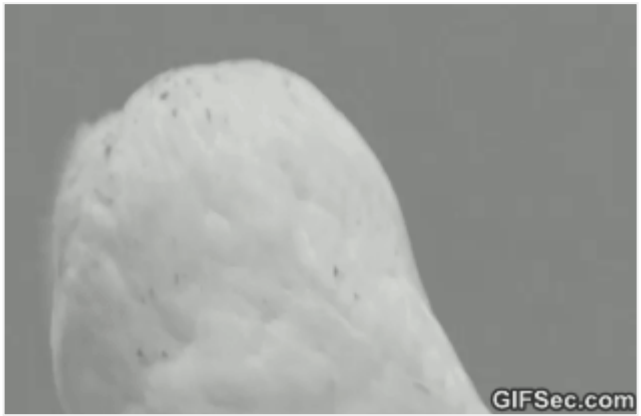
CLI, VIM, GIT, OMG, WTF, BBQ!?

THE MISSING MANUAL TO OPEN SOURCE TOOLS

Slides available: http://goo.gl/Qo9VsC

Insert Cygwin-Install loop here



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~

timimg

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

Setting up

check on:

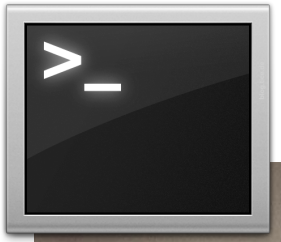
- Internet
- shell

any people considerations:

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

Wifi & password:

Why should I learn about this stuff? - CLI



Power!

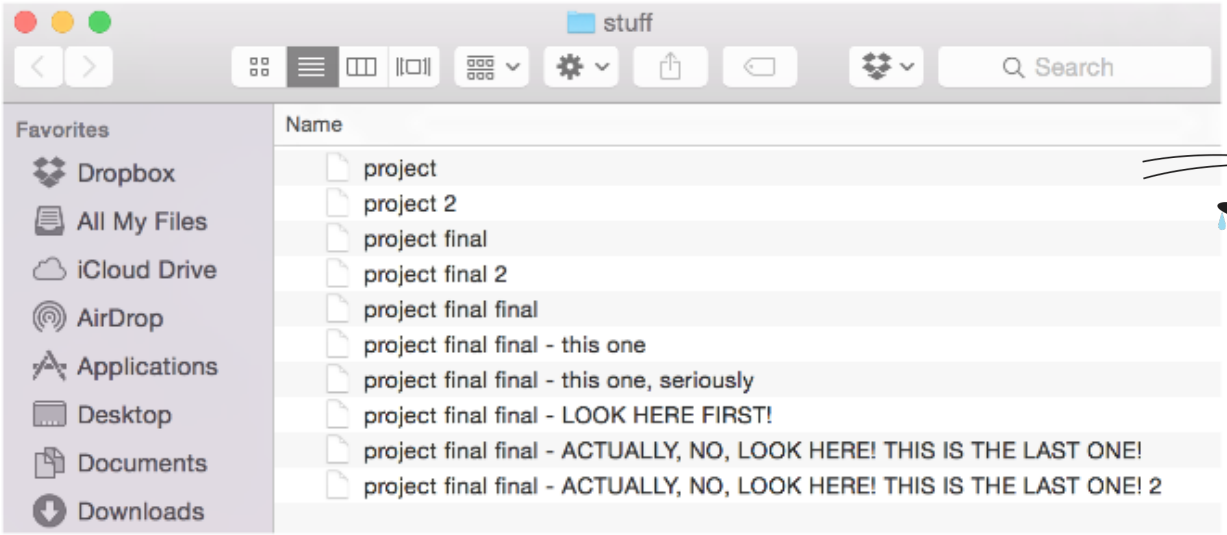
- “GUIs make easy tasks easy-- CLI makes difficult tasks possible”
- rename files en masse
- log onto a computer remotely and access its files
- see or change the permissions on a file
- find all files of a given extension or any file that was created within the last week
- set up a websever
- schedule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



- versioning
- standard for OS collaboration
- small commits = best practice

LET’S PARTY!

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

A more direct way to communicate with your computer

Interested?

In The Beginning was the Command Line, Neal Stephenson

very basics of CLI: moving, manipulating, reading

pwd	present working directory
ls	list
cd (..)	change directory (up one)
mkdir a	make directory named a
touch b	make file named b

how's the font size?

very basics of CLI: moving, manipulating, reading

pwd	present working directory
ls	list
cd (..)	change directory (up one)
mkdir a	make directory named a
touch b	make file named b
less a	show the contents of a
cp a b	copy a and name it b
mv a b	move a and rename it b
rm a	remove (forever) a

