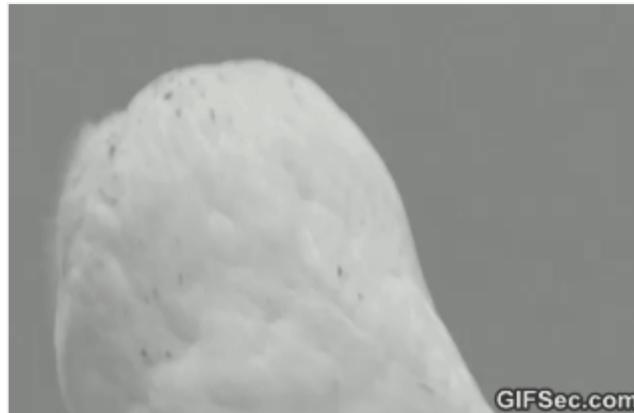


CLI, VIM, GIT, OMG, WTF, BBQ!?

THE MISSING MANUAL TO OPEN
SOURCE TOOLS

Insert Cygwin-Install loop here



GIFSec.com

OBJECTIVES:

- very basics of CLI
 - navigate
 - create
- very basics of vim
 - navigate
 - basic editing
- very basics of git
 - create/clone repos
 - push & pull remotes



~Break~

- CLI part deux
 - discover
 - modify
- vim part deux
 - .bashrc
 - editing tips

Setting up

check on:

- Internet
- shell

any people considerations:

- Can everyone see?
- Hear?
- Am I talking too quickly?

Wifi & password:



Why should I learn about this stuff? - CLI

Power!

- “GUIs make easy tasks easy-- CLI makes difficult tasks possible”
- rename files en masse
- log onto a computer remotely and access its files
- see or change the permissions on a file
- find all files of a given extension or any file that was created within the last week
- set up a webserver
- schedule a script to run at set intervals (cron job)



Why should I learn about this stuff? - Vim

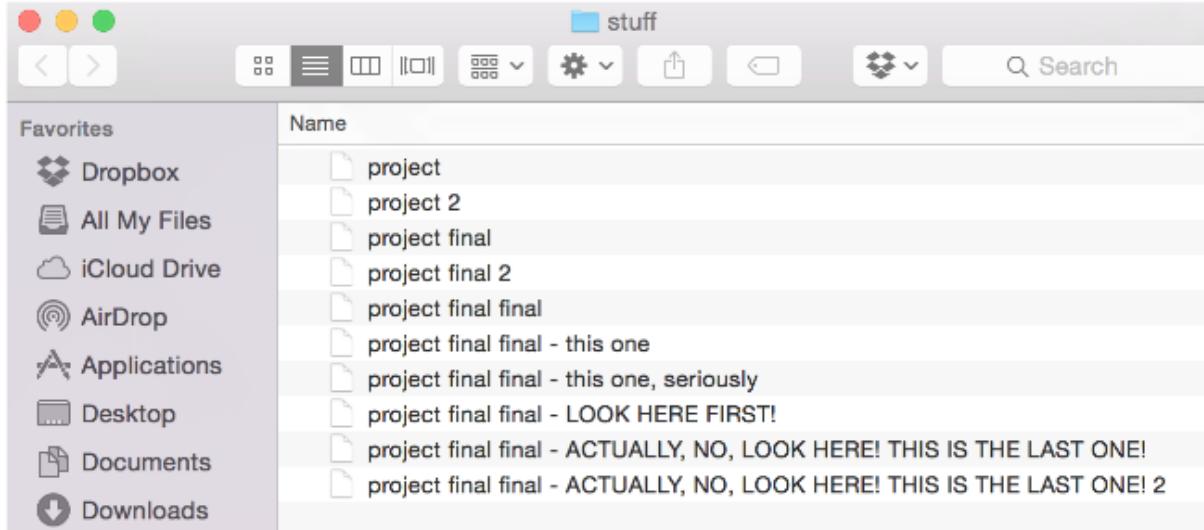
*disclaimer

- ubiquity
- home key speed
- lightweight



Why should I learn about this stuff? - Git

The Only Naming Convention That Works



- standard for OS collaboration
 - small commits = best practice

DOGHOUSEDIARIES

LET'S PARTY!

PARTY HARD



very basics of CLI: Table of Comments (Toc)

- what is CLI
- moving, manipulating,
reading commands
- flags, help
- q/a/cats



very basics of CLI: what is CLI?

- command-line interpreter
- command line shell*
- command language interpreter

Interested?

[*In The Beginning was the Command Line, Neal Stephenson*](#)

A more direct way to communicate
with your computer

very basics of CLI: moving, manipulating, reading

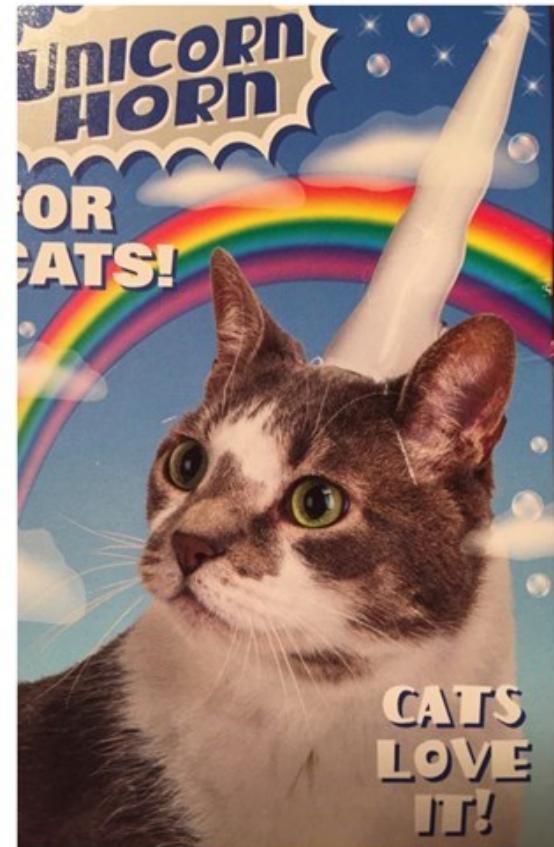
- pwd
- ls
- cd(..)
- mkdir
- touch
- less
- cp
- mv
- rm



very basics of CLI: flags, help

- rm -r -f (--recursive --force)
- ls -a -l (--all --long)
- cp -r (--recursive)
- q (quit)
- d (done)
- ctl+c (close) (ping)
- ctl+d (done) (cat)
- apropos disk
- man mkdir
- help cd (for shell
built in types)

