

Lecture 10

Stash, Revert, and Searching History

Schedule

- Only a few classes left 😞
- I hope you had fun and learned a lot
- Today (11/7): Stash, Revert, History
- Next Time (11/14): SourceTree and other Visual Tools
- 11/21: NO CLASS – Thanksgiving
- Last Lecture (11/28): Reflog, Tags, Plumbing and Porcelain
- Last Class (12/5): Final Exam

Scenario: you want to switch branches, but you have uncommitted changes

```
:( Andrew@Gengar ~/temp
04:08 PM (coolfeature)$ git checkout master
error: Your local changes to the following files would be overwritten by checkout:
    A
Please, commit your changes or stash them before you can switch branches.
Aborting
```

What if you don't want to commit?

git stash

Example use:

git stash



- Makes a “pseudo-commit” and puts it on a **stack** of stashed pseudo-commit.
- Message for stash is “WIP on *<branchname>*...”
- Use git stash save *<message>* to store stashes with better messages

git stash list

Example use:

git stash list

- Lists all stashed changes on the stash stack

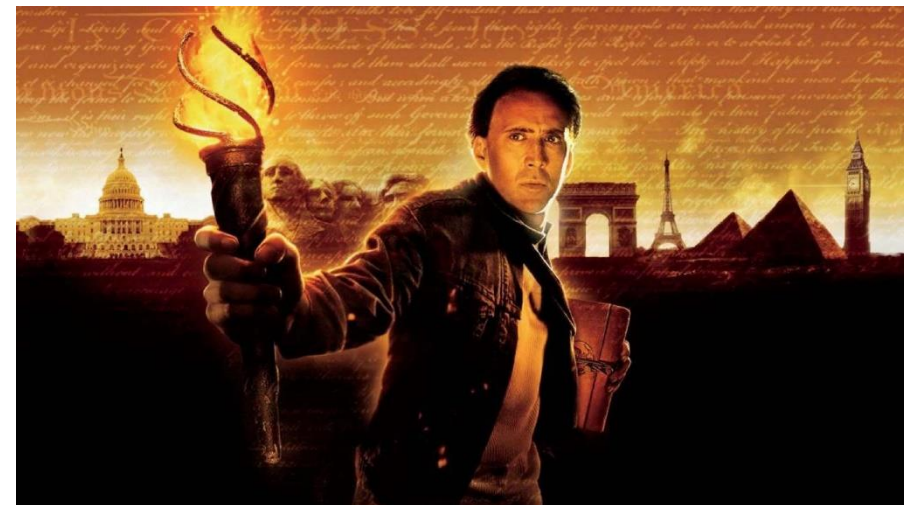


git stash apply (stash@{<depth>})

Example use:

```
git stash apply stash@{2}
```

- Reapplies the stashed change at the specified depth, if given.
- Depth is just another way of choosing from a list of saved stashes



git stash pop

Example use:

git stash pop

- Reapplies the top stashed change and removes it from the stash stack.

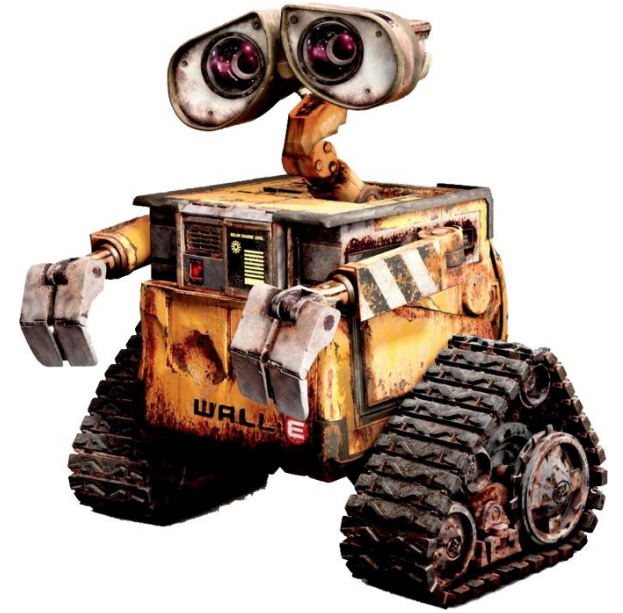


git stash drop (stash@{<depth>})

Example use:

```
git stash drop stash@{2}
```

- Removes the stashed change at the specified depth, if given.



git stash show (-p) (stash@{<depth>})

Example use:

```
git stash show stash@{2}
```

- Show details about the stashed change at the specified depth, if given.



git revert <commit hash> ...

Example use:


```
git revert HEAD (putting HEAD there is optional)
```


- Makes a new commit that “undoes” the commit at HEAD
- For when you made a mistake and want to completely reverse a commit
- Can cause a conflict (just like a merge) that you have to manually resolve
- Cause use with a list of commits






File History on Github

Branch: master ▾ Autolab / app / models / annotation.rb Find file Copy path

 icanberk Annotations for PDF submission ce723aa on Jul 8, 2015

2 contributors 


Executable File | 26 lines (23 sloc) | 564 Bytes Raw Blame **History**   

```
1 ##
2 # Annotations are Submission and Problem specific.
- - - - -
```











- Same as `git log <filename>`

git blame <filename>

- Shows which commit last modified a specific line

Autolab / app / models / annotation.rb 

100755 | 26 lines (23 sloc) | 564 Bytes

 Rubocop'd A models	2 years ago		1 ##
			2 # Annotations are Submission and Problem specific.
			3 # Currently, they are just text, but it would be nice if they could be used in
			4 # score calculations in the future.
			5 #
 Initial commit for open source Autolab	2 years ago		6 class Annotation < ActiveRecord::Base
			7 belongs_to :submission
			8 belongs_to :problem
			9
 Annotations for PDF submission	2 years ago		10 validates :comment, :filename, :submission_id, presence: true
 Initial commit for open source Autolab	2 years ago		11
 Fixes for viewFeedback	2 years ago		12 def as_text
 Ran rubocop --autocorrect	2 years ago		13 if value
			14 if problem

git log -G "regex to search for"

Searches for commits whose diff contains the text

```
$ git log --pretty=short -G "validates :comment"  
commit ce723aad8a5ab59994f8f55110d7de9c404e2fc8  
Author: Ilder Canberk <iltercanberk@gmail.com>  
  
    Annotations for PDF submission  
  
commit 1494872c7f780a406d5aad20087b2b9840fef6  
Author: Daniel Bucci <dlbucci@superftc.com>  
  
    Rubocop'd A models
```



(called pickaxe diff in the git documentation)

Activity/Homework Part 1!



Blame and Revert

1. Fork <https://github.com/aperley/Autolab> (my fork of Autolab)
2. Take a look at line #30 of the file
`app/views/layouts/application.html.erb`
3. Revert the commit that added that line, and push this change to your fork on a branch called `<andrewid>-revert-footer`
4. Make a PR to my fork

Activity/Homework Part 2!

Stash

1. Download this repo: <https://www.andrew.cmu.edu/course/98-174/homework/stash-practice.zip> (why couldn't you clone it?)
2. The repo contains 1 commit with a simple story and a few attempts to add to it. On the master branch, use git stash and make commits as necessary to include all of the attempts EXCEPT the one about squirrels
3. Create a new repo on Github called stash-practice and push your changes to its master branch