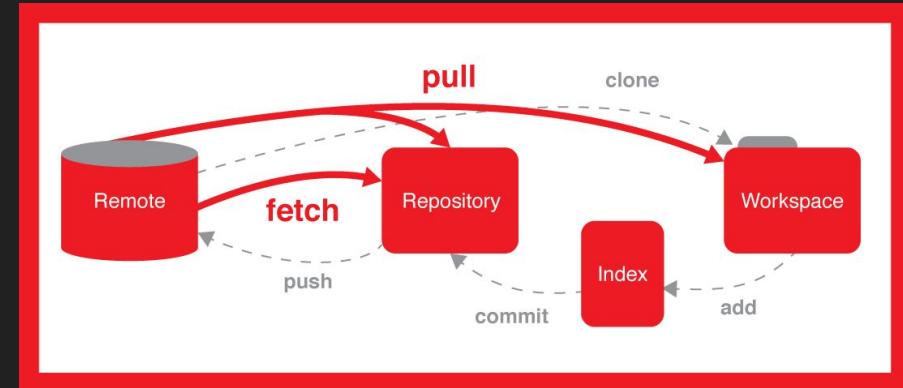
Popular online service for hosting Git repositories, public or private

Add features that enhance an open-source software workflow

- Issues
- PRs
- Support for Continuous Integration / CD

Ways to use Git

- Locally: Repo only on your machine
- Synced: Repo synced to remote
 - Remotes are often hosted*



- Command Line Interface (Default)
 - Fully featured
 - Steeper learning curve
- Graphical User Interface
 - May not be fully featured
 - Easier to use
 - IDE Plugins, GitHub Desktop, etc.

*Remote hosting

- GitHub
- GitLab
- Bitbucket
- SourceForge
- Kiln
- AWS CodeCommit
- etc...

Usage & Commands

How to use git

Version Control Vocabulary

Repo / Repository

A set of files, containing commits.

Commit

A set of changes applied to files, linked to the commit before it, with a message and author.

Diff

Changes between current state and last commit.

Branch

A set of commits that are linked to one another.

Remote

A server that hosts a Git repository.

Push

Sync local changes to a branch with a remote.

Pull

Retrieve changes from remote.

Git Commands Reference

Check out a cheat sheet here:

<https://education.github.com/git-cheat-sheet-education.pdf>

Shortlink:

<https://bit.ly/1LHKj5a>

Git Config (First-Time Setup)

Git needs to know your username and e-mail to associate your work with you.

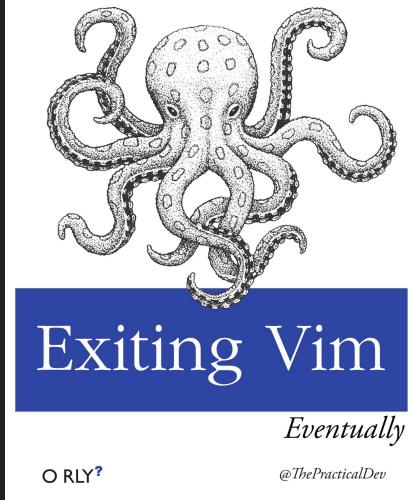
Use the following commands:

```
git config --global user.name "Bobby Tables"
```

```
git config --global user.email bobby@tables.com
```

```
git config --global core.editor "nano"
```

Just memorize these fourteen contextually dependant instructions



Commands: Create a repo

git init : Makes the current directory into a Git repository.

```
:) lizzy@cranberries: ~/code $ mkdir git-test
:) lizzy@cranberries: ~/code $ cd git-test/
:) lizzy@cranberries: ~/code/git-test $ git init
Initialized empty Git repository in /home/lizzy/code/git-test/.git/
:) lizzy@cranberries: ~/code/git-test $ git status
On branch master

No commits yet

nothing to commit (create/copy files and use "git add" to track)
:) lizzy@cranberries: ~/code/git-test $
```

git clone [url] : Downloads a repo from the given URL into a new folder.

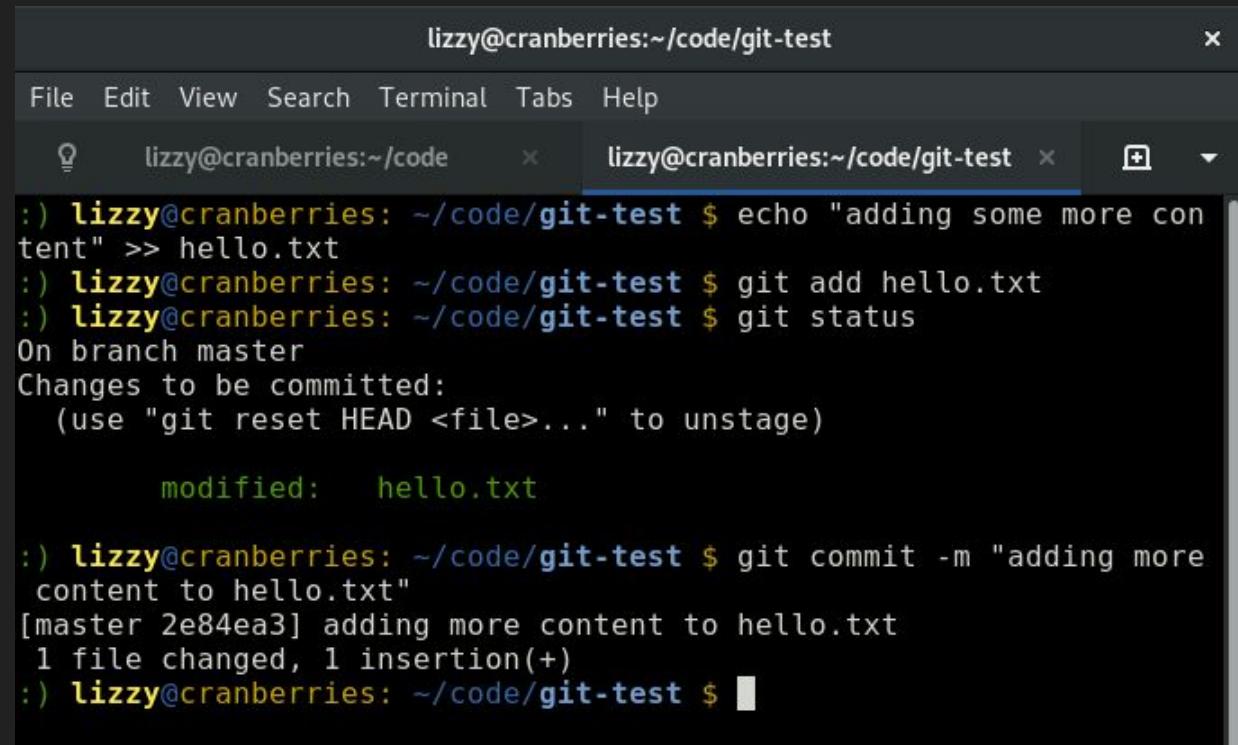
```
:) lizzy@cranberries: ~/code $ git clone https://www.github.com/UWB-ACM/Git-Gud-Workshop.git
Cloning into 'Git-Gud-Workshop'...
```