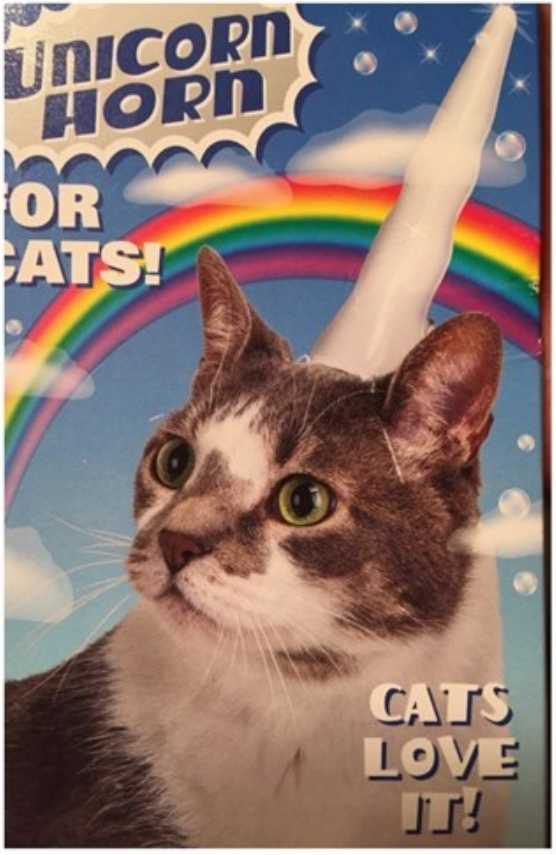
very basics of CLI: flags, help

- rm -rf (--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



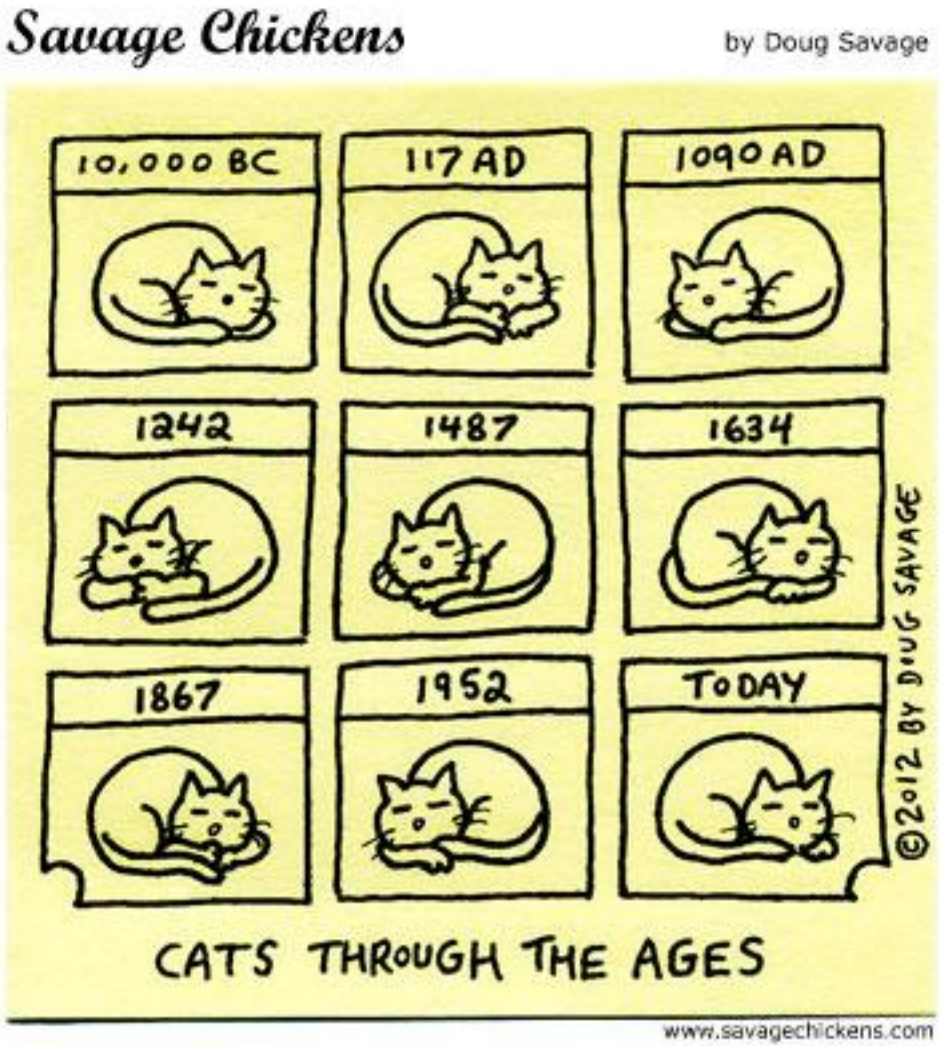
very basics of vim: ToC

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

very basics of vim:

a very, very, very brief history of vim

- Vi
- Bram Moolenaar wrote Vi-Improved



very basics of vim: modes

- Command
- Insert
- Visual

BASICS	
:wq	write and quit
:q!	quit without writing
i	enter Insert mode from Command mode
esc	exit Insert mode & return to Command mode
v	from command mode to visual mode
u	undo!
ctl + r	redo
DIRECTIONS	
h	to move the cursor one space left
j	to go down one line
k	to go up one line
l	to go one space right

I LIKE TO MOVE IT MOVE IT	
gg / G	takes you to the beginning / end of the file
0 / $	takes you to the beginning / end of the line
w / b	takes you to the next / previous word
DELETE!	
dw	to delete a word
d$	to delete from where you are to the end of the line
VISUAL EFFECTS	
y	ends and yanks your selection
p	pastes your selection
yy	yanks the line you are on
yap	yanks around paragraph

very basics of vim: q/a/cats

very basics of git

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

“it’s not you, it’s git”

very basics of git: a brief history

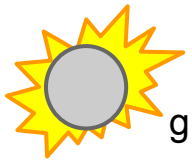
BitKeeper went private, and Linus Torvalds needed a version control system for the Linux Kernel

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



git commit

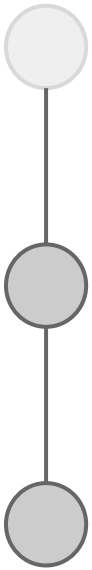
Commit message conventions!

