

 This repository Search

Pull requestsIssuesGist

 +

tiimgreen / github-cheat-sheet

Watch932Star18,380Fork2,111

CodeIssues 3Pull requests 2PulseGraphs

Branch: mastergithub-cheat-sheet / README.zh-cn.mdFind fileCopy path

tiimgreen Remove erroring links75b034d on 1 Mar

14 contributors

929 lines (649 sloc)34.1 KBRawBlameHistory

# GitHub秘籍

Git 和 Github 秘籍，灵感来自于 Zach Holman 在 2012 年 Aloha Ruby Conference 和 2013 年 WDCNZ 上所做的演讲：[Git and GitHub Secrets\(slides\)](#) 和 [More Git and GitHub Secrets\(slides\)](#)。

其他语言版本: [English](#), [한국어](#), [日本語](#), [简体中文](#), [正體中文](#)。

## 目录

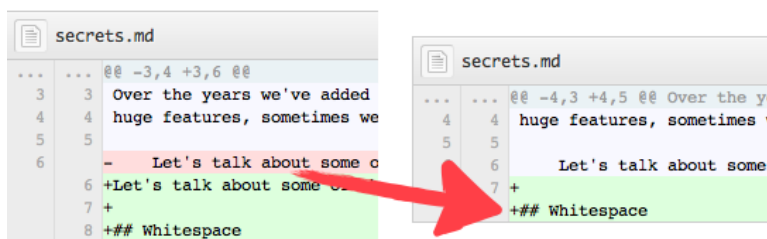
- GitHub
  - 不比较空白字符
  - 调整 Tab 字符所代表的空格数
  - 查看某个用户的 Commit 历史
  - 仓库克隆
  - 分支
    - 将某个分支与其他所有分支进行对比
    - 比较分支
    - 比较不同派生库的分支
  - Gists
  - Git.io
  - 键盘快捷键
  - 整行高亮
  - 用 Commit 信息关闭 Issue
  - 链接其他仓库的 Issue
  - 锁定项目对话功能
  - 设置 CI 对每条 Pull Request 都进行构建
  - Markdown 文件语法高亮
  - 表情符
  - 图片 / GIF 动画
    - 在 GitHub Wiki 中引用图片
  - 快速引用
  - 粘贴剪贴板中的图片到评论
  - 快速添加许可证文件
  - 任务列表
    - Markdown 文件中的任务列表
  - 相对链接
  - GitHub Pages 的元数据与插件支持
  - 查看 YAML 格式的元数据
  - 渲染表格数据
  - 撤销 Pull Request
  - Diffs
    - 可渲染文档的Diffs

- 可比较的地图数据
- 在 Diff 中展开查看更多的上下文
- 获取 Pull Request 的 diff 或 patch 文件
- 显示图片以及比较图片
- Hub
- 贡献者指南
- Octicons
- GitHub 资源
  - GitHub 相关演讲视频
- Git
  - 从工作区去除大量已删除文件
  - 上一个分支
  - 去除空白
  - 检出 Pull Requests
  - 没有任何改动的提交
  - 美化 Git Status
  - 美化 Git Log
  - Git 查询
  - 合并分支
  - 修复有问题的提交以及自动合并
  - 以网站方式查看本地仓库
  - Git 配置
    - Git 命令自定义别名
    - 自动更正
    - 颜色输出
  - Git 资源
    - Git 参考书籍

## GitHub

### 不比较空白字符

在任意 diff 页面的 URL 后加上 `?w=1`，可以去掉那些只是空白字符的改动，使你能更专注于代码改动。



详见 *GitHub secrets*.

### 调整 Tab 字符所代表的空格数

在 diff 或文件的 URL 后面加上 `?ts=4`，这样当显示 tab 字符的长度时就会是 4 个空格的长度，不再是默认的 8 个空格。ts 后面的数字还可以根据你个人的偏好进行修改。这个技巧不适用于 Gists，或者以 Raw 格式查看文件，但有浏览器扩展插件可以帮你自动调整: [Chrome 扩展](#)，[Opera 扩展](#)。

下面以一个 Go 语言源文件为例，看看在 URL 里添加 `?ts=4` 参数的效果。添加前：

file69 lines (57 sloc)1.86 kb

OpenEditRawBlameHistoryDelete

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

... 添加后的样子：

file69 lines (57 sloc)1.86 kb

OpenEditRawBlameHistoryDelete

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


## 查看用户的全部 Commit 历史

在 Commits 页面 URL 后加上 `?author={user}` 查看用户全部的提交。


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

branch: master


Commits on Jan 13, 2015

 Merge pull request #18476 from Alamo/scaffold\_index\_view\_grammar ...  
dhh authored 5 days ago


78a4884<>

 Stop promoting rack-cache usage at the moment (not so common or impor... ...  
dhh authored 5 days ago

1302edf<>


 Show how to change the queuing backend for ActiveJob in production  
dhh authored 5 days ago

6463495<>

 Set all asset options together  
dhh authored 5 days ago

b9b28d8<>

Commits on Jan 9, 2015

 Merge pull request #18413 from brainopia/automatic\_inverse\_of\_for\_bel... ...  
dhh authored 9 days ago

6eb499f<>

深入了解提交视图之间的区别

## 仓库克隆

当克隆仓库时可以不要那个 `.git` 后缀。

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

更多对 `Git clone` 命令的介绍.

## 分支

将某个分支与其他所有分支进行对比

当你查看某个仓库的分支（Branches）页面（紧挨着 Commits 链接）时

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

你会看到一个包含所有未合并的分支的列表。

在这里你可以访问分支比较页面或删除某个分支。

rails / rails

Watch

1,732

Unstar

24,618

Fork

9,521

OverviewActiveStaleAll branches

Search branches...

Default branch

master

Updated an hour ago by fixn

✓

Default

Active branches

4-2-stable

Updated 2 hours ago by sgrif

✗

1147269

Compare

4-1-stable

Updated 4 days ago by rafaelfranca

✓

6892672

Compare

fix\_nested\_transactions\_for\_re...