Commands: Make changes to repo

git add [files] : Adds the specified files to the staging area.

git rm [files] : Removes the specified files from the staging area.

git commit : Associates a message with the changes in the staging area, and add these changes to the repo's history.



The screenshot shows a terminal window titled "lizzy@cranberries:~/code/git-test". It has two tabs open, both labeled "lizzy@cranberries:~/code". The current tab is active and displays the following command-line session:

```
:) lizzy@cranberries: ~/code/git-test $ echo "adding some more content" >> hello.txt
:) lizzy@cranberries: ~/code/git-test $ git add hello.txt
:) lizzy@cranberries: ~/code/git-test $ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   hello.txt

:) lizzy@cranberries: ~/code/git-test $ git commit -m "adding more content to hello.txt"
[master 2e84ea3] adding more content to hello.txt
 1 file changed, 1 insertion(+)
:) lizzy@cranberries: ~/code/git-test $
```

Commands: Branches

git checkout [reference] : Gets the contents of the repo from a specific existing commit or branch.

git checkout -b [branch-name] : Create a new branch locally (note the -b flag)

```
:) lizzy@cranberries: ~/code/git-test $ git checkout -b new_feature
Switched to a new branch 'new_feature'
:) lizzy@cranberries: ~/code/git-test $ echo "fun new feature" >> new_feature.txt
:) lizzy@cranberries: ~/code/git-test $ git status
On branch new_feature
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    new_feature.txt

nothing added to commit but untracked files present (use "git add" to track)
:) lizzy@cranberries: ~/code/git-test $ git checkout master
Switched to branch 'master'
:) lizzy@cranberries: ~/code/git-test $ █
```

Commands: Syncing with remote repo

git remote -v : get information about remote repositories associated with the local repository

git remote add <nickname> <URL> : associates a local repository with a remotely hosted repository (GitHub, GitLab, etc)

git push <nickname> <branch> : Uploads changes.

git pull <nickname> <branch> : Downloads changes.

```
:) lizzy@cranberries: ~/code/git-test $ git remote -v
:) lizzy@cranberries: ~/code/git-test $ git remote add origin https://github.com/etcadinfinity/musical-carnival.git
:) lizzy@cranberries: ~/code/git-test $ git remote -v
origin https://github.com/etcadinfinity/musical-carnival.git (fetch)
origin https://github.com/etcadinfinity/musical-carnival.git (push)
:) lizzy@cranberries: ~/code/git-test $ git push origin master
Username for 'https://github.com': etcadinfinity
Password for 'https://etcadinfinity@github.com':
Enumerating objects: 6, done.
```

In case of fire

1. git commit

2. git push

3. leave the building



Workshop #1

Create your first commit

Workshop #1: Our First Commit

1. Create a local repo

```
git init
```

2. Create new files

```
echo "Hi!" >> hello.txt
```

(or equivalent)

3. Add files to our index

```
git add hello.txt
```

4. Commit changes to local repo

```
git commit
```

Workshop #1: Example

1. Initialize repo
2. Create text file
3. Add file to index
4. Commit file to repo

```
12:13:20:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ git init
Initialized empty Git repository in /home/spacekatt/ACM/test/.git/
12:13:22:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ echo "Hello, World!" >> hello.txt
12:13:33:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ git add hello.txt
12:13:38:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ git commit -m "Add hello world doc"
[master (root-commit) 93e019a] Add hello world doc
 1 file changed, 1 insertion(+)
 create mode 100644 hello.txt
12:13:50:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ □
```

Workshop #2

Sync Local Repository to Remote Repository

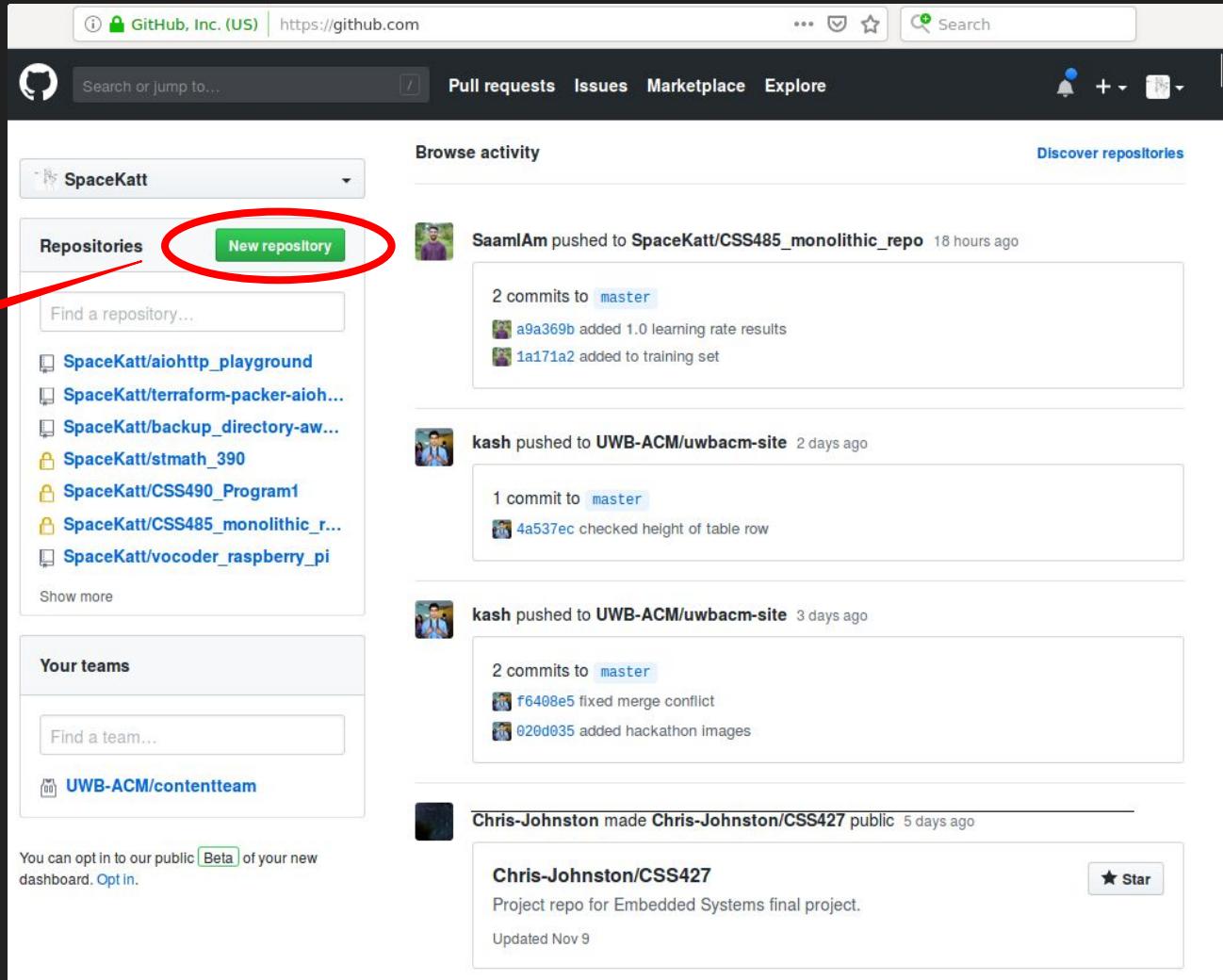
Workshop #2: Syncing to remote (GitHub)

