

A list of cool features of Git and GitHub. <http://git.io/sheet>

428 commits

1 branch

0 releases

65 contributors

Branch: master github-cheat-sheet / +

Awesome badge

tiimgreen authored on Aug 5 latest commit 8a7dd3435b

CONTRIBUTING.md	CONTRIBUTING.md	8 months ago
LICENSE	Add LICENSE	a year ago
README.ja.md	Awesome badge	a month ago
README.ko.md	Awesome badge	a month ago
README.md	Awesome badge	a month ago
README.zh-cn.md	Awesome badge	a month ago
README.zh-tw.md	Awesome badge	a month ago

README.md

# GitHub Cheat Sheet

A collection of cool hidden and not so hidden features of Git and GitHub. This cheat sheet was inspired by [Zach Holman's Git and GitHub Secrets](#) talk at Aloha Ruby Conference 2012 ([slides](#)) and his [More Git and GitHub Secrets](#) talk at WDCNZ 2013 ([slides](#)).

Shortlink: <http://git.io/sheet>

Read this in other languages: [English](#), [한국어](#), [日本語](#), [简体中文](#), [正體中文](#).

## Table of Contents

- GitHub
  - Ignore Whitespace
  - Adjust Tab Space
  - Commit History by Author
  - Cloning a Repository
  - Branch
    - Compare all Branches to Another Branch
    - Comparing Branches
    - Compare Branches across Forked Repositories
  - Gists
  - Git.io
  - Keyboard Shortcuts
  - Line Highlighting in Repositories
  - Closing Issues via Commit Messages

<> Code

Issues 3

Pull requests 1

Pulse

Graphs

HTTPS clone URL

<https://github.com/>

You can clone with [HTTPS](#) or [Subversion](#).

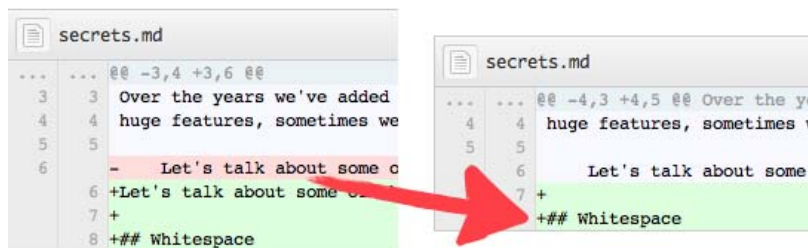
Clone in Desktop

Download ZIP

- [Cross-Link Issues](#)
- [Locking Conversations](#)
- [CI Status on Pull Requests](#)
- [Filters](#)
- [Syntax Highlighting in Markdown Files](#)
- [Emojis](#)
- [Images/GIFs](#)
  - [Embedding Images in GitHub Wiki](#)
- [Quick Quoting](#)
- [Pasting Clipboard Image to Comments](#)
- [Quick Licensing](#)
- [Task Lists](#)
  - [Task Lists in Markdown Documents](#)
- [Relative Links](#)
- [Metadata and Plugin Support for GitHub Pages](#)
- [Viewing YAML Metadata in your Documents](#)
- [Rendering Tabular Data](#)
- [Rendering PDF](#)
- [Revert a Pull Request](#)
- [Diffs](#)
  - [Rendered Prose Diffs](#)
  - [Diffable Maps](#)
  - [Expanding Context in Diffs](#)
  - [Diff or Patch of Pull Request](#)
  - [Rendering and diffing images](#)
- [Hub](#)
- [Contributing Guidelines](#)
- [Octicons](#)
- [GitHub Resources](#)
  - [GitHub Talks](#)
- [Git](#)
  - [Remove All Deleted Files from the Working Tree](#)
  - [Previous Branch](#)
  - [StripSpace](#)
  - [SSH keys](#)
  - [Checking out Pull Requests](#)
  - [Empty Commits](#)
  - [Styled Git Status](#)
  - [Styled Git Log](#)
  - [Git Query](#)
  - [Git Grep](#)
  - [Merged Branches](#)
  - [Fixup and Autosquash](#)
  - [Web Server for Browsing Local Repositories](#)
  - [Git Configurations](#)
    - [Aliases](#)
    - [Auto-Correct](#)
    - [Color](#)
  - [Git Resources](#)
    - [Git Books](#)

## Ignore Whitespace

Adding `?w=1` to any diff URL will remove any changes only in whitespace, enabling you to see only that code that has changed.



[Read more about GitHub secrets.](#)

## Adjust Tab Space

Adding `?ts=4` to a diff or file URL will display tab characters as 4 spaces wide instead of the default 8. The number after `ts` can be adjusted to suit your preference. This does not work on Gists, or raw file views, but a [Chrome](#) or [Opera](#) extension can automate this.

Here is a Go source file before adding `?ts=4` :

```
file 69 lines (57 sloc) 1.86 kb
1 package flint
2
3 import (
4     "path/filepath"
5 )
6
7 type lintError struct {
8     Level int
9     Message string
10 }
11
```

...and this is after adding `?ts=4` :

```
file 69 lines (57 sloc) 1.86 kb
1 package flint
2
3 import (
4     "path/filepath"
5 )
6
7 type lintError struct {
8     Level int
9     Message string
10 }
```

## Commit History by Author

To view all commits on a repo by author add `?author={user}` to the URL.

```
https://github.com/rails/rails/commits/master?author=dhh
```

branch: master

Commits on Jan 13, 2015

- Merge pull request #18476 from AlamoZ/scaffold\_index\_view\_grammar  
dhh authored 5 days ago [78a4884](#) [Compare](#)
- Stop promoting rack-cache usage at the moment (not so common or impor...  
dhh authored 5 days ago [1302edf](#) [Compare](#)
- Show how to change the queuing backend for ActiveJob in production  
dhh authored 5 days ago [6463495](#) [Compare](#)
- Set all asset options together  
dhh authored 5 days ago [b9b28d8](#) [Compare](#)

Commits on Jan 9, 2015

- Merge pull request #18413 from brainopia/automatic\_inverse\_of\_for\_bel...  
dhh authored 9 days ago [6eb499f](#) [Compare](#)

[Read more about the differences between commits views.](#)

# Cloning a Repository

When cloning a repository the `.git` can be left off the end.

```
$ git clone https://github.com/tiimgreen/github-cheat-sheet
```

[Read more about the Git `clone` command.](#)

# Branch

## Compare all Branches to Another Branch

If you go to the repo's [Branches](#) page, next to the Commits button:

```
https://github.com/{user}/{repo}/branches
```

... you would see a list of all branches which are not merged into the main branch.

From here you can access the compare page or delete a branch with a click of a button.

rails / rails

Watch

1,732

Unstar

24,618

Fork

9,521

Overview

Active

Stale

All branches

Search branches...

Default branch

master

Updated an hour ago by tbn

✓

Default

Active branches

4-2-stable	Updated 2 hours ago by sgrif	✗	1147   269	Compare
4-1-stable	Updated 4 days ago by rafaelfranca	✓	6892   672	Compare
fix_nested_transactions_for_re...	Updated 8 days ago by chancancode	✗	149   3	Compare
3-2-stable	Updated 12 days ago by rafaelfranca	✗	22478   1757	Compare
4-0-stable	Updated 12 days ago by rafaelfranca	✓	11974   1081	Compare

View more active branches

## Comparing Branches

To use GitHub to compare branches, change the URL to look like this:

```
https://github.com/{user}/{repo}/compare/{range}
```

Where `{range}` = `master...4-1-stable`

For example:

```
https://github.com/rails/rails/compare/master...4-1-stable
```

We're showing branches in this repository, but you can also [compare across forks](#).

base: `master` ... compare: `4-1-stable`

Create pull request Discuss and review the changes in this comparison with others.

