



# GitHub秘籍

---

极客学院出版

# 前言

---

Git 是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目。GitHub 可以托管各种 Git 库，是一个网站，给用户提供 Git 服务。本文主要讲解了 Git 和 Github 比较炫酷的功能，能够帮助读者深入了解 Github 这个网站的功能。

## 适用人群

本书为 Github 中级教程，适用在 Github 上做开源项目、制作自己的博客和协同做项目的开发者。

## 学习前提

学习本书前，需要你熟悉基本的 Github 操作和 Git 命令，了解 Markdown 编译器的使用。

鸣谢：[GitHub 秘籍提供方[href="https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.zh-cn.md"](https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.zh-cn.md))]

# 目录

---

前言 .....	1
第 1 章    Github 实用秘籍 .....	3
Github 秘籍一 .....	4
Github 秘籍二 .....	15
Github 秘籍三 .....	23
第 2 章    Git 实用秘籍 .....	33
Git 实用功能 .....	34
Git 配置 .....	41
Git 资源 .....	44



1

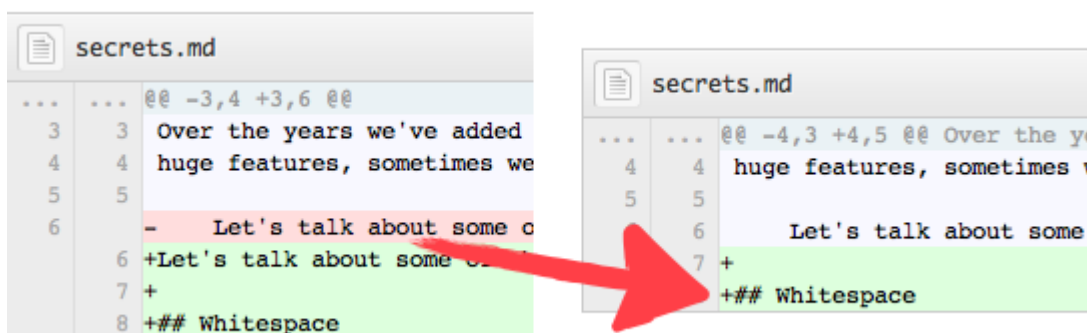
# Github 实用秘籍



## Github 秘籍一

### 不比较空白字符

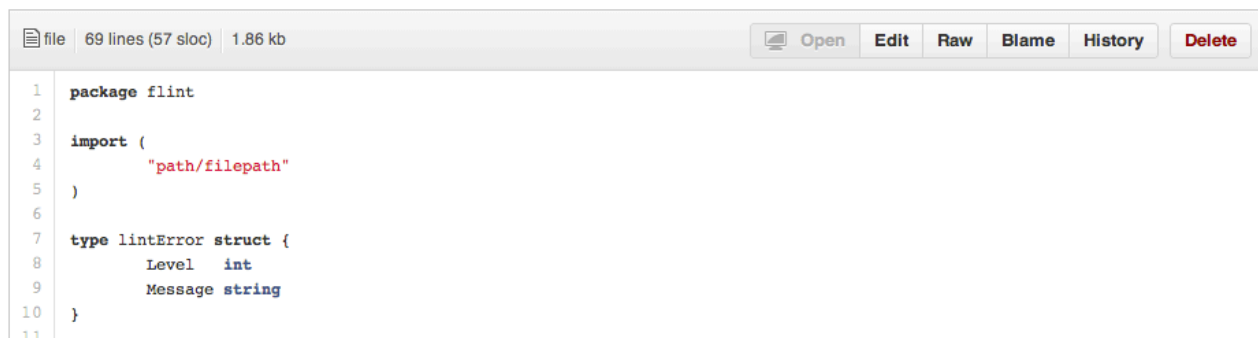
在任意 diff 页面的 URL 后加上 `?w=1`，可以去掉那些只是空白字符的改动，使你能更专注于代码改动。



### 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 \(https://chrome.google.com/webstore/detail/github-tab-size/ofjbgncegkdemndciafjngjbdpfbmbkn\)](https://chrome.google.com/webstore/detail/github-tab-size/ofjbgncegkdemndciafjngjbdpfbmbkn)，[Opera 扩展 \(https://addons.opera.com/en/extensions/details/github-tab-size/\)](https://addons.opera.com/en/extensions/details/github-tab-size/)。

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：



添加后的样子：

```

file 69 lines (57 sloc) 1.86 kb
Open Edit Raw Blame History Delete

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

```









## 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** ...  78a4884 <>
-  **Stop promoting rack-cache usage at the moment (not so common or impor...** ...  1302edf <>
-  **Show how to change the queuing backend for ActiveJob in production** ...  6463495 <>
-  **Set all asset options together** ...  b9b28d8 <>

Commits on Jan 9, 2015

-  **Merge pull request #18413 from brainopia/automatic\_inverse\_of\_for\_bel...** ...  6eb499f <>

深入了解提交视图之间的区别 (<https://help.github.com/articles/differences-between-commit-views/>)

## 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 (<http://git-scm.com/docs/git-clone>)

## 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

Overview Active Stale All branches Search branches...

**Default branch**

**master** Updated an hour ago by fxn ✓ **Default**

**Active branches**

Branch Name	Updated	By	Status	Commits	Action
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 >](#)

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.

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

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**

lantropov	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 (<https://help.github.com/articles/comparing-commits-across-time/>)