Using the repository from Workshop #1...

- Create remote repo (on GitHub) <Create repo on GitHub>
- Add reference to remote (URL) locally git remote add origin <remote url>
- Sync local changes to remote git push -u origin master

Create New Repository

Click da' button!



The screenshot shows the GitHub dashboard for the user 'SpaceKatt'. A large red arrow points from the text 'Click da' button!' to the 'New repository' button, which is highlighted with a red oval. The dashboard includes sections for 'Repositories' (listing several repositories like 'SpaceKatt/aiohttp_playground' and 'SpaceKatt/terraform-packer-aio...'), 'Your teams' (listing 'UWB-ACM/contentteam'), and 'Browse activity' (showing recent pushes from other users like 'SaamIAm' and 'kash' to their repositories). The top navigation bar includes links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'.

SpaceKatt

Repositories **New repository**

Find a repository...

- SpaceKatt/aiohttp_playground
- SpaceKatt/terraform-packer-aio...
- SpaceKatt/backup_directory-aw...
- SpaceKatt/stmath_390
- SpaceKatt/CSS490_Program1
- SpaceKatt/CSS485_monolithic_r...
- SpaceKatt/vocoder_raspberry_pi

Show more

Your teams

Find a team...

UWB-ACM/contentteam

Beta

You can opt in to our public Beta of your new dashboard. [Opt in](#).

Pull requests Issues Marketplace Explore

Browse activity

Discover repositories

SaamIAm pushed to **SpaceKatt/CSS485_monolithic_repo** 18 hours ago

- 2 commits to master
- a9a369b added 1.0 learning rate results
- 1a171a2 added to training set

kash pushed to **UWB-ACM/uwbacm-site** 2 days ago

- 1 commit to master
- 4a537ec checked height of table row

kash pushed to **UWB-ACM/uwbacm-site** 3 days ago

- 2 commits to master
- f6408e5 fixed merge conflict
- 020d035 added hackathon images

Chris-Johnston made **Chris-Johnston/CSS427** public 5 days ago

Chris-Johnston/CSS427

Project repo for Embedded Systems final project.

Updated Nov 9

Star

Name New Repository

Give your new repository some name...

If you're hosting a cool project, then use a descriptive name!

Don't include a README, .gitignore, or license yet.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

 SpaceKatt ▾

Repository name

glowing-guacamole



Great repository names are short and memorable. Need inspiration? How about [glowing-guacamole](#).

Description (optional)



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.



Initialize this repository with a README

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾

Add a license: **None** ▾



Create repository

Copy URL of repo

Click da' button to copy
to your clipboard

Keep this page open

We will refresh it after
the next step!

SpaceKatt / glowing-guacamole

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

Quick setup — if you've done this kind of thing before

HTTPS SSH <https://github.com/SpaceKatt/glowing-guacamole.git> 

Get started by creating a new file or uploading an existing file. We recommend every repository has a README, LICENSE, and .gitignore.

Or, create a new repository on the command line

```
echo "# glowing-guacamole" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/SpaceKatt/glowing-guacamole.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/SpaceKatt/glowing-guacamole.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#) 

Add remote and push!

Add reference to the remote repository

Sync remote with local repo with push