Updated 8 days ago by chancancode

✗

1493

Compare

3-2-stable

Updated 12 days ago by rafaelfranca

✗

224781757

Compare

4-0-stable

Updated 12 days ago by rafaelfranca

✓

119741081

Compare

View more active branches

比较分支

如果要在 GitHub 上直接比较两个分支，可以使用如下形式的 URL：

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

其中 {range} = master...4-1-stable

例如：

```
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.

Commits672

Files changed509

Commit comments5

45 contributors

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

Commits on Jun 19, 2014

rafaelfranca

Merge pull request #15813 from DNNX/valid-action-name-refactoring ...

✓6413eb4

rafaelfranca

Fix has\_and\_belongs\_to\_many in a namespaced model pointing to a non n... ...

✓0769465

rafaelfranca

Fix has\_and\_belongs\_to\_many in a namespaced model pointing to a non n... ...

✓1a12bee

rafaelfranca

Merge pull request #15772 from nbudin/sti\_through\_bug ...

✓8f76511

rafaelfranca

Merge pull request #15772 from nbudin/sti\_through\_bug ...

✓a44feed

rafaelfranca

Merge pull request #15729 from sgrif/sg-double-save-hm-t-4-1-stable ...

✓ae0d952

{range} 参数还可以使用下面的形式：

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

日期格式 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

emre-basala

Add tests to ActiveSupport:XmlMini to\_tag method

05d7cde

Nov 17, 2013

iantropov

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

ec09280

Jan 07, 2014

roderickvd

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

9330631

在 diff 和 patch 页面里也可以比较分支：

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

了解更多关于基于时间的 Commit 比较.

### 比较不同派生库的分支

想要对派生仓库（Forked Repository）之间的分支进行比较，可以使用如下的 URL：

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

例如：

<https://github.com/rails/rails/compare/byroot:master...master>

byroot:master ... rails:master

Edit

Please review the [guidelines for contributing](#) to this repository.

Create Pull Request

Open a Pull Request for this comparison to discuss and review your changes with others.

?

6,802 commits

309 files changed

14 comments

64 contributors

Commits

Files changed

Commit comments

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

Apr 02, 2014

kastiglione

PostgreSQL, Support for materialized views. [Dave Lee & Yves Senn] ...

def6071

rajcybage

We can conditional define the tests depending on the adapter or ...

ee36af1

rafaelfranca

Merge pull request #14565 from rajcybage/conditional\_test\_cases ...

c82483a

rwz

DRY AS::SafeBuffer a bit using existing helper

8482895

alex88

Fixed small documentation typo ...

8ae3f24

rafaelfranca

Merge pull request #14568 from alex88/patch-1 ...

3bcc51a

## Gists

Gists 方便我们管理代码片段，不必使用功能齐全的仓库。

GitHub Gist

Search...

Discover Gists

rafaelchmiel

PUBLIC

tiimgreen

/ app.rb

Created 4 days ago

Star 0

Fork 0

A simple Ruby program.

Gist Detail

Revisions 1

app.rb

Ruby

1 puts 'Hello World'

Download Gist

Clone this gist

https://gist.github.com

Embed this gist

<script src="https:

Link to this gist

https://gist.github.com

rafaelchmiel commented

2 days ago

Wow, dat is some engineering.

Write

Preview

Comments are parsed with [GitHub Flavored Markdown](#)

Leave a comment

Add Comment

Gist 的 URL 后加上 .pibb （像这样）可以得到便于嵌入到其他网站 的 HTML 代码。

Gists 可以像任何标准仓库一样被克隆。

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



```
tim — bash — 80x17
Last login: Sun Jan 18 22:05:37 on ttys002
Tims-MacBook-Pro:~ tim$ git clone https://gist.github.com/tiimgreen/10545817
Cloning into '10545817'...
remote: Counting objects: 3, done.
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
Checking connectivity... done.
Tims-MacBook-Pro:~ tim$
```

这意味着你可以像 Github 仓库一样修改和更新 Gists：

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

但是，Gists 不支持目录。所有文件都必须添加在仓库的根目录下。 [进一步了解如何创建 Gists](#)。

## Git.io

Git.io 是 Github 的短网址服务。



你可以通过 Curl 命令以普通 HTTP 协议使用它：

```
$ 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/...
```

[进一步了解 Git.io](#)。

## 键盘快捷键

在仓库页面上提供了快捷键方便快速导航。

- 按 `t` 键打开一个文件浏览器。
- 按 `w` 键打开分支选择菜单。
- 按 `s` 键聚焦光标到当前仓库的搜索框。此时按退格键就会从搜索当前仓库切换到搜索整个 Github 网站。

- 按 `l` 键编辑 Issue 列表页的标签。
- 查看文件内容时（如：<https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.md>），按 `y` 键将会冻结这个页面，这样就算代码被修改了也不会影响你当前看到的。

按 `?` 查看当前页面支持的快捷键列表：

Keyboard shortcuts

Site wide shortcuts

`s`

Focus command bar, search all repositories

`/`

Focus command bar, search the current repository

`gn`

Go to Notifications

`?`

Bring up this help dialog

`j`

Move selection down

`k`

Move selection up

`x`

Toggle selection

Issues

`% enter`

Submit comment

`% shift p`

Preview comment

`% shift l`

Go fullscreen

`c`

Create issue

`l`

Create label

`u`

Back to issues

`/`

Focus issues search

`r`

Reply (quoting selected text)

Pull request list

`r`

Reply (quoting selected text)

`o / enter`

Open issue

`% enter`

Submit comment

`% shift p`

Preview comment

`% shift l`

Go fullscreen

Network Graph

`← / h`

Scroll left

进一步了解可用的搜索语法。

## 整行高亮

在代码文件地址 URL 后加上 `#L52` 或者单击行号 52 都会将第 52 行代码高亮显示。

多行高亮也可以，比如用 `#L53-L60` 选择范围，或者按住 `shift` 键，然后再点击选择的两行。

[https://github.com/rails/rails/blob/master/activemodel/lib/active\\_model.rb#L53-L60](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
```

## 用 Commit 信息关闭 Issue

如果某个提交修复了一个 Issue，当提交到 `master` 分支时，提交信息里可以使用 `fix/fixes/fixed`，`close/closes/closed` 或者 `resolve/resolves/resolved` 等关键词，后面再跟上 Issue 号，这样就会关闭这个 Issue。

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

这将会关闭 Issue #12，并且在 Issue 讨论列表里关联引用这次提交。



tiimgreen closed this issue from a commit 9 days ago

Fix screwup, fixes #12

36f4317

tiimgreen closed this in 36f4317 9 days ago

进一步了解通过提交信息关闭 Issue.