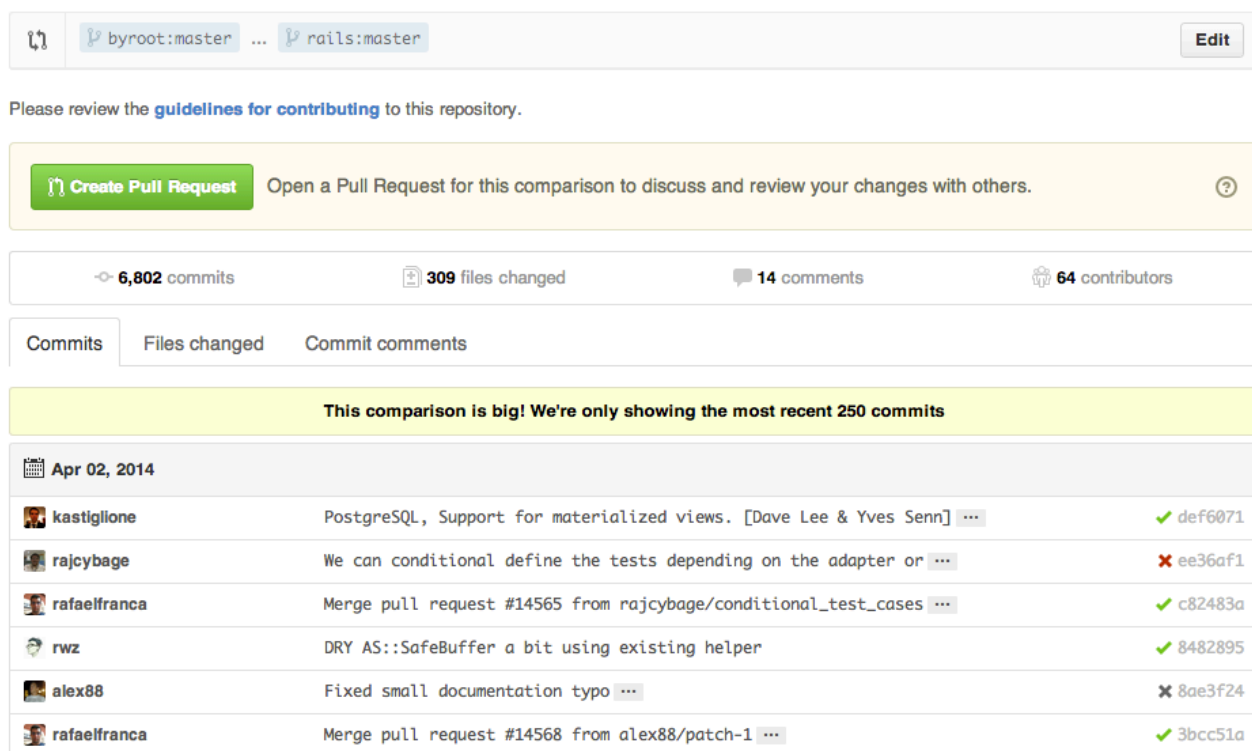
## 比较不同派生库的分支

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
```









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 \(https://gist.github.com/\)](https://gist.github.com/) 方便我们管理代码片段，不必使用功能齐全的仓库。

The screenshot shows a GitHub Gist page for a public gist titled "app.rb" by user "tiimgreen". The gist contains a single line of Ruby code: `puts 'Hello World'`. The page includes a search bar, a "Discover Gists" button, and a sidebar with options to "Download Gist", "Clone this gist", "Embed this gist", and "Link to this gist". A comment by "rafalchmiel" is visible, stating "Wow, dat is some engineering." The page also features a "Write" tab for adding comments and a "Preview" tab. The gist is created 4 days ago and has 0 stars and 0 forks.

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
```

The screenshot shows a terminal window titled "tim - bash - 80x17". The terminal output shows the command `git clone https://gist.github.com/tiimgreen/10545817` being executed. The output indicates that the repository was successfully cloned into a directory named "10545817". The terminal shows the following output:

```
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 \(https://help.github.com/articles/creating-gists\)](https://help.github.com/articles/creating-gists)

## Git.io

[Git.io \(http://git.io/\)](http://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 \(https://github.com/blog/985-git-io-github-url-shortene\)](https://github.com/blog/985-git-io-github-url-shortene)

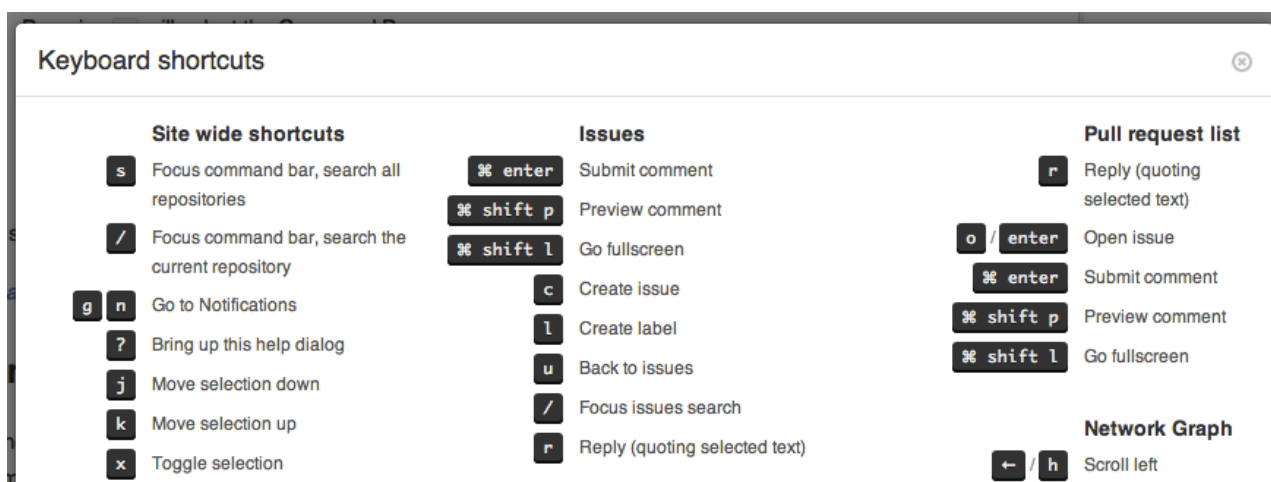
## 键盘快捷键

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：



[进一步了解可用的搜索语法 \(https://help.github.com/articles/search-syntax/\)](https://help.github.com/articles/search-syntax/)

## 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`

```

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    ActiveSupport::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