```
03:18:23:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ git remote add origin https://github.com/SpaceKatt/glowing-guacamole.git
03:18:25:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ git push origin master
Username for 'https://github.com': spacekatt
Password for 'https://spacekatt@github.com':
Counting objects: 6, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (6/6), 506 bytes | 0 bytes/s, done.
Total 6 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:     https://github.com/SpaceKatt/glowing-guacamole/pull/new/master
remote:
To https://github.com/SpaceKatt/glowing-guacamole.git
 * [new branch]      master -> master
03:18:41:spacekatt@spacekatt-Latitude-E6540:~/ACM/test$ 
```

Refresh Repo Page on GitHub

We now see the file we pushed to our remote, origin

SpaceKatt / **glowing-guacamole**

Sample repository for the GitGud workshop (16/Nov/2018)

Manage topics

Edit

1 commit | 1 branch | 0 releases | 1 contributor

Tree: 93e019ae16 ▾ New pull request

Create new file Upload files Find file Clone or download ▾

SpaceKatt Add hello world doc Latest commit 93e019a 2 days ago

hello.txt Add hello world doc 2 days ago

Exciting!

Much open source!

WOW

Heckin' origin

Best Practices

Git Gud, Git Better

Writing Good Commit Messages

Qualities of a good commit message:

- Use the imperative mood
 - "Add config file"
 - "Fix nginx SSL bug"
- Subject is clear and descriptive, while under 50 characters in length.
- Body adds detail and explanations
- Avoid using git commit -m
(do as I say, not as I do)

COMMENT	DATE
O CREATED MAIN LOOP & TIMING CONTROL	14 HOURS AGO
O ENABLED CONFIG FILE PARSING	9 HOURS AGO
O MISC BUGFIXES	5 HOURS AGO
O CODE ADDITIONS/EDITS	4 HOURS AGO
O MORE CODE	4 HOURS AGO
O HERE HAVE CODE	4 HOURS AGO
O AAAAAAAA	3 HOURS AGO
O ADKFJSLKDFJSOKLFJ	3 HOURS AGO
O MY HANDS ARE TYPING WORDS	2 HOURS AGO
O HAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

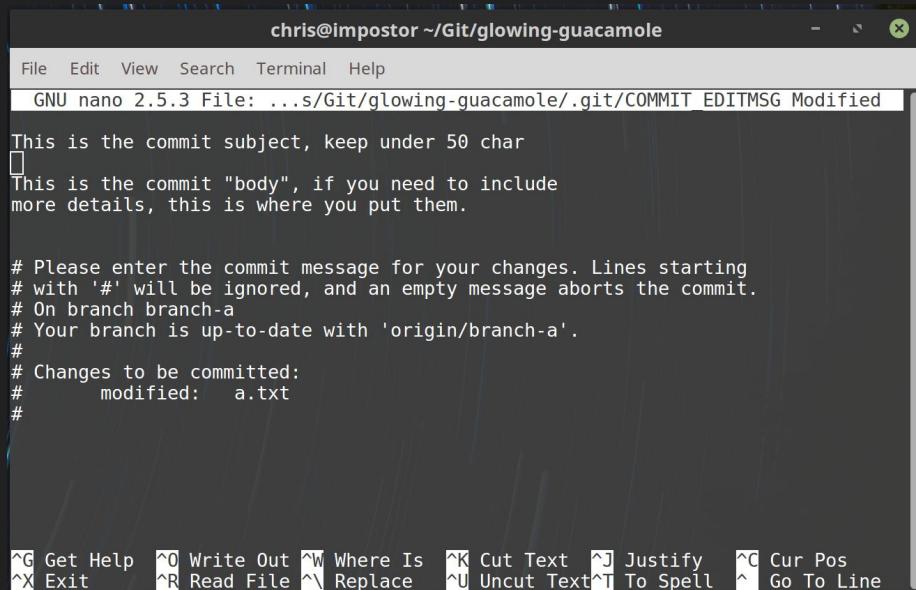
Further reading:

<https://help.github.com/articles/closing-issues-using-keywords/>

<https://code.likeagirl.io/useful-tips-for-writing-better-git-commit-messages-808770609503>

Commit Message Example

What you will see:



chris@impostor ~/Git/glowing-guacamole

File Edit View Search Terminal Help

GNU nano 2.5.3 File: ...s/Git/glowing-guacamole/.git/COMMIT_EDITMSG Modified

```
This is the commit subject, keep under 50 char
This is the commit "body", if you need to include
more details, this is where you put them.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch branch-a
# Your branch is up-to-date with 'origin/branch-a'.
#
# Changes to be committed:
#       modified:   a.txt
#
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

How others will see your message in GitHub:

This is the commit subject, keep under 50 char ...

 **Chris-Johnston** committed 7 minutes ago

This is the commit "body", if you need to include
more details, this is where you put them.

Add branch a files'

 **Chris-Johnston** committed 39 minutes ago

get outta here, foo!

 **Chris-Johnston** committed an hour ago

Viewing the log

View previous commit messages with...

git log

Benefits of a well-kempt log...

- Workflow transparency
- Process traceability
- Bug tracking
- Smaller commits
- Easier merging

```
commit 3199695f77bdb35fd9d510208b8ee95f04e7f3a3
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 18:56:23 2018 -0800

    Update HTML style

commit 08fa02f8155d73dedbec152e05e3640b5632e408
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 16:49:47 2018 -0800

    Fix bug where service starts before loading profile

    Since AWS credentials are provisioned at deployment, I believe
    the gunicorn service was being started by systemd before the
    credential files existed, causing an error (sometimes). Restarting
    the service after Terraform provisions everything seems to fix it.

commit b0299d535f0b1013c399a9ecaee75e7f21d722ef
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 13:08:49 2018 -0800

    Add alert message to user for empty input

commit ece5c4048e93913ef26d5cf0a324124190a6297f
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 12:56:30 2018 -0800

    Handle ommisions in query field

commit 29011a7ab3923bfa5d07ddd1bcf8dcf77ecfd172
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 12:34:01 2018 -0800

    Add all attributes to table

commit ac282eaac3871148bb52e7b8bd87dc2f001254fa
Author: Thomas Kercheval <spacekattpoispin@gmail.com>
Date:   Thu Nov 8 11:45:08 2018 -0800

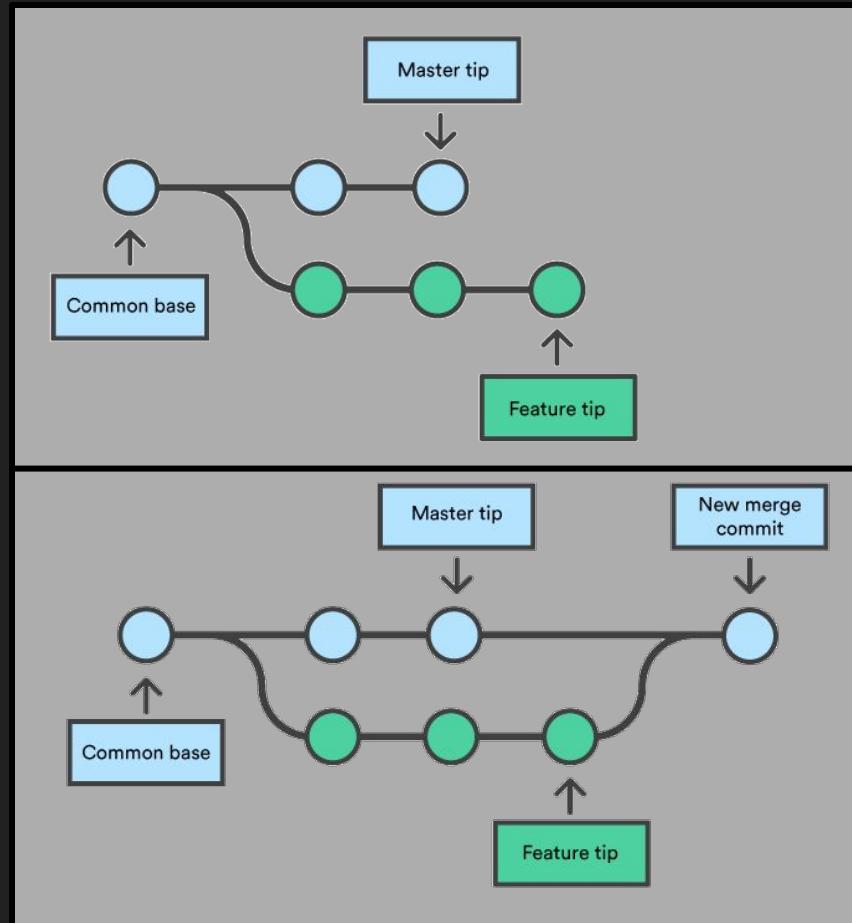
    Add simple HTML table in React
```