Commits 672 Files changed 509 Commit comments 5 45 contributors

This comparison is big! We're only showing the most recent 250 commits

Commits on Jun 19, 2014

	Merge pull request #15813 from DNXX/valid-action-name-refactoring ...	✓ 6413eb4
	Fix has_and_belongs_to_many in a namespaced model pointing to a non n...	✓ 0769465
	Fix has_and_belongs_to_many in a namespaced model pointing to a non n...	✓ 1a12bee
	Merge pull request #15772 from nbudin/sti_through_bug ...	✓ 8f76511
	Merge pull request #15772 from nbudin/sti_through_bug ...	✓ a44feed
	Merge pull request #15729 from sgrif/sg-double-save-hm-t-4-1-stable ...	✓ ae0d952

`{range}` can be changed to things like:

```
https://github.com/rails/rails/compare/master@{1.day.ago}...master
https://github.com/rails/rails/compare/master@{2014-10-04}...master
```

Dates are in the format `YYYY-MM-DD`

master@{2014-10-04} ... master Edit

130 commits 123 files changed 5 comments 39 contributors

Commits Files changed Commit comments

Aug 07, 2013

	Add tests to ActiveSupport::XmlMini to_tag method	✓ 05d7cde
--	---	-----------

Nov 17, 2013

	Fix insertion of records for hmt association with scope, fix #3548	✗ ec09280
--	--	-----------

Jan 07, 2014

	Auto-generate stable fixture UUIDs on PostgreSQL. ...	✓ 9330631
--	---	-----------

Branches can also be compared in `diff` and `patch` views:

```
https://github.com/rails/rails/compare/master...4-1-stable.diff
https://github.com/rails/rails/compare/master...4-1-stable.patch
```

[Read more about comparing commits across time.](#)

## Compare Branches across Forked Repositories

To use GitHub to compare branches across forked repositories, change the URL to look like this:

```
https://github.com/{user}/{repo}/compare/{foreign-user}:{branch}...{own-branch}
```

For example:

```
https://github.com/rails/rails/compare/byroot:master...master
```

The screenshot shows a GitHub pull request comparison for the rails/rails repository. At the top, it shows a comparison between 'byroot:master' and 'rails:master'. Below this, there's a green button to 'Create Pull Request' with the text 'Open a Pull Request for this comparison to discuss and review your changes with others.' Below that, statistics are shown: 6,802 commits, 309 files changed, 14 comments, and 64 contributors. There are tabs for 'Commits', 'Files changed', and 'Commit comments'. A yellow banner states 'This comparison is big! We're only showing the most recent 250 commits'. Below this, a list of commits is shown, including one by 'kastiglione' for PostgreSQL support and another by 'rajael' for conditional test cases.

Commit	Author	Message	Status
PostgreSQL, Support for materialized views. [Dave Lee & Yves Senn] ...	kastiglione	PostgreSQL, Support for materialized views. [Dave Lee & Yves Senn] ...	✓ def6071
We can conditional define the tests depending on the adapter or ...	rajael	We can conditional define the tests depending on the adapter or ...	✗ ee36af1
Merge pull request #14565 from rajael/conditional_test_cases ...	rajael	Merge pull request #14565 from rajael/conditional_test_cases ...	✓ c82483a
DRY AS::SafeBuffer a bit using existing helper	rwz	DRY AS::SafeBuffer a bit using existing helper	✓ 8482895
Fixed small documentation typo ...	alex88	Fixed small documentation typo ...	✗ 80e3f24
Merge pull request #14568 from alex88/patch-1 ...	rajael	Merge pull request #14568 from alex88/patch-1 ...	✓ 3bcc51a

## Gists

[Gists](#) are an easy way to work with small bits of code without creating a fully fledged repository.

The screenshot shows a GitHub Gist page for a Ruby program named 'app.rb'. The gist is public and was created 4 days ago by 'tiimgreen / app.rb'. It contains a single line of code: 'puts 'Hello World''. Below the code, there's a comment from 'rafaelchmiel' saying 'Wow, dat is some engineering.' and a form to add a new comment. The page also includes a 'Download Gist' button and a 'Clone this gist' link.

**Gist Detail**

Revisions: 1

Download Gist

Clone this gist  
<https://gist.github.com>

Embed this gist  
`<script src="https://gist.github.com/rajael/conditional_test_cases">`

Link to this gist  
<https://gist.github.com>

**app.rb** Ruby

```
1 puts 'Hello World'
```

**rafaelchmiel** commented 2 days ago

Wow, dat is some engineering.

**Write** **Preview**

Comments are parsed with GitHub Flavored Markdown

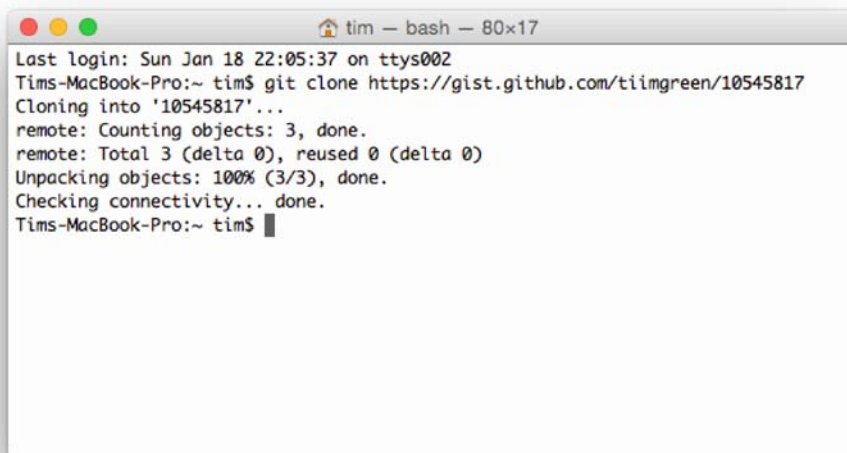
Leave a comment

Add Comment

Add `.pibb` to the end of any Gist URL ([like this](#)) in order to get the *HTML only* version suitable for embedding in any other site.

Gists can be treated as a repository so they can be cloned like any other:

```
$ git clone https://gist.github.com/tiimgreen/10545817
```



This means you also can modify and push updates to Gists:

```
$ git commit
$ git push
Username for 'https://gist.github.com':
Password for 'https://tiimgreen@gist.github.com':
```

However, Gists do not support directories. All files need to be added to the repository root. [Read more about creating Gists.](#)

## Git.io

[Git.io](#) is a simple URL shortener for GitHub.



You can also use it via pure HTTP using Curl:

```
$ curl -i http://git.io -F "url=https://github.com/..."
HTTP/1.1 201 Created
Location: http://git.io/abc123
```

```
$ curl -i http://git.io/abc123
HTTP/1.1 302 Found
Location: https://github.com/...
```

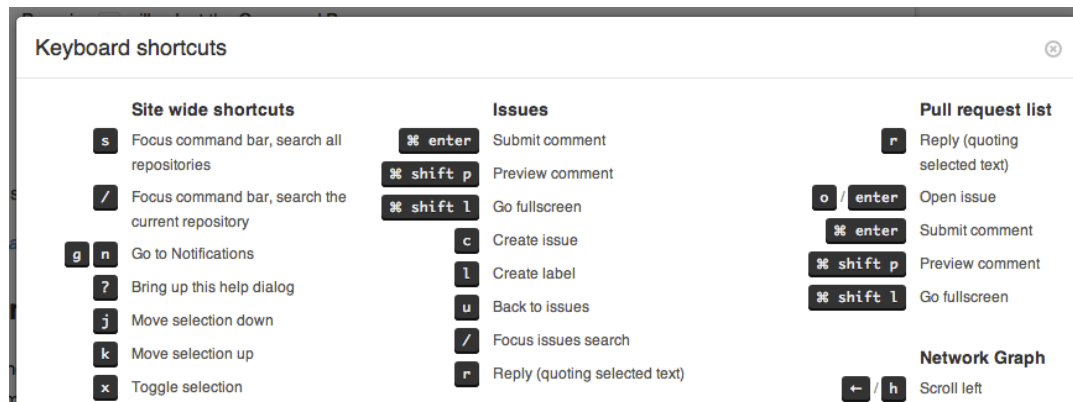
[Read more about Git.io.](#)

## Keyboard Shortcuts

When on a repository page, keyboard shortcuts allow you to navigate easily.

- Pressing `t` will bring up a file explorer.
- Pressing `w` will bring up the branch selector.
- Pressing `s` will focus the search field for the current repository. Pressing Backspace to delete the “This repository” pill changes the field to search all of GitHub.
- Pressing `l` will edit labels on existing Issues.
- Pressing `y` **when looking at a file** (e.g. `https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.md`) will change your URL to one which, in effect, freezes the page you are looking at. If this code changes, you will still be able to see what you saw at that current time.

To see all of the shortcuts for the current page press `?` :



[Read more about search syntax you can use.](#)

## Line Highlighting in Repositories

Either adding `#L52` to the end of a code file URL or simply clicking the line number will highlight that line number.

It also works with ranges, e.g. `#L53-L60`, to select ranges, hold `shift` and click two lines:

```
https://github.com/rails/rails/blob/master/activemodel/lib/active_model.rb#L53-L60
```



```

43  autoload :Serialization
44  autoload :TestCase
45  autoload :Translation
46  autoload :Validations
47  autoload :Validator
48
49  eager_autoload do
50    autoload :Errors
51  end
52
53  module Serializers
54    extend ActiveSupport::Autoload
55
56    eager_autoload do
57      autoload :JSON
58      autoload :Xml
59    end
60  end
61
62  def self.eager_load!
63    super
64    ActiveModel::Serializers.eager_load!
65  end
66 end
67
68 ActiveSupport.on_load(:i18n) do
69   I18n.load_path << File.dirname(__FILE__) + '/active_model/locale/en.yml'
70 end

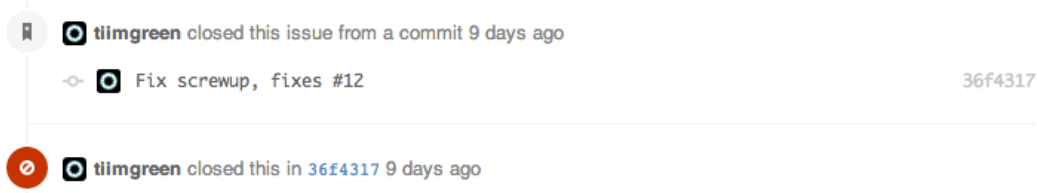
```

## Closing Issues via Commit Messages

If a particular commit fixes an issue, any of the keywords `fix/fixes/fixed`, `close/closes/closed` or `resolve/resolves/resolved`, followed by the issue number, will close the issue once it is committed to the master branch.

```
$ git commit -m "Fix screwup, fixes #12"
```

This closes the issue and references the closing commit.



