git branch -D local-branch
git push origin :local-branch
Much like James Cameron’s *Avatar™*, Git doesn’t make any goddamn sense.
This talk throws a ton of stuff at you
This talk throws a ton of stuff at you
(Not unlike James Cameron’s Avatar™, btw)
Learn the bits that make you more productive
merge strategies
merge strategies

strategy defaults:
one head: recursive
multiple heads: octopus
merge strategies

git merge master -s strategy
merge strategies

git merge master -s ours
merge strategies

`git merge master -s ours`

“Give me master, but my final state should match my current branch”
merge strategies

git merge master -s ours

Great for when your long-running branch will supersede master
merge strategies

git merge master \\ -s recursive
merge strategies

git merge master \n-s recursive -X patience

Better line matching for logical groups
merge strategies — patience

```ruby
{
  :a => :b
}
```
merge strategies — patience

{  
  :a => :b
}

{  
  :c => :d
}
merge strategies — patience

{
  :a => :b
+}
+
+
+{
++
++ :c => :d
}

merge strategies — patience

{  
  :a => :b
}

+{  
  +  :c => :d
  +}

merge strategies

git merge master \ -s recursive -X patience

Can also set as a global option
merge strategies

› man git-merge
# look at “Merge Strategies”
The second-order-diff trick
the second-order-diff trick

Tom Moertel

(cl.ly/QNaG)
the second-order-diff trick

**Goal**: transform strings with a regex in project
the second-order-diff trick

Approach: re-run `git diff` and `git reset` until the regex is correct
the second-order-diff trick

Solution: stash, then diff against \texttt{stash}\{0\}
the **second-order-diff** trick

```bash
❯ git status
# clean
```
the second-order-diff trick

> ./run-regex-script
the second-order-diff trick

git diff

# review and check output
the second-order-diff trick

❯ git stash

# stash your reviewed diff
the second-order-diff trick

> ./run-regex-script
the second-order-diff trick

```bash
$ git diff stash{0}
# only incremental changes
```
the *second-order-diff* trick

Find and replace in projects without *worrying*
emoji diffs
Yeah this is important
emoji diffs

Git diffs unicode poorly:
@@ -1 +1 @@
-MINDLESS COUPLE ABOUT TO BEHEAD CHILD
+<U+1F46A>
Git uses **less** as its pager
emoji diffs

git --no-pager diff
emoji diffs

@@ -1 +1 @@
-MINDLESS COUPLE ABOUT TO BEHEAD CHILD
+
This couple about to behead child
git stripspace
Fixes horrible coworkers and contributors
git **strip-space**

Strips trailing whitespace, collapses newlines, adds a final newline
git strip space

git strip space < file
`echo "hi \n" | git stripspace`
Run this in your editor after file saves
git diff --check

Fixes horrible you.
git diff --check

run.sh:1: trailing whitespace.
+echo "hi"
git diff --check

run.sh:1: space before tab in indent.
  +    ls -la
git diff --cached
git diff --cached


git diff only diffs unstaged changes
git diff --cached

Shows **staged**
(but uncommitted)
changes
HOT, HOT, HOT, TRIVIA BREAK
Which of these commit messages actually happened on GitHub?
1. A reasoned argument, citing sources

2. A 500MB db dump
1. A reasoned argument, citing sources

2. A 500MB db dump
“Why is my repository slow?”
git merge --abort
git merge --abort

Aborts a conflicted merge
git merge --abort

Resets your tree
git merge -m "message"
git merge -m

Like `commit -m` but populates the merge commit
git merge --no-commit
git merge --no-commit

Merges without a commit message
git status --ignored
git status --ignored

Shows the status of everything you’re ignoring
--assume-unchanged

Temporarily ignore a file
git update-index
  --assume-unchanged
  
  path_to_file
--assume-unchanged

git update-index \
--no-assume-unchanged \
path_to_file
proper messages
Linus has thoughts on your commit messages
proper messages

```
git commit -m "fixes shit"
```
proper messages

```
git commit -m "fixes shit"
```
proper messages

A good commit has:
proper messages

Short summary (title)
proper messages

Short summary (title)
Short summary (title)

A longer description of the change follows.
proper messages

Short summary (title)

A longer description of the change follows.