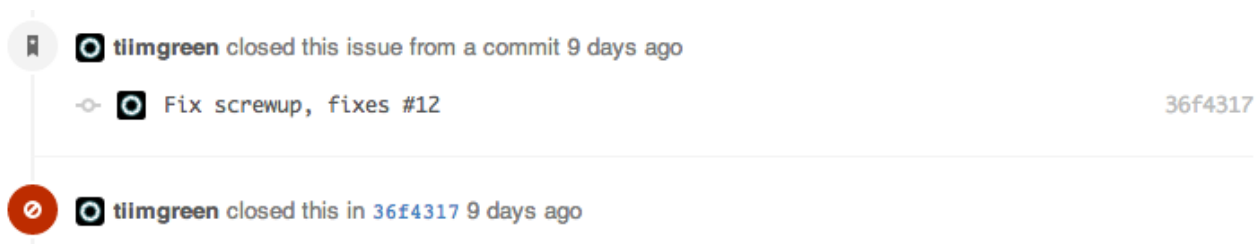
```

## 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。



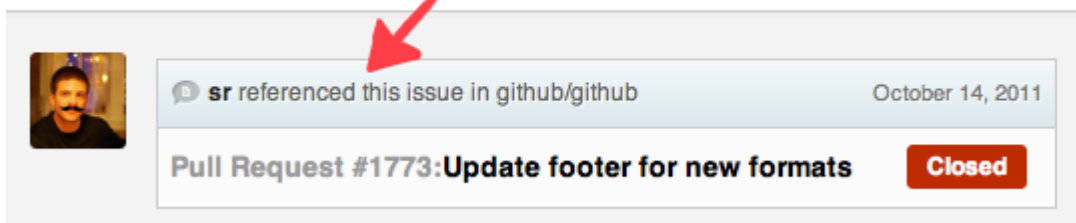
进一步了解通过提交信息关闭 Issue (<https://help.github.com/articles/closing-issues-via-commit-messages>)

## 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 的链接。

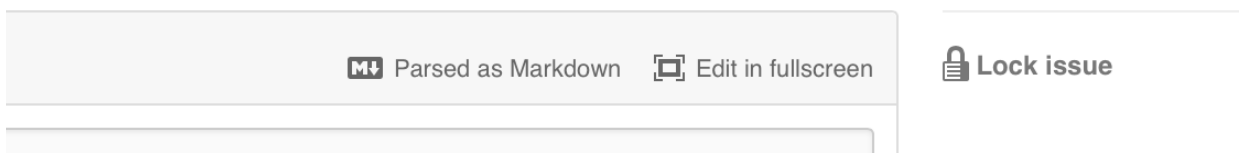
要链接到其他仓库的 Issue，就使用 {user}/{repo}#ISSUE\_NUMBER 的方式，例如 `tiimgreen/toc#12`。

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

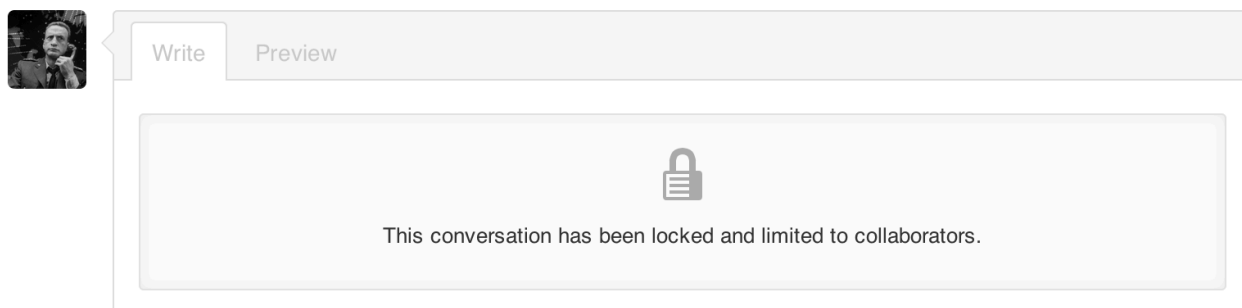


## 锁定项目对话功能

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。



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。



进一步了解对话锁定功能 (<https://github.com/blog/1847-locking-conversations>)

## 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 \(http://docs.travis-ci.com/user/getting-started/\)](http://docs.travis-ci.com/user/getting-started/)。

The screenshot shows a GitHub pull request interface. At the top, the repository is 'octokit / octokit.rb' with 68 watches, 1,510 stars, and 291 forks. The pull request title is 'Lazy create test repos #452'. A green 'Open' button indicates the pull request is ready for review. The description states: 'Alternative to mass repo creation in #429. Checks for and creates test repositories before recording new cassettes.' The pull request is from the 'lazy-create-test-re...' branch to the 'master' branch. The commit history shows two commits by 'joeyw' a month ago: 'Lazily create test repositories on new cassette recording' (commit dfce8db) and 'Ignore test repo setup interactions so they arent saved in cassettes' (commit 269b525). A green status bar at the bottom indicates 'All is well — The Travis CI build passed'. The right sidebar shows 'Labels' (None yet), 'Milestone' (No milestone), 'Assignee' (No one assigned), and 'Notifications' (Subscribe button).

Public octokit / octokit.rb Watch 68 Unstar 1,510 Fork 291

## < Lazy create test repos #452

**New Issue**

**Open** joeyw wants to merge 2 commits into master from lazy-create-test-re...

Conversation 0 Commits 2 Files changed 1 +17 -0

joeyw commented a month ago Collaborator

Alternative to mass repo creation in #429

Checks for and creates test repositories before recording new cassettes.

joeyw added some commits a month ago

- Lazily create test repositories on new cassette recording ✓ dfce8db
- Ignore test repo setup interactions so they arent saved in cassettes ✓ 269b525

✓ All is well — The Travis CI build passed · Details

Write Preview Comments are parsed with GitHub Flavored Markdown

Labels: None yet

Milestone: No milestone

Assignee: No one assigned

Notifications: Subscribe

You'll be notified when you participate or are @mentioned in this thread.

进一步了解提交状态 API (<https://github.com/blog/1227-commit-status-api>)

## Github 秘籍二

---

### 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：

<

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