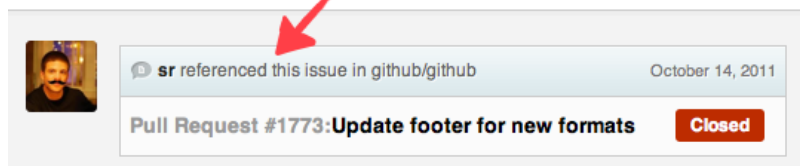
[Read more about closing Issues via commit messages.](#)

## Cross-Link Issues

If you want to link to another issue in the same repository, simply type hash `#` then the issue number, and it will be auto-linked.


To link to an issue in another repository, `{user}/{repo}#ISSUE_NUMBER` e.g. `tiimgreen/toc#12`.

We should probably handle this with [github/enterprise#59](#)



## Locking Conversations

Pull Requests and Issues can now be locked by owners or collaborators of the repo.

 Parsed as Markdown  Edit in fullscreen

 Lock issue

This means that users who are not collaborators on the project will no longer be able to comment.



Write Preview

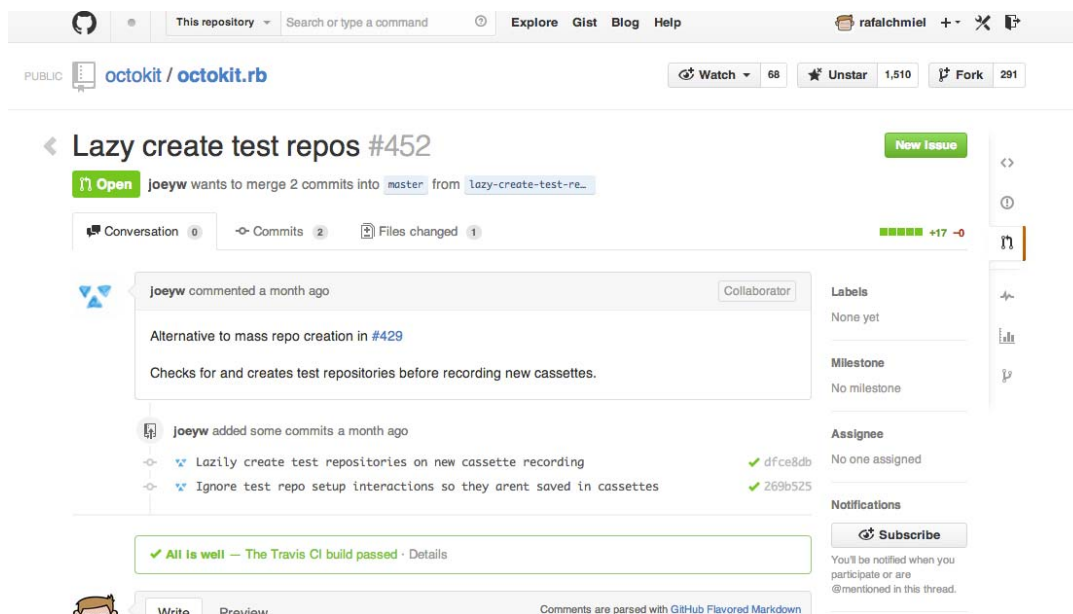


This conversation has been locked and limited to collaborators.

[Read more about locking conversations.](#)

## CI Status on Pull Requests

If set up correctly, every time you receive a Pull Request, [Travis CI](#) will build that Pull Request just like it would every time you make a new commit. Read more about how to [get started with Travis CI](#).



The screenshot shows a GitHub pull request interface. At the top, the repository is 'octokit / octokit.rb'. The pull request title is 'Lazy create test repos #452'. It shows 'Open' status, '2 commits', and '1 file changed'. A comment by 'joeyw' is visible, mentioning 'Alternative to mass repo creation in #429' and 'Checks for and creates test repositories before recording new cassettes.' Below the comment, two commits are listed: 'Lazily create test repositories on new cassette recording' and 'Ignore test repo setup interactions so they arent saved in cassettes'. A green status bar indicates 'All is well - The Travis CI build passed'. On the right, there are sections for 'Labels', 'Milestone', 'Assignee', and 'Notifications'.

[Read more about the commit status API.](#)

## Filters

Both issues and pull requests allow filtering in the user interface.

For the Rails repo: <https://github.com/rails/rails/issues>, the following filter is built by selecting the label "activerecord":

```
is:issue label:activerecord
```

But, you can also find all issues that are NOT labeled activerecord:

```
is:issue -label:activerecord
```

Additionally, this also works for pull requests:

```
is:pr -label:activerecord
```

Github has tabs for displaying open or closed issues and pull requests but you can also see merged pull requests. Just put the following in the filter:

```
is:merged
```

[Read more about searching issues.](#)

Finally, github now allows you to filter by the Status API's status.

Pull requests with only successful statuses:

```
status:success
```

[Read more about searching on the Status API.](#)

## Syntax Highlighting in Markdown Files

For example, to syntax highlight Ruby code in your Markdown files write:

```
```ruby
require 'tabbit'
table = Tabbit.new('Name', 'Email')
table.add_row('Tim Green', 'tiimgreen@gmail.com')
puts table.to_s
```
```

This will produce:

```
require 'tabbit'
table = Tabbit.new('Name', 'Email')
table.add_row('Tim Green', 'tiimgreen@gmail.com')
puts table.to_s
```

GitHub uses [Linguist](#) to perform language detection and syntax highlighting. You can find out which keywords are valid by perusing the [languages YAML file](#).

[Read more about GitHub Flavored Markdown.](#)

## Emojis

Emojis can be added to Pull Requests, Issues, commit messages, repository descriptions, etc. using

```
:name_of_emoji: .
```

The full list of supported Emojis on GitHub can be found at [emoji-cheat-sheet.com](#) or [scotch-io/All-Github-Emoji-Icons](#). A handy emoji search engine can be found at [emoji.muan.co](#).

The top 5 used Emojis on GitHub are:

1. `:shipit:`
2. `:sparkles:`
3. `:-1:`
4. `:+1:`
5. `:clap:`

## Images/GIFs

Images and GIFs can be added to comments, READMEs etc.:

```
![Alt Text](http://www.sheawong.com/wp-content/uploads/2013/08/keephatin.gif)
```

Raw images from the repo can be used by calling them directly.:

```
![Alt Text](https://github.com/{user}/{repo}/raw/master/path/to/image.gif)
```



All images are cached on GitHub, so if your host goes down, the image will remain available.

## Embedding Images in GitHub Wiki


There are multiple ways of embedding images in Wiki pages. There's the standard Markdown syntax (shown above). But there's also a syntax that allows things like specifying the height or width of the image:

```
[[ http://www.sheawong.com/wp-content/uploads/2013/08/keephatin.gif | height = 100px ]]
```

Which produces:

[Home](#) [Pages](#) [History](#) [New Page](#)

## Home (Preview)



### README not found

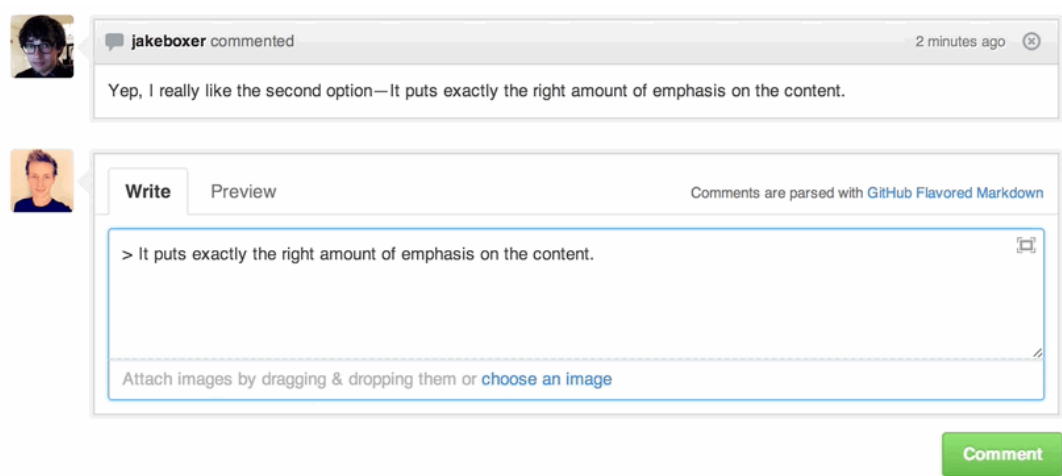
While a README isn't a required part of an open source project, it is a very good idea to have one. READMEs are a great place to describe your project or add some documentation such as how to install or use your project. You might want to include contact information - if your project becomes popular people will want to help you out. See [GitHub's article on creating repositories](#). See [Tom Preston-Werner's blog post on README driven development](#). Also see the [Wikipedia article on READMEs](#).

