

Beyond Merge

Git best practices for working in a team environment

**Clara Bennett (tmfpolarbear)
Christian Sauer (tmfcs)**

Why Go Beyond Merge?

- "Those who cannot remember the past are condemned to repeat it."
- Improves teamwork, collaboration, and communication

Good Commits

What Makes a Good Commit

- Related to just one "changeset"
 - Easy to review
 - Easy to revert
 - Use git add -i or GUI tool

Good Commit Messages

Why should you care?

- Not for you, for your coworkers
- You may be unavailable and unreachable (another dimension, a zombie, etc).
- You may not work here anymore.
- You may forget.

Explain what the changes DO and WHY, not what the changes ARE

- Code and version control already shows *what* you changed
- Need to explain business/use case--what problem are you solving?
- Include names of people, dates, etc of justification
- CYA!!

Sample Template

Summarize what changes do < 50 chars

More detailed explanation, wrapping at ~72 chars, with an empty line between the "title" and the "body". Explain motivation for changes, and describe difference between previous state and current state.

- Optionally with bullet points

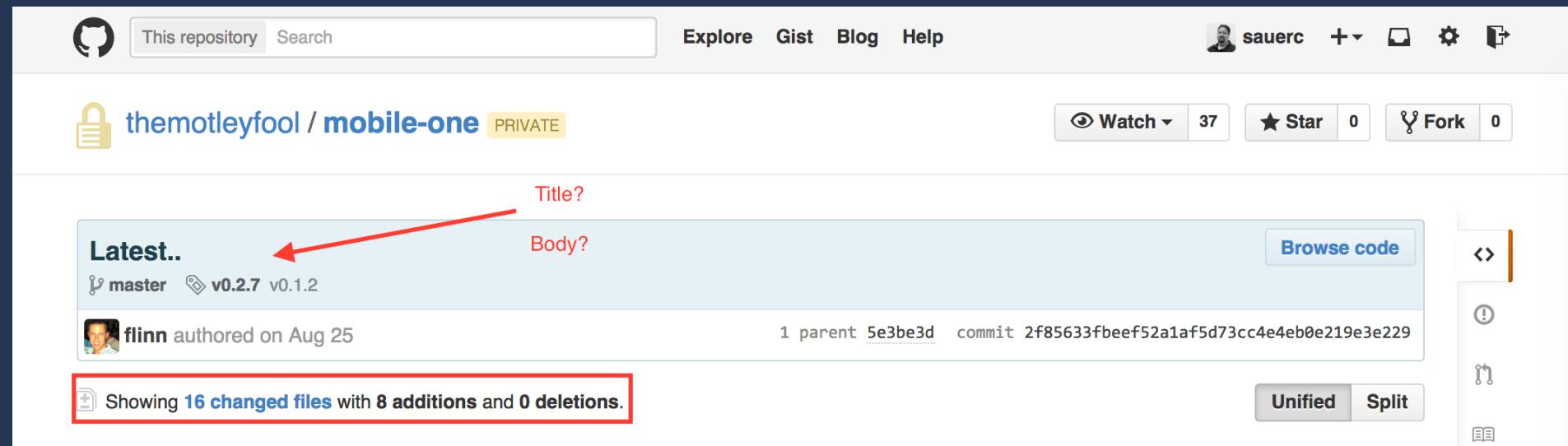
Good Example

App icon color change for the business

Meeting with Tom Gardner on 15-Oct-1999. He wants the icon to be cornflower blue. His reasoning: "Efficiency is priority number one, people. Because waste is a thief"

Moving the control of the icon color to a setting in the config file just in case he changes his mind.