```
</code></pre>
```

效果如下：

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

Github 使用 [Linguist \(https://github.com/github/linguist\)](https://github.com/github/linguist) 做语言识别和语法高亮。你可以仔细阅读 [languages YAML file \(https://github.com/github/linguist/blob/master/doc/quickstart.md\)](https://github.com/github/linguist/blob/master/doc/quickstart.md)

进一步了解 [GitHub Flavored Markdown \(https://help.github.com/articles/github-flavored-markdown/\)](https://help.github.com/articles/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](http://www.emoji-cheat-sheet.com/) (<http://www.emoji-cheat-sheet.com/>) 或 [scotch-io/All-Github](https://scotch-io.com/All-Github-emojis/)

Github 上使用最多的 5 个表情符号是：

- :shipit:
- :sparkles:
- :-1:
- :+1:
- :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：

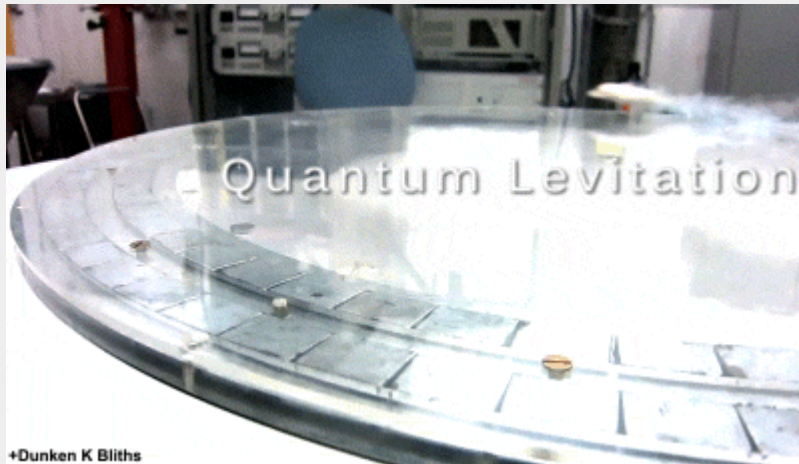


图片 1.20 Alt Text

仓库中的图片可以被直接引用：

Alt Text

图片 1.21 Alt Text



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。

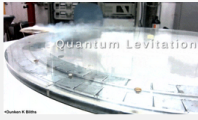
## 在 GitHub Wiki 中引用图片

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

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

结果：

Welcome to the http-api-design-ZH\_CN wiki!



▼ Pages 1

Home

Clone this wiki locally

<https://github.com/wenquan0hf/>

Clone in Desktop

© 2015 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Contact](#)



[Status](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

## 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。



jakeboxer commented

2 minutes ago

Yep, I really like the second option—it puts exactly the right amount of emphasis on the content.



Write

Preview

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

Leave a comment

Attach images by dragging & dropping them or [choose an image](#)

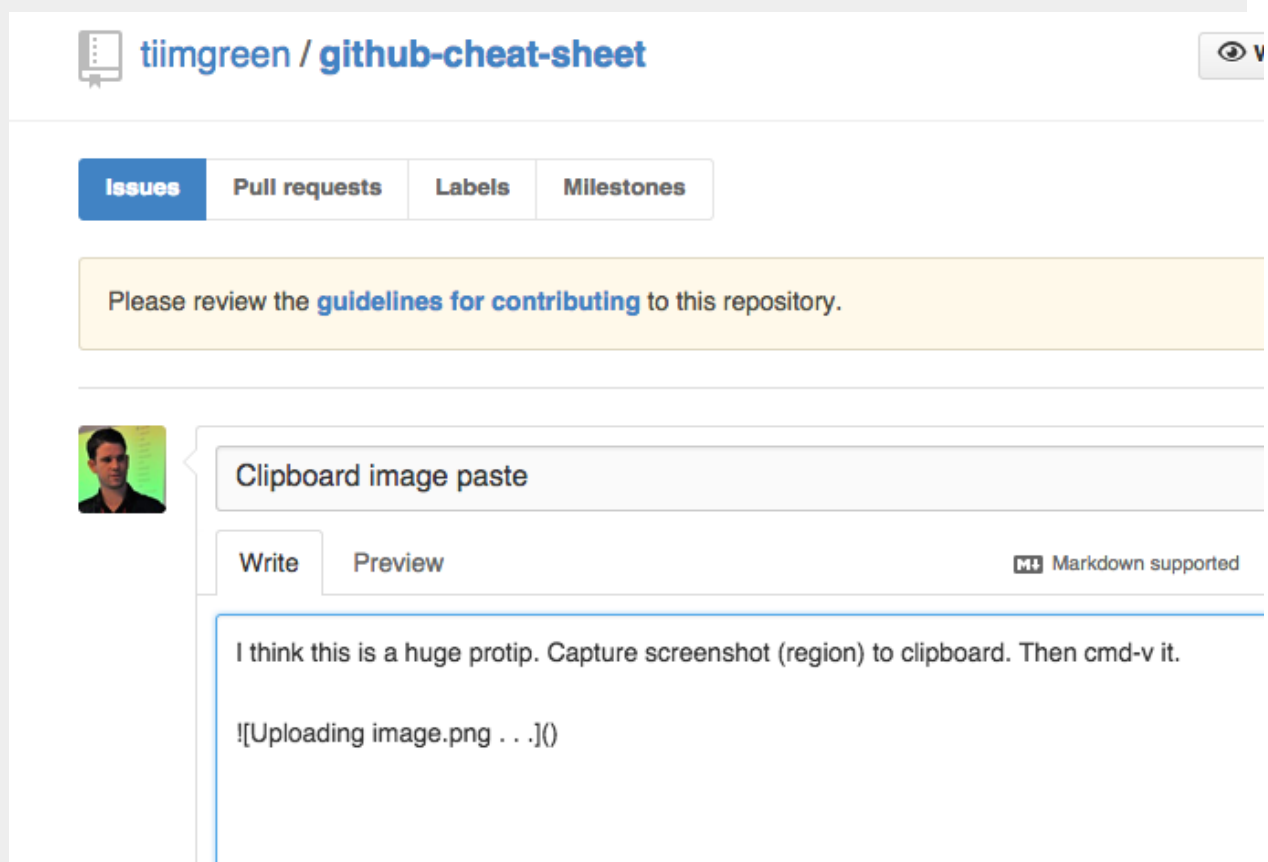
Comment

进一步了解快速引用 (<https://github.com/blog/1399-quick-quotes>)

## 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



进一步了解在 issue 中使用附件 (<https://help.github.com/articles/issue-attachments>)

## 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：

☐ **Initialize this repository with a README**

This will allow you to `git clone` the repository immediately. Skip this step if you have already run `git`

Add .gitignore: **None** ▼

Add a license: **None** ▼ ⓘ

Create repository

**Licenses** ⓘ

Filter licenses...

GPL v2

✓ MIT License

Affero GPL

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选择一个作为模板。

tiimgreen.github.io / LICENSE

or cancel

Want to use a template?

Choose a license: **None** ▼ ⓘ

**License** ⓘ

Filter licenses...

✓ Apache v2 License

GPL v2

MIT License

Affero GPL

Artistic License 2.0

BSD (3-Clause) License

BSD 2-Clause license

CC0 1.0 Universal

Eclipse Public License v1.0

这个技巧也适用于 .gitignore 文件。

进一步了解开源许可证 (<https://help.github.com/articles/open-source-licensing>)

## 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 (<https://help.github.com/articles/writing-on-github#task-lists>)

## 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 – ☒ Venus – ☒ Earth – ☒ Moon – ☒ Mars – ☐ Deimos – ☐ Phobos