[↩](#)  
[🔍](#)  
[🔖](#)  
[📖](#)  
[🔗](#)  
[📊](#)  
[🔖](#)  
[✂](#)

## Quick Quoting

When on a comment thread and you want to quote something someone previously said, highlight the

text and press `>`, this will copy it into your text box in the block-quote format.

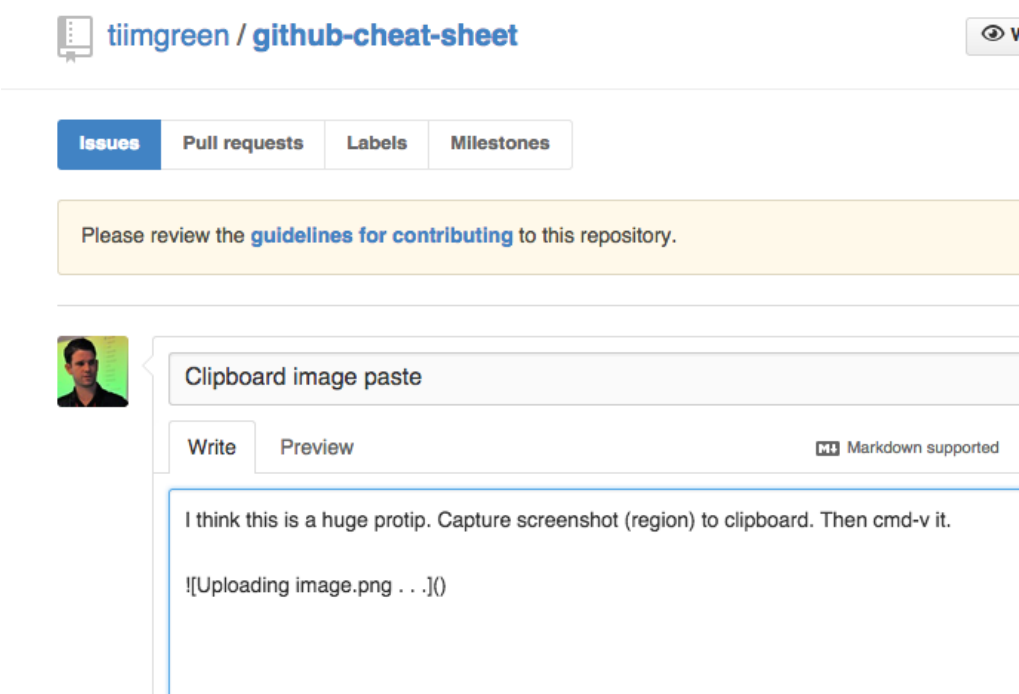


[Read more about quick quoting.](#)

## Pasting Clipboard Image to Comments

*(Works on Chrome browsers only)*

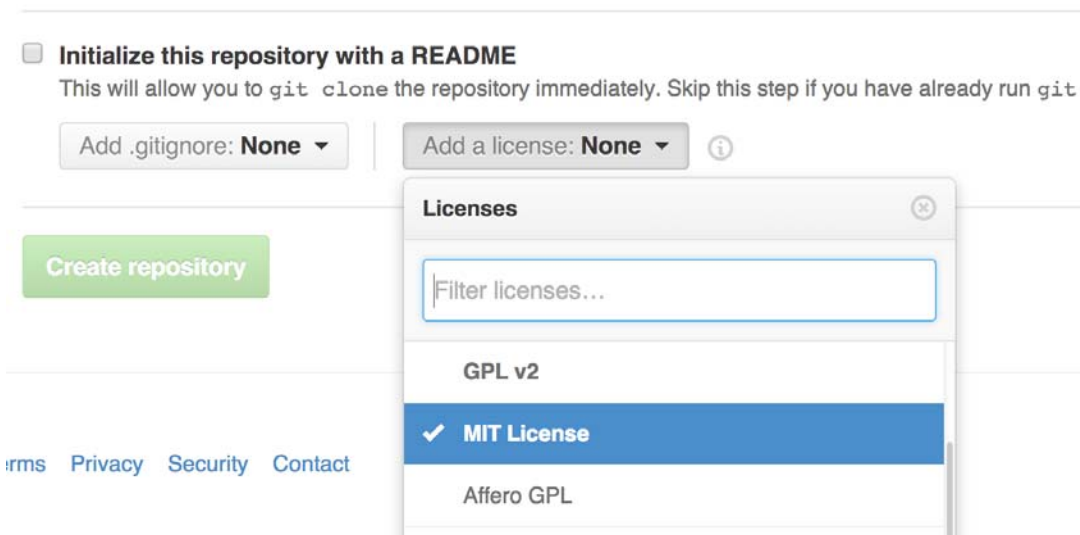
After taking a screenshot and adding it to the clipboard (mac: `cmd-ctrl-shift-4`), you can simply paste (`cmd-v` / `ctrl-v`) the image into the comment section and it will be auto-uploaded to github.



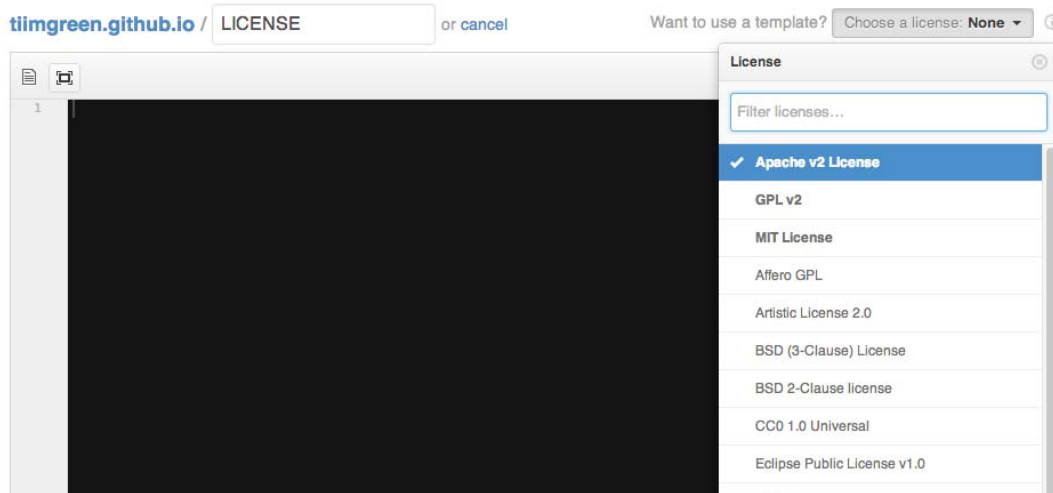
[Read more about issue attachments.](#)

## Quick Licensing

When creating a repository, GitHub gives you the option of adding in a pre-made license:



You can also add them to existing repositories by creating a new file through the web interface. When the name `LICENSE` is typed in you will get an option to use a template:



Also works for `.gitignore`.

[Read more about open source licensing.](#)

## Task Lists

In Issues and Pull requests check boxes can be added with the following syntax (notice the space):

- [ ] Be awesome
- [ ] Prepare dinner
  - [ ] Research recipe
  - [ ] Buy ingredients
  - [ ] Cook recipe
- [ ] Sleep

## < TODO #1



AlexandreArpin opened this issue just now · 0 comments



AlexandreArpin commented just now

- ☐ Be awesome
- ☐ Prepare dinner
  - ☐ Research recipe
  - ☐ Buy ingredients
  - ☐ Cook recipe
- ☐ Sleep

When they are clicked, they will be updated in the pure Markdown:

- [x] Be awesome
- [ ] Prepare dinner
  - [x] Research recipe
  - [x] Buy ingredients
  - [ ] Cook recipe
- [ ] Sleep

[Read more about task lists.](#)

## Task Lists in Markdown Documents

In full Markdown documents **read-only** checklists can now be added using the following syntax:

- [ ] Mercury
- [x] Venus
- [x] Earth
  - [x] Moon
- [x] Mars
  - [ ] Deimos
  - [ ] Phobos

- ☐ Mercury
- ☒ Venus
- ☒ Earth
  - ☒ Moon
- ☒ Mars
  - ☐ Deimos
  - ☐ Phobos

[Read more about task lists in markdown documents.](#)

## Relative Links

Relative links are recommended in your Markdown files when linking to internal content.

```
[Link to a header](#awesome-section)
[Link to a file](docs/readme)
```

Absolute links have to be updated whenever the URL changes (e.g. repository renamed, username

changed, project forked). Using relative links makes your documentation easily stand on its own.