very basics of CLI: q/a/cats



very basics of vim: ToC

- modes
- moving
- visual mode
- replacing
- q/a/cats

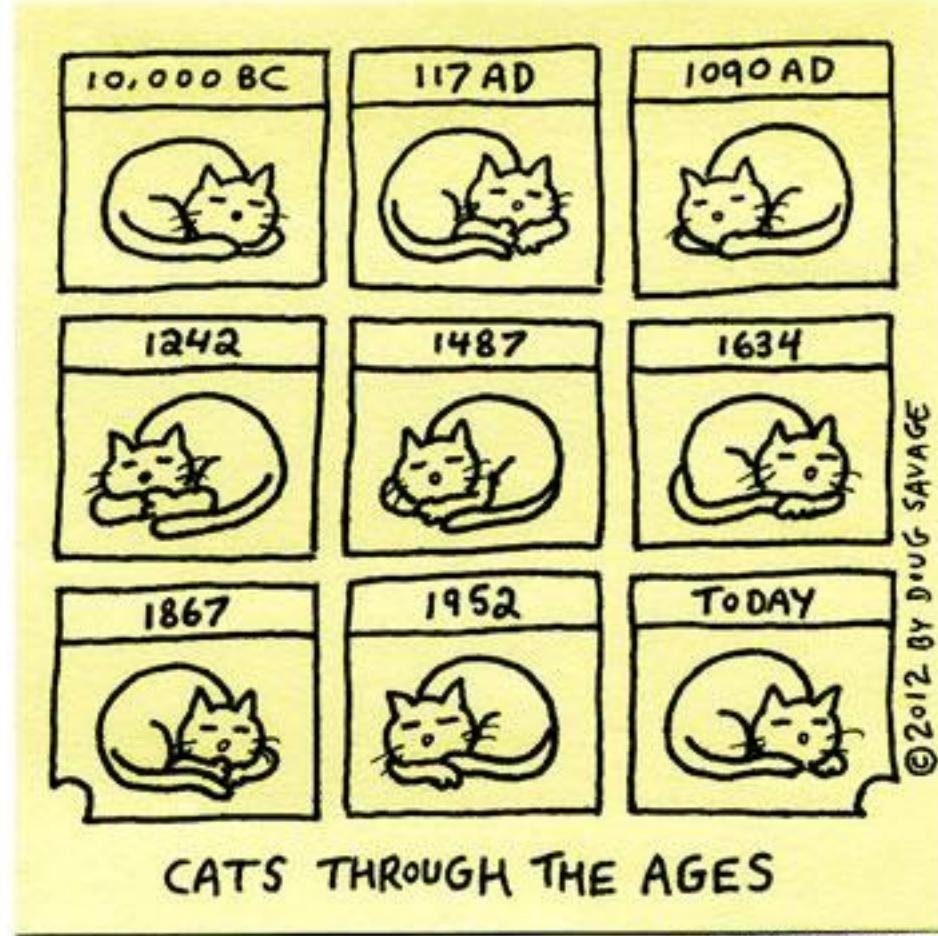


very basics of vim: a very, very, very brief history of vim

- Vi
- Bram Moolenaar wrote V
Improved

Savage Chickens

by Doug Savage



very basics of vim: modes

- Command
- Insert
- Visual



BASICS		I LIKE TO MOVE IT MOVE IT	
:wq	write and quit	gg / G	takes you to the beginning / end of the file
:q!	quit without writing	0 / \$	takes you to the beginning / end of the line
i	enter Insert mode from Command mode	w / b	takes you to the next / previous word
esc	exit Insert mode & return to Command mode	DELETE!	
v	from command mode to visual mode	dw	to delete a word
u	undo!	d\$	to delete from where you are to the end of the line
ctl + r	redo	VISUAL EFFECTS	
DIRECTIONS		y	ends and yanks your selection
h	to move the cursor one space left	p	pastes your selection
j	to go down one line	yy	yanks the line you are on
k	to go up one line	yap	yanks around paragraph
l	to go one space right		

very basics of vim: q/a/cats



GIFsBOOM.net

very basics of git

- a brief history
- setup github accounts
- git basics
- q/a/cats

*“it’s not you,
it’s git”*



Randall Hansen @sonofhans · 3h

@benjammingh "Git: it doesn't only hurt the first time."



[View conversation](#)

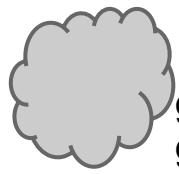
very basics of git: a brief history

BitKeeper went private, and Linus Torvalds needed a version control system for the Linux Kernel



very basics of git: set up github accounts

- `git config --global --list`
- `git config --global user.name "bob"`
- `git config --global user.email "bob@youruncle.com"`

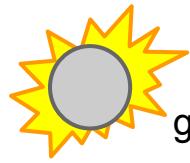


git init

git log (don't panic)



```
git status  
git add foo  
git status
```



git commit

Commit message conventions!



touch bar
git add bar



git init
git log (don't panic)
touch foo
git status
git add foo
git status
git commit



touch bar
git add bar
git commit

git init
git log (don't panic)
touch foo
git status
git add foo
git status
git commit



vim bar (edit)
git add bar



touch bar
git add bar
git commit



git init
git log (don't panic)
touch foo
git status
git add foo
git status
git commit



vim bar (edit)
git add bar
git commit



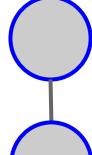
touch bar
git add bar
git commit



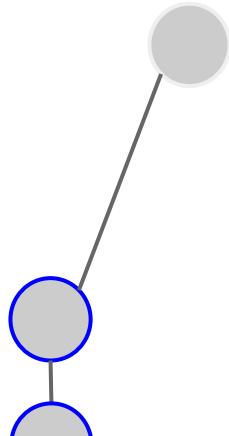
git init
git log (don't panic)
touch foo
git status
git add foo
git status
git commit

branches



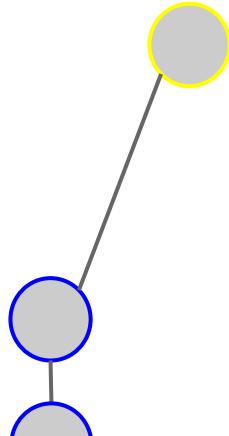


```
git init  
touch foo  
git add foo  
git status  
git commit
```