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

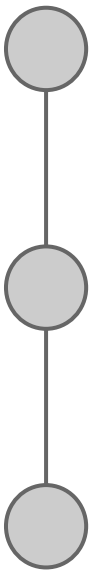
git init
git log (don’t panic)
touch foo
git status
git add foo
git status
git commit



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

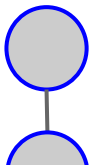


vim bar (edit)
git add bar
git commit

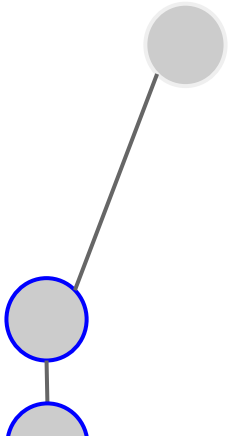
touch bar
git add bar
git commit

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

branches

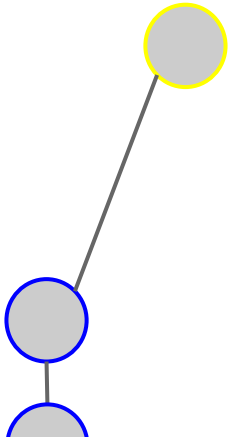


git init
touch foo
git add foo
git status
git commit



git init
touch foo
git add foo
git status
git commit

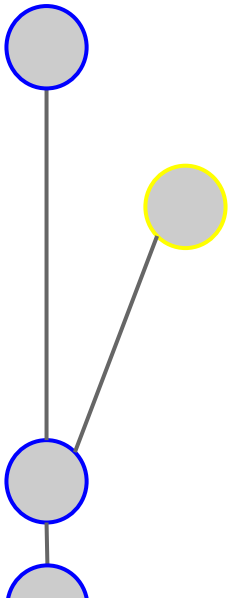
git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz



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

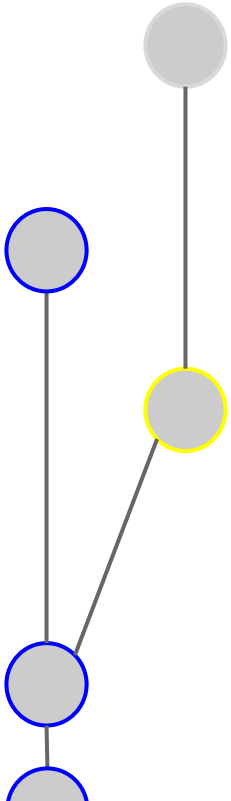
git status
git diff
git branch



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

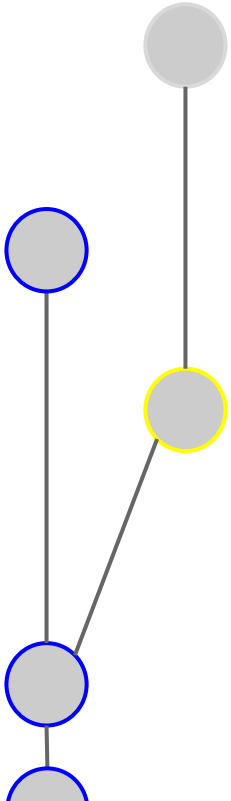


git checkout science ⇒ git branch

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

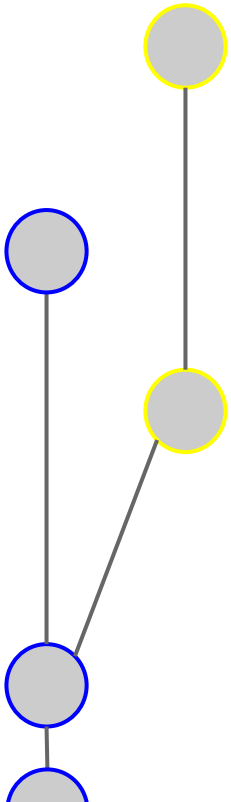


git checkout science ⇒ git branch
git merge master

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

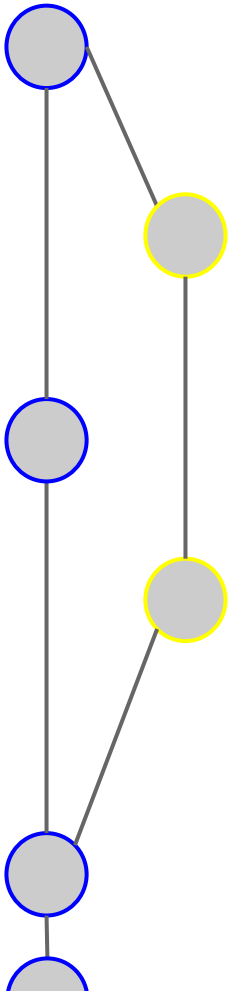


git checkout science ⇒ git branch
git merge master
git commit

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 merge science

git status
git diff
git branch

git checkout science ⇒ git branch
git merge master
git commit

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

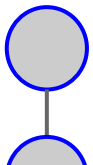
git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log

git init
touch foo
git add foo
git status
git commit

remote repositories (yours)

local
(your laptop)

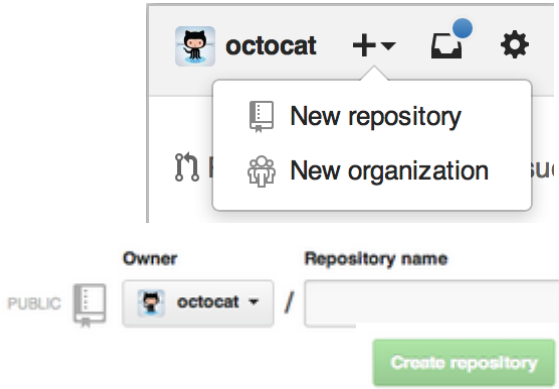
origin
(your fork)



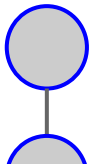
git commit -m ‘last thing we did’

local
(your laptop)

origin
(your fork)