Branches

Branches keep track of changes, from a shared starting point.

Branches can:

- Separate from another branch
- Merge into another branch
- Contain different contents from another branch



Using Branches

Why are branches useful?

- Add new features or potentially hazardous changes to the codebase without affecting the stable version of the repository
- Allow multiple developers to make changes to the same files without overwriting each others' work

How do branches get merged together?

- GitHub: *pull request*
- GitLab: *merge request*

Workshop #3

Making a Pull Request
(Forking, Branching & Pull Requests)

Workshop #3: Making a Pull Request

Using git, we can contribute to a pre-existing codebase.

Fork the sample repo first

We will clone our sample repo, make a new branch, add some commits, upload them to our forked copy, and make a new pull request.

Create a file called:

jokes/<FirstName><LastInitial>.txt

Include a funny joke inside it, commit, and push it

Then, we open a pull request on GitHub, to signal upstream that we wish them to merge our work in.

[Fork repo on GitHub]

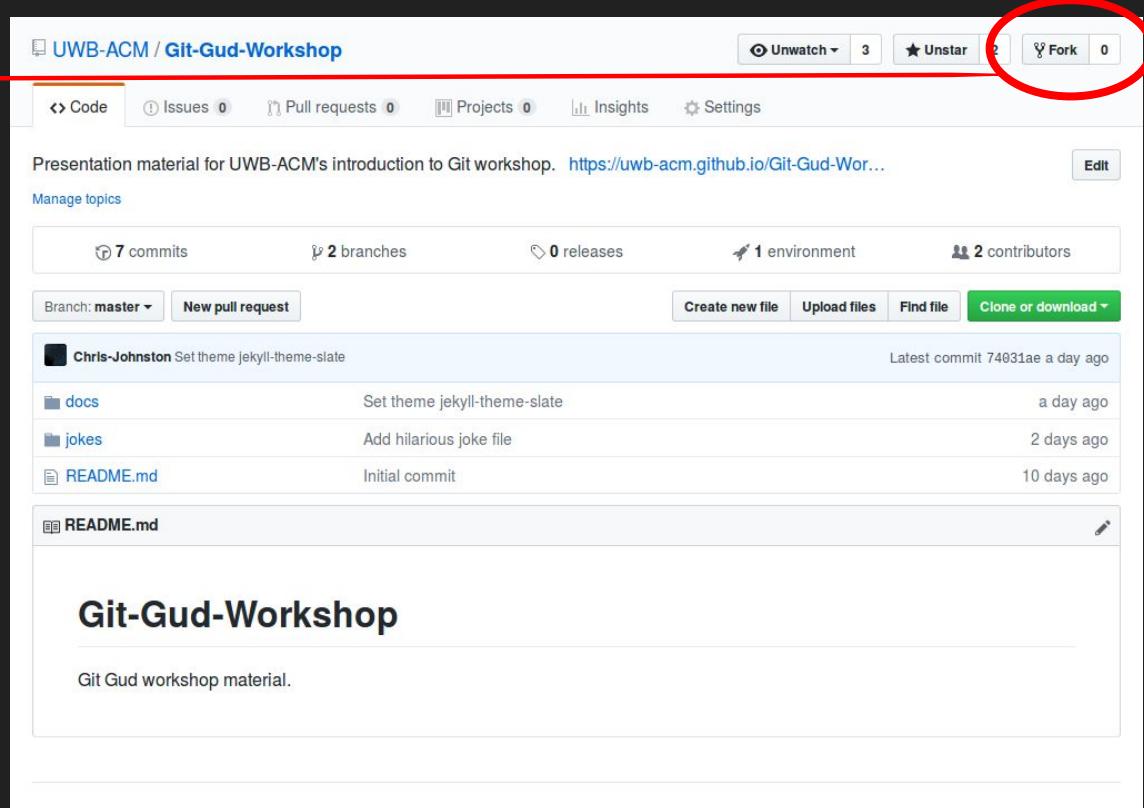
```
git clone <remote_forked_url>
cd folder
git checkout -b <branch_name>
echo "<joke>" >> jokes/BobbyT.txt
git add jokes/BobbyT.txt
git commit -m "<comment>"
git push origin <branch_name>
Open pull request [on GitHub]
```

(1 of 6) Forking the repo

Use the Fork button!

This will give us a copy

Use the forked copy to modify project without interrupting upstream development



The screenshot shows a GitHub repository page for "UWB-ACM / Git-Gud-Workshop". A red circle highlights the "Fork" button in the top right corner of the header. The repository has 7 commits, 2 branches, 0 releases, 1 environment, and 2 contributors. The README.md file contains the text "Git Gud workshop material.".

UWB-ACM / Git-Gud-Workshop

Code Issues 0 Pull requests 0 Projects 0 Insights Settings

Unwatch 3 Unstar 2 Fork 0

Presentation material for UWB-ACM's introduction to Git workshop. <https://uwb-acm.github.io/Git-Gud-Wor...>

Edit

Manage topics

7 commits 2 branches 0 releases 1 environment 2 contributors

Branch: master New pull request Create new file Upload files Find file Clone or download

Chris-Johnston Set theme jekyll-theme-slate Latest commit 74031ae a day ago

docs Set theme jekyll-theme-slate a day ago

jokes Add hilarious joke file 2 days ago

README.md Initial commit 10 days ago

README.md

Git-Gud-Workshop

Git Gud workshop material.