Bad Examples



A screenshot of a GitHub diff interface. The commit title is 'holy crap i don't even'. The commit message contains several exclamation marks: '?!?!?!'. The commit details show it was authored by Aaron Torgerson on April 4, 2014, at 2:57:27 PM. The commit hash is 98897c6. The diff shows changes in the file 'folios/static/folios/html/portfolio/_active-custom.html'. A specific hunk of code is highlighted, showing additions and removals. The code includes AngularJS-like directives like ng-if and ng-class.

```
holy crap i don't even
Commit: 98897c6e06f9e3cdadc611fd7fd8423d2fa71f5c [98897c6]
Parents: f20cc84c31
Author: Aaron Torgerson <atorgerson@fool.com>
Date: Friday, April 4, 2014 2:57:27 PM

holy crap i don't even ?!?!?!

Context: 3 Lines ▾ □ Ignore Whitespace External Diff
folios/static/folios/html/portfolio/_active-custom.html
Modified file, 2 lines added, 2 lines removed Reverse File

Hunk 1 : Lines 71-78
71 71 <td class="expander" ng-class="{expanded: position.expanded}"><i class="icon-ct
72 72 <td ng-repeat="column in currentView.columns" column-cell row="position" col="0">
73 73 </td>
74 - <td ng-if="position.UnrealizedRelatedPositions.length !== 1 || position.IsWatched"
75 - <td ng-if="position.UnrealizedRelatedPositions.length === 1 && !position.IsWatched"
74 + <td ng-if="position.UnrealizedRelatedPositions.length !== 1 || (position.UnrealizedRe
75 + <td ng-if="position.UnrealizedRelatedPositions.length === 1 && !(position.UnrealizedRe
76 76 lot-utility-sprocket="position.UnrealizedRelatedPositions[0]"
77 77 portfolios="portfolios"
78 78 lot-edited="lotUpdated(position.UnrealizedRelatedPositions[0])"

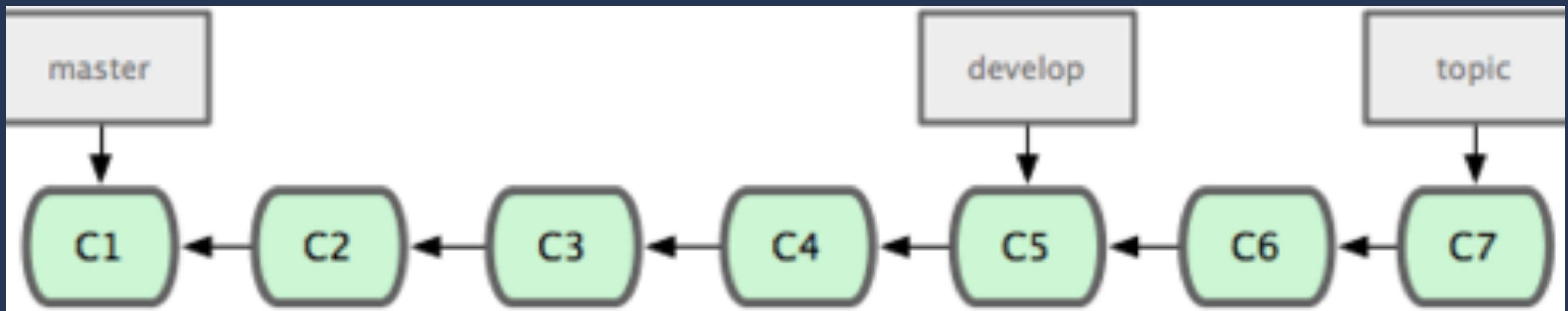
Reverse Hunk Reverse Selected Lines
```

Local Branches

Why always working on a branch makes life easier

Branches are cheap!

- Branches are just pointers, easy to create and delete
- Edits on branch recorded as changes from branch point
- Many branches, single working copy



You'll be prepared!

- Never know how long a change will take
- Working on multiple changes/fixes at the same time
- Easy to switch to emergency work
- Local master always reflects publicly-available code

Easy to keep up-to-date!

- Pull often (rebase or merge, discussed later)
- Easier to deal with lots of small changes/conflicts, if any, daily

Easy to remove!

- If and when priorities change mid-development, there's no annoying cleanup of now-irrelevant code on master
- Delete the branch, or keep it for reference/potential code reuse--either way, the master branch is uncluttered

Examples

In the beginning:

```
$ git checkout -b fix-divide-by-zero origin/master # descriptive name!
```

While working:

```
$ git pull # or git pull --rebase
```

In the end:

```
$ git rebase -i  
$ git checkout master  
$ git merge --no-ff fix-divide-by-zero  
$ git push  
$ git branch -d fix-divide-by-zero
```



Stash

Squirrel!?