< 50c

wrap at 72c
yay a commit message look at how cool i am with my commit message let me talk a bunch because i think i deserve it

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch v3
# You are currently editing a commit while rebasing branch 'master' on '4628474'.
# (use "git commit --amend" to amend the current commit)
# (use "git rebase --continue" once you are satisfied with your changes)
#
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
proper messages

git-commit helps you write perfect messages
proper messages

Just keep typing until you notice your text changes colors

# Please enter the commit message for your changes. Lines starting
# with '!' will be ignored, and an empty message aborts the commit.
# On branch v3
# You are currently editing a commit while rebasing branch 'master' on
# (use "git commit --amend" to amend the current commit)
# (use "git rebase --continue" once you are satisfied with your chang
Just keep typing until you notice your text changes colors

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch v3
# You are currently editing a commit while rebasing branch 'master' on
# (use "git commit --amend" to amend the current commit)
# (use "git rebase --continue" once you are satisfied with your chang
proper messages

Just keep typing until you notice your text changes colors

From this point on you can type a normal description and, lo and behold, it's going to word wrap you just the way you like it. See? Doesn't this make you feel all warm and fuzzy and shit inside of your body?

# Please enter the commit message for your changes. Lines starting with '# will be ignored, and an empty message aborts the commit.
# On branch v3
# You are currently editing a commit while rebasing branch 'master' on
# (use "git commit --amend" to amend the current commit)
# (use "git rebase --continue" once you are satisfied with your changes)
proper messages

wraps at 72c

Just keep typing until you notice your text changes colors.

From this point on you can type a normal description and, lo and behold, it's going to word wrap you just the way you like it. See? Doesn't this make you feel all warm and fuzzy and shit inside of your body?

# Please enter the commit message for your changes. Lines starting # with '#' will be ignored, and an empty message aborts the commit. # On branch v3 # You are currently editing a commit while rebasing branch 'master' on # (use "git commit --amend" to amend the current commit) # (use "git rebase --continue" once you are satisfied with your change)
proper messages

Assumes your $EDITOR is vim
proper messages

export EDITOR=vim
proper messages

Assumes you have syntax highlighting on
proper messages

syntax on

~/.vimrc
proper messages

Assumes you have plugin indenting on
proper messages

filetype indent plugin on

~/.vimrc
canonical URLs
Play

Play is an employee-powered iTunes-based client-driven distributed music server for your office. Also it can prepare your taxes.

Background

Did you know that listening to music while you work produces better and faster code? It’s true. It’s in a README.

We listen to music constantly at GitHub. So I wrote Play. Initially it was a modest shell-oriented music server, but we’ve since grown it to quite the setup:
Play

Play is an employee-powered iTunes-based client-driven distributed music server for your office. Also it can prepare your taxes.

Background

Did you know that listening to music while you work produces better and faster code? It's true. It's in a README.

We listen to music constantly at GitHub. So I wrote Play. Initially it was a modest shell-oriented music server, but we've since grown it to quite the setup:
Play

Play is an employee-powered iTunes-based client-driven distribution service.

Background

Did you know that listening to music while you work produces better and faster code? It's true. It's in a README.

We listen to music constantly at GitHub. So I wrote Play. Initially it was a modest shell-oriented music server, but we've since grown it to quite the setup:
Play

Play is an employee-powered iTunes-based client-driven distribution platform.

Background

Did you know that listening to music while you work produces better and faster code? It's true. It's in a README.

We listen to music constantly at GitHub. So I wrote Play. Initially it was a modest shell-oriented music server, but we've since grown it to quite the setup:
This page won’t change, even if the branch does.
curlable public keys
Plaintext, non-authed access to public keys
Programmatically add SSH keys
fetching pulls
Accepting a Pull Request: Add a remote, fetch branch, merge
fetching pulls

Lots of boring commands
GitHub stores all Pull Requests in your repository.
This means we can track code even if the fork is deleted.
You can abuse this, too!
fetching pulls

git fetch origin pull/id/head:head:name
git fetch origin pull/12/head:pr
fetching pulls

git fetch origin pull/12/head:pr

fetches Pull Request #12 into a branch named “pr”
fetching pulls

Automate this within helper scripts or an alias
live updates
<table>
<thead>
<tr>
<th>holman</th>
<th>Hook up JavaScript again</th>
<th>edzcube</th>
</tr>
</thead>
<tbody>
<tr>
<td>holman</td>
<td>Remove scaffolded css</td>
<td>f199c60</td>
</tr>
<tr>
<td>holman</td>
<td>Make sure errors get cleared</td>
<td>✔ 6199391</td>
</tr>
</tbody>
</table>
holman commented

Probably finna merge after lunch.
holman commented

Probably finna merge after lunch.

holman referenced this pull request from a commit

holman merged commit 6fb63e1 into v3 from v3-rails 2 months ago

holman closed the pull request 2 months ago

holman deleted the v3-rails branch 2 months ago Restore
holman commented

Probably finna merge after lunch.