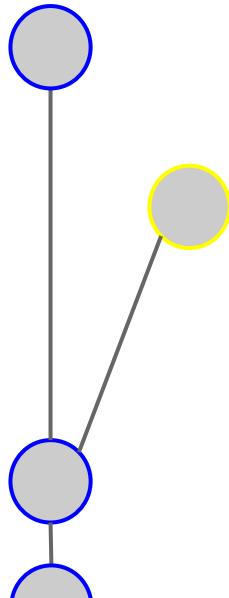
git init
touch foo
git add foo
git status
git commit

git branch => git branch science
git checkout science => git branch
touch baz
git add baz



git init
touch foo
git add foo
git status
git commit

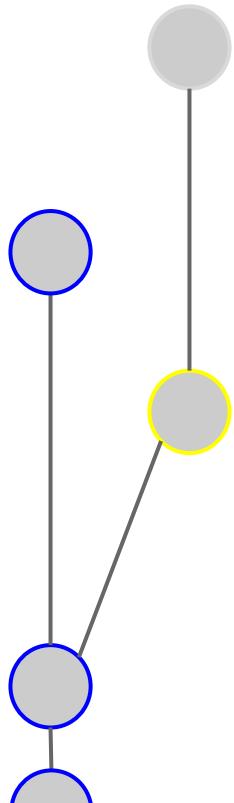
git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log



git init
touch foo
git add foo
git status
git commit

git checkout master
git log
touch, add, & commit qux

git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log

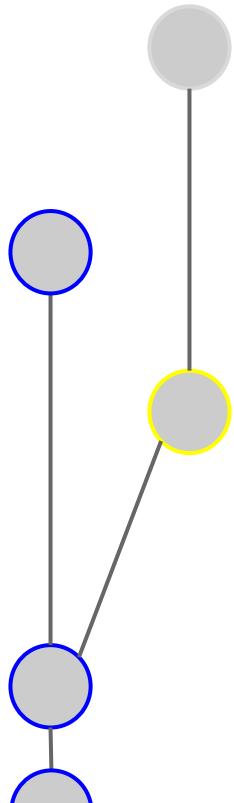


git init
touch foo
git add foo
git status
git commit

git checkout master
git log
touch, add, & commit qux

git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log

git checkout science => git branch
edit & add baz

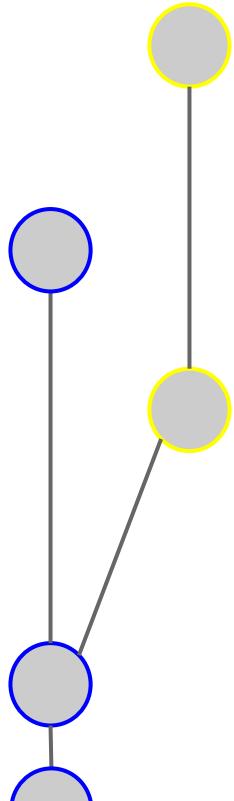


git init
touch foo
git add foo
git status
git commit

git checkout master
git log
touch, add, & commit qux

git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log

git checkout science => git branch
edit & add baz
git merge master

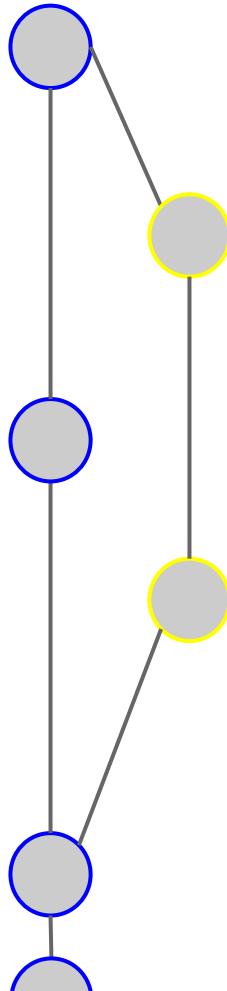


git init
touch foo
git add foo
git status
git commit

git checkout master
git log
touch, add, & commit qux

git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log

git checkout science => git branch
edit & add baz
git merge master
git commit



git checkout master
git merge science

git checkout science ⇒ git branch
git rebase master
edit, add, & commit baz
git commit

git checkout master
git log
touch, add, & commit qux

git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log

git init
touch foo
git add foo
git status
git commit

remote repositories (yours)

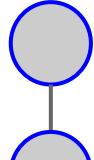


local

(your laptop)

origin

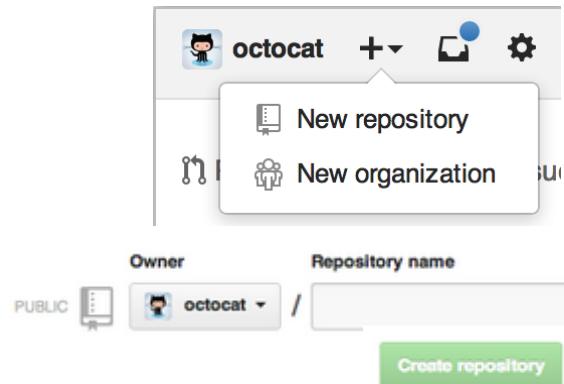
(your fork)



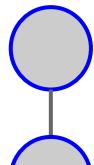
git commit -m 'last thing we did'

local
(your laptop)

origin
(your fork)



create a new repository on github.com

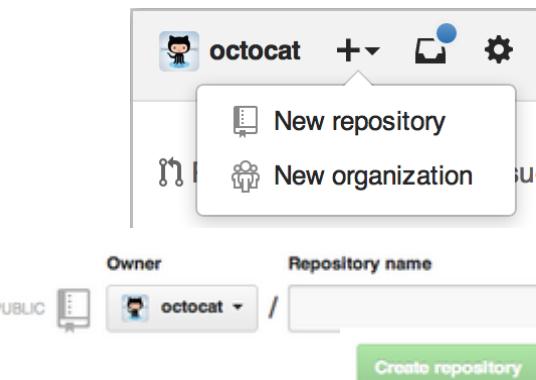


git commit -m 'last thing we did'

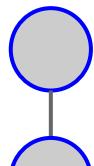


local
(your laptop)

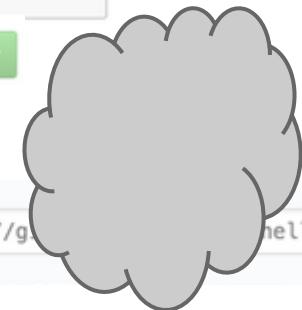
origin
(your fork)



create a new repository on github.com
copy the URL

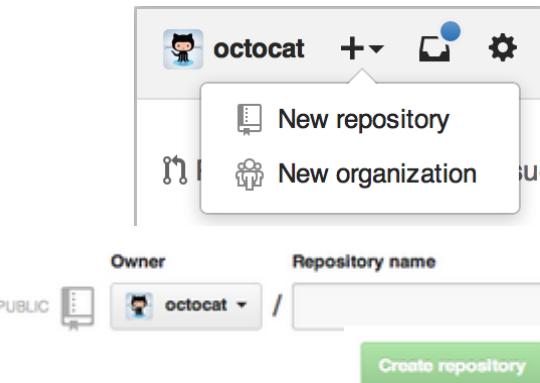


git commit -m 'last thing we did'

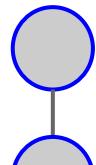


local
(your laptop)