- Holding onto uncommitted changes without dirtying history
- Specific to the branch
- Multiple stashes possible, give them names
- Can be turned into branches
- Useful when need to pull latest without committing

```
stash@{0}: WIP on master: e203df2 slide rebalance
stash@{1}: On master: Possible image changes
stash@{2}: On master: Potential slides on filter-branch
```

Save

```
$ git stash  
$ git stash save "descriptive name"
```

Apply

```
$ git stash pop  
$ git stash apply
```

Creating a branch from stash

```
$ git stash branch wip-new-stuff stash@{2}
```

Getting latest without committing

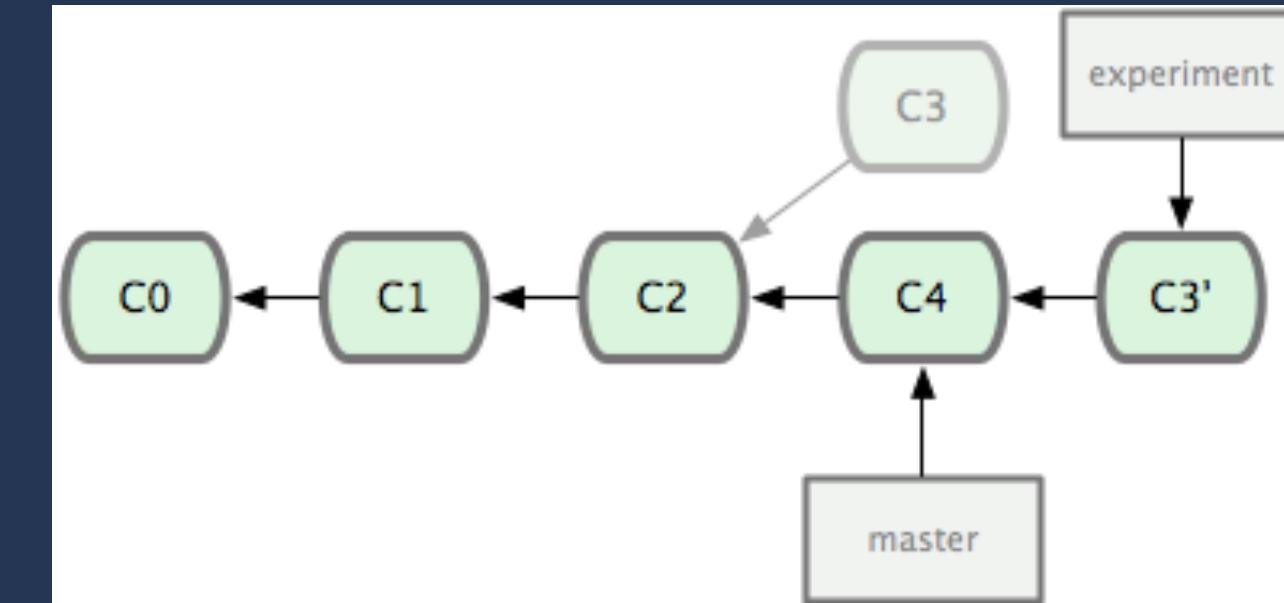
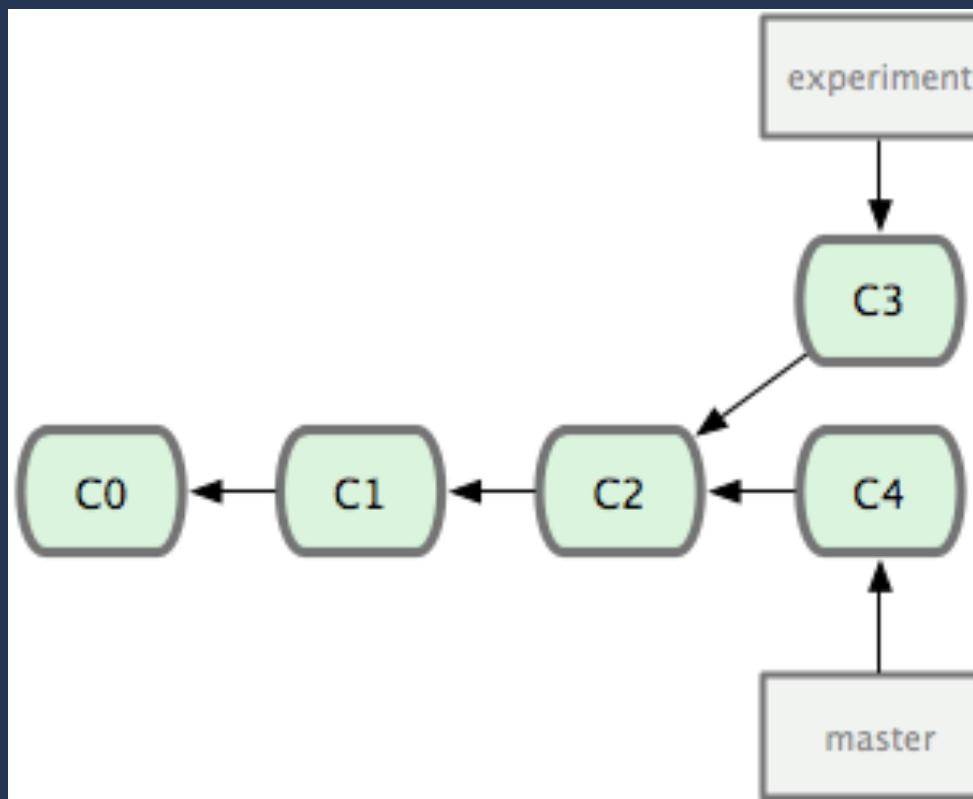
```
$ git stash && git pull && git stash pop
```