## 链接其他仓库的 Issue

如果你想引用到同一个仓库中的一个 Issue，只需使用井号 # 加上 Issue 号，这样就会自动创建到此 Issue 的连接。

要链接到其他仓库的 Issue，就使用 {user}/{repo}#ISSUE\_NUMBER 的方式，例如 tiimgreen/toc#12。

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

sr referenced this issue in github/github

October 14, 2011

Pull Request #1773:Update footer for new formats

Closed

## 锁定项目对话功能

现在仓库的管理员和合作者可以将 Pull Requests 和 Issue 的评论功能关闭。

**Parsed as Markdown**

**Edit in fullscreen**

**Lock issue**

这样，不是项目合作者的用户就不能在这个项目上使用评论功能。

Write

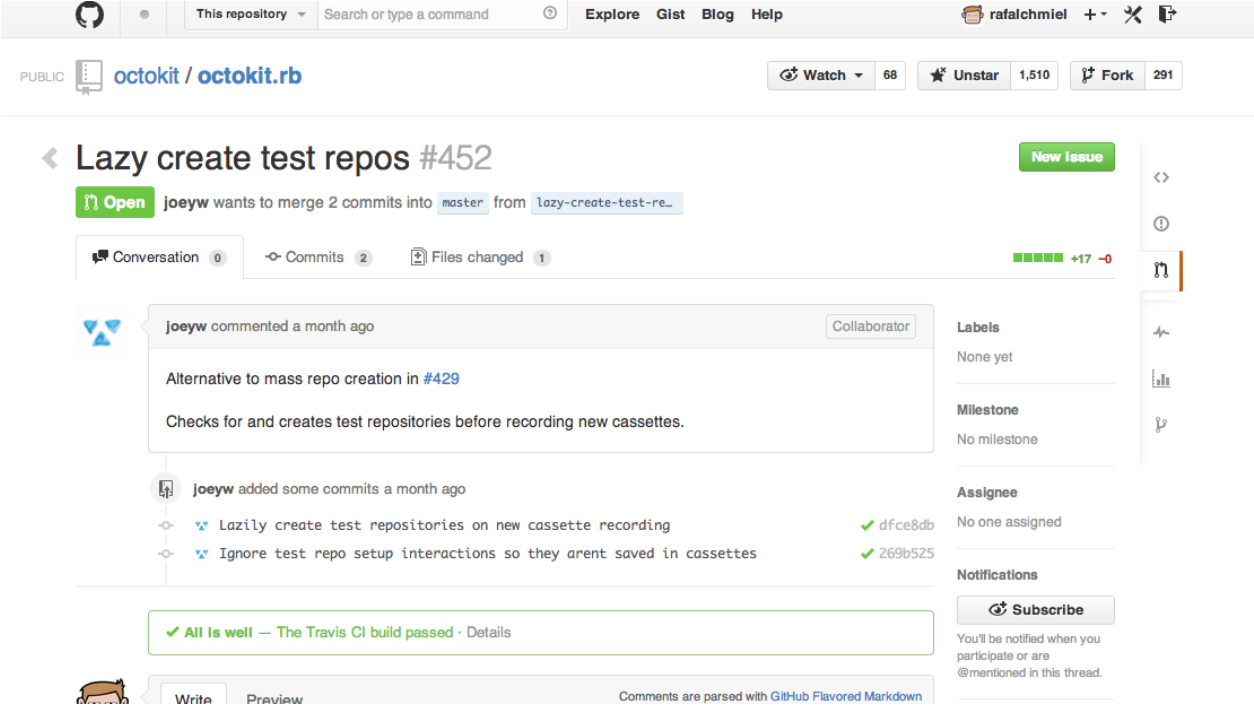
Preview

This conversation has been locked and limited to collaborators.

进一步了解对话锁定功能.

## 设置 CI 对每条 Pull Request 都进行构建

如果配置正确，Travis CI 会为每个你收到的 Pull Request 执行构建，就像每次提交也会触发构建一样。想了解更多关于 Travis CI 的信息，请参考 [Travis CI入门](#)。



进一步了解提交状态 API.

## Markdown 文件语法高亮

例如，可以像下面这样在你的 Markdown 文件里为 Ruby 代码添加语法高亮：

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

效果如下：

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

Github使用 [Linguist](#) 做语言识别和语法高亮。你可以仔细阅读 [languages YAML file](#)，了解有哪些可用的关键字。

进一步了解 [GitHub Flavored Markdown](#).

## 表情符

可以在 Pull Requests, Issues, 提交消息, Markdown 文件里加入表情符。使用方法 `:name_of_emoji:`

```
:smile:
```

将输出一个笑脸：



Github 支持的完整表情符号列表详见[emoji-cheat-sheet.com](#) 或 [scotch-io/All-Github-Emoji-Icons](#)。

Github 上使用最多的5个表情符号是：

1. `:shipit:`
2. `:sparkles:`
3. `:-1:`

4. `:+1:`
5. `:clap:`

## 图片 / GIF 动画

注释和README等文件里也可以使用图片和 GIF 动画：

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

仓库中的图片可以被直接引用：

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



所有图片都缓存在 Github，不用担心你的站点不能访问时就看不到图片了。

### 在 GitHub Wiki 中引用图片


有多种方法可以在 Wiki 页面里嵌入图片。既可以像上一条里那样使用标准的 Markdown 语法，也可以像下面这样指定图片的高度或宽度：

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

结果：

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

## Home (Preview)

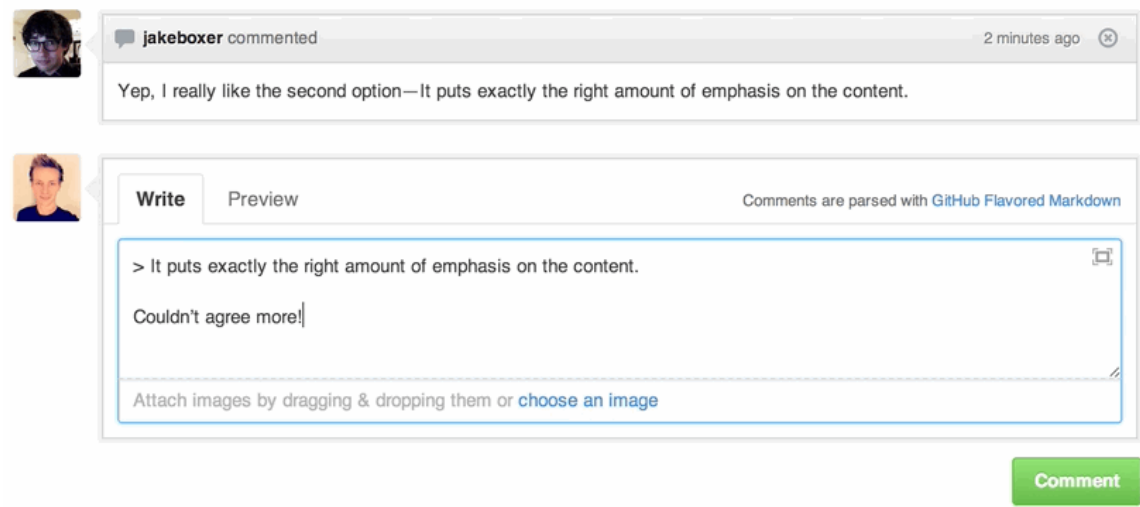


### README not found

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](#).

## 快速引用

在主题评论中引用之前某个人所说的，只需选中文本，然后按 `r` 键，想要的就会以引用的形式复制到你的输入框里。

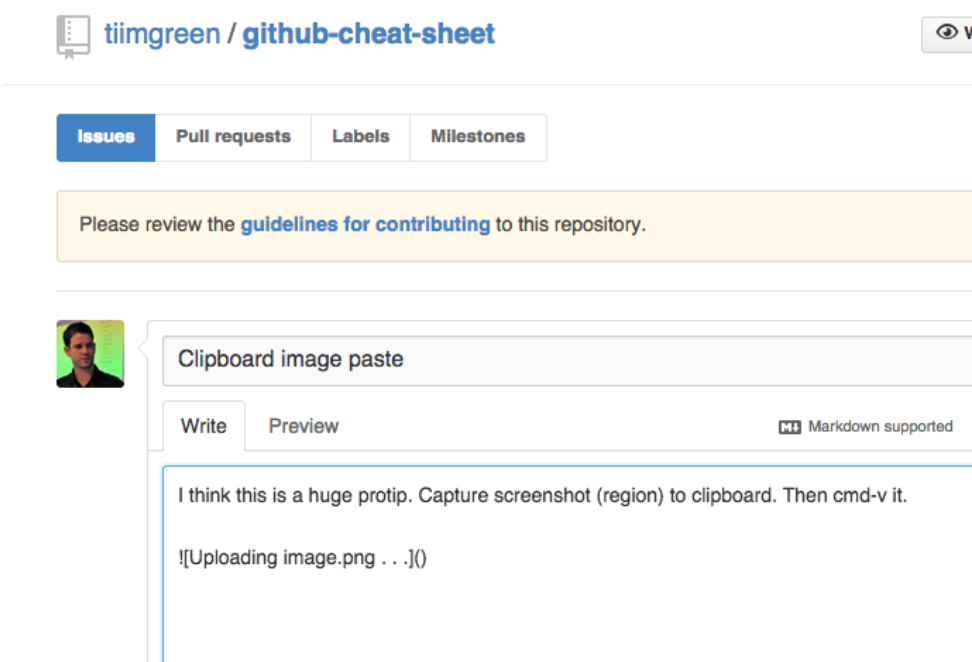


进一步了解快速引用.