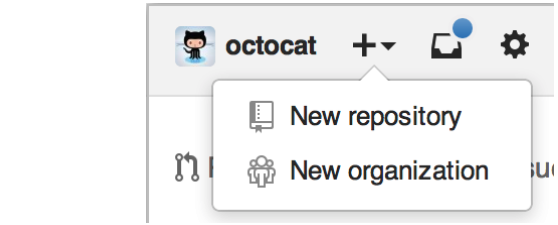
create a new repository on github.com



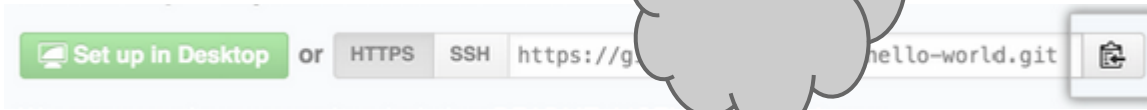
git commit -m ‘last thing we did’

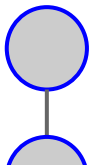
local
(your laptop)

origin
(your fork)



create a new repository on github.com
copy the URL

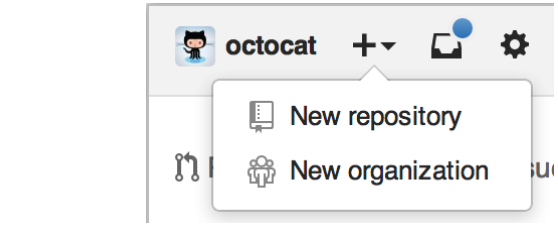




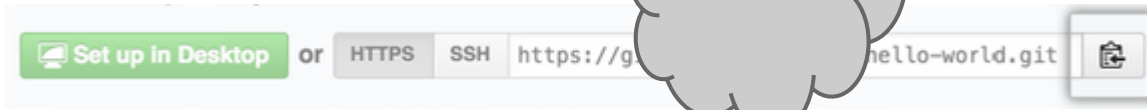
git commit -m ‘last thing we did’

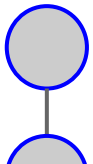
local
(your laptop)

origin
(your fork)



create a new repository on github.com
copy the URL

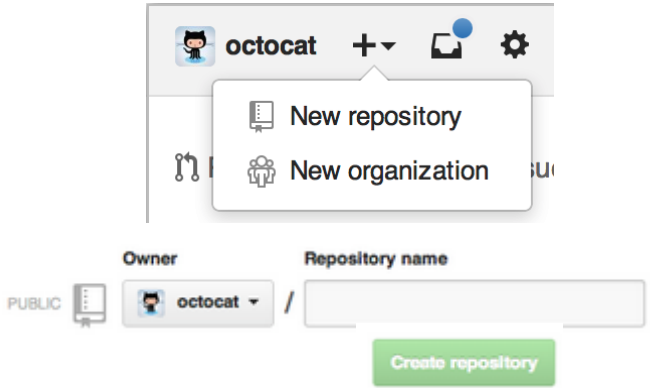




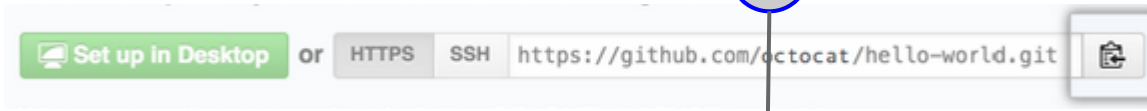
git commit -m ‘last thing we did’
git remote add origin *URL*

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...

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)

local
(your laptop)

origin
(your fork)

local
(your laptop)

origin
(your fork)

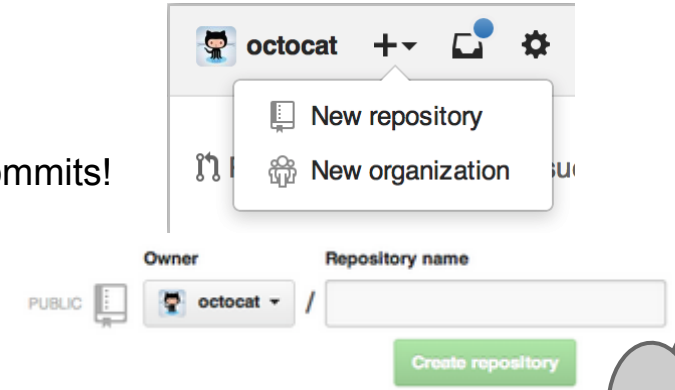
local
(your laptop)

origin
(your fork)

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!

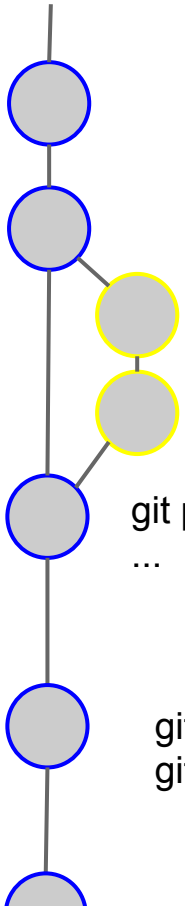


create a new repository on github.com
copy the URL

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

remote repositories (others')

local
(your laptop)

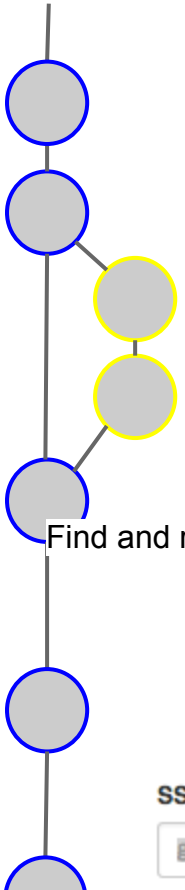


git pull upstream
…
git push origin science

git pull upstream
...