(2 of 6) Cloning the repo (part 1)

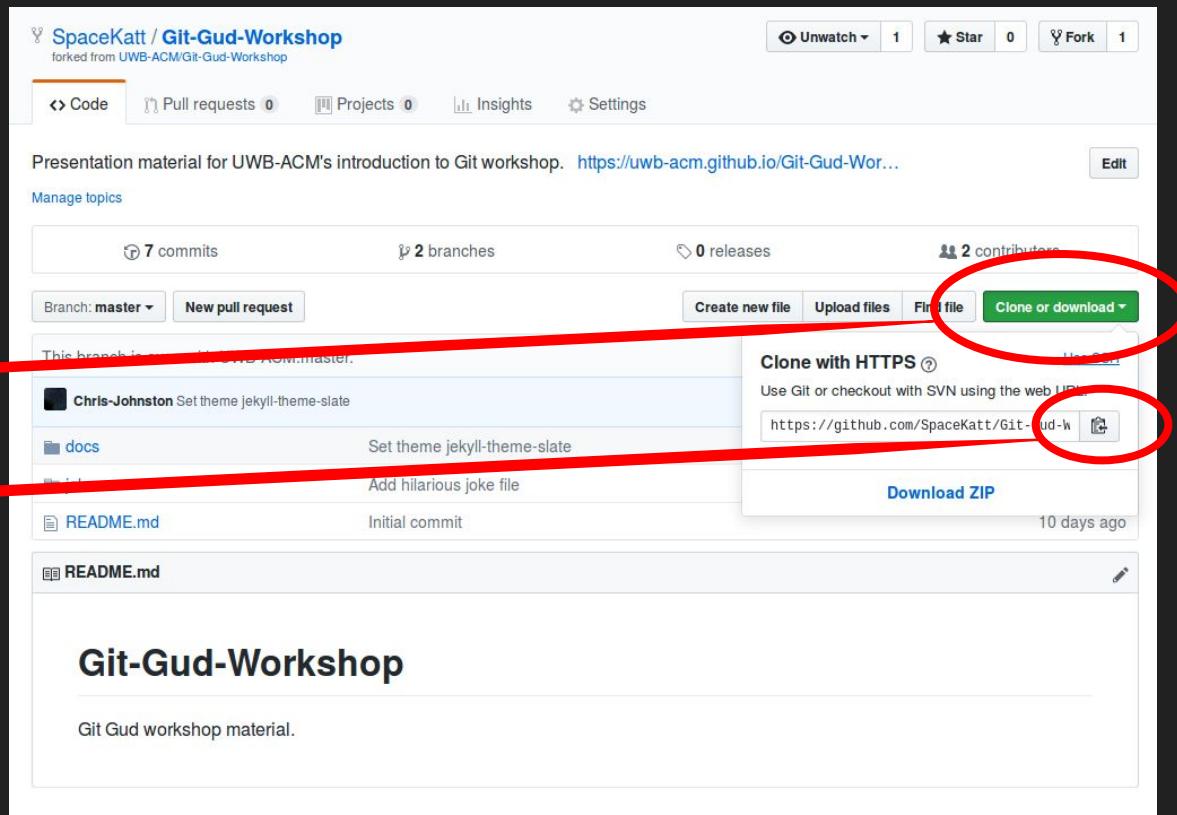
To clone your forked copy...

Copy remote's URL

View URL

Copy to clipboard

We use this URL on next slide



(3 of 6) Cloning the repo (part 2)

Cloning using HTTPS (Use this!)

```
12:13:38:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp$ git clone https://github.com/SpaceKatt/Git-Gud-Workshop.git
Cloning into 'Git-Gud-Workshop'...
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 25 (delta 2), reused 14 (delta 0), pack-reused 0
Unpacking objects: 100% (25/25), done.
Checking connectivity... done.
12:13:42:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp$ ls
. . . Git-Gud-Workshop
```

[Don't use this for the workshop] Cloning using SSH (requires setting up SSH keys)

```
12:11:46:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp$ git clone git@github.com:SpaceKatt/Git-Gud-Workshop.git
Cloning into 'Git-Gud-Workshop'...
Enter passphrase for key '/home/spacekatt/.ssh/id_rsa':
remote: Enumerating objects: 25, done.
remote: Counting objects: 100% (25/25), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 25 (delta 2), reused 14 (delta 0), pack-reused 0
Receiving objects: 100% (25/25), done.
Resolving deltas: 100% (2/2), done.
Checking connectivity... done.
12:11:52:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp$ ls
. . . Git-Gud-Workshop
```

(4 of 6) Making a Joke

Create a new branch, with your first name and last initial

Create a file inside the jokes directory, with your first name and last initial

jokes/<FirstName><LastInitial>.txt

Inside it, include a funny joke. If it is not funny, you **will** be asked to leave.

Commit it

```
12:19:19:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp$ cd Git-Gud-Workshop/jokes/
12:19:21:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git checkout -b SpaceK
Switched to a new branch 'SpaceK'
12:19:30:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ vi SpaceK.txt
12:19:38:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git add SpaceK.txt
12:19:42:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git status
On branch SpaceK
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   SpaceK.txt

12:19:44:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git commit -m "Add best joke, ever"
[SpaceK 7b52c35] Add best joke, ever
 1 file changed, 2 insertions(+)
```