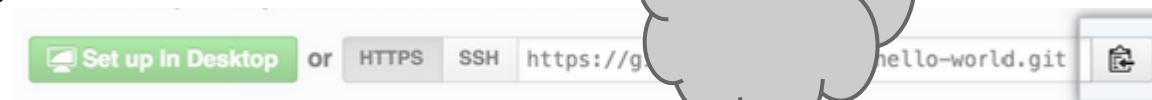
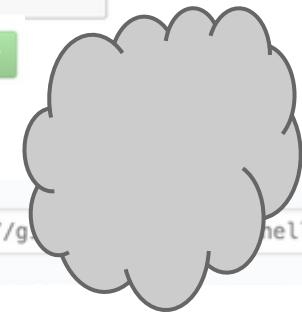
origin
(your fork)



create a new repository on github.com
copy the URL

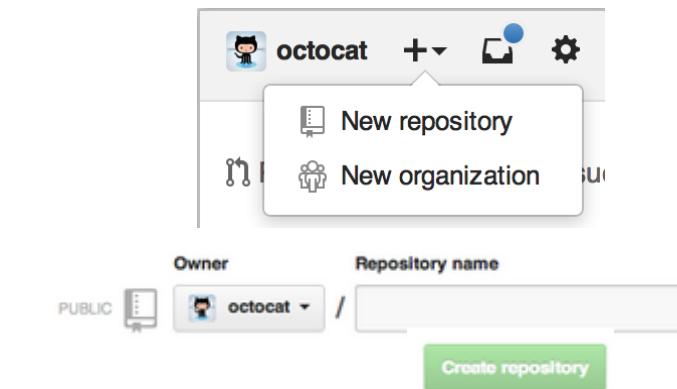


git commit -m 'last thing we did'
git remote add origin *URL*



local
(your laptop)

origin
(your fork)



create a new repository on github.com
copy the URL

git commit -m 'last thing we did'
git remote add origin *URL*
git push origin master

local
(your laptop)

origin
(your fork)



change, add, & commit...



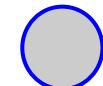
```
git commit -m 'last thing we did'  
git remote add origin URL  
git push origin master
```

create a new repository on github.com
copy the URL

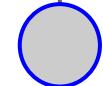


local

(your laptop)



change, add, & commit...
git push origin master



git commit -m 'last thing we did'
git remote add origin *URL*
git push origin master

origin

(your fork)



oh look! all of our commits!

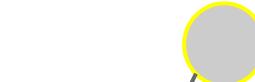


create a new repository on github.com
copy the URL



local

(your laptop)



```
git checkout -b moar_science  
git commit
```



```
change, add, & commit...  
git push origin master
```

origin
(your fork)



oh look! all of our commits!

Create repository

create a new repository on github.com
copy the URL



```
git commit -m 'last thing we did'  
git remote add origin URL  
git push origin master
```



local

(your laptop)



```
git checkout -b moar_science  
git commit  
git push origin moar_science
```



change, add, & commit...
git push origin master

oh look! all of our commits!

origin

(your fork)



Create repository

create a new repository on github.com
copy the URL



```
git commit -m 'last thing we did'  
git remote add origin URL  
git push origin master
```



local

(your laptop)

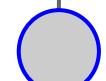


```
git checkout -b moar_science  
git commit  
git push origin moar_science
```



change, add, & commit...
git push origin master

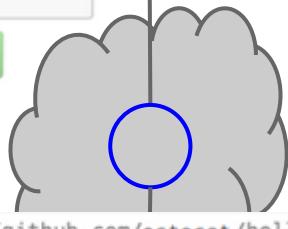
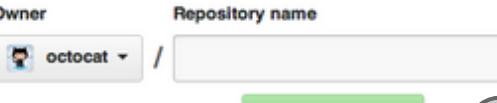
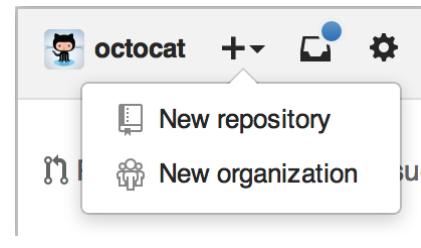
oh look! all of our commits!



```
git commit -m 'last thing we did'  
git remote add origin URL  
git push origin master
```

origin

(your fork)



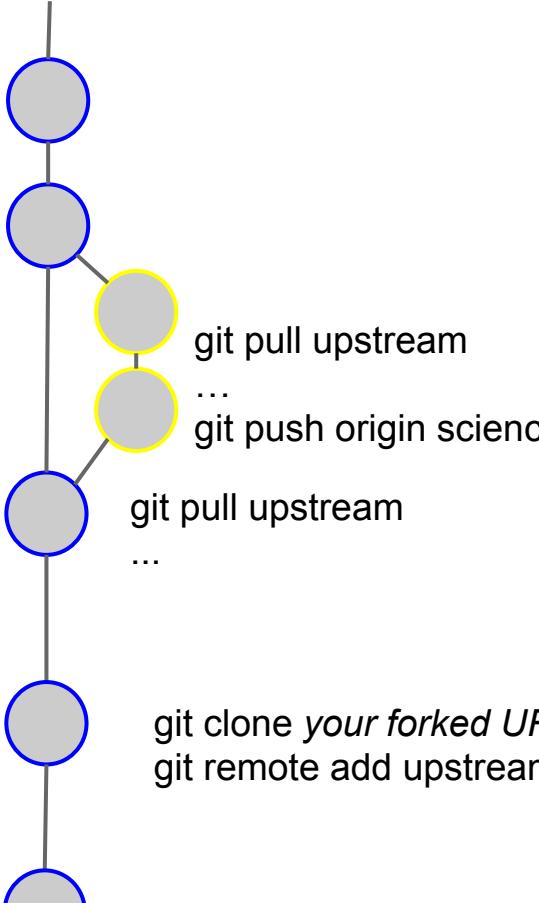
create a new repository on github.com
copy the URL



remote repositories (others')

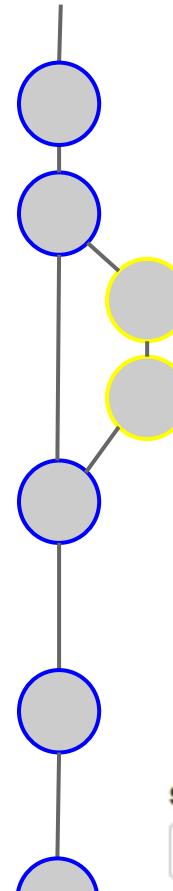


local (your laptop)

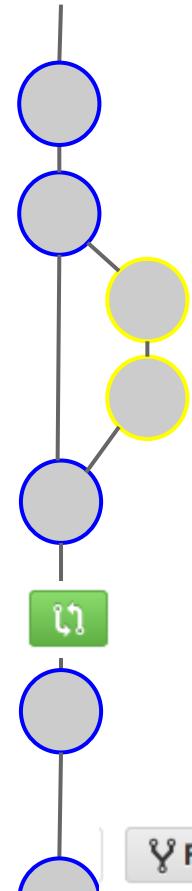


origin (your fork)

base fork = home base
head fork = on your shoulders



upstream (the main repo)



