

 無料電子ブック

学習

github

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#github



.....	9
.....	9
<b>SHA</b> .....	<b>9</b>
.....	10
<b>2: Gist</b> .....	<b>11</b>
.....	11
.....	11
Examples.....	11
.....	11
.....	12
<b>3: Gitflow</b> .....	<b>13</b>
.....	13
.....	13
.....	13
Examples.....	13
5.....	13
<b>4: GitHub</b> .....	<b>15</b>
.....	15
Examples.....	15
.....	15
<b>5: GitHub</b> .....	<b>17</b>
.....	17
Examples.....	17
.....	17
.....	17
.....	18
.....	18
<b>6: GitHub</b> .....	<b>20</b>
Examples.....	20
.....	20

<b>7: GitHub</b>	<b>21</b>
Examples	21
.....	21
Git	21
GitHubURL	21
.....	22
<b>8: GitHub</b>	<b>23</b>
.....	23
.....	23
Examples	23
.....	23
.....	25
<b>9: GitHub1</b>	<b>28</b>
Examples	28
.....	28
URL	28
<b>10: GitHub/</b>	<b>29</b>
Examples	29
GitHub/	29
<b>11: GitHub</b>	<b>31</b>
Examples	31
GitHub	31
<b>12:</b>	<b>33</b>
.....	33
Examples	33
.....	33
<b>13:</b>	<b>34</b>
Examples	34
.....	34
.....	34
h21	35

<b>14:</b> .....	<b>