(5 of 6) Pushing your joke

Push your branch to remote

Remote is by default “origin”, from when we cloned the repo

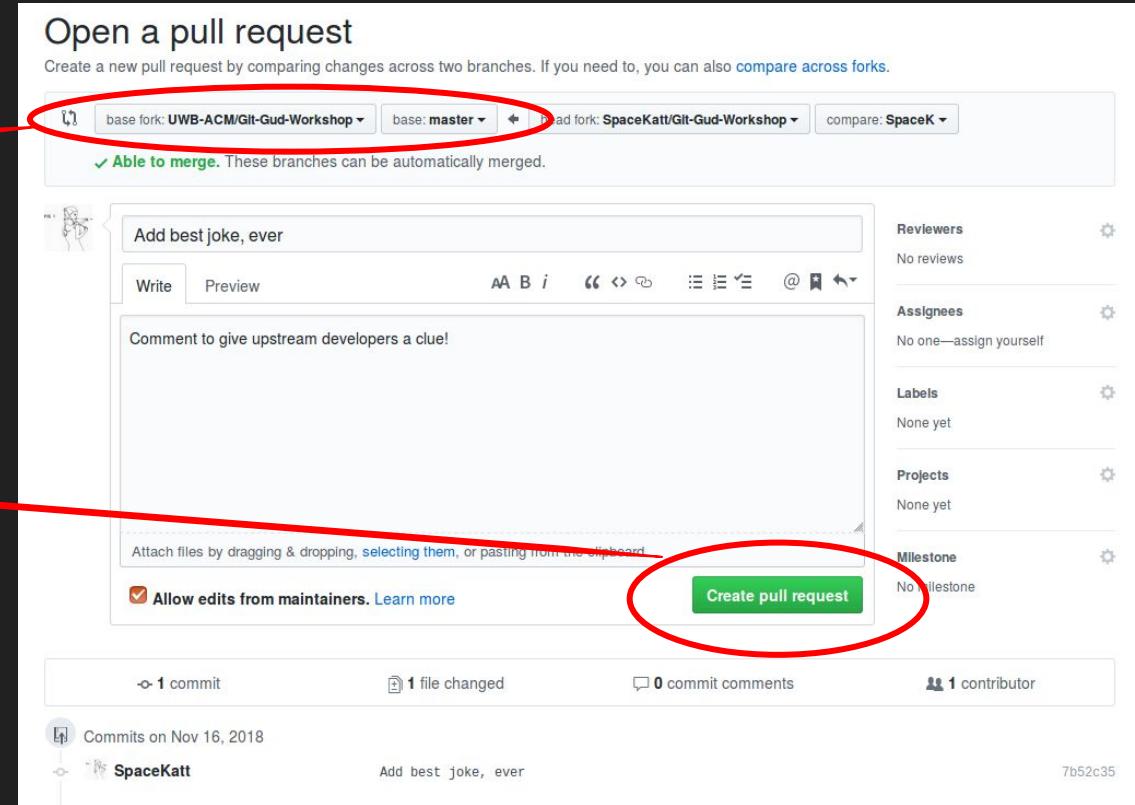
For the next step (making a pull request) visit the URL shown below

```
12:20:01:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git push origin SpaceK
Username for 'https://github.com': spacekatt
Password for 'https://spacekatt@github.com':
Counting objects: 4, done.
Delta compression using up to 8 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 424 bytes | 0 bytes/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
remote:
remote: Create a pull request for 'SpaceK' on GitHub by visiting:
remote:     https://github.com/SpaceKatt/Git-Gud-Workshop/pull/new/SpaceK
remote:
To https://github.com/SpaceKatt/Git-Gud-Workshop.git
 * [new branch]      SpaceK -> SpaceK
```

(6 of 6) Creating a pull request

- Ensure “base fork” is set to master from “upstream”
 - Upstream is remote you forked repo from
- Leave a comment
- Click “Create pull request”

Now others can comment on your changes and approve them for integration into the code base.



Collaborate!!!

Pull requests are a great way to collaborate with your teammates

- Code reviews
- Code state management
- Greater visibility of changes

Add my favorite joke #1

Merged Chris-Johnston merged 2 commits into master from branchy_mcface 9 days ago

Conversation 3 Commits 2 Checks 0 Files changed 1 +2 -0

SpaceKatt commented 9 days ago Member + ...
What's the deal with commit messages?

Add my favorite joke ... 89e1b97

SpaceKatt commented 9 days ago Member + ...
Pull requests are such a great way to collaborate! 1

Chris-Johnston requested changes 9 days ago View changes
Please use Pascal Case in the filename: jokes/ThomasK.txt

Chris-Johnston assigned SpaceKatt 9 days ago

Chris-Johnston added the Joke label 9 days ago Conform to coding standards ...

Chris-Johnston approved these changes 9 days ago View changes
LGTM

Chris-Johnston merged commit 13f24d2 into master 9 days ago Revert

Chris-Johnston deleted the branchy_mcface branch 9 days ago Restore branch

Reviewers: Chris-Johnston ✓
Assignees: SpaceKatt
Labels: Joke
Projects: None yet
Milestone: No milestone
Notifications: Unsubscribe
You're receiving notifications because you were assigned.
2 participants: SpaceKatt
Lock conversation
You can opt in to our Beta for issue deletion. Try it.

Edit

After your changes get *merged*...

To sync back with master after someone merges your PR...

```
git remote add upstream <upstream_url>
```

(Add remote reference to fork's upstream)

```
git checkout master
```

(Switch back to master branch)

```
git pull upstream master
```

(Pull changes from remote [GitHub])

```
12:31:11:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git remote add upstream git@github.com:UWB-ACM/Git-Gud-Workshop.git
12:31:22:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
12:31:28:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git pull upstream master
Enter passphrase for key '/home/spacekatt/.ssh/id_rsa':
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From github.com:UWB-ACM/Git-Gud-Workshop
 * branch           master      -> FETCH_HEAD
 * [new branch]     master      -> upstream/master
Updating 74031ae..9634c6c
 fast forward
 jokes/SpaceK.txt | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 jokes/SpaceK.txt
```

See our new file!?!?

Part II: After your changes get merged...

Update our remote fork after we fetch upstream changes

git push origin master (Synchronize our forked remote with changes from upstream)

```
12:31:50:spacekatt@spacekatt-Latitude-E6540:~/ACM/tmp/Git-Gud-Workshop/jokes$ git push origin master
Username for 'https://github.com': spacekatt
Password for 'https://spacekatt@github.com':
Counting objects: 1, done.
Writing objects: 100% (1/1), 655 bytes | 0 bytes/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/SpaceKatt/Git-Gud-Workshop.git
 74031ae..9634c6c  master -> master
```

Git Goodness

More useful things to know about git

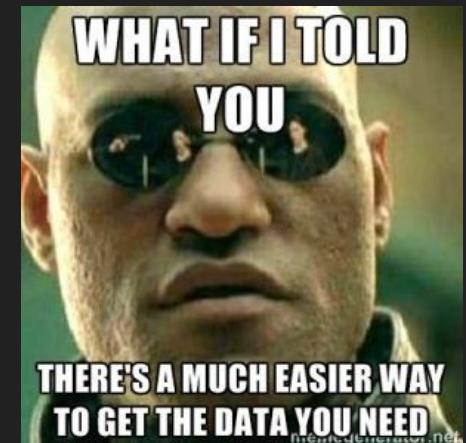
Git Flow

“Git flow” is a pattern for how you can collaborate with others on the same codebase. We just did this!