[Read more about relative links.](#)

## Metadata and Plugin Support for GitHub Pages

Within Jekyll pages and posts, repository information is available within the `site.github` namespace, and can be displayed, for example, using `{{ site.github.project_title }}`.

The Jemoji and jekyll-mentions plugins enable [emoji](#) and [@mentions](#) in your Jekyll posts and pages to work just like you'd expect when interacting with a repository on GitHub.com.

[Read more about repository metadata and plugin support for GitHub Pages.](#)

## Viewing YAML Metadata in your Documents

Many blogging websites, like [Jekyll](#) with [GitHub Pages](#), depend on some YAML-formatted metadata at the beginning of your post. GitHub will render this metadata as a horizontal table, for easier reading

file61 lines (39 slocc)1.415 kb

OpenEditRawBlameHistoryDelete

| layout | title                         | description                   | tags             | path                                      | eventdate  |
|--------|-------------------------------|-------------------------------|------------------|---|------------|
| bare   | Git and GitHub Review for NYU | Git and GitHub Review for NYU | notesonlineclass | classnotes/2013-02-13-NYU-github-class.md | 2013-02-13 |

### Your instructors for the evening are:

- Matthew McCullough ([Twitter](#), [GitHub](#))
- Tim Berglund ([Twitter](#), [GitHub](#))

[Read more about viewing YAML metadata in your documents.](#)

## Rendering Tabular Data

GitHub supports rendering tabular data in the form of `.csv` (comma-separated) and `.tsv` (tab-separated) files.



| file   869 lines (868 sloc)   188.222 kb                         |                        |              |                                       |                                       |                           |
|--|------------------------|--------------|---------------------------------------|---------------------------------------|---------------------------|
| <a>Edit</a> <a>Raw</a> <a>Blame</a> <a>History</a> <a>Delete</a> |                        |              |                                       |                                       |                           |
| Q Search this file...  |                        |              |                                       |                                       |                           |
| 1  | Title                  | Release Year | Locations                             | Fun Facts                             | Production Company        |
| 2  | 180                    | 2011         | 555 Market St.                        |                                       | SPI Cinemas               |
| 3  | 180                    | 2011         | Epic Roasthouse (399 Embarcade...     |                                       | SPI Cinemas               |
| 4  | 180                    | 2011         | Mason & California Streets (Nob Hill) |                                       | SPI Cinemas               |
| 5  | 180                    | 2011         | Justin Herman Plaza                   |                                       | SPI Cinemas               |
| 6  | 180                    | 2011         | 200 block Market Street               |                                       | SPI Cinemas               |
| 7  | 180                    | 2011         | City Hall                             |                                       | SPI Cinemas               |
| 8  | 180                    | 2011         | Polk & Larkin Streets                 |                                       | SPI Cinemas               |
| 9  | 180                    | 2011         | Randall Musuem                        |                                       | SPI Cinemas               |
| 10   | 24 Hours on Craigslist | 2005         |                                       |                                       | Yerba Buena Productions   |
| 11   | 48 Hours               | 1982         |                                       |                                       | Paramount Pictures        |
| 12   | 50 First Dates         | 2004         | Rainforest Café (145 Jefferson Str... |                                       | Columbia Pictures Corpora |
| 13   | A Jitney Elopement     | 1915         | Golden Gate Park                      | During San Francisco's Gold Rush...   | The Essanay Film Manufact |
| 14   | A Jitney Elopement     | 1915         | 20th and Folsom Streets               |                                       | The Essanay Film Manufact |
| 15   | A Night Full of Rain   | 1978         | San Francisco Chronicle (901 Mis...   | The San Francisco Zodiac Killer of... | Liberty Film              |

[Read more about rendering tabular data.](#)

## Rendering PDF

GitHub supports rendering PDF:



[Read more about rendering PDF.](#)

## Revert a Pull Request

After a pull request is merged, you may find it does not help anything or it was a bad decision to merge the pull request.

You can revert it by clicking the **Revert** button on the right side of a commit in the pull request page to create a pull request with reverted changes to this specific pull request.

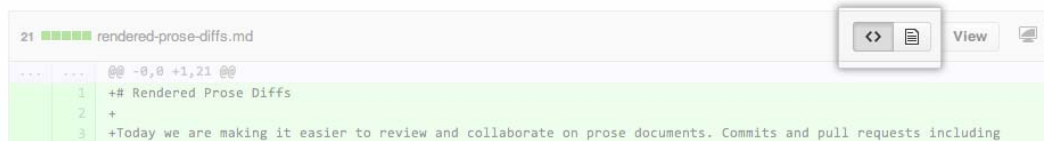


[Read more about reverting pull requests](#)

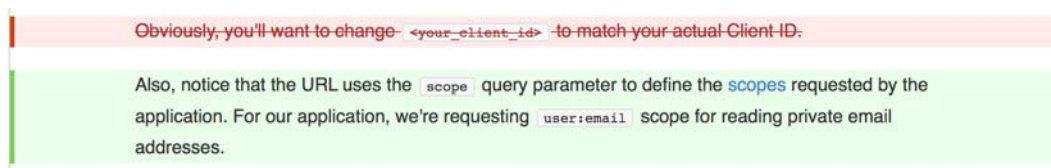
## Diffs

### Rendered Prose Diffs

Commits and pull requests, including rendered documents supported by GitHub (e.g. Markdown), feature *source* and *rendered* views.



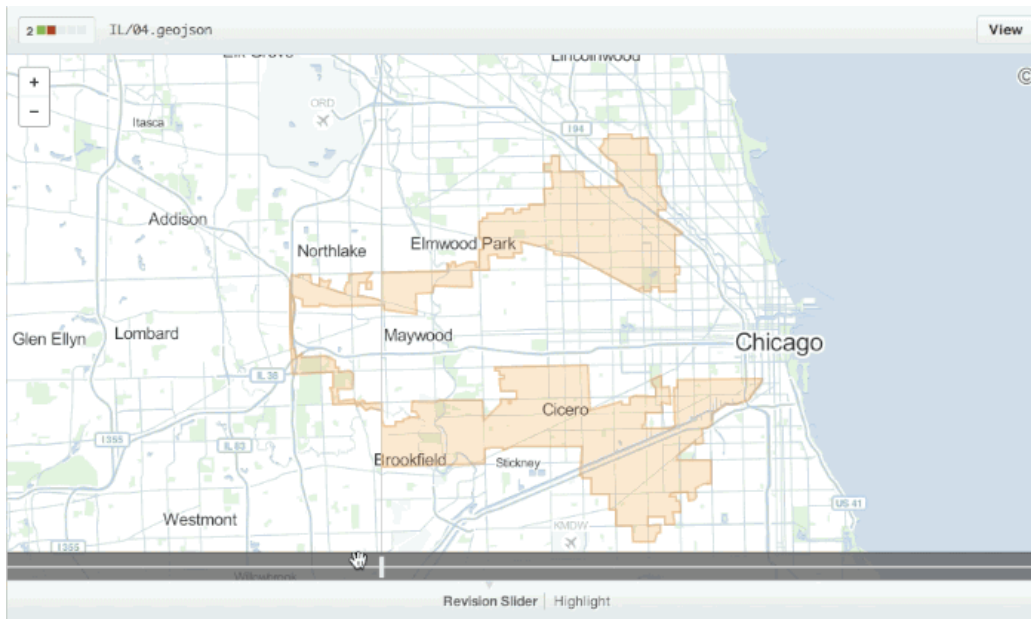
Click the "rendered" button to see the changes as they'll appear in the rendered document. Rendered prose view is handy when you're adding, removing, and editing text:



[Read more about rendered prose diffs.](#)

### Diffable Maps

Any time you view a commit or pull request on GitHub that includes geodata, GitHub will render a visual representation of what was changed.