## 粘贴剪贴板中的图片到评论

(仅适用于 Chrome 浏览器)

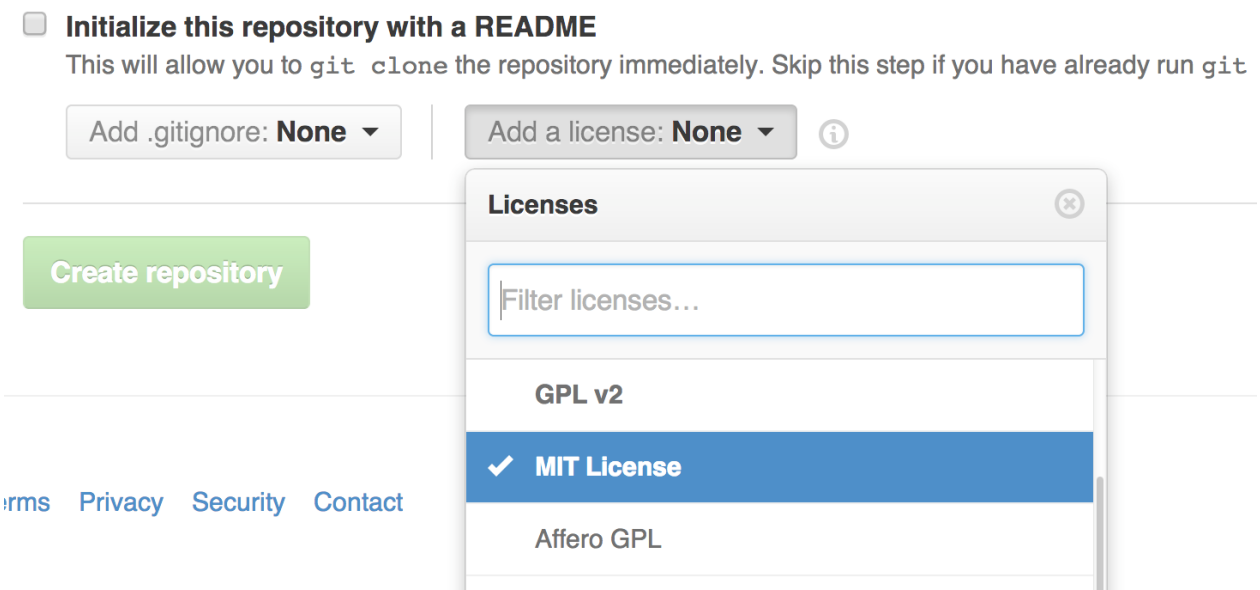
当截屏图片复制到剪贴板后（mac 上用 `cmd-ctrl-shift-4`），你可以用(`cmd-v` / `ctrl-v`)把图片粘贴到评论框里，然后它就会自动上传到 Github。



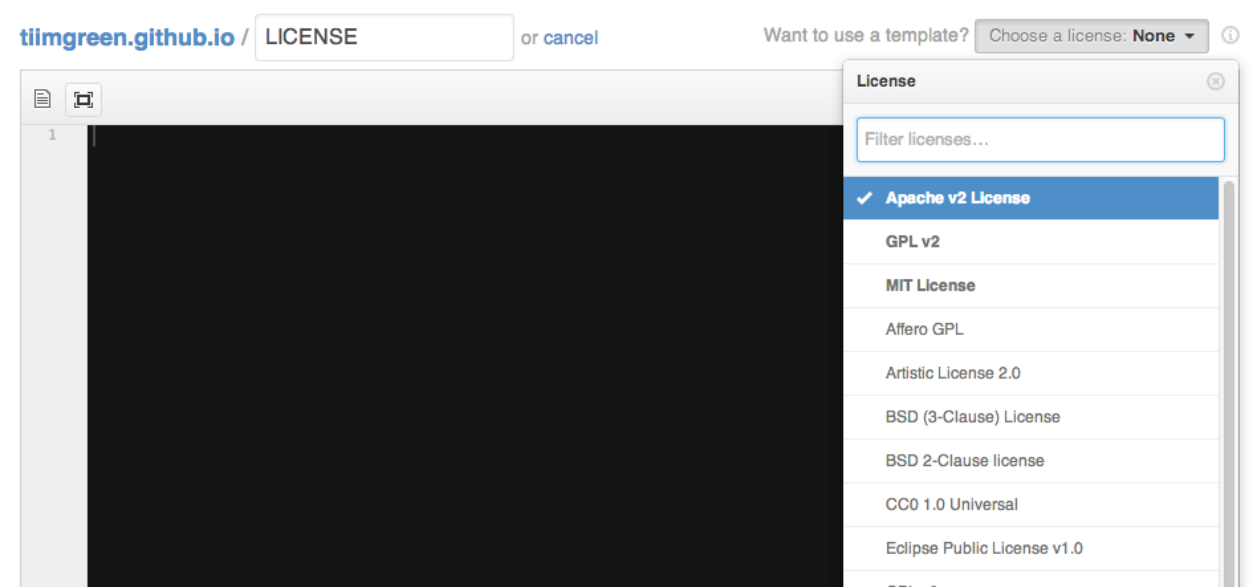
进一步了解在 issue 中使用附件

## 快速添加许可证文件

创建一个仓库时，Github会为你提供一个预置的软件许可列表：



对于已有的仓库，可以通过 web 界面创建文件来添加软件许可。输入 LICENSE 作为文件名后，同样可以从预置的列表中选择一个作为模板。



这个技巧也适用于 .gitignore 文件。


[进一步了解开源许可证](#)

## 任务列表

Issues 和 Pull requests 里可以添加复选框，语法如下（注意空白符）：

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

## < TODO #1

 Open

 AlexandreArpin opened this issue just now · 0 comments

AlexandreArpin commented just now

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

当项目被选中时，它对应的 Markdown 源码也被更新了：

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

进一步了解任务列表.