```
– ☐ Mercury
– ☒ Venus
– ☒ Earth
– ☒ Moon
– ☒ Mars
– ☐ Deimos
– ☐ Phobos
```

进一步了解 Markdown 文件中的任务列表 (<https://github.com/blog/1825-task-lists-in-all-markdown-documents>)

## 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 \(页 0\)](#) [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 (<https://help.github.com/articles/relative-links-in-readmes>)

# Github 秘籍三

## 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 }} 显示仓库名称。Jemoji 和 jekyll-mentions 插件为你的 Jekyll 文章和页面增加了 [emojihref="https://github.com/tiimgreen/github-cheat-sheet/"] 和 [mention="https://github.com/tiimgreen/github-cheat-sheet/"] 支持。了解更多 GitHub Pages 的元数据和插件支持 (<https://github.com/blog/1797-repository-metadata-and-plugin-support-for-github-pages>)

## 查看 YAML 格式的元数据

许多博客站点，比如基于 Jekyll (<http://jekyllrb.com/>) 的 GitHub Pages (<http://pages.github.com/>)，都依赖于一些文章头部的 YAML 元数据。

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	notes	online	class	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 (<https://github.com/blog/1647-viewing-yaml-metadata-in-your-documents>)

## 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。

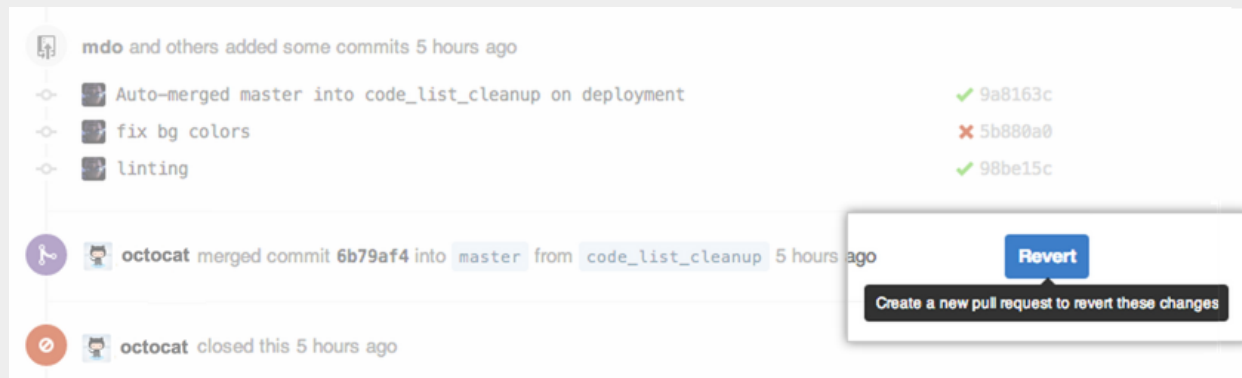
file   869 lines (868 sloc)   188.222 kb					
Search this file...					
1	Title	Release Year	Locations	Fun Facts	Production Company
2	180	2011	555 Market St.		SPI Cinemas
3	180	2011	Epic Roasthouse (399 Embarcade...		SPI Cinemas
4	180	2011	Mason & California Streets (Nob Hill)		SPI Cinemas
5	180	2011	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

进一步了解渲染表格数据 (<https://github.com/blog/1601-see-your-csvs>)

## 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

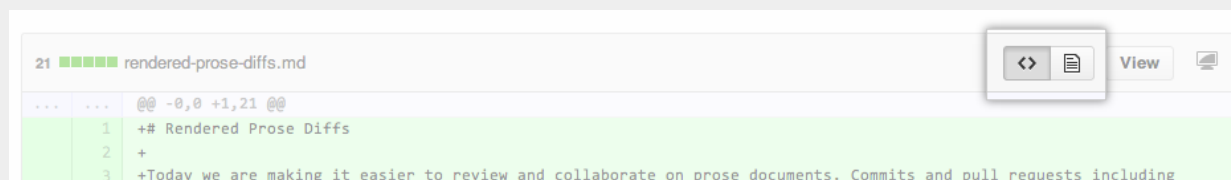


进一步了解“撤销”按钮 (<https://github.com/blog/1857-introducing-the-revert-button>)

## 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。

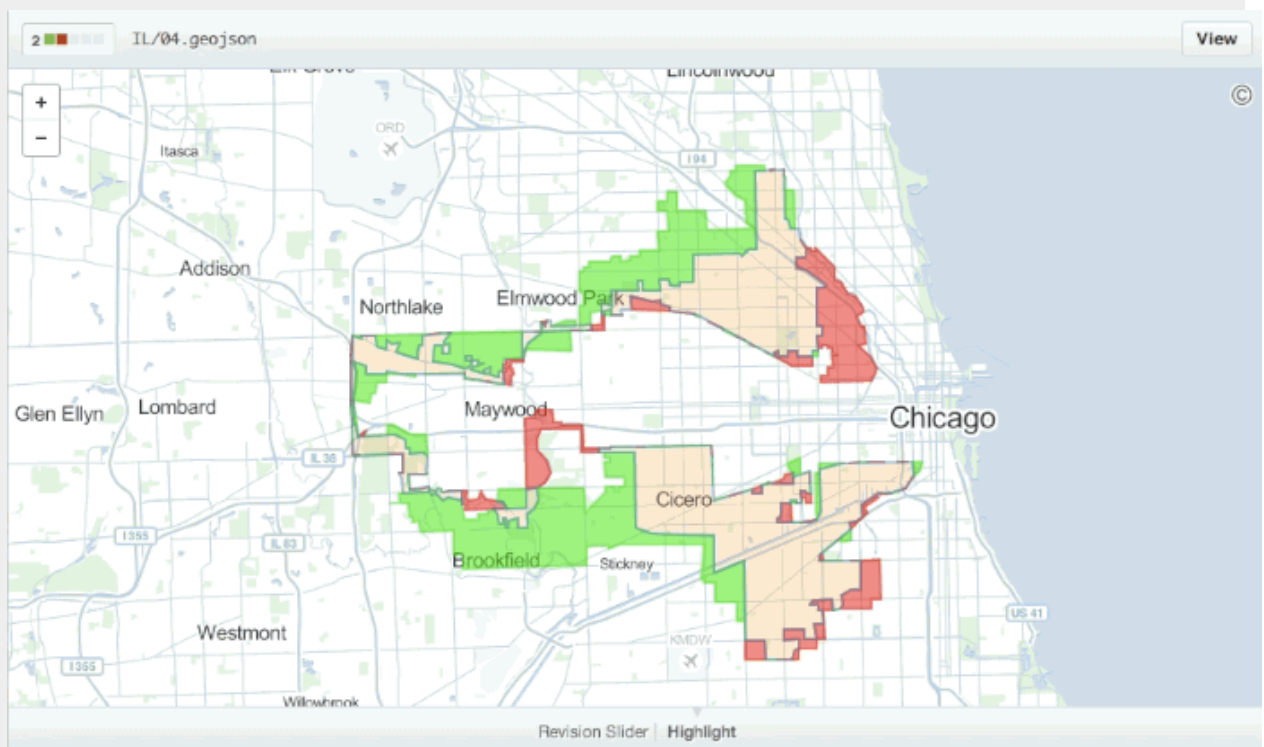
Obviously, you'll want to change `<your_client_id>` to match your actual Client ID.

Also, notice that the URL uses the `scope` query parameter to define the `scopes` requested by the application. For our application, we're requesting `user:email` scope for reading private email addresses.