[Read more about diffable maps.](#)

## Expanding Context in Diffs

Using the *unfold* button in the gutter of a diff, you can reveal additional lines of context with a click. You can keep clicking *unfold* until you've revealed the whole file, and the feature is available anywhere GitHub renders diffs.

|     |     |  |
|-----|-----|--|
| 235 | 249 | - (id)initWithServer:(OCTServer *)server {   |
| 236 | 250 | NSParameterAssert(server != nil);  |
| 237 | 251 |  |
| 238 | 252 | self = [super initWithBaseURL:server.APIEndpoint];   |
| 239 | 253 | if (self == nil) return nil;   |
| 240 | 254 | NSString *userAgent = self.class.userAgent;  |
| 241 | 255 | if (userAgent != nil) [self setDefaultHeader:@"User-Agent" value:userAgent];                                     |
| 242 | 256 |  |
| 243 | 257 |  |
| 244 | 258 | - self.parameterEncoding = AFJSONParameterEncoding;  |
| 245 | 259 | - [self setDefaultHeader:@"Accept" value:@"application/vnd.github.beta+json"];                                   |
| 246 | 260 | -  |
| 247 | 261 | [AFHTTPRequestOperation addAcceptableStatusCodes:[NSIndexSet indexSetWithIndex:OCTClientNotModifiedStatusCode]]  |
| 248 | 262 | - [AFJSONRequestOperation addAcceptableContentTypes:[NSSet initWithObject:@"application/vnd.github.beta+json"]]; |
| 249 | 263 | NSString *contentType = [NSString stringWithFormat:@"application/vnd.github.%@+json", OCTClientAPIVersion];      |
| 250 | 264 | [self setDefaultHeader:@"Accept" value:contentType];   |
| 251 | 265 | [AFJSONRequestOperation addAcceptableContentTypes:[NSSet initWithObject:contentType]];                           |
| 252 | 266 | self.parameterEncoding = AFJSONParameterEncoding;  |
| 253 | 267 | [self registerHTTPOperationClass:AFJSONRequestOperation.class];  |
| 254 | 268 |  |
| 255 | 269 | return self;   |

[Read more about expanding context in diffs.](#)

## Diff or Patch of Pull Request

You can get the diff of a Pull Request by adding a `.diff` or `.patch` extension to the end of the URL. For example:

```
https://github.com/tiimgreen/github-cheat-sheet/pull/15
https://github.com/tiimgreen/github-cheat-sheet/pull/15.diff
https://github.com/tiimgreen/github-cheat-sheet/pull/15.patch
```

The `.diff` extension would give you this in plain text:

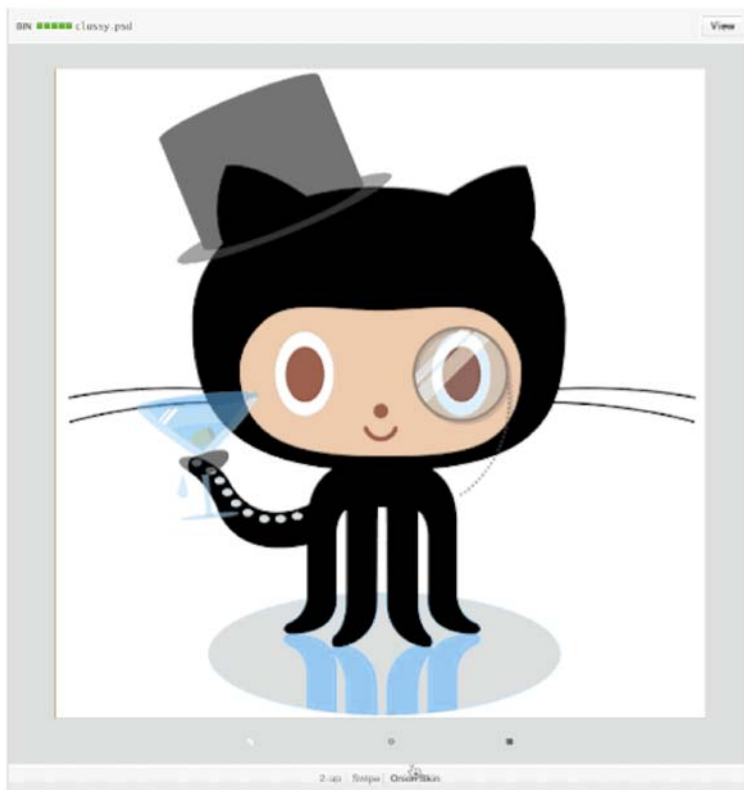
```
diff --git a/README.md b/README.md
index 88fcf69..8614873 100644
--- a/README.md
+++ b/README.md
@@ -28,6 +28,7 @@ All the hidden and not hidden features of Git and GitHub. This cheat sheet wa
- [Merged Branches](#merged-branches)
- [Quick Licensing](#quick-licensing)
- [TODO Lists](#todo-lists)
+- [Relative Links](#relative-links)
- [.gitconfig Recommendations](#gitconfig-recommendations)
  - [Aliases](#aliases)
  - [Auto-correct](#auto-correct)
@@ -381,6 +382,19 @@ When they are clicked, they will be updated in the pure Markdown:
- [ ] Sleep

(...)

```

## Rendering and diffing images

GitHub can display several common image formats, including PNG, JPG, GIF, and PSD. In addition, there are several ways to compare differences between versions of those image formats.



[Read more about rendering and diffing images.](#)

## Hub

[Hub](#) is a command line Git wrapper that gives you extra features and commands that make working with GitHub easier.

This allows you to do things like:

```
$ hub clone tiimgreen/toc
```

[Check out some more cool commands Hub has to offer.](#)

## Contributing Guidelines

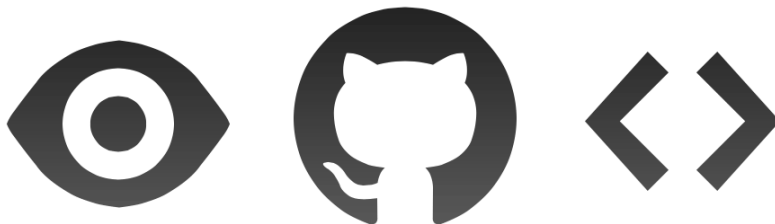
Adding a `CONTRIBUTING` file to the root of your repository will add a link to your file when a contributor creates an Issue or opens a Pull Request.



[Read more about contributing guidelines.](#)

## Octicons

GitHub's icons (Octicons) have now been open sourced.



[Read more about GitHub's Octicons](#)

## GitHub Resources

| Title  | Link  |
|--|---|
| GitHub Explore   | <a href="https://github.com/explore">https://github.com/explore</a>       |
| GitHub Blog  | <a href="https://github.com/blog">https://github.com/blog</a>             |
| GitHub Help  | <a href="https://help.github.com/">https://help.github.com/</a>           |
| GitHub Training  | <a href="https://training.github.com/">https://training.github.com/</a>   |
| GitHub Developer   | <a href="https://developer.github.com/">https://developer.github.com/</a> |
| Github Education (Free Micro Account and other stuff for |   |

students)

<https://education.github.com/>

## GitHub Talks

| Title   | Link  |
|---|---|
| How GitHub Uses GitHub to Build GitHub          | <a href="https://www.youtube.com/watch?v=qyz3jkOBbQY">https://www.youtube.com/watch?v=qyz3jkOBbQY</a> |
| Introduction to Git with Scott Chacon of GitHub | <a href="https://www.youtube.com/watch?v=ZDR433b0HJY">https://www.youtube.com/watch?v=ZDR433b0HJY</a> |
| How GitHub No Longer Works                      | <a href="https://www.youtube.com/watch?v=gXD1ITW7iZI">https://www.youtube.com/watch?v=gXD1ITW7iZI</a> |
| Git and GitHub Secrets                          | <a href="https://www.youtube.com/watch?v=Foz9yvMkvIA">https://www.youtube.com/watch?v=Foz9yvMkvIA</a> |
| More Git and GitHub Secrets                     | <a href="https://www.youtube.com/watch?v=p50xsL-iVgU">https://www.youtube.com/watch?v=p50xsL-iVgU</a> |

## Git

### Remove All Deleted Files from the Working Tree

When you delete a lot of files using `/bin/rm` you can use the following command to remove them from the working tree and from the index, eliminating the need to remove each one individually:

```
$ git rm $(git ls-files -d)
```

For example:

```
$ git status
On branch master
Changes not staged for commit:
  deleted:    a
  deleted:    c

$ git rm $(git ls-files -d)
rm 'a'
rm 'c'

$ git status
On branch master
Changes to be committed:
  deleted:    a
  deleted:    c
```

### Previous Branch

To move to the previous branch in Git:

```
$ git checkout -
# Switched to branch 'master'

$ git checkout -
# Switched to branch 'next'

$ git checkout -
```

```
# Switched to branch 'master'
```

[Read more about Git branching.](#)

## Stripspace

Git Stripspace:

- Strips trailing whitespace
- Collapses newlines
- Adds newline to end of file

A file must be passed when calling the command, e.g.:

```
$ git stripspace < README.md
```

[Read more about the Git `stripspace` command.](#)

## SSH keys

You can get a list of public ssh keys in plain text format by visiting:

```
https://github.com/{user}.keys
```

e.g. <https://github.com/tiimgreen.keys>

[Read more about accessing public ssh keys.](#)

## Checking out Pull Requests

Pull Requests are special branches on the GitHub repository which can be retrieved locally in several ways:

Retrieve a specific Pull Request and store it temporarily in `FETCH_HEAD` for quickly `diff` ing or `merge` ing:

```
$ git fetch origin refs/pull/[PR-Number]/head
```

Acquire all Pull Request branches as local remote branches by refspec:

```
$ git fetch origin '+refs/pull/*/head:refs/remotes/origin/pr/'
```

Or setup the remote to fetch Pull Requests automatically by adding these corresponding lines in your repository's `.git/config`:

```
[remote "origin"]
  fetch = +refs/heads/*:refs/remotes/origin/*
  url = git@github.com:tiimgreen/github-cheat-sheet.git
```

```
[remote "origin"]
  fetch = +refs/heads/*:refs/remotes/origin/*
  url = git@github.com:tiimgreen/github-cheat-sheet.git
  fetch = +refs/pull/*/head:refs/remotes/origin/pr/*
```

For Fork-based Pull Request contributions, it's useful to `checkout` a remote branch representing the Pull Request and create a local branch from it:

```
$ git checkout pr/42 pr-42
```

Or should you work on more repositories, you can globally configure fetching pull requests in the global git config instead.

```
git config --global --add remote.origin.fetch "+refs/pull/*/head:refs/remotes/origin/pr/*"
```

This way, you can use the following short commands in all your repositories:

```
git fetch origin
```

```
git checkout pr/42
```

[Read more about checking out pull requests locally.](#)

## Empty Commits

Commits can be pushed with no code changes by adding `--allow-empty` :

```
$ git commit -m "Big-ass commit" --allow-empty
```

Some use-cases for this (that make sense), include:

- Annotating the start of a new bulk of work or a new feature.
- Documenting when you make changes to the project that aren't code related.
- Communicating with people using your repository.
- The first commit of a repo, as the first commit cannot be rebased later: `git commit -m "init repo" --allow-empty` .