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

origin
(your fork)



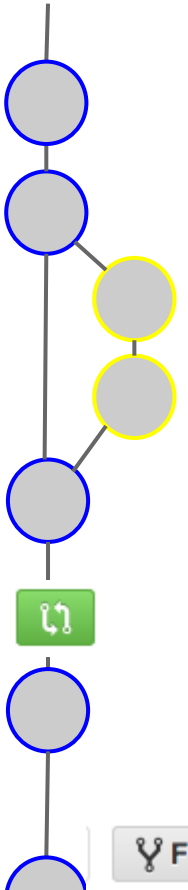
base fork = home base
head fork = on your shoulders

Find and replace

SSH clone URL

git@github.com:thel

upstream
(the main repo)

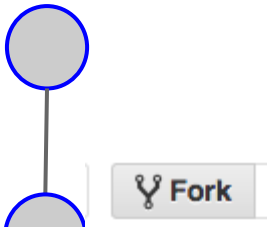


Fork

local
(your laptop)

origin
(your fork)

upstream
(the main repo)



local
(your laptop)

origin
(your fork)

upstream
(the main repo)

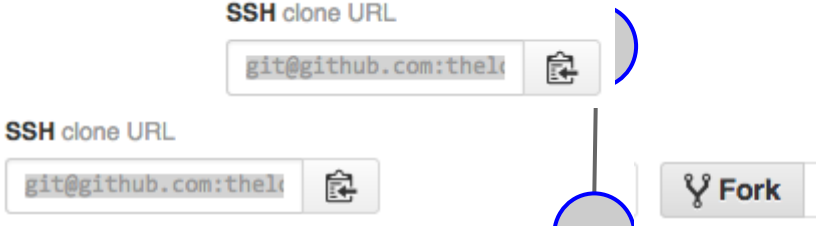
local
(your laptop)

origin
(your fork)

upstream
(the main repo)

where do you want it?

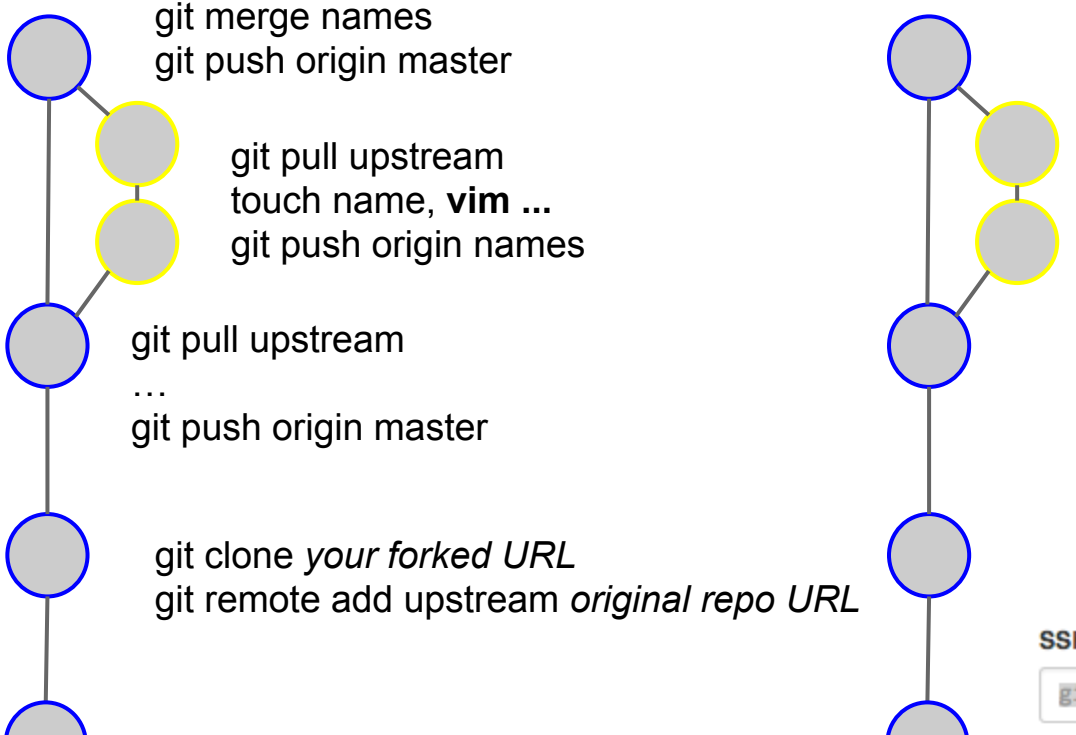
git clone *your forked URL*
git remote add upstream *original repo URL*
git remote

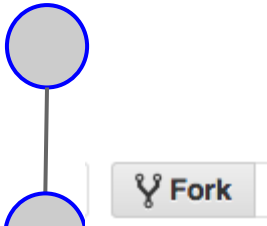


local
(your laptop)

origin
(your fork)

upstream
(the main repo)





local
(your laptop)

origin
(your fork)