进一步了解渲染纯文本视图 Diffs (<https://github.com/blog/1784-rendered-prose-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 (<https://github.com/blog/1772-diffable-more-customizable-maps>)

## 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

	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	//
✖		@@ -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 setWithObject:@"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 setWithObject: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 (<https://github.com/blog/1705-expanding-context-in-diffs>)

## 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：

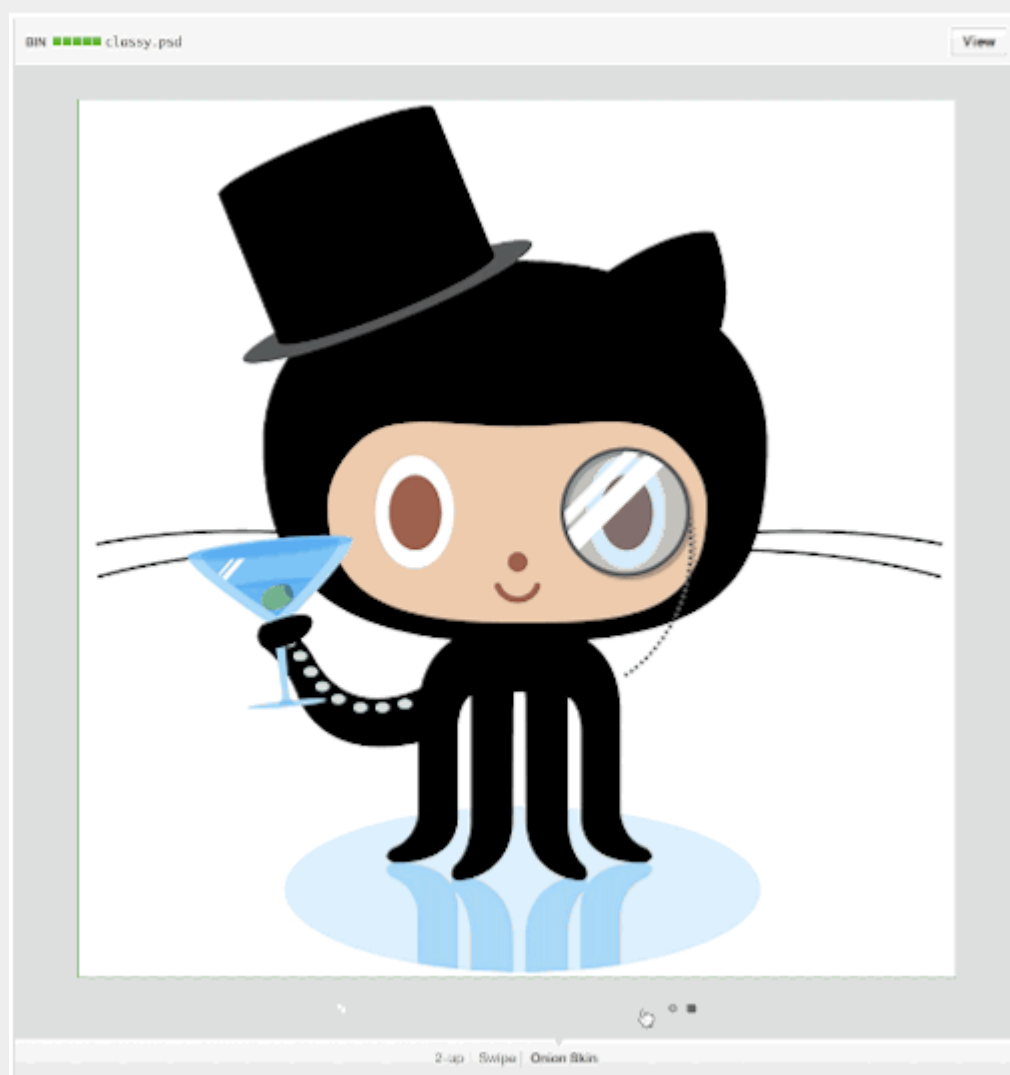
<https://github.com/tiimgreen/github-cheat-sheet/pull/15> <https://github.com/tiimgreen/github-cheat-sheet/pull/15.diff> <https://github.com/tiimgreen/github-cheat-sheet/pull/15.patch>

.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 \(页 0\) - Quick Licensing \(页 0\) - TODO Lists \(页 0\) +- Relative Links \(页 0\) - .gitconfig Recommendations \(页 0\) - Aliases \(页 0\) - Auto-correct \(页 0\) @@ -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 (<https://help.github.com/articles/rendering-and-diffing-images>)

## Hub

Hub (<https://github.com/github/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。

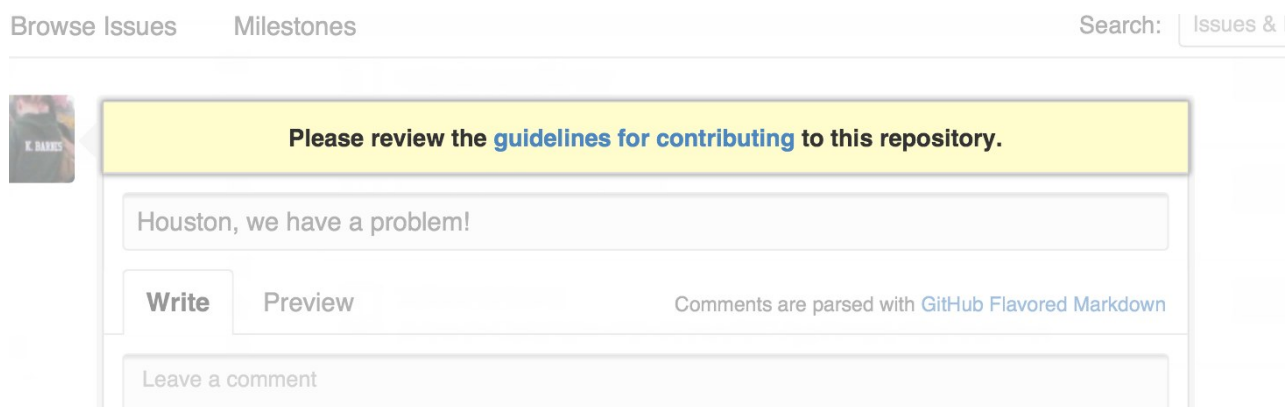
例如可以像下面这样进行克隆：

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

查看更多 Hub 提供的超酷命令 (<https://github.com/github/hub#commands>)

## 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 (<https://github.com/blog/1184-contributing-guidelines>)

## Octicons

GitHub 图标库 (Octicons) 现已开源。



进一步了解 GitHub 图标库 (<https://octicons.github.com/>)

## 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 实用秘籍



## Git 实用功能

---

### 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 (<http://git-scm.com/book/en/Git-Branching-Basic-Branching-and-Merging>)

## 去除空白

Git Stripospace 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 stripospace < README.md
```