`git clone your forked URL`
`git remote add upstream original repo URL`

SSH clone URL

git@github.com:the

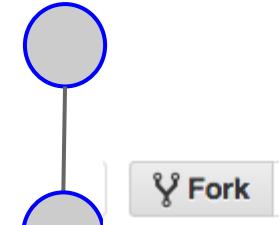


 Fork

local
(your laptop)

origin
(your fork)

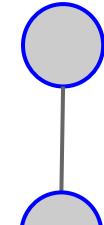
upstream
(the main repo)



local
(your laptop)

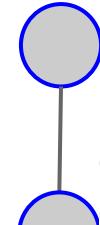
origin
(your fork)

upstream
(the main repo)



SSH clone URL

git@github.com:thel

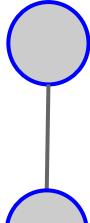


Fork

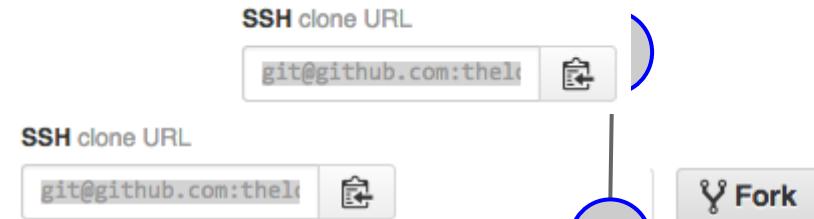
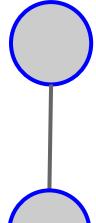
local
(your laptop)

origin
(your fork)

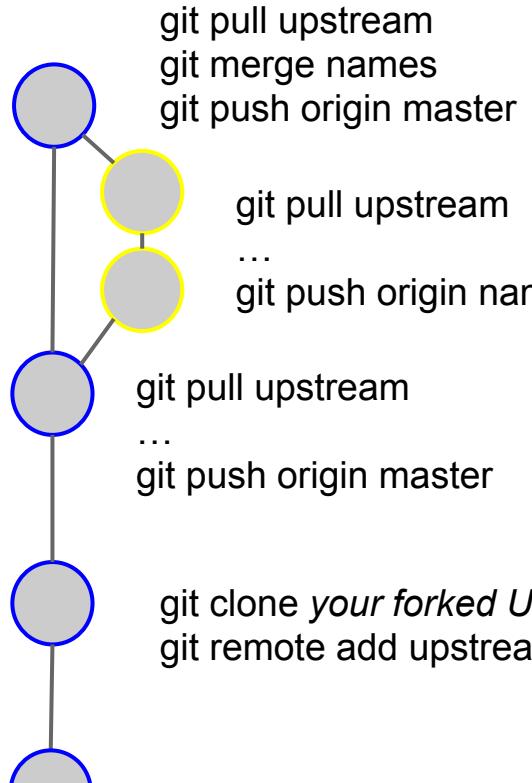
upstream
(the main repo)



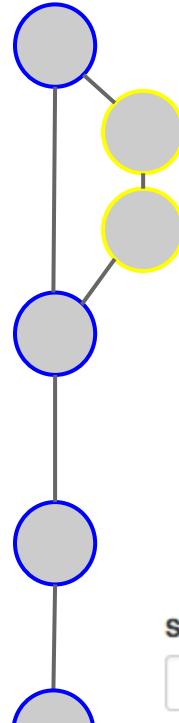
git clone *your forked URL*
git remote add **upstream** *original repo URL*
git remote



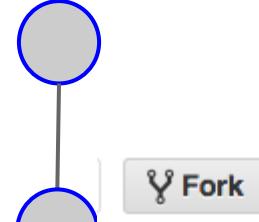
local (your laptop)



origin (your fork)



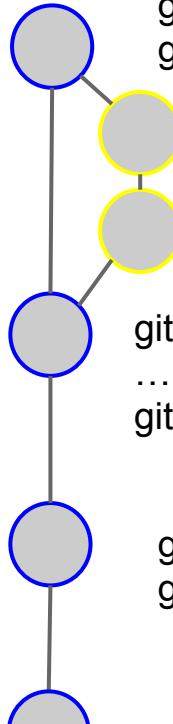
upstream (the main repo)



local (your laptop)

origin (your fork)

upstream (the main repo)



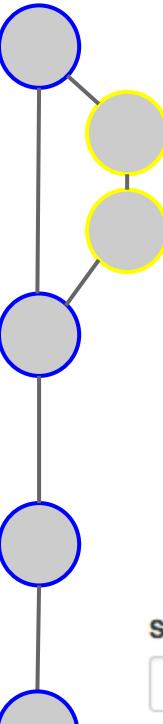
```
graph TD; Root1(( )) --- Node1(( )); Node1 --- Node2(( )); Node2 --- Node3(( )); Node3 --- Node4(( ));
```

git pull upstream
git merge science
git push origin master

git pull upstream
...
git push origin science

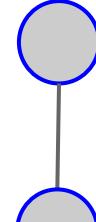
git pull upstream
...
git push origin master

git clone *your forked URL*
git remote add upstream *original repo URL*

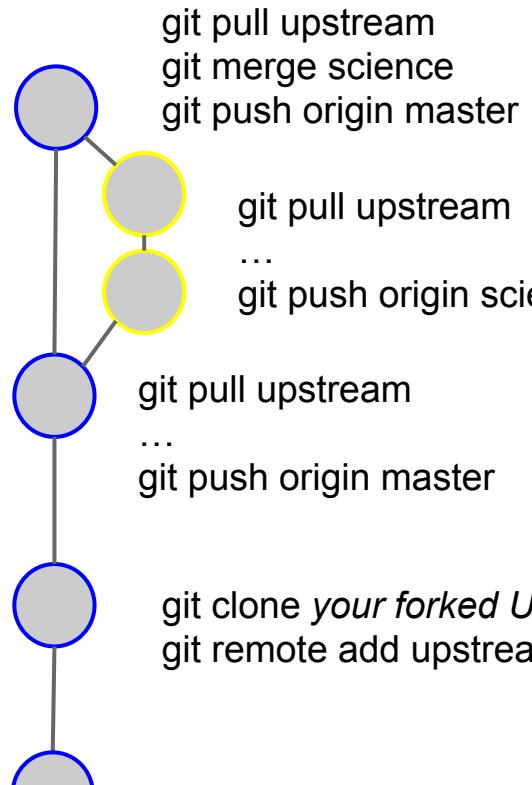


SSH clone URL

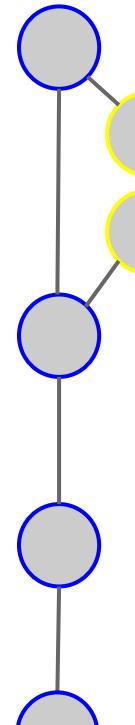
`git@github.com:thel...`



local (your laptop)