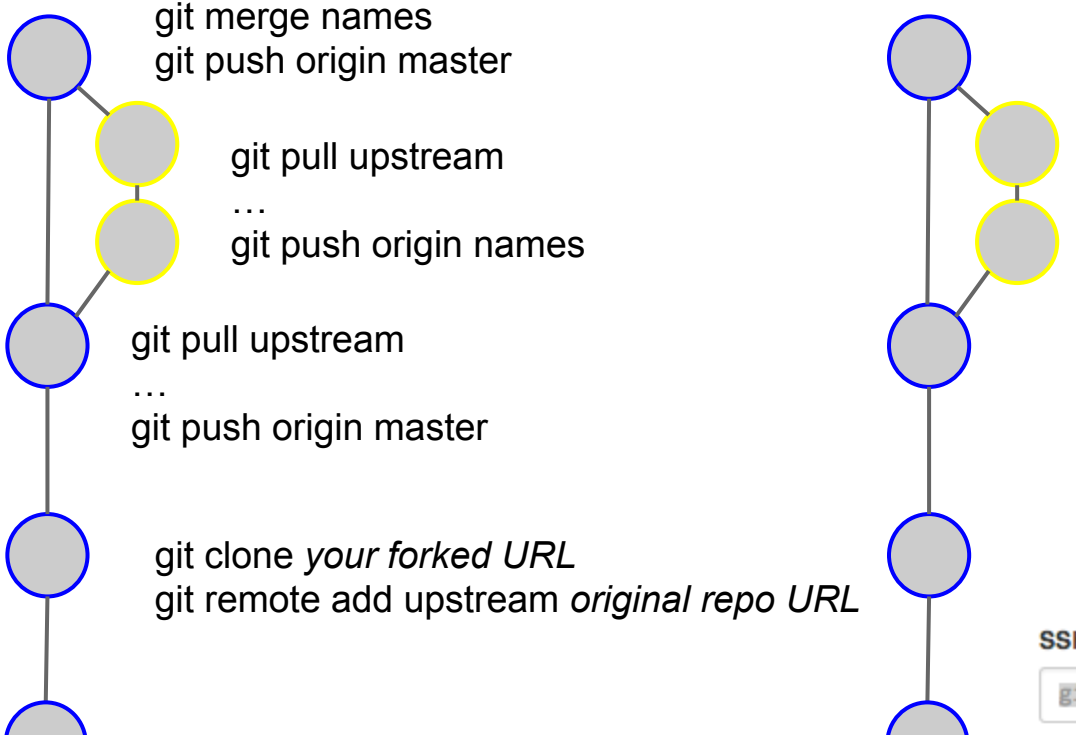
upstream
(the main repo)

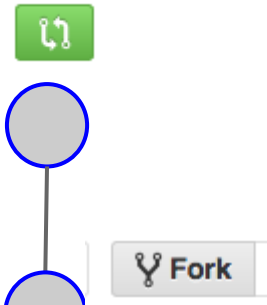
git pull upstream
git merge names
git push origin master

git pull upstream
…
git push origin names

git pull upstream
…
git push origin master

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





local
(your laptop)

origin
(your fork)

upstream
(the main repo)

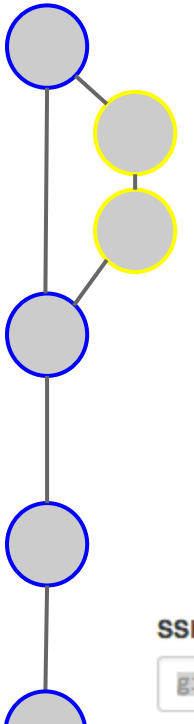
git pull upstream
git merge names
git push origin master

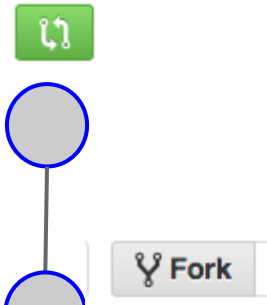
git pull upstream
…
git push origin names

git pull upstream
…
git push origin master

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

base fork = home base
head fork = on your shoulders





local
(your laptop)

origin
(your fork)

upstream
(the main repo)

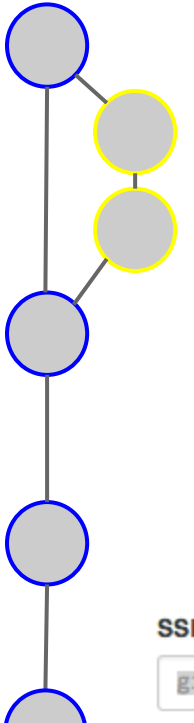
git pull upstream
git merge names
git push origin master

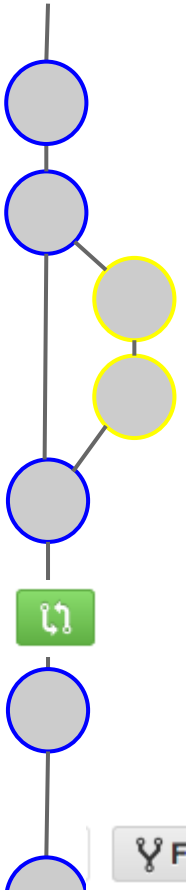
git pull upstream
…
git push origin names