The background image shows a vast, icy landscape, likely a glacier or frozen sea, with numerous small icebergs and larger, more prominent ones. In the distance, a range of rugged mountains is visible under a clear sky.

Learn to Love (and Fear) Rebase

What is a rebase, anyway?

- Rewinds commits and replays them on top of target commit
- Rewrites history to be linear, and thus more readable



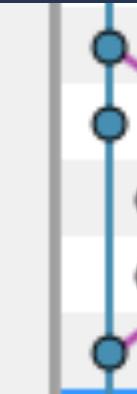
Merge vs Rebase on git pull

Scenario:

- Your changes are committed but not pushed.
- Changes from others committed and pushed.
- You need to pull before you can push.

Merge

- Your changes come first and then everyone else's
- Chronologically might be correct
- Repo perspective wrong

	Merge branch 'master' of http://rc.foolhq.com/Django/folios-ui	1 Aug 2014 15:38	Chris Quinn <cqui@foc>	ae0a8ef
	Changes to vstrackit	1 Aug 2014 15:38	Chris Quinn <cqui@foc>	e8bd7bd
	Setting up concat to be used instead of uglify for the vagrant box.	1 Aug 2014 15:34	Flinn <mflinn@foc>	1d1f656
	changes for making source maps	1 Aug 2014 15:09	Christian Sauer <csauer@foc>	24a019a
	Wiring up cancel modal event	31 Jul 2014 18:55	Chris Quinn <cqui@foc>	039cb10

Rebase

- Their changes come first and then your changes
- Since repo is "The Truth", more accurate depiction of events

Changes to vstrackit	1 Aug 2014 15:38	Chris Quinn <cqui@foc>	e8bd7bd
Setting up concat to be used instead of uglify for the vagrant box.	1 Aug 2014 15:34	Flinn <mflinn@foc>	1d1f656
changes for making source maps	1 Aug 2014 15:09	Christian <csauer@foc>	24a019a
Wiring up cancel modal event	31 Jul 2014 18:55	Chris Quinn <cqui@foc>	039cb10

Merge Crazy

Close commission for lots	14 Jul 2014 11:47	Joe Niedzwiecki <jniedzwiecki@fool.com>	789cde9
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:25	Greg Robleto <grobleto@fool.com>	a7a43e0
Styles for Investigator	14 Jul 2014 11:25	Greg Robleto <grobleto@fool.com>	aa20c69
Merge branch 'master' of http://rc.foolhq.com/Django/folios-ui	14 Jul 2014 11:14	unknown <cquinn@VAL-CQuinn.hq.fool.net>	2ea2f86
Adding CSP header to index view	14 Jul 2014 11:14	unknown <cquinn@VAL-CQuinn.hq.fool.net>	58f73d9
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:06	Aaron Torgerson <atorgerson@fool.com>	323eeef7
fix grunt	14 Jul 2014 11:06	Aaron Torgerson <atorgerson@fool.com>	18daf81
Notification Admin: basic table look for list page	14 Jul 2014 11:06	Christian <csauer@fool.com>	e25bcac
adding admin urls	14 Jul 2014 11:06	Christian <csauer@fool.com>	28a9bca
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:00	Aaron Torgerson <atorgerson@fool.com>	911d3e0
disabling source maps unless explicitly enabled	14 Jul 2014 10:59	Aaron Torgerson <atorgerson@fool.com>	f5d7a95
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 10:56	unknown <CHarris@vad-charris.hq.fool.net>	3487944
added the investigator	14 Jul 2014 10:55	unknown <CHarris@vad-charris.hq.fool.net>	02562fc
fixed minification of non-fool js files	14 Jul 2014 10:26	Aaron Torgerson <atorgerson@fool.com>	c996ed4
adding admin urls	11 Jul 2014 16:17	Christian <csauer@fool.com>	d3a5194
Notification Admin: form validation	11 Jul 2014 15:41	Christian <csauer@fool.com>	b2c5787
Notification Admin: display error messages when the API goes wrong.	11 Jul 2014 11:59	Christian <csauer@fool.com>	d1691af