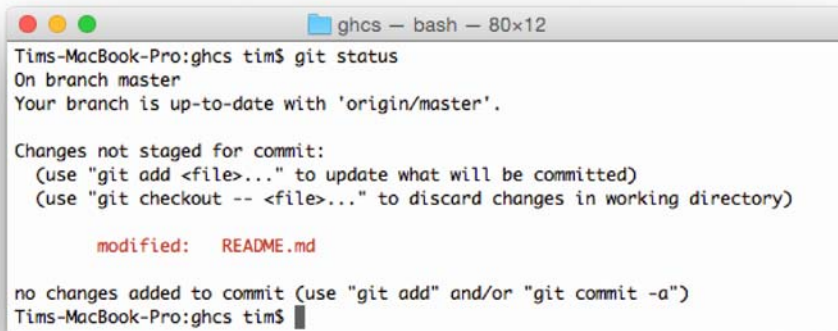


## Styled Git Status

Running:

```
$ git status
```

Produces:



```
ghcs — bash — 80x12
Tims-MacBook-Pro:ghcs tim$ git status
On branch master
Your branch is up-to-date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

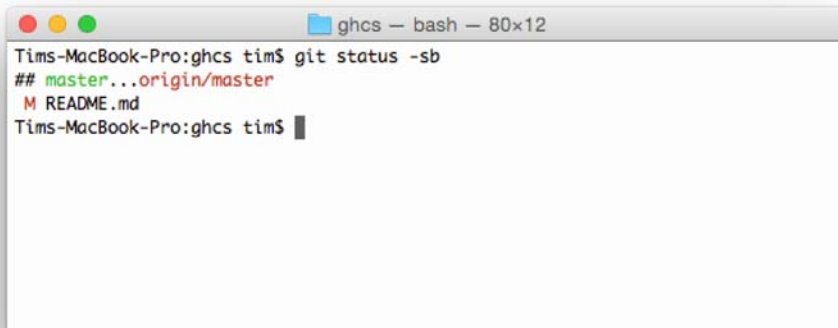
        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
Tims-MacBook-Pro:ghcs tim$
```

By adding `-sb`:

```
$ git status -sb
```

This is produced:



```
ghcs — bash — 80x12
Tims-MacBook-Pro:ghcs tim$ git status -sb
## master...origin/master
M README.md
Tims-MacBook-Pro:ghcs tim$
```

[Read more about the Git `status` command.](#)

## Styled Git Log

Running:

```
$ git log --all --graph --pretty=format:'%Cred%h%Creset -%C(auto)%d%Creset %s %Cgreen(%cr) %C(b
```

Produces:

```
ghcs -- less -- 155x33
Tims-MacBook-Pro:ghcs tim$ git log --all --graph --pretty=format:'%Cred%h%Creset -%C(auto)%d%Creset %s %Cgreen(%cr) %C(bold blue)<Xon>%Creset' --abbrev-com
mit --date=relative
* f122fd1 - (HEAD, origin/master, origin/HEAD, master) Remove note about Emojis in Markdown files, closes #93 (25 minutes ago) <Tim Green>
* 704d07f - CONTRIBUTING.md (29 minutes ago) <Tim Green>
* 3db4f63 - Posting Clipboard Image to Comments, closes #92 (33 minutes ago) <Tim Green>
* 0cfc20 - Remove out-dated feature, fixes #87 (48 minutes ago) <Tim Green>
* 57f1b82 - Update links/images (54 minutes ago) <Tim Green>
* fba6fa9 - Merge pull request #91 from paulirish/patch-1 (73 minutes ago) <Tim Green>
| \
| * 1abd8af - Chrome extension for GH tab size (89 minutes ago) <Paul Irish>
| \
| * 77ece52 - Merge pull request #89 from hail2u/ja-translation (5 months ago) <Tim Green>
| \
| * abdfef8 - Update Ja translation to 6f5cfae (5 months ago) <Kyo Nagashima>
| * edea1aa - Merge pull request #88 from marocchino/ko-update (5 months ago) <Tim Green>
| \
| \
| \
| * ee65723 - Update to 6c9cf82 (5 months ago) <Shim Tio>
| \
| * 6c9cf82 - Merge pull request #86 from tonyxue/master (6 months ago) <Tim Green>
| \
| * 3961f0b - zh-cn update for 9c1f837 (6 months ago) <Tony Xue>
| \
| * 6f5cfae - Needs more :trollface: (6 months ago) <Rafal Chmiel>
| * 9c1f837 - Merge pull request #85 from gWorldz/master (6 months ago) <Rafal Chmiel>
| \
| * b957df8 - Update README.md (6 months ago) <Doc gWorldz>
| * aa8d100 - Update README.md (6 months ago) <Doc gWorldz>
| \
| * 6a1e7aa - Merge pull request #84 from TheCellarRoom/master (6 months ago) <Tim Green>
| \
| * 1eafd88 - Update README.md (6 months ago) <ChelseaStats>
```

Credit to [Palesz](#)

This can be aliased using the instructions found [here](#).

[Read more about the Git Log command.](#)

## Git Query

A Git query allows you to search all your previous commit messages and find the most recent one matching the query.

```
$ git show :/query
```

Where `query` (case-sensitive) is the term you want to search, this then finds the last one and gives details on the lines that were changed.

```
$ git show :/typo
```

```
ghcs -- less -- 140x36
Tims-MacBook-Pro:ghcs tim$ git show :/typo
commit e70c204e3d7cb154f5866b575d01c1da14b68a6c
Author: Rafal Chmiel <hi@rafalchmiel.com>
Date: Mon Apr 14 12:37:57 2014 +0100

    Fix typo

diff --git a/README.md b/README.md
index 2bc494e..40022d2 100644
--- a/README.md
+++ b/README.md
@@ -118,7 +118,7 @@ To see all of the shortcuts for the current page press '?'.

## Closing Issues with Commits

-If a particular commit fixes an issue, any of the keywords 'fix/fixes/fixed' or 'close/closes/closed', followed by the issue number, will c
+If a particular commit fixes an issue, any of the keywords 'fix/fixes/fixed' or 'close/closes/closed', followed by the issue number, will c

'''bash
$ git commit -m "Fix cock up, fixes #12"
(END)
```

Press `q` to quit.

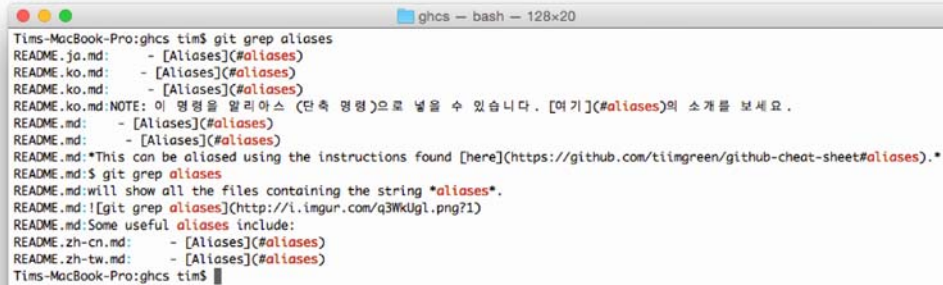
## Git Grep

Git Grep will return a list of lines matching a pattern.

Running:

```
$ git grep aliases
```

will show all the files containing the string *aliases*.



```
Tims-MacBook-Pro:ghcs tim$ git grep aliases
README.ja.md: - [Aliases](#aliases)
README.ko.md: - [Aliases](#aliases)
README.ko.md: - [Aliases](#aliases)
README.ko.md:NOTE: 이 명령을 알리아스 (단축 명령)으로 넣을 수 있습니다. [여기](https://github.com/tiimgreen/github-cheat-sheet#aliases)의 소개를 보세요.
README.md: - [Aliases](#aliases)
README.md: - [Aliases](#aliases)
README.md:*This can be aliased using the instructions found [here](https://github.com/tiimgreen/github-cheat-sheet#aliases).
README.md:$ git grep aliases
README.md:will show all the files containing the string *aliases*.
README.md:![git grep aliases](http://i.imgur.com/q3WkJgl.png?1)
README.md:Some useful aliases include:
README.zh-cn.md: - [Aliases](#aliases)
README.zh-tw.md: - [Aliases](#aliases)
Tims-MacBook-Pro:ghcs tim$
```

Press `q` to quit.