git pull upstream
…
git push origin master

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

base fork = home base
head fork = on your shoulders





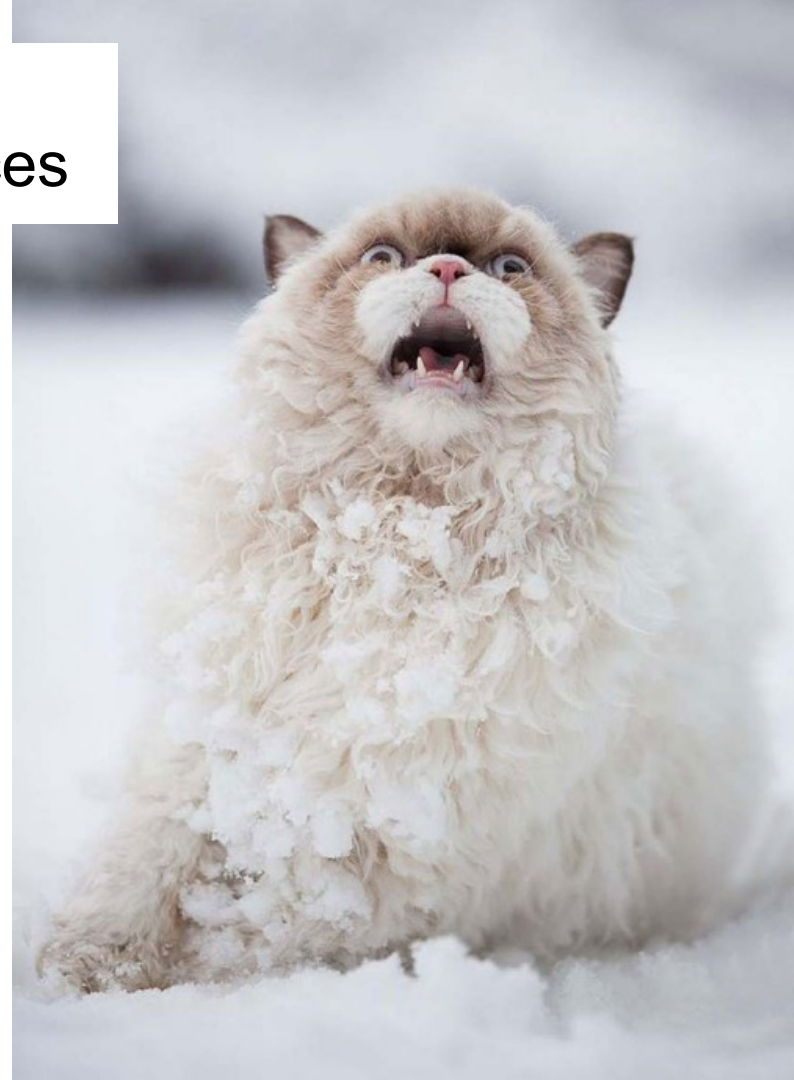
very basics of git: q/a/cats

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

CLI part deux: cat

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

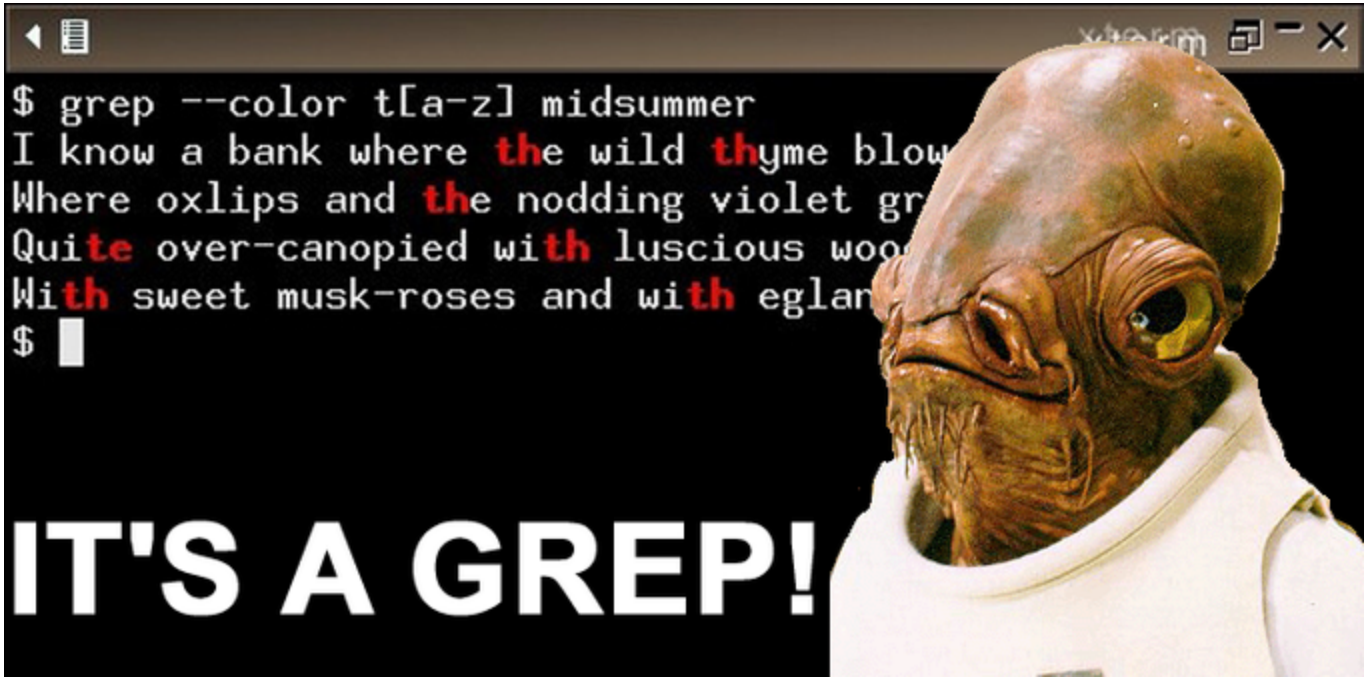
CLI part deux: wildcards

*	any number of things
?	one thing
[abc]	a, b, or c
[a..d]	a through d
[!abc]	anything except a, b, or c
[[:alnum:]]	any alpha-numeric character • [[:alpha:]] • [[:digit:]] • [[:lower:]] • [[:upper:]]

CLI part deux: grep

- grep = **g**lobal **r**egular **e**xpression **p**rint
- grep [-options] what_ur_lookin_4 where_ur_lookin

.	single character
^a	begins with a
a$	ends with a



- diff -c foo bar
  - difference
- comm file1 file2
  - compare

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

- find & replace
- more moving
- .vimrc
- folds
- extra vim resources
- q/a/cats

vim part deux: find & replace

FIND	
/bird	finds a bird!
p / n	takes you to the previous / next bird
REPLACE	
:%s/bird/cat/gc	replace all the birds with cats, but ask first
:2,5s/bird/cat/g	between lines 2 & 5, replace all birds with cats, no asking
:%s_bird_cat_g	replace all the birds with cats (changing the / to something else helps w/ pathnames)

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 beginning / 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

vim part deux: moar moving

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?