### Markdown 文件中的任务列表

在完全适配Markdown语法的文件中可以使用以下语法加入一个只读的任务列表

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

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

进一步了解 *Markdown* 文件中的任务列表

### 相对链接

Markdown文件里链接到内部内容时推荐使用相对链接。

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

绝对链接会在 URL 改变时（例如重命名仓库、用户名改变，建立分支项目）被更新。使用相对链接能够保证你的文档不受此影响。

进一步了解*相对链接*.

### GitHub Pages 的元数据与插件支持

在 Jekyll 页面和文章里，仓库信息可在 `site.github` 命名空间下找到，也可以显示出来，例如，使用 `{{ site.github.project_title }}` 显示项目标题。

Jemoji 和 jekyll-mentions 插件为你的 Jekyll 文章和页面增加了emoji和@mentions功能。

了解更多 *GitHub Pages* 的元数据和插件支持。

## 查看 YAML 格式的元数据

许多博客站点，比如基于 Jekyll的GitHub Pages，都依赖于一些文章头部的 YAML 格式的元数据。Github 会将其渲染成一个水平表格，方便阅读。

file | 61 lines (39 sloc) | 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)

进一步了解 在文档里查看 YAML 元数据。

## 渲染表格数据

GitHub 支持将 `.csv` (逗号分隔)和 `.tsv` (制表符分隔)格式的文件渲染成表格数据。

file | 869 lines (868 sloc) | 188.222 kb

EditRawBlameHistoryDelete

Search this file...

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

进一步了解渲染表格数据。

## 撤销 Pull Request

合并一个 Pull Request 之后，你可能会反悔：要么是这次 Pull Request 没什么用处，要么是还不到合并的时候。

此时可以通过 Pull Request 中的 Revert 按钮来撤销一个已合并的 Pull Request 中的 commit。按下按钮后将自动生成一个进

行逆操作的 Pull Request。

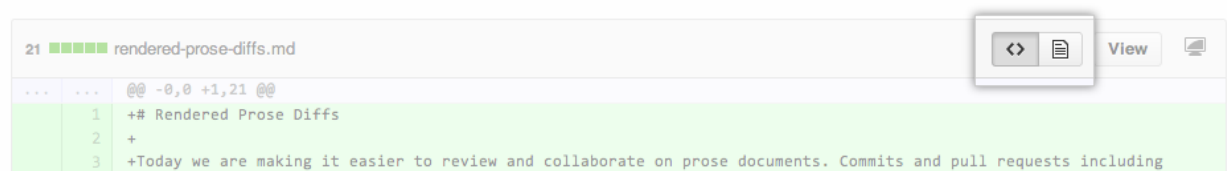


\*进一步了解“撤销”按钮

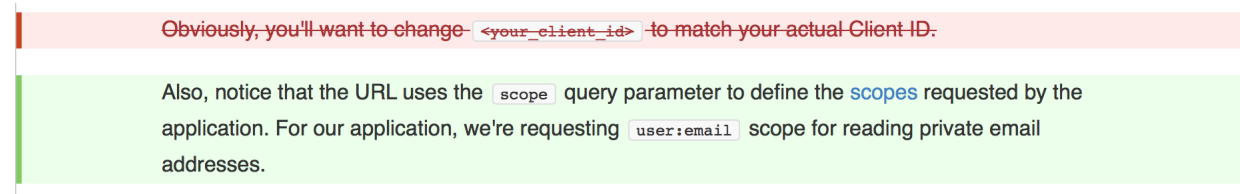
Diff

可渲染文档的Diffs

Commit 和 Pull Request 里包含有 Github 支持的可渲染文档（比如 Markdown）会提供source 和 rendered 两个视图功能。



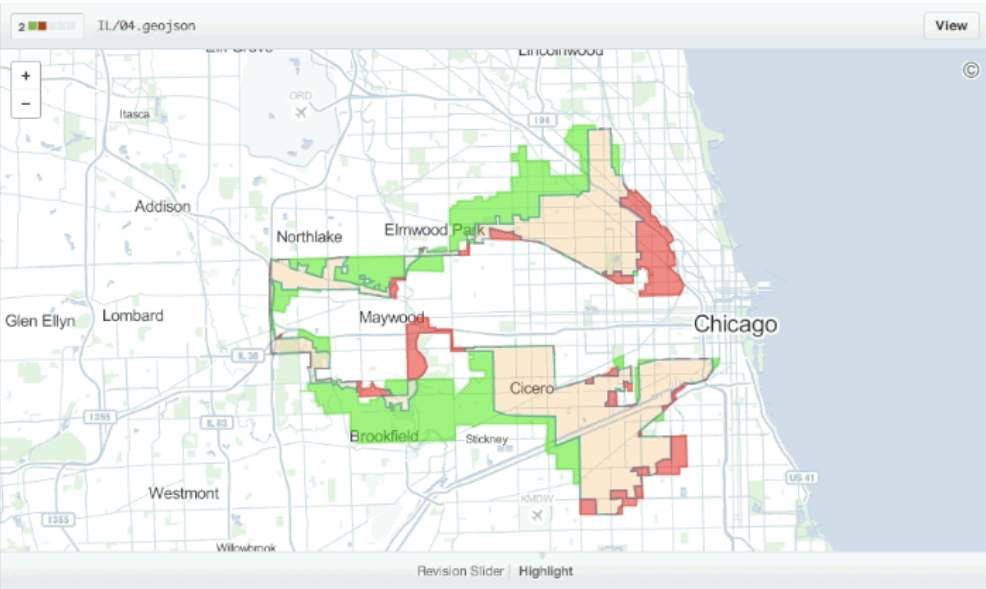
点击 "rendered" 按钮，看看改动在渲染后的显示效果。当你添加、删除或修改文本时，渲染纯文本视图非常方便。



进一步了解渲染纯文本视图Diffs.

可比较的地图数据

当你在GitHub上查看一个包含地理数据的 commit 或 pull request时，Github 将以可视化的方式对比版本之间的差异。



进一步了解可比较的地图数据.



### 在 Diff 中展开查看更多的上下文

你可以通过点击 diff 边栏里的 *unfold* 按钮来多显示几行上下文。你可以一直点击 *unfold* 按钮直到显示了文件的全部内容。这个功能在所有 GitHub 的 diff 功能中都可以使用。

|     |     |  |
|-----|-----|--|
|     | 94  | + (RACSignal *)enqueueRequest:(NSURLRequest *)request fetchAllPages:(BOOL)fetchAllPages;                         |
|     | 95  | +  |
| 82  | 96  | // Enqueues a request to fetch information about the current user by accessing                                   |
| 83  | 97  | // a path relative to the user object.   |
| 84  | 98  | //   |
|     | 99  | @@ -241,11 +255,13 @@ - (id)initWithServer:(OCTServer *)server {   |
| 241 | 255 | NSString *userAgent = self.class.userAgent;  |
| 242 | 256 | if (userAgent != nil) [self setDefaultHeader:@"User-Agent" value:userAgent];                                     |
| 243 | 257 |  |
| 244 |     | - self.parameterEncoding = AFJSONParameterEncoding;  |
| 245 |     | - [self setDefaultHeader:@"Accept" value:@"application/vnd.github.beta+json"];                                   |
| 246 |     | -  |
| 247 | 258 | [AFHTTPRequestOperation addAcceptableStatusCodes:[NSIndexSet indexSetWithIndex:OCTClientNotModifiedStatusCode]]  |
| 248 |     | - [AFJSONRequestOperation addAcceptableContentTypes:[NSSet initWithObject:@"application/vnd.github.beta+json"]]; |
|     | 259 | +  |
|     | 260 | + NSString *contentType = [NSString stringWithFormat:@"application/vnd.github.%%+json", OCTClientAPIVersion];    |
|     | 261 | + [self setDefaultHeader:@"Accept" value:contentType];   |
|     | 262 | + [AFJSONRequestOperation addAcceptableContentTypes:[NSSet initWithObject:contentType]];                         |
|     | 263 | +  |
|     | 264 | + self.parameterEncoding = AFJSONParameterEncoding;  |
| 249 | 265 | [self registerHTTPOperationClass:AFJSONRequestOperation.class];  |
| 250 | 266 |  |
| 251 | 267 | return self;   |

进一步了解展开 Diff 上下文.