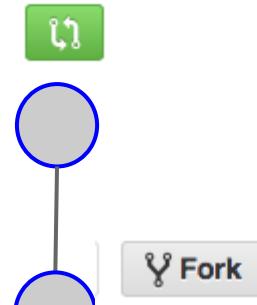


origin (your fork)

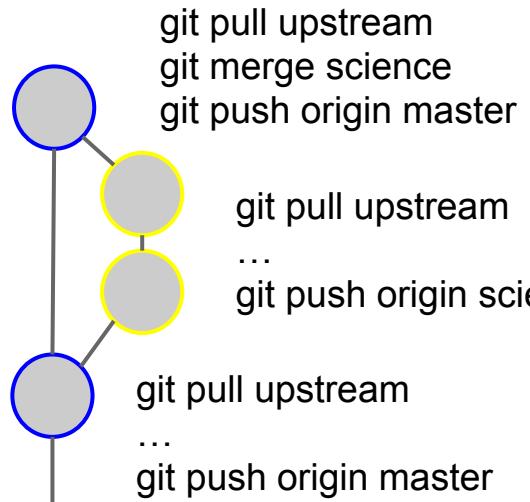


upstream (the main repo)

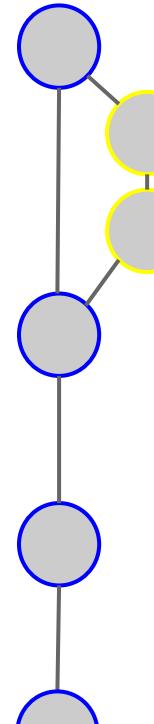
base fork = home base
head fork = on your shoulders



local (your laptop)



origin (your fork)

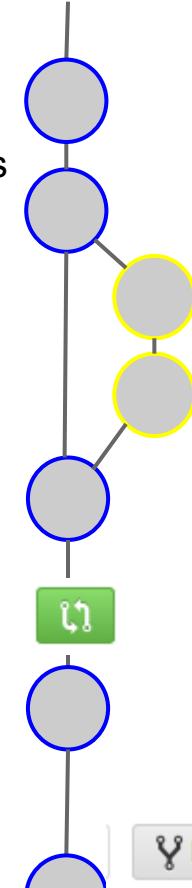


base fork = home base
head fork = on your shoulders

SSH clone URL

git@github.com:thele...

upstream (the main repo)



Fork

very basics of git: q/a/cats

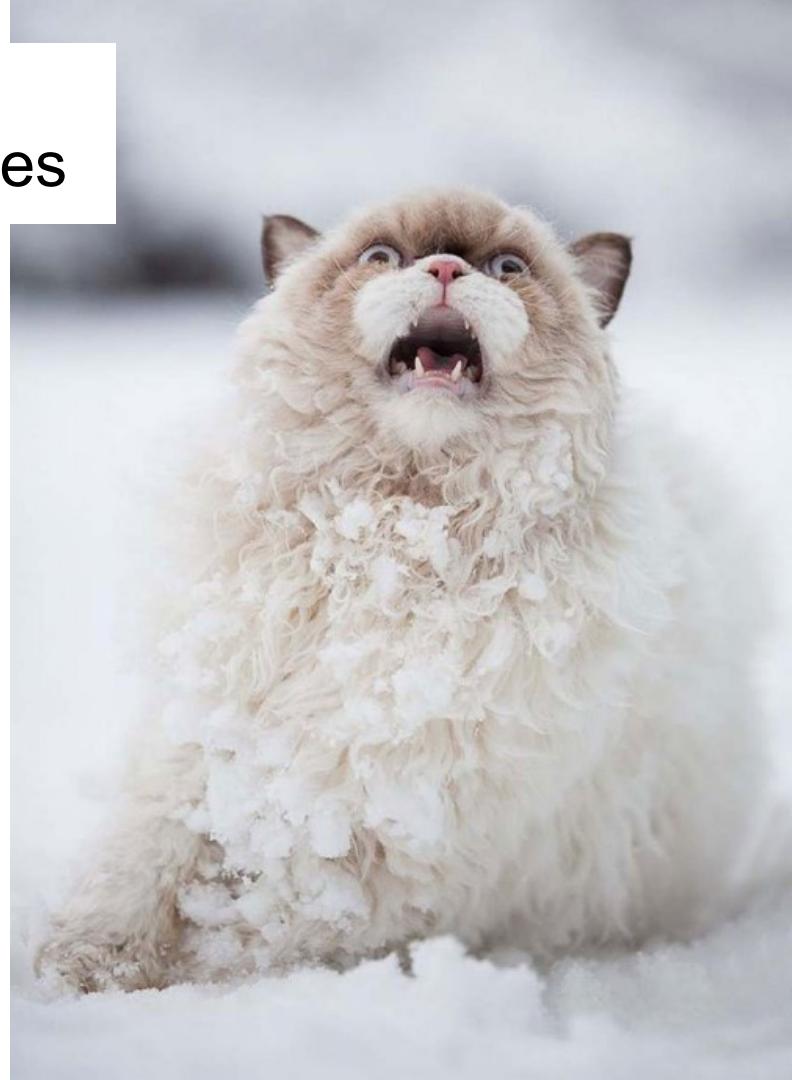


very basics of git: extra resources

<http://git-scm.com/book/en/v2>

Pro Git book, written by Scott Chacon and Ben Straub (in class repo under git directory)

cheat sheet - <https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf>



BREAK!
you did it!
now go on a break!