You can also use multiple flags for more advanced search. For example:

- `-e` The next parameter is the pattern (e.g. regex)
- `--and`, `--or` and `--not` Combine multiple patterns.

Use it like this:

```
$ git grep -e pattern --and -e anotherpattern
```

[Read more about the Git `grep` command.](#)

## Merged Branches

Running:

```
$ git branch --merged
```

Will give you a list of all branches that have been merged into your current branch.

Conversely:

```
$ git branch --no-merged
```

Will give you a list of branches that have not been merged into your current branch.

[Read more about the Git `branch` command.](#)

## Fixup and Autosquash

If there is something wrong with a previous commit (can be one or more from HEAD), for example `abcde`, run the following command after you've amended the problem:

```
$ git commit --fixup=abcde
$ git rebase abcde^ --autosquash -i
```

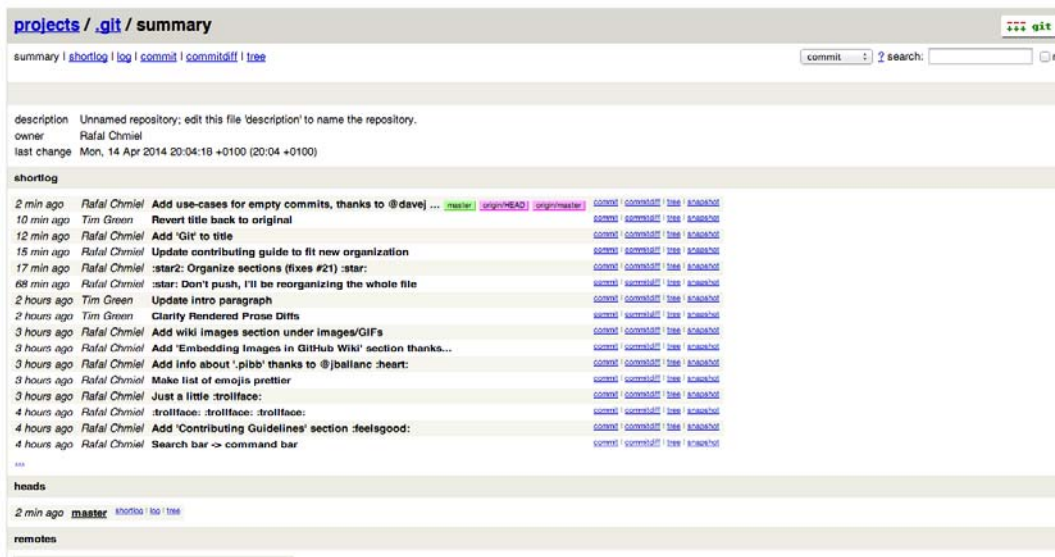
[Read more about the Git `commit` command.](#) [Read more about the Git `rebase` command.](#)

## Web Server for Browsing Local Repositories

Use the Git `instaweb` command to instantly browse your working repository in `gitweb`. This command is a simple script to set up `gitweb` and a web server for browsing the local repository.

```
$ git instaweb
```

Opens:



[Read more about the Git `instaweb` command.](#)

## Git Configurations

Your `.gitconfig` file contains all your Git configurations.

### Aliases

Aliases are helpers that let you define your own git calls. For example you could set `git a` to run `git add --all`.

To add an alias, either navigate to `~/.gitconfig` and fill it out in the following format:

```
[alias]
co = checkout
cm = commit
p = push
# Show verbose output about tags, branches or remotes
```

```
tags = tag -l
branches = branch -a
remotes = remote -v
```

...or type in the command-line:

```
$ git config --global alias.new_alias git_function
```

For example:

```
$ git config --global alias.cm commit
```

For an alias with multiple functions use quotes:

```
$ git config --global alias.ac 'add -A . && commit'
```

Some useful aliases include:

| Alias        | Command  | What to Type  |
|--------------|--|---|
| git cm       | git commit   | git config --global alias.cm commit   |
| git co       | git checkout   | git config --global alias.co checkout   |
| git ac       | git add . -A git commit  | git config --global alias.ac '!git add -A && git commit'  |
| git st       | git status -sb   | git config --global alias.st 'status -sb'   |
| git tags     | git tag -l   | git config --global alias.tags 'tag -l'   |
| git branches | git branch -a  | git config --global alias.branches 'branch -a'  |
| git cleanup  | git branch --merged   grep -v '*'<br>  xargs git branch -d   | git config --global alias.cleanup "!git branch --merged   grep -v '*'   xargs git branch -d"  |
| git remotes  | git remote -v  | git config --global alias.remotes 'remote -v'   |
| git lg       | git log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit -- | git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-commit --" |

Some Aliases are taken from [@mathiasbynens](https://github.com/mathiasbynens/dotfiles) dotfiles:

<https://github.com/mathiasbynens/dotfiles/blob/master/.gitconfig>

## Auto-Correct

If you type `git comit` you will get this:

```
$ git comit -m "Message"
# git: 'comit' is not a git command. See 'git --help'.
```

```
# Did you mean this?
#   commit
```

To call `commit` when `comit` is typed, just enable auto-correct:

```
$ git config --global help.autocorrect 1
```

So now you will get this:

```
$ git comit -m "Message"
# WARNING: You called a Git command named 'comit', which does not exist.
# Continuing under the assumption that you meant 'commit'
# in 0.1 seconds automatically...
```

## Color

To add more color to your Git output:

```
$ git config --global color.ui 1
```

*Read more about the Git `config` command.*

## Git Resources

| Title                                     | Link  |
|---|---|
| Official Git Site                         | <a href="http://git-scm.com/">http://git-scm.com/</a>   |
| Official Git Video Tutorials              | <a href="http://git-scm.com/videos">http://git-scm.com/videos</a>   |
| Code School Try Git                       | <a href="http://try.github.com/">http://try.github.com/</a>   |
| Introductory Reference & Tutorial for Git | <a href="http://gitref.org/">http://gitref.org/</a>   |
| Official Git Tutorial                     | <a href="http://git-scm.com/docs/gittutorial">http://git-scm.com/docs/gittutorial</a>   |
| Everyday Git                              | <a href="http://git-scm.com/docs/everyday">http://git-scm.com/docs/everyday</a>   |
| Git Immersion                             | <a href="http://gitimmersion.com/">http://gitimmersion.com/</a>   |
| Ry's Git Tutorial                         | <a href="http://rypress.com/tutorials/git/index.html">http://rypress.com/tutorials/git/index.html</a>                         |
| Git for Designers                         | <a href="http://hoth.entp.com/output/git_for_designers.html">http://hoth.entp.com/output/git_for_designers.html</a>           |
| Git for Computer Scientists               | <a href="http://eagain.net/articles/git-for-computer-scientists/">http://eagain.net/articles/git-for-computer-scientists/</a> |
| Git Magic                                 | <a href="http://www-cs-students.stanford.edu/~blynn/gitmagic/">http://www-cs-students.stanford.edu/~blynn/gitmagic/</a>       |
| GitHub Training Kit                       | <a href="http://training.github.com/kit">http://training.github.com/kit</a>   |
| Git Visualization Playground              | <a href="http://onlywei.github.io/explain-git-with-d3/#freeplay">http://onlywei.github.io/explain-git-with-d3/#freeplay</a>   |

## Git Books

| Title                               | Link  |
|-------------------------------------|---|
| Pragmatic Version Control Using Git | <a href="http://www.pragprog.com/titles/tsgit/pragmatic-version-control-using-git">http://www.pragprog.com/titles/tsgit/pragmatic-version-control-using-git</a> |

|                                   |   |
|-----------------------------------|---|
| Pro Git                           | <a href="http://git-scm.com/book">http://git-scm.com/book</a>   |
| Git Internals PluralSight         | <a href="https://github.com/pluralsight/git-internals-pdf">https://github.com/pluralsight/git-internals-pdf</a>   |
| Git in the Trenches               | <a href="http://cbx33.github.com/gitt/">http://cbx33.github.com/gitt/</a>   |
| Version Control with Git          | <a href="http://www.amazon.com/Version-Control-Git-collaborative-development/dp/1449316387">http://www.amazon.com/Version-Control-Git-collaborative-development/dp/1449316387</a> |
| Pragmatic Guide to Git            | <a href="http://www.pragprog.com/titles/pg_git/pragmatic-guide-to-git">http://www.pragprog.com/titles/pg_git/pragmatic-guide-to-git</a>   |
| Git: Version Control for Everyone | <a href="http://www.packtpub.com/git-version-control-for-everyone/book">http://www.packtpub.com/git-version-control-for-everyone/book</a>   |