### 获取 Pull Request 的 diff 或 patch 文件

在 Pull Request 的 URL 后面加上 .diff 或 .patch 的扩展名就可以得到它的 diff 或 patch 文件，例如：

```
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
```

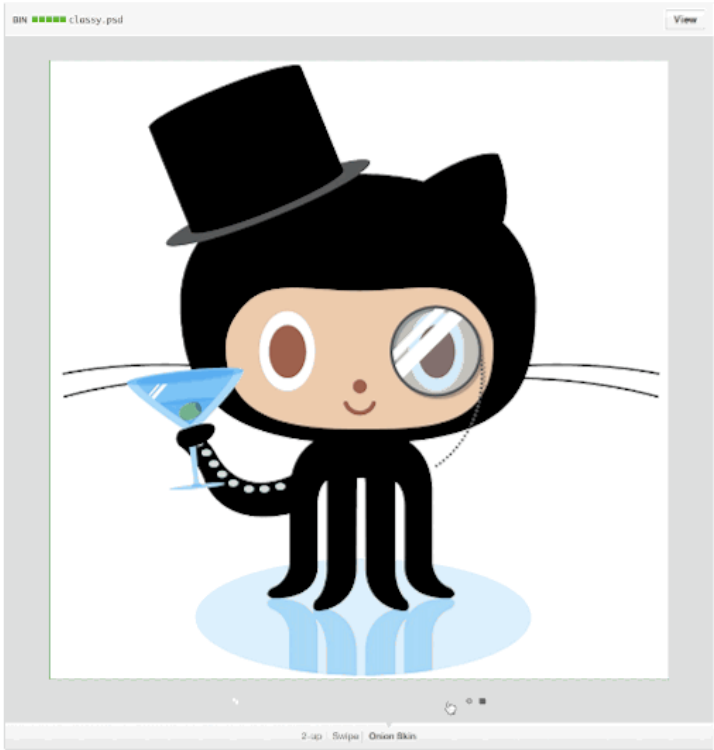
.diff 扩展会使用普通文本格式显示如下内容：

```
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 was i
- [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

(...)
```

### 显示图片以及比较图片

GitHub 可以显示包括 PNG、JPG、GIF、PSD 在内的多种图片格式并提供了几种方式来比较这些格式的图片文件版本间的不同。



查看更多关于图片显示和比较

## Hub

Hub 是一个对 Git 进行了封装的命令行工具，可以帮助你更方便的使用 Github。

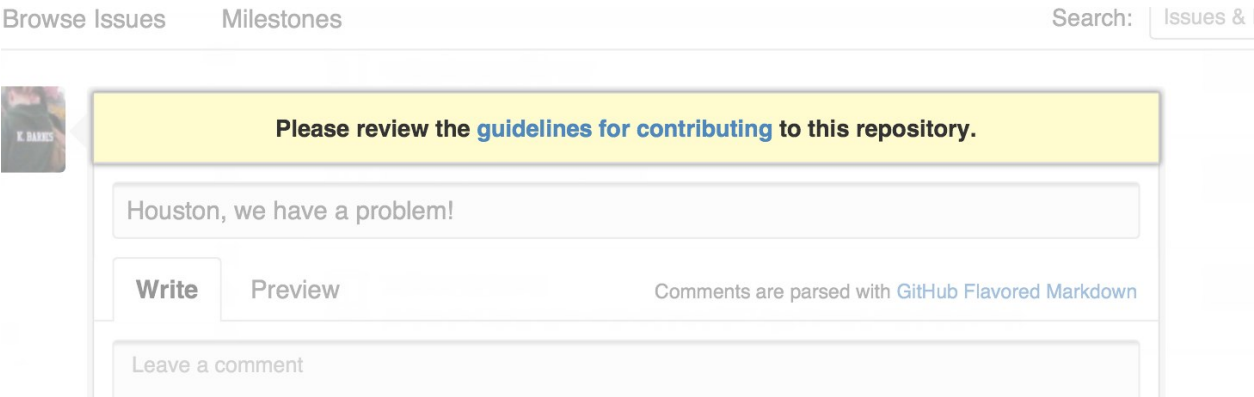
例如可以像下面这样进行克隆：

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

查看更多 Hub 提供的超酷命令.

## 贡献者指南

在仓库的根目录添加一个名为 CONTRIBUTING 的文件后，贡献者在新建 Issue 或 Pull Request 时会看到一个指向这个文件的链接。



进一步了解贡献者指南.

## Octicons

GitHubs 图标库 (Octicons) 现已开源。



进一步了解 *GitHub* 图标库

### GitHub 资源

| 内容         | 链接  |
|------------|---|
| 探索 GitHub  | <a href="https://github.com/explore">https://github.com/explore</a>       |
| GitHub 博客  | <a href="https://github.com/blog">https://github.com/blog</a>             |
| GitHub 帮助  | <a href="https://help.github.com/">https://help.github.com/</a>           |
| GitHub 培训  | <a href="http://training.github.com/">http://training.github.com/</a>     |
| GitHub 开发者 | <a href="https://developer.github.com/">https://developer.github.com/</a> |

### GitHub 相关演讲视频

| 内容  | 链接  |
|---|---|
| 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=Foz9yvMkvlA">https://www.youtube.com/watch?v=Foz9yvMkvlA</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

### 从工作区去除大量已删除文件

当用 `/bin/rm` 命令删除了大量文件之后，你可以用下面一条命令从工作区和索引中去除这些文件，以免一个一个的删除：

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

例如：

```
$ 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
```

## 上一个分支

快速检出上一个分支：

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

$ git checkout -
# Switched to branch 'next'

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

[进一步了解 Git 分支.](#)

## 去除空白

Git Stripspace 命令可以：

- 去掉行尾空白符
- 多个空行压缩成一行
- 必要时在文件末尾增加一个空行

使用此命令时必须传入一个文件，像这样：

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

[进一步了解 Git stripspace 命令.](#)