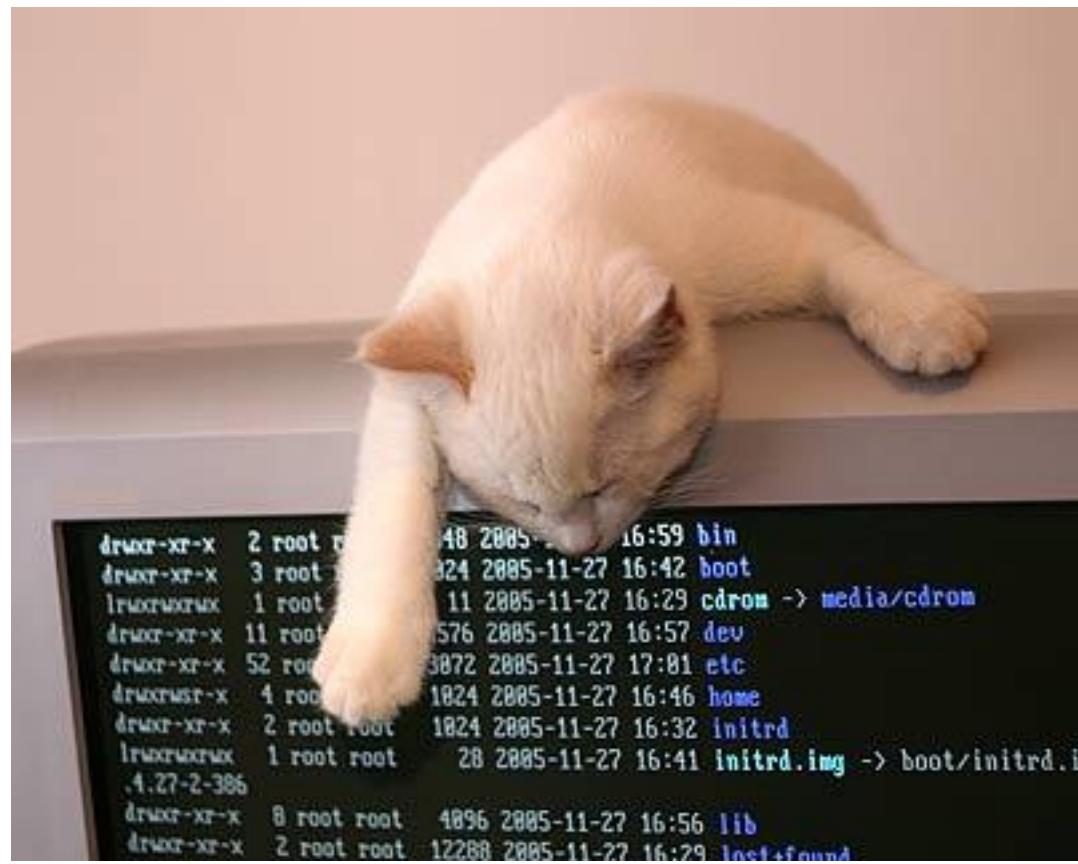


[INSERT LOOPING CAT GIFS HERE](#)



CLI part deux:

- cat
- wildcards
- grep
- ssh
- extra CLI resources
- q/a/cats



```
drwxr-xr-x 2 root root 48 2005-11-27 16:59 bin
drwxr-xr-x 3 root root 324 2005-11-27 16:42 boot
lrwxrwxrwx 1 root root 11 2005-11-27 16:29 cdrom -> media/cdrom
drwxr-xr-x 11 root root 576 2005-11-27 16:57 dev
drwxr-xr-x 52 root root 3872 2005-11-27 17:01 etc
drwxrwsr-x 4 root root 1024 2005-11-27 16:46 home
drwxr-xr-x 2 root root 1024 2005-11-27 16:32 initrd
lrwxrwxrwx 1 root root 28 2005-11-27 16:41 initrd.img -> boot/initrd.img
drwxr-xr-x 4.27-2-386 8 root root 4896 2005-11-27 16:56 lib
drwxr-xr-x 2 root root 12288 2005-11-27 16:29 lost+found
```

CLI part deux: cat

I must go.

cat file1	display the contents of file1
cat > file1 (text, enter, ctl+d)	write over file1 with text
cat file1 > file2	write over file1 with file2
cat file1 >> file2	add the content of file1 to the end of file2

My planet needs me.

CLI part deux: wildcards

*

?

[abc]

[a..d]

[!abc]

[:alnum:]

(& alpha/digit/lower/upper)



CLI part deux: grep

- grep = global regular expression print
- grep [-options] what_ur_lookin_4 where_ur_lookin
- . = single character
- ^ = begins with
- \$ = ends with
- diff -c file1 file2
 - - = file1
 - + = file2)
- comm file1 file2

```
$ grep --color t[a-z] midsummer
I know a bank where the wild thyme blow
Where oxlips and the nodding violet grow
Quite te over-canopied with luscious wood
With th sweet musk-roses and with eglan
$ █
```

IT'S A GREP!



```
ssh tiffany@remoteServer  
(enter password for server)  
exit
```

```
ssh-keygen -t rsa  
cd ~/.ssh  
id_rsa.pub = public key = lock  
id_rsa = private key = key
```

```
ssh-copy-id login@remoteServer  
      or  
scp ~/.ssh/id_dsa.pub  
login@remoteServer:.  
ssh/authorized_keys
```

CLI part deux: ssh

DO NOT SHARE YOUR PRIVATE KEY WITH ANYONE, NOT EVEN YOUR MOM AND ESPECIALLY NOT YOUR CAT!

Let's practice in GitHub

```
ssh-keygen -t rsa -C "tiffany@example.com"  
eval "$(ssh-agent -s)"
```

```
ssh-add ~/.ssh/id_rsa
```

```
pbcopy < ~/.ssh/id_rsa.pub (or just highlight &  
copy)
```

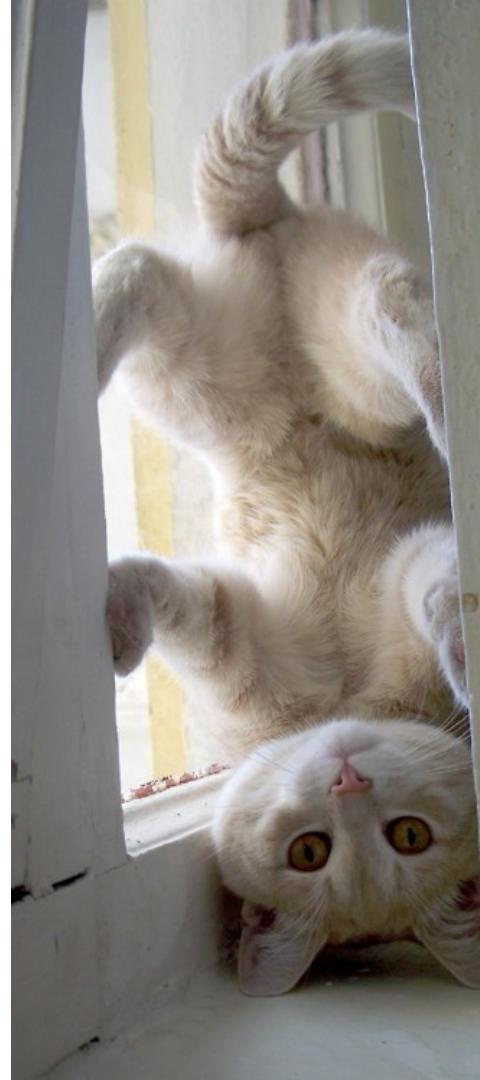
```
from github: settings - ssh - new key - paste  
ssh -T git@github.com
```

CLI part deux: stdin (0),
stdout (1), stderr (2)



CLI part deux: sorting out heads & tails

head ls-output	first 10 lines of ls-output
tail -5 ls-output	last 5 lines of ls-output
tail -f ls-output	continually refreshing last lines of ls-output
ls .../omg/ sort	sorts the results alphabetically
ls .../omg/ sort uniq	sorts, then removes duplicates



CLI part deux: extra CLI resources

100 Useful Unix Commands

http://www.oliverelliott.org/article/computing/ref_unix/

In The Beginning was the Command Line, Neal Stephenson

The Linux Command Line, William Shotts (in class repo under cli directory)