holman referenced this pull request from a commit

holman merged commit 6fb63e1 into v3 from v3-rails

holman closed the pull request 2 months ago

holman deleted the v3-rails branch 2 months ago Restore

holman commented

I had lunch.
HOT, HOT, HOT, TRIVIA BREAK
Most frequent Emoji used on GitHub:
#4

:+1:
quick quotes
quick quotes

Markdown uses \[ > \] for blockquotes
Um, he's sick. My best friend's sister's boyfriend's brother's girlfriend heard from this guy who knows this kid who's going with the girl who saw Ferris pass out at 31 Flavors last night. I guess it's pretty serious.
quick quotes

Highlight and press r
Um, he's sick. My best friend's sister's boyfriend's brother's girlfriend heard from this guy who knows this kid who's going with the girl who saw Ferris pass out at 31 Flavors last night. I guess it's pretty serious.
Um, he's sick. My best friend's sister's boyfriend's brother's girlfriend heard from this guy who knows this kid who's going with the girl who saw Ferris pass out at 31 Flavors last night. I guess it's pretty serious.

> Um, he's sick. My best friend's sister's boyfriend's brother's girlfriend heard from this guy who knows this kid who's going with the girl who saw Ferris pass out at 31 Flavors last night. I guess it's pretty serious.
browse stars
http://github/stars/brianmario
web flow
You can do a lot without leaving the browser
Documentation, prose, quick fixes, switching away from nosql
Create branch: add-gitignore
from 'master'
dotfiles / .gitignore

Want to use a .gitignore template? Choose .gitignore: Ruby

File
1. *.gem
2. *.rbc
3. .bundle
4. .config
5. coverage
6. InstalledFiles
7. lib/bundler/man
8. pkg
9. rdoc
10. spec/reports
11. test/tmp
12. test/version_tmp
13. tmp
14. # YARD artifacts
15. .yardoc
16. _yardoc
17. doc/
18. 
Your recently pushed branches:

- Compare & pull request: add-gitignore (less than a minute ago)
You can add more commits to this pull request by pushing to the `add-gitignore` branch on `defunkt/dotfiles`.

This pull request can be automatically merged.
You can also merge branches on the command line.

Merge pull request
Pull request successfully merged and closed
You're all set—the add-gitignore branch can be safely deleted.

Delete branch
Frees you from `git clone`
issue autocomplete
Type #, get a search
We broke something in
We broke something in #
We broke something in #rail

- #280 Rails 4
- #293 V3 api rails
- #279 Use Rails URL helpers
- #210 404 on redirect to API login
- #126 gem rb-fsevent fail to build native extension

Attach images by dragging & dropping or selecting them.
We broke something in #280
repo redirects
Rename your repository and GitHub will redirect old URLs
in-repo licensing
Initialize this repository with a README
This will allow you to git clone the repository immediately.

Add .gitignore: None
Add a license: None

Create repository

Filter licenses...

- GPL v2
- **MIT License**
- Affero GPL
- Artistic License 2.0
You can add licenses to existing repos, too.
in-repo licensing
in-repo licensing
in-repo licensing

(also works for .gitignore)
file finder
file finder

Type t, find in project
task lists
task lists

Use issues and pulls as todo lists
- [ ] Use NoSQL
- [ ] Use today's trendiest language
- [ ] Do what our VC tells us
- [ ] Comment on Hacker News
- [ ] Profit
holman opened this pull request a minute ago

Make a successful startup

- Use NoSQL
- Use today's trendiest language
- Do what our VC tells us
- Comment on Hacker News
- Profit
holman opened this pull request 2 minutes ago

Make a successful startup

<table>
<thead>
<tr>
<th>Task</th>
<th>Assigned</th>
<th>Milestone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use NoSQL</td>
<td>No one is assigned</td>
<td>No milestone</td>
</tr>
<tr>
<td>Use today's trendiest language</td>
<td>No one is assigned</td>
<td>No milestone</td>
</tr>
<tr>
<td>Do what our VC tells us</td>
<td>No one is assigned</td>
<td>No milestone</td>
</tr>
<tr>
<td>Comment on Hacker News</td>
<td>No one is assigned</td>
<td>No milestone</td>
</tr>
<tr>
<td>Profit</td>
<td>No one is assigned</td>
<td>No milestone</td>
</tr>
</tbody>
</table>
- [x] Use NoSQL
- [x] Use today's trendiest language
- [x] Do what our VC tells us
- [x] Comment on Hacker News
- [ ] Profit
Make a successful startup
5 tasks (4 completed, 1 remaining)
by holman 7 minutes ago

haha-business!
fork URLss
fork URLs

https://github/user/repo/fork
Where should we fork this repository?

@holman

@github

@play

@libgit2
fork URLs

README links
A lot of stuff. Seriously, a lot of stuff. Check them out in the file browser above and see what components may mesh up with you. Fork it, remove what you don't use, and build on what you do use.
Remember:
Git is probably terrible
But it’s also neato
Thanks!