[进一步了解 Git stripospace 命令 \(http://git-scm.com/docs/git-stripospace\)](http://git-scm.com/docs/git-stripospace)

## 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 (<https://help.github.com/articles/checking-out-pull-requests-locally>)

## 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：

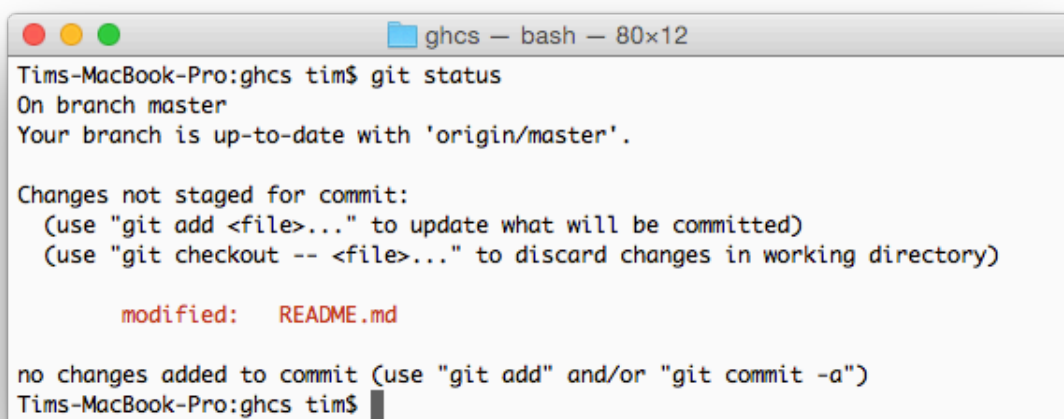
- 标记新的工作或一个新功能的开始。
- 记录对项目的跟代码无关的改动。
- 跟使用你仓库的其他人交流。
- 作为仓库的第一次提交，因为第一次提交后不能被 rebase：`git commit -m "init repo" --allow-empty`

## 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. The output indicates the user is on the 'master' branch and it is up-to-date with 'origin/master'. It then shows 'Changes not staged for commit:' with instructions to use 'git add' or 'git checkout --' to stage or discard changes. A file named 'README.md' is listed as 'modified'. Finally, it states 'no changes added to commit' and provides instructions to use 'git add' and '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)

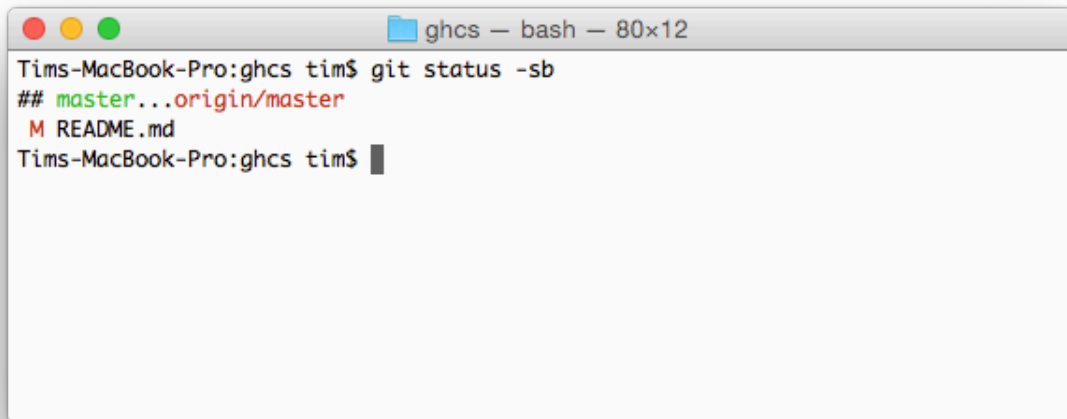
        modified:   README.md

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:

A screenshot of a macOS terminal window titled 'ghcs - bash - 80x12'. The window shows the output of the command 'git status -sb'. The output is: 'Tims-MacBook-Pro:ghcs tim\$ git status -sb', followed by '## master...origin/master' in green, then 'M README.md' in red, and finally 'Tims-MacBook-Pro:ghcs tim\$' with a cursor.