Look Familiar?

Close commission for lots	14 Jul 2014 11:47	Joe Niedzwiecki <jniedzwiecki@fool.com>	789cde9
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:25	Greg Robleto <grobledo@fool.com>	a7443e0
Styles for Investigator	14 Jul 2014 11:25	Greg Robleto <grobledo@fool.com>	aa20c69
Merge branch 'master' of http://rc.foolhq.com/Django/folios-ui	14 Jul 2014 11:14	unknown <cquinn@VAL-CQuinn.hq.fool.net>	2ea2f86
Adding CSP header to index view	14 Jul 2014 11:14	unknown <cquinn@VAL-CQuinn.hq.fool.net>	58f73d9
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:06	Aaron Torgerson <atorgerson@fool.com>	323eeff7
fix grunt	14 Jul 2014 11:06	Aaron Torgerson <atorgerson@fool.com>	18daf81
Notification Admin: basic table look for list page	14 Jul 2014 11:06	Christian <csauer@fool.com>	e25bcac
adding admin urls	14 Jul 2014 11:06	Christian <csauer@fool.com>	28a9bca
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 11:00	Aaron Torgerson <atorgerson@fool.com>	911d3e0
disabling source maps unless explicitly enabled	14 Jul 2014 10:59	Aaron Torgerson <atorgerson@fool.com>	f5d7a95
Merge branch 'master' of http://rhodecode.foolhq.com/Django/folios-ui	14 Jul 2014 10:56	unknown <CHarris@vad-charris.hq.fool.net>	3487944
added the investigator	14 Jul 2014 10:55	unknown <CHarris@vad-charris.hq.fool.net>	02562fc
fixed minification of non-fool js files	14 Jul 2014 10:26	Aaron Torgerson <atorgerson@fool.com>	c996ed4
adding admin urls	11 Jul 2014 16:17	Christian <csauer@fool.com>	d3a5194
Notification Admin: form validation	11 Jul 2014 15:41	Christian <csauer@fool.com>	b2c5787
Notification Admin: display error messages when the API does wrong.	11 Jul 2014 11:59	Christian <csauer@fool.com>	d1691af



Rebase Happiness

Close commission for lots	14 Jul 2014 11:47	Joe Niedzwiecki <jniedzwiecki@fool.com>	789cde9
Styles for Investigator	14 Jul 2014 11:25	Greg Robleto <grobleto@fool.com>	aa20c69
Adding CSP header to index view	14 Jul 2014 11:14	unknown <cquinn@VAL-CQuinn.hq.fool.net>	58f73d9
fix grunt	14 Jul 2014 11:06	Aaron Torgerson <atorgerson@fool.com>	18daf81
Notification Admin: basic table look for list page	14 Jul 2014 11:06	Christian <csauer@fool.com>	e25bcac
adding admin urls	14 Jul 2014 11:06	Christian <csauer@fool.com>	28a9bca
disabling source maps unless explicitly enabled	14 Jul 2014 10:59	Aaron Torgerson <atorgerson@fool.com>	f5d7a95
added the investigator	14 Jul 2014 10:55	unknown <CHarris@vad-charris.hq.fool.net>	02562fc
fixed minification of non-fool js files	14 Jul 2014 10:26	Aaron Torgerson <atorgerson@fool.com>	c996ed4
adding admin urls	11 Jul 2014 16:17	Christian <csauer@fool.com>	d3a5194
Notification Admin: form validation	11 Jul 2014 15:41	Christian <csauer@fool.com>	b2c5787
Notification Admin: display error messages when the API goes wrong.	11 Jul 2014 11:59	Christian <csauer@fool.com>	d1691af

Edit history interactively with rebase -i

- Allows you to clean up / edit / reorder commits
- Don't have to care about local history until ready to push
- Commit often locally and reorganize into logical commits later!