GitHub explains this better than we can: (open this link)

<https://guides.github.com/introduction/flow/>

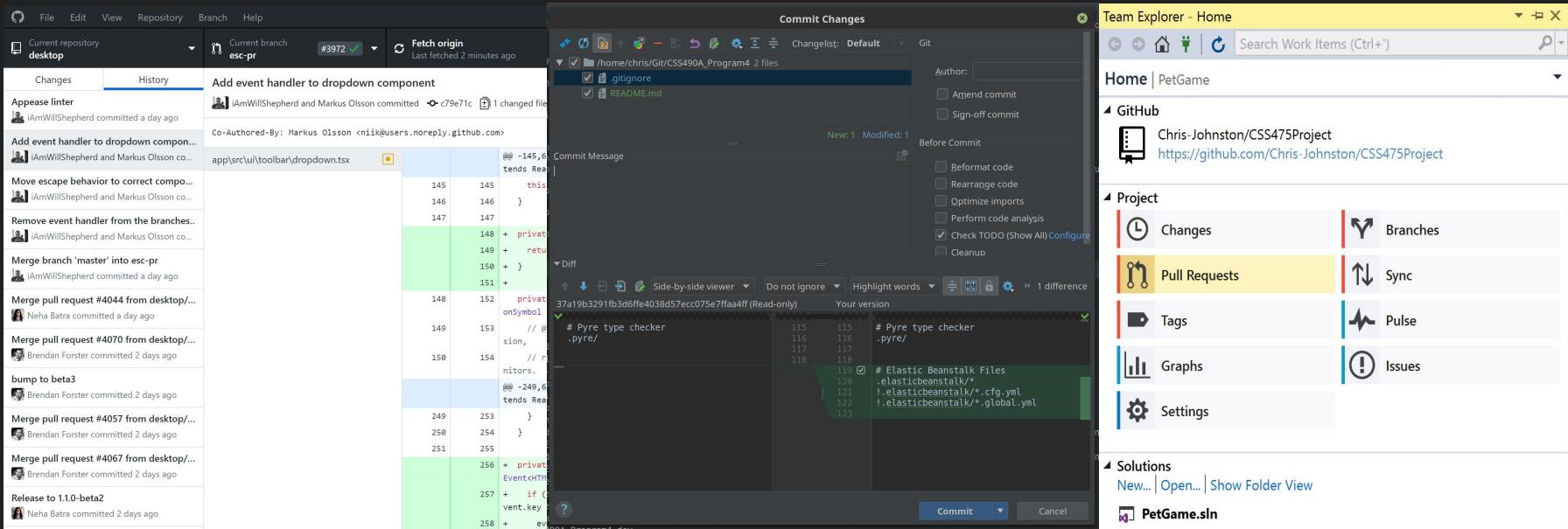
<https://goo.gl/jgeusu>



Git GUIs

Nearly every modern IDE will have a Git plugin

(so does vim)



GitHub Desktop is a Git client that is oriented around GitHub. It works with non-GitHub repos too.

The JetBrains IDEs (IntelliJ, PyCharm, CLion) have pretty good Git integration.

Visual Studio has a pretty solid Git integration.

Further Topics

GitHub Issues

Markdown & making a
good readme

Continuous Integration &
Continuous Deployment

Releases / tags

Licensing

In-Depth Merging

Rebasing/merge conflicts

Git Stash

Forking a repo

.gitignore

.gitattributes

Github-specific stuff

... and a lot more.

If you have *questions*
right now, we can try to
give a brief explanation.

We encourage you to learn these on your own!
Practice! Try using Git for your programming
assignments.



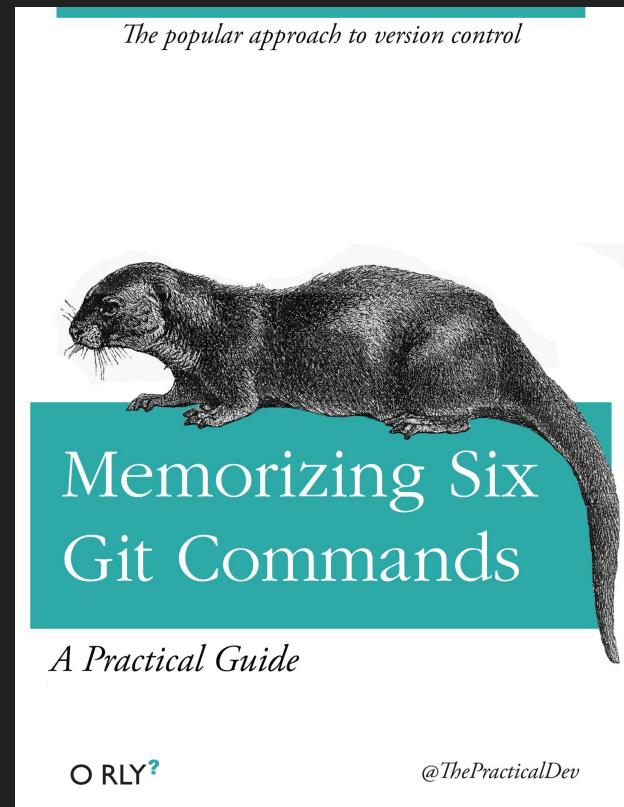
Great Resources for Further Reading

The official Git documentation: <https://git-scm.com/doc>

GitHub documentation: <https://guides.github.com/>

Atlassian Git tutorials: <https://www.atlassian.com/git/tutorials>

And of course: Google, Stack Overflow, and your peers!



Lessons Learned

- Why git is important
- How to configure git CLI
- How to make a commit
- How to push local repository to remote (GitHub)
- How to manage files with git
- How to contribute to open source projects
- Identified more advanced topics for further consideration
- **Git is fun!**

Anything from the previous
slides or beyond?

Questions?
Comments?
Concerns?