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

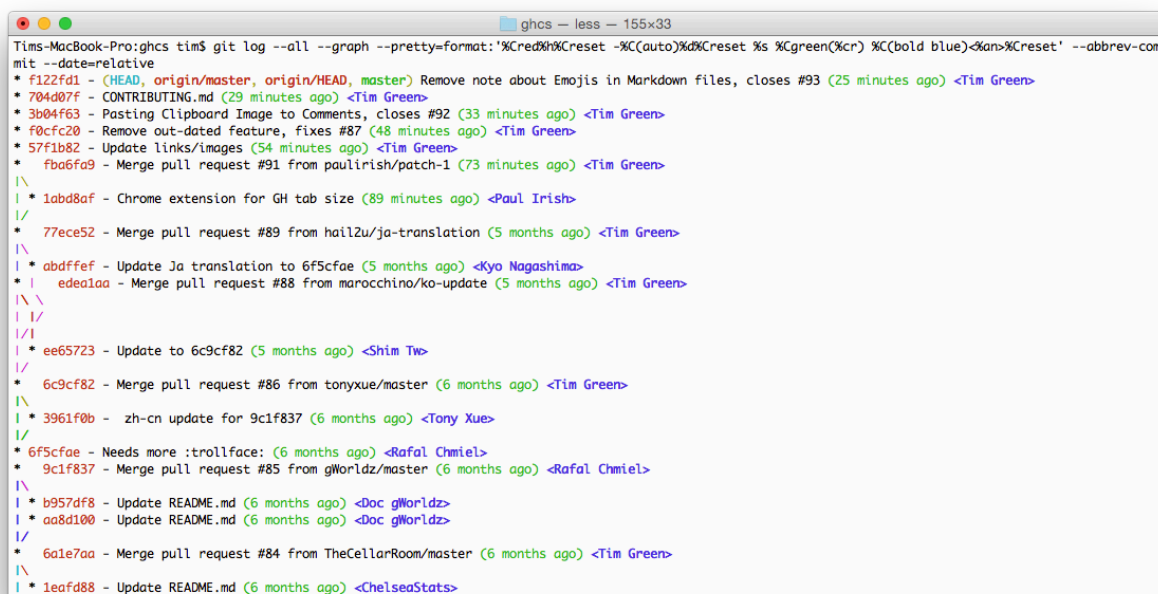
进一步了解 Git status 命令 (<http://git-scm.com/docs/git-status>)

## 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)<
```

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)<kan>%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>
* 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 hailzu/ja-translation (5 months ago) <Tim Green>
| \
| * abdfef - 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 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>
| \
| * 1eaf888 - Update README.md (6 months ago) <ChelseaStats>

```

这要归功于Palesz (<http://stackoverflow.com/users/88355/palesz>) 在 stackoverflow 的回答。

这个命令可以被用作别名，详细做法见[这里](https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.zh-cn.md)。

[进一步了解 Git log 命令](http://git-scm.com/docs/git-log)

## 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
```

```

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`, 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)

```

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 \(http://git-scm.com/docs/git-branch\)](http://git-scm.com/docs/git-branch)

## 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 \(http://git-scm.com/docs/git-commit\)](http://git-scm.com/docs/git-commit)



[进一步了解 Git rebase 命令 \(http://git-scm.com/docs/git-rebase\)](http://git-scm.com/docs/git-rebase)

## 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：

The screenshot shows the Git web interface for a repository named `projects / .git / summary`. The interface includes a search bar and a navigation menu. The main content area displays the repository's description, owner (Rafal Chmiel), and last change (Mon, 14 Apr 2014 20:04:18 +0100). Below this, the **shortlog** section lists recent commits with their authors, commit messages, and links to the commit, diff, tree, and snapshot. The **heads** section shows the `master` branch. The **remotes** section is currently empty.

Time	Author	Commit Message	Commit	Diff	Tree	Snapshot
2 min ago	Rafal Chmiel	Add use-cases for empty commits, thanks to @davej ...	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
10 min ago	Tim Green	Revert title back to original	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
12 min ago	Rafal Chmiel	Add 'Git' to title	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
15 min ago	Rafal Chmiel	Update contributing guide to fit new organization	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
17 min ago	Rafal Chmiel	:star2: Organize sections (fixes #21) :star:	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
68 min ago	Rafal Chmiel	:star: Don't push, I'll be reorganizing the whole file	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
2 hours ago	Tim Green	Update intro paragraph	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
2 hours ago	Tim Green	Clarify Rendered Prose Diffs	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
3 hours ago	Rafal Chmiel	Add wiki images section under images/GIFs	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
3 hours ago	Rafal Chmiel	Add 'Embedding Images in GitHub Wiki' section thanks...	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
3 hours ago	Rafal Chmiel	Add info about 'plibb' thanks to @jballanc :heart:	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
3 hours ago	Rafal Chmiel	Make list of emojis prettier	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
3 hours ago	Rafal Chmiel	Just a little :trollface:	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
4 hours ago	Rafal Chmiel	:trollface: :trollface: :trollface:	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
4 hours ago	Rafal Chmiel	Add 'Contributing Guidelines' section :feelsgood:	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>
4 hours ago	Rafal Chmiel	Search bar -> command bar	<a href="#">commit</a>	<a href="#">diff</a>	<a href="#">tree</a>	<a href="#">snapshot</a>

[进一步了解 Git instaweb 命令 \(http://git-scm.com/docs/git-instaweb\)](http://git-scm.com/docs/git-instaweb)

## 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 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

别名 Alias	命令 Command	如何设置 What to Type
git add	<code>git add . -A</code> <code>git commit</code>	<code>git config --global alias.ac '!git add -A &amp;&amp; git commit'</code>
git status	<code>git status -sb</code>	<code>git config --global alias.st 'status -sb'</code>
git tags	<code>git tag -l</code>	<code>git config --global alias.tags 'tag -l'</code>
git branches	<code>git branch -a</code>	<code>git config --global alias.branches 'branch -a'</code>
git cleanup	<code>git branch --merged   grep -v '*'   xargs git branch -d</code>	<code>git config --global alias.cleanup '!git branch --merged   grep -v '*'   xargs git branch -d'</code>
git remote	<code>git remote -v</code>	<code>git config --global alias.remotes 'remote -v'</code>
git log	<code>git log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)&lt;%an&gt;%Creset' --abbrev-commit --</code>	<code>git config --global alias.lg "log --color --graph --pretty=format:'%Cred%h%Creset -%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)&lt;%an&gt;%Creset' --abbrev-commit --"</code>

## 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 \(http://git-scm.com/docs/git-config\)](http://git-scm.com/docs/git-config)

## 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 Designer	<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 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 Peepcode	<a href="http://peepcode.com/products/git-internals-pdf">http://peepcode.com/products/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>

# 极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台



更多信息请访问 

<http://wiki.jikexueyuan.com/project/github-secret/>