How does it work?

```
$ git rebase -i <branch> <target-commit>
```

Original list from rebase -i

```
pick 00afbba Commit 1
pick b0073c4 Commit 2
pick f647576 Commit 3
pick a9d9b7c Combine with Commit 1
pick 37eb65b Commit 4
pick e142ce6 Commit 5
pick 7f1a8dc Fix commit 4
```

Reordered commits

```
pick b0073c4 Commit 2
pick 00afbba Commit 1
squash a9d9b7c Combine with Commit 1
edit f647576 Commit 3
pick 37eb65b Commit 4
fixup 7f1a8dc Fix commit 4
pick e142ce6 Commit 5
```

What's to fear?

- Creates new commits
 - Rebase already pushed commits, others will have a bad day
- Conflicts nonintuitive
 - "Local"/"Ours"/"Mine" is the repository
 - "Remote"/"Theirs" is **YOUR** changes
 - Potential for more conflicts than merge

Bonus: autosquash

- To easily do a fixup or squash on an earlier commit, run:

```
$ git commit --fixup=<commit-to-fix>
$ git rebase -i --autosquash <branch> <target>
```

- Automatically places fixup/squash commit on rebase
- To default autosquash on interactive rebase, run:

```
$ git config --global rebase.autosquash=true
```

Reflog: your safety net

- Local to the checkout
- Records every action git performed where data is stored
- Gives you SHA1 to checkout/cherry-pick/merge from any point in history
- **WARNING** 'git gc' which runs automagically periodically, cleans the reflog!

```
21a3a6b HEAD@{0}: commit: moved images to subfolder for better organization
2c9bd34 HEAD@{1}: commit: Updating commit reorder image to be more realistic
485f1bb HEAD@{2}: commit: Acknowledging what `git pull` with merge looks like
5d0dd5b HEAD@{3}: rebase finished: returning to refs/heads/master
5d0dd5b HEAD@{4}: pull: checkout 5d0dd5b280f99083d33d97c214ae1b594ee98cf3
87ffa35 HEAD@{5}: commit: Speaking order for presentation
3a3359a HEAD@{6}: commit: verb tense change on Master is Deployable
ac55667 HEAD@{7}: commit: Title change for details of Good Commit Messages
28d3a42 HEAD@{8}: commit: Add commands for creating a good commit
94d4592 HEAD@{9}: commit: Removing "Getting out of Trouble" slides
a3935e3 HEAD@{10}: commit: Shortening commet to make line bigger
a458c95 HEAD@{11}: commit: New title slide
ee8a804 HEAD@{12}: commit: Converted rebase -i slide from inline text to image
b995847 HEAD@{13}: commit: Cleaned up "What's to fear" slide
8761389 HEAD@{14}: commit: Beginnings of Good Commit slides
090d422 HEAD@{15}: commit: Improve local branch example
32c5cac HEAD@{16}: commit: screenshot of branching game
cd66cb1 HEAD@{17}: commit: Change wording on Hands on Workshop slide
0a612b2 HEAD@{18}: commit: Need space at the end to format properly
2c48bdb HEAD@{19}: commit: More screenshots for git pull merge vs rebase
14e9f53 HEAD@{20}: commit: Changed to be more bullet point-y and presentation-y
191eca8 HEAD@{21}: commit: Merae vs Rebase durina ait pull
d5aae38 HEAD@{22}: commit (amend): Commit messages, expanding to good and bad examples
a2559a3 HEAD@{23}: commit: Commit messages, expanding to good and bad examples
aa8f182 HEAD@{24}: commit: Branch point vs Snapshot
dbb2b79 HEAD@{25}: commit: Include rebase as an option for local branches
2570e4f HEAD@{26}: commit: Being a zombie more humorous than being dead
d37cbf8 HEAD@{27}: commit: Fix/Standardized code blocks
fa1fe3b HEAD@{28}: commit: Humorous good commit message example
```

Releases

2 Strategies

1. Release Branches
2. Master is Deployable

Release Branches

- New work lands in master
- Each release has a branch
- Specific commits are cherry-picked ("back-ported") into release branches as necessary
- Tags on branches when deployed

Master is Deployable

- New work is done on feature branches
- Feature branches not merged until tests passed and code reviewed
- All deployments come from master

Doesn't matter. Team needs to pick one and stick to it.