## 检出 Pull Requests

对 Github 仓库来说，Pull Request 是种特殊分支， 可以通过以下多种方式取到本地：

取出某个特定的 Pull Request 并临时作为本地的 FETCH\_HEAD 中以便进行快速查看更改( diff )以及合并( merge )：

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

通过 refspec 获取所有的 Pull Request 为本地分支：

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

或在仓库的 .git/config 中加入下列设置来自动获取远程仓库中的 Pull Request

```
[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/*
```

对基于派生库的 Pull Request，可以通过先 `checkout` 代表此 Pull Request 的远端分支再由此分支建立一个本地分支：

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

操作多个仓库的时候，可以在 Git 中设置获取 Pull Request 的全局选项。

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

此时可以在任意仓库中使用以下命令：

```
git fetch origin
```

```
git checkout pr/42
```

*进一步了解如何本地检出 pull request.*

## 没有任何改动的提交

可以使用 `--allow-empty` 选项强制创建一个没有任何改动的提交：

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

这样做在如下几种情况下是有意义的：

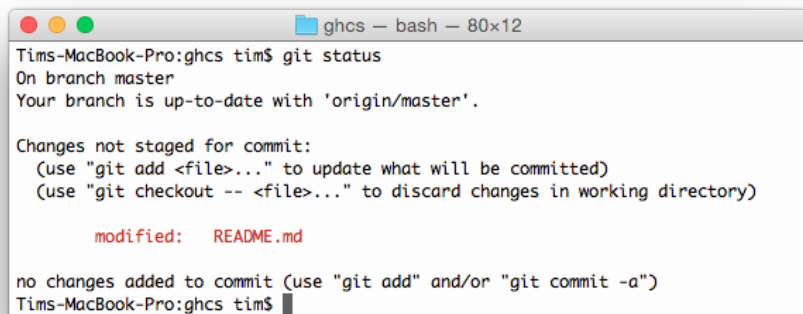
- 标记新的工作或一个新功能的开始。
- 记录对项目的跟代码无关的改动。
- 跟使用你仓库的其他人交流。
- 作为仓库的第一次提交，因为第一次提交后不能被 `rebase`：`git commit -m "init repo" --allow-empty` .

## 美化 Git Status

在命令行输入如下命令：

```
$ git status
```

可以看到：

A screenshot of a macOS terminal window titled 'ghcs - bash - 80x12'. The terminal shows the output of the 'git status' command. It indicates the user is on the 'master' branch and that the branch is up-to-date with 'origin/master'. It then shows that there are changes not staged for commit, specifically 'modified: README.md'. It provides instructions on how to stage changes with 'git add' or discard them with 'git checkout --'. Finally, it shows that no changes were added to the commit and prompts the user to use 'git add' and/or 'git commit -a' to create a commit. The prompt 'Tims-MacBook-Pro:ghcs tim\$' is visible at the bottom.

```
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)

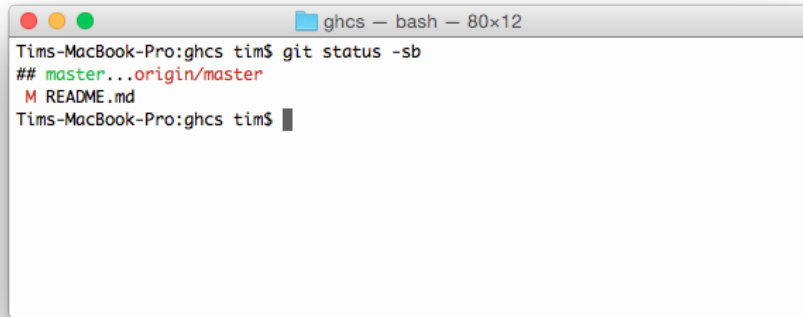
        modified:   README.md

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

加上 `-sb` 选项：

```
$ git status -sb
```

这会得到:



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

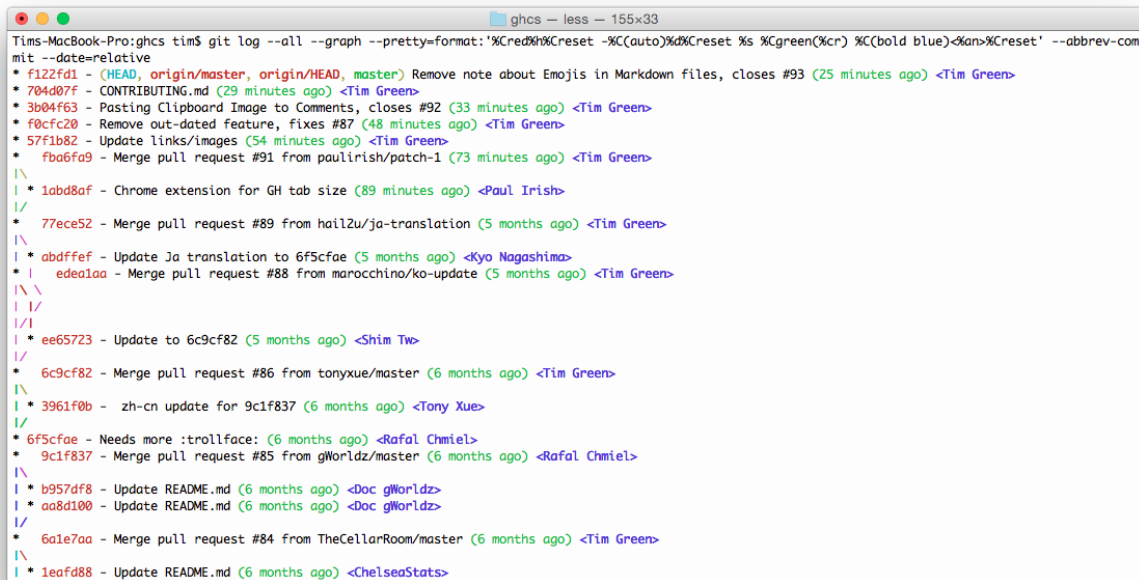
进一步了解 `Git status` 命令.

## 美化 Git Log

输入如下命令:

```
$ git log --all --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset'
```

可以看到:



```
Tims-MacBook-Pro:ghcs tim$ git log --all --graph --pretty=format:'%Cred%h%Creset -%C(auto)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%Creset' --abbrev-commit --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>
* 3b04f63 - Pasting Clipboard Image to Comments, closes #92 (33 minutes ago) <Tim Green>
* f0cfc20 - 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>
|/
|
| * abdfef - Update Ja translation to 6f5cfae (5 months ago) <Kyo Nagashima>
| * edea1aa - Merge pull request #88 from marocchina/ko-update (5 months ago) <Tim Green>
|/
|
| * ee65723 - Update to 6c9cf82 (5 months ago) <Shim Tw>
|/
|
| * 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>
```

这要归功于 [Palesz](#) 在 [stackoverflow](#) 的回答。

这个命令可以被用作别名, 详细做法见[这里](#)。

进一步了解 `Git Log` 命令.

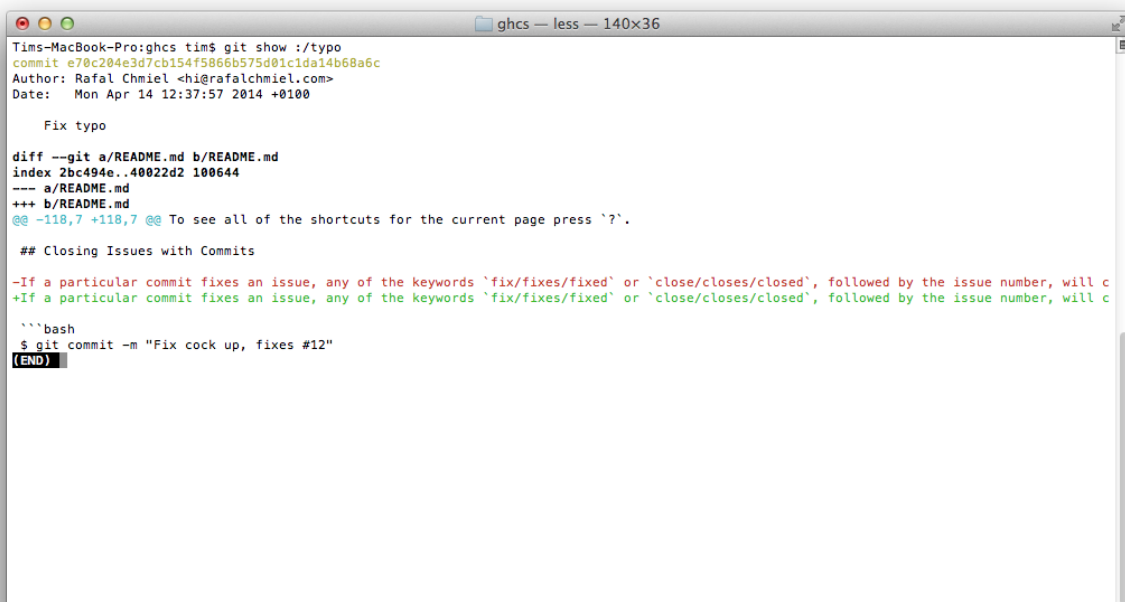
## Git 查询

Git 查询运行你在之前的所有提交信息里进行搜索, 找到其中和搜索条件相匹配的最近的一条。

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

这里 `query` (区别大小写) 是你想要搜索的词语, 这条命令会找到包含这个词语的最后那个提交并显示变动详情。

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



- 按 `q` 键退出命令。\*

## 合并分支

输入命令:

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

这会显示所有已经合并到你当前分支的分支列表。

相反地:

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

会显示所有还没有合并到你当前分支的分支列表。

*进一步了解 `Git branch` 命令.*

## 修复有问题的提交以及自动合并

如果上一个或多个提交包含了错误,可以在你修复问题后使用下列命令处理(假设要修复的提交版本是 `abcde`):

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

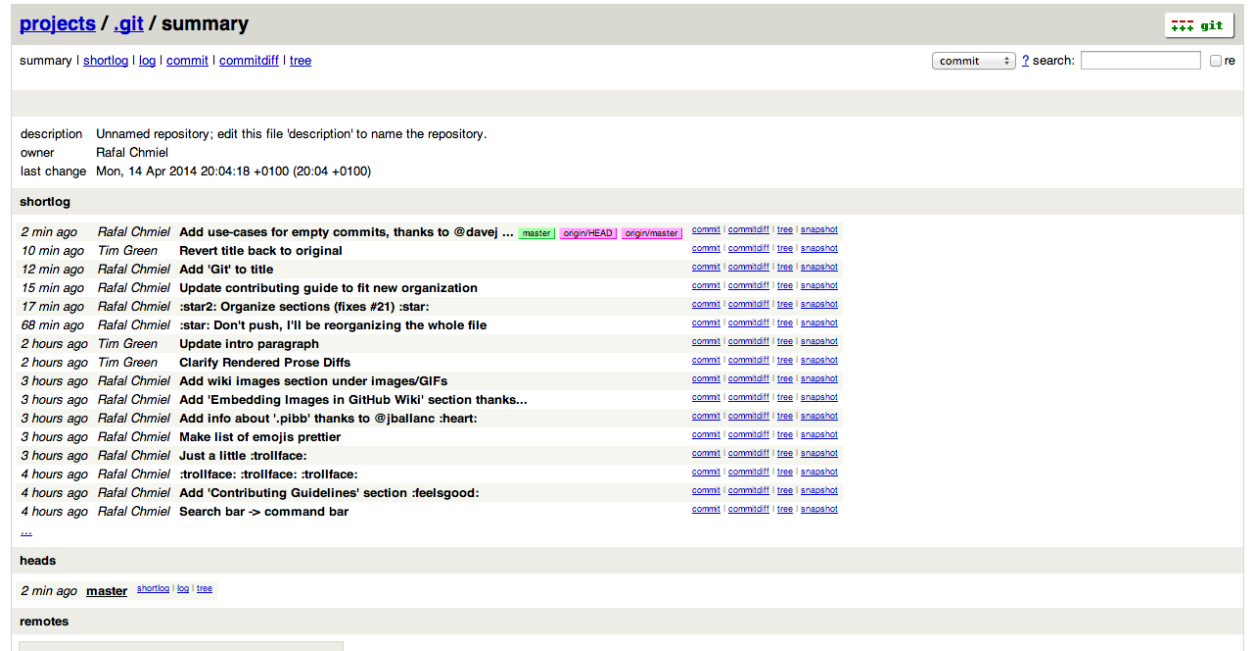
*进一步了解 `Git commit` 命令. 进一步了解 `Git rebase` 命令.*

## 以网站方式查看本地仓库

使用 `Git` 的 `instaweb` 可以立即在 `gitweb` 中浏览你的工作仓库。这个命令是个简单的脚本,配置了 `gitweb` 和用来浏览本地仓库的Web服务器。(译者注:默认需要`lighttpd`支持)

```
$ git instaweb
```

执行后打开:



进一步了解 `Git instaweb` 命令。

## Git 配置

所有 Git 配置都保存在你的 `.gitconfig` 文件中。

## Git 命令自定义别名

别名用来帮助你定义自己的 `git` 命令。比如你可以定义 `git a` 来运行 `git add --all`。

要添加一个别名，一种方法是打开 `~/.gitconfig` 文件并添加如下内容：

```
[alias]
  co = checkout
  cm = commit
  p = push
  # Show verbose output about tags, branches or remotes
  tags = tag -l
  branches = branch -a
  remotes = remote -v
```

...或者在命令行里键入：

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

例如：

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

指向多个命令的别名可以用引号来定义：

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

下面列出了一些有用的别名：

| 别名<br>Alias | 命令 Command   | 如何设置 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 '*'   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 --" |

### 自动更正

如果键入 `git comit` 你会看到如下输出：

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

# Did you mean this?
#   commit
```

为了在键入 `comit` 调用 `commit` 命令，只需启用自动纠错功能：

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

现在你就会看到：

```
$ 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...
```

### 颜色输出

要在你的 `Git` 命令输出里加上颜色的话，可以用如下命令：

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

*进一步了解 `Git config` 命令.*

### Git 资源

| 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 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 参考书籍

| 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>   |