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

zf	fold highlighted text
za	unfold
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”

http://vim-adventures.com/

http://vim.wikia.com/

http://www.vimgolf.com/ (thanks, Tom!)

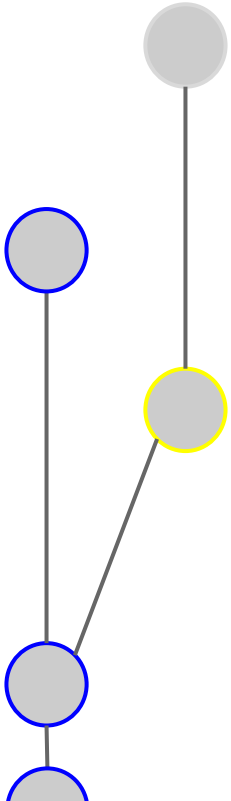
vim part deux: q/a/cats

closing out: on the importance of practice

Gratuitous cats

link to LOL-show

REBASE

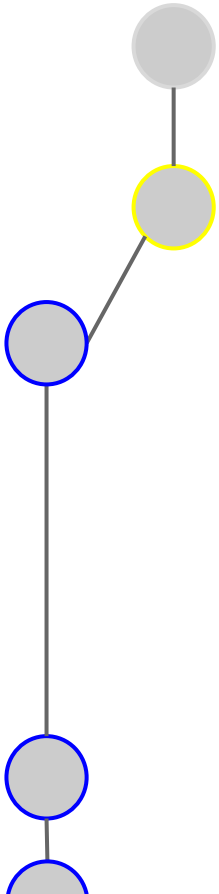


git checkout science ⇒ git branch
edit & add baz

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

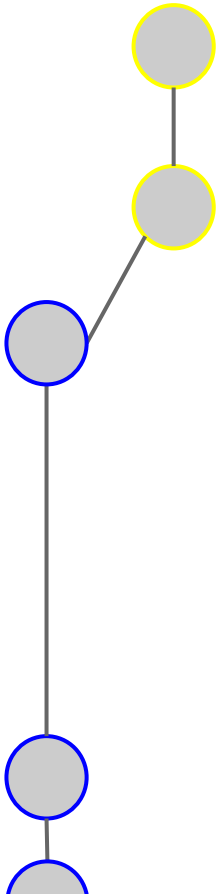


git checkout science ⇒ git branch
edit & add baz
git rebase master

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

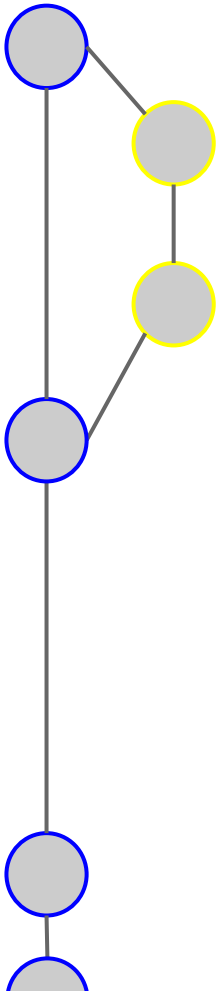


git checkout science ⇒ git branch
edit & add baz
git rebase master
git commit

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 merge science

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

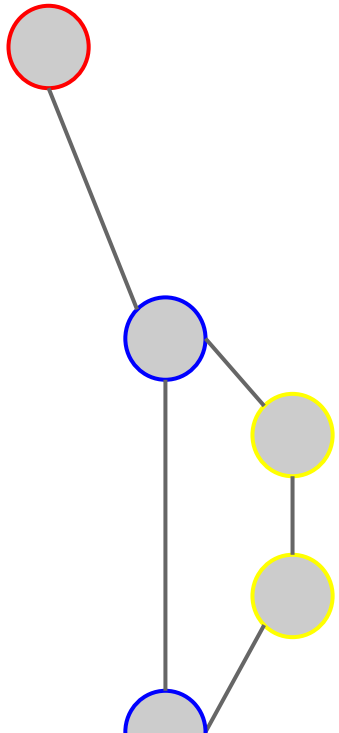
git checkout master
git log
touch, add, & commit qux

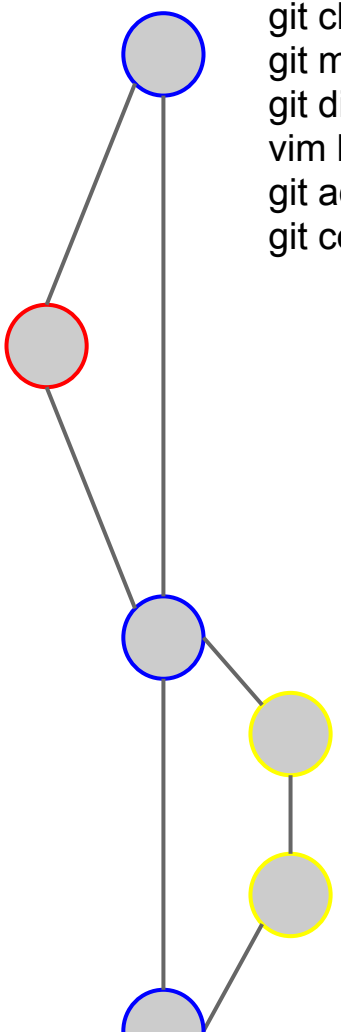
git branch ⇒ git branch science
git checkout science ⇒ git branch
touch baz
git add baz
git commit
git log

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
```





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