CLI part deux: q/a/cats



vim part deux: ToC

- * more moving
- * .vimrc
- * folds
- * extra vim resources
- * q/a/cats

vim part deux: moar moving

SCROLL	
control + b / f	takes you back / forward a page
control + d / u	takes you down / up half a page
GETTING IN GOOD	
I / A	insert begining / end of current line
P	pastes *before* your cursor (yy includes a /n)
o / O	adds a new line below / above the current line and goes into insert mode

MOVES LIKE JAGGER	
) / (forwad / backward 1 sentence
} / {	forward/backward one paragraph
" (2 apostrophes)	move to the last line you jumped from
DUAL WEILDING	
:e /OMG/WTF	while already in vim, open up /OMG/WTF for editing
:bp / :bn	move to previous / next opened file to edit

vim part deux: .vimrc & ...language?

esklimo kitteh



has 10 wordz 4 sno

Action	Preposition	Object	Explanation
d	i	w	delete (in) word
c	i	w	cut word
y	i	w	yank word
d	i	{	delete everything in { }
d	i	t	delete everything in an HTML tag
d	a	[delete everything in and including []
d	t	"	delete everything to next occurrence of "
y	T	(yank everything to prev occurrence of (

vim part deux: folds & other nifty things

highlight in visual mode, zf

za

f* - to the next '*' in the line (not above or below)

F* - previous '*'

; scroll through your f command



vim part deux: extra resources

from command line, type `vimtutor`
google “best vimrc configuration”



vim part deux: q/a/cats



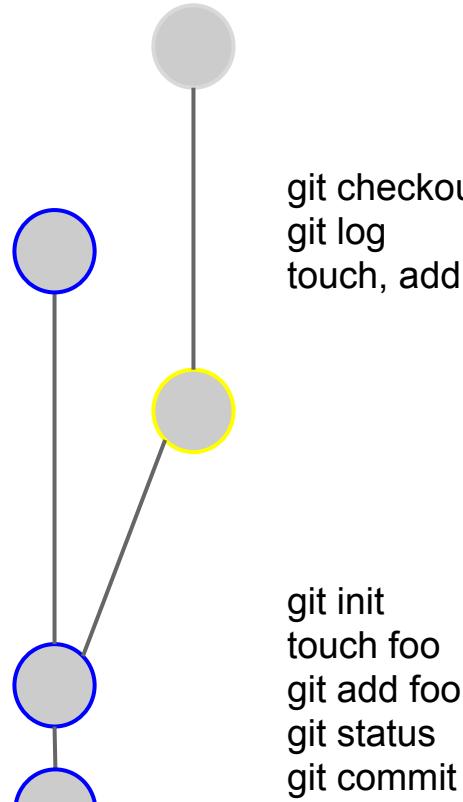
**I'M FROM TECH SUPPORT. I'M
HERE TO DELETE YOUR COOKIES**

closing out: on the importance of practice

Gratuitous cats

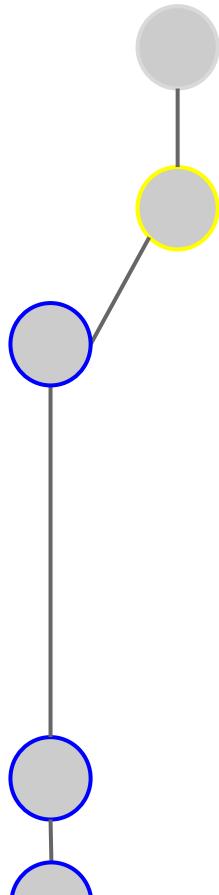
[link to LOL-show](#)

REBASE



git checkout science ⇒ git branch
edit & add baz

git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log

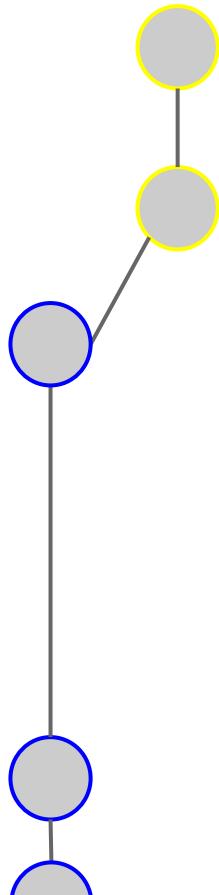


git init
touch foo
git add foo
git status
git commit

git checkout master
git log
touch, add, & commit qux

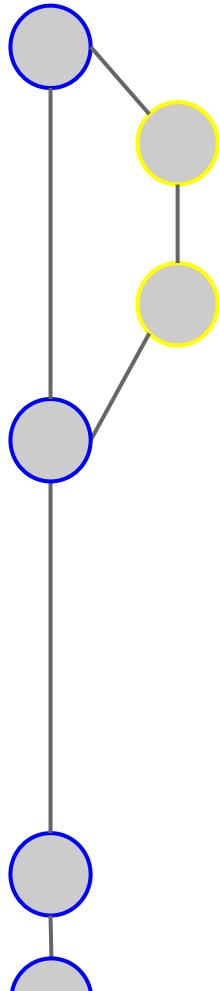
git branch => git branch science
git checkout science => git branch
touch baz
git add baz
git commit
git log

git checkout science => git branch
edit & add baz
git rebase master



git checkout science ⇒ git branch
edit & add baz
git rebase master
git commit

git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log



git checkout master
git merge science

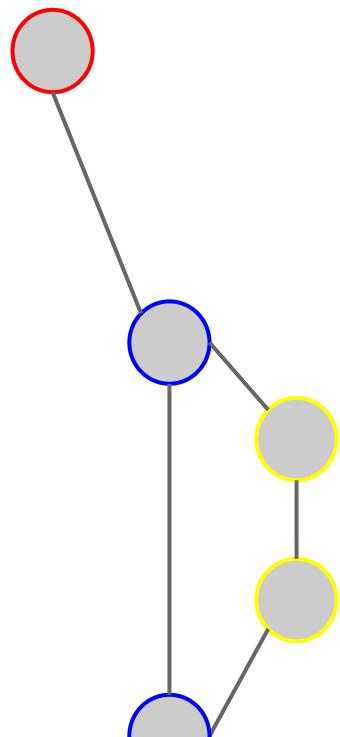
git checkout science ⇒ git branch
git rebase master
edit, add, & commit baz
git commit

git checkout master
git log
touch, add, & commit qux

git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log

git init
touch foo
git add foo
git status
git commit

MERGE CONFLICT



```
git branch mad_science  
git checkout mad_science  
vim baz  
git add baz  
git commit
```

```
graph TD; A(( )) --- B(( )); B --- C(( )); C --- D(( )); D --- E(( ));
```

git checkout master
git merge mad_science
git diff
vim baz (choose wisely)
git add baz
git commit

git branch mad_science
git checkout mad_science
vim baz
git add baz
git commit