

Git tooty  
(fart joke?)

Silas Tittes

15 September, 2016

# Outline

- ▶ I don't git git, why should I do this?

# Outline

- ▶ I don't git git, why should I do this?
- ▶ Working with a remote repository – GitHub

# Outline

- ▶ I don't `git git`, why should I do this?
- ▶ Working with a remote repository – GitHub
- ▶ Working with a local repository – `git`

# Outline

- ▶ I don't `git git`, why should I do this?
- ▶ Working with a remote repository – GitHub
- ▶ Working with a local repository – `git`
- ▶ Version control *basics* – reviewing file histories

# Outline

- ▶ I don't git git, why should I do this?
- ▶ Working with a remote repository – GitHub
- ▶ Working with a local repository – git
- ▶ Version control *basics* – reviewing file histories
- ▶ Where to go from here

# I don't git it, why should I do this?

- ▶ Version control – because you *will* screw something up.



# I don't git it, why should I do this?

- ▶ Version control – because you *will* screw something up.
- ▶ Collaboration – because *everything* is better together.





# I don't git it, why should I do this?

- ▶ Version control – because you *will* screw something up.
- ▶ Collaboration – because *everything* is better together.
- ▶ Remote backup – because your computer *will* die.



# I don't git it, why should I do this?

- ▶ Version control – because you *will* screw something up.
- ▶ Collaboration – because *everything* is better together.
- ▶ Remote backup – because your computer *will* die.
- ▶ Public portfolio – because you *should* demonstrate your awesomeness to the world.

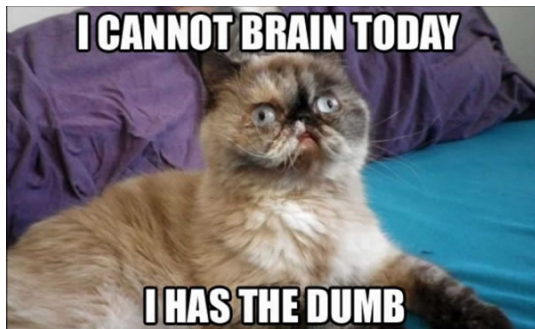


# I don't git it, why should I do this?

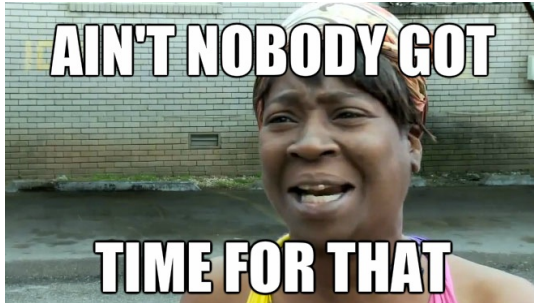
- ▶ Version control – because you *will* screw something up.
- ▶ Collaboration – because *everything* is better together.
- ▶ Remote backup – because your computer *will* die.
- ▶ Public portfolio – because you *should* demonstrate your awesomeness to the world.
- ▶ Encourages reproducibility and creativity.



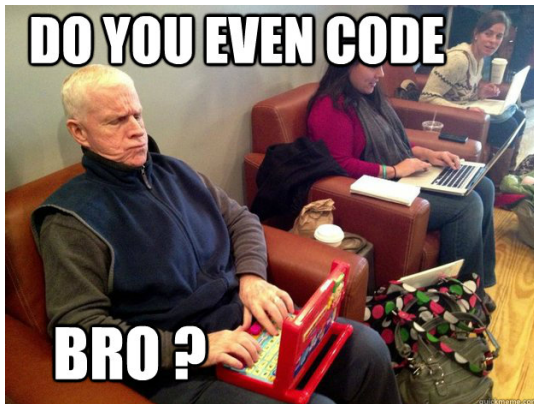
What if people look at my code and think I'm dumb?



What if people look at my code and think I'm dumb?



I'm not really a coder



# You are, but even if you're "not"

"GitHub is used to manage the collaborative development of recipes, musical scores, books, fonts, legal documents, lessons and tutorials, and data sets"

## GitHub for the rest of us



*Credit: Flickr/Nick Guaranto*

Git made it possible for programmers to coordinate distributed work across teams -- now GitHub makes it possible for everyone else



By **Jon Udell** | Follow  
InfoWorld | Feb 24, 2015

Working with a remote repository – GitHub (or  
BitBucket)

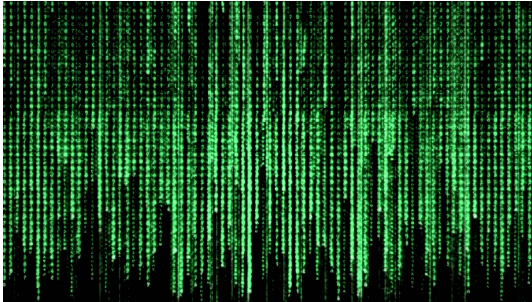


Let's make a GitHub log-in and repository!



Working with a local repository – git

Enter the matrix



## Clone your new remote repo

```
git clone https://github.com/octocat/lovecatz.git
```

Now make some changes to your README.md on your computer, or create a new file. Perhaps `hello.cpp` if you're feeling real crazy.

## Git used to it – the basics

You've made some local repo edits to some code or documentation, what now?

*#See what's changed*

`git status`

*#prepare the content for the next commit -- "staging"*

`git add --all` *# --all, or specify files*

*#"The Rub" -- Record changes to the repository*

*#This is what creates version control*

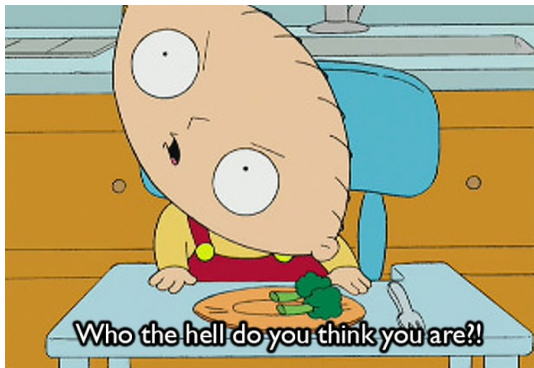
`git commit -m "short discription of code changes"`

*#sync local and remote repos*

`git push -u origin master`

## Error!

```
git config --global user.name "Octocat69"  
git config --global user.email "Octocat69@bbqchikn.com"
```



## Back on track

*#sync local and remote repos*

```
git push -u origin master
```

*#Up to date?*

```
git status
```

Check out remote GitHub repo to see updates.

## Practice the “add, commit, push” cycle 4X



add, commit, push

add, commit, push

add, commit, push

add, commit, push



Version control basics – reviewing file histories

Where to go from here