Putting it all together

Github Flow

Process used by Github to work on Github

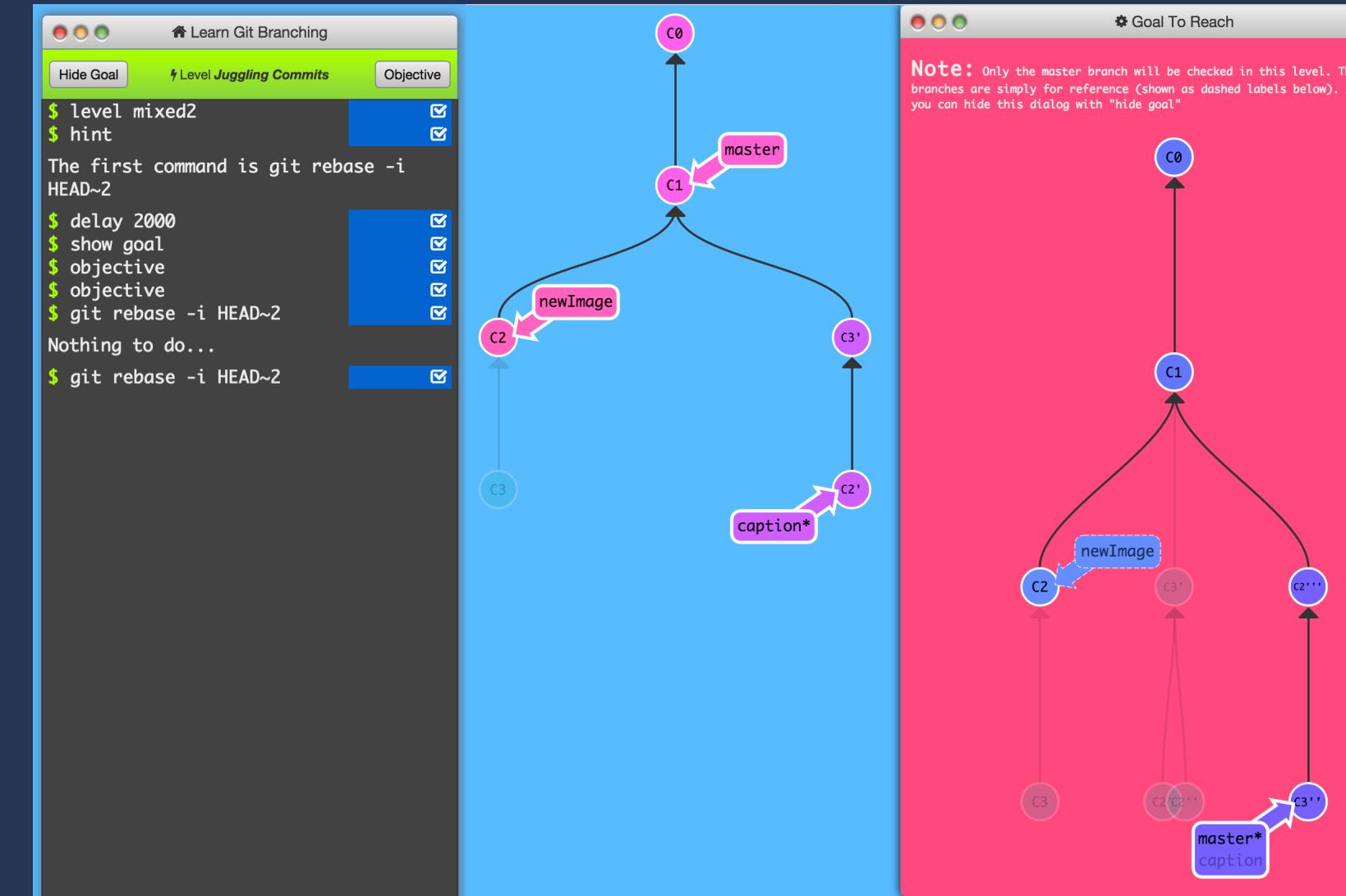
1. Create a branch
2. Add commits
3. Rebase -i
4. Open a Pull Request (or ask for Code Review)
5. Discuss/Review
6. Merge (--no-ff)
7. Deploy

Hands on Workshop

Any Interest? Let us know!

Would you like to play a game?

<http://pcottle.github.io/learnGitBranching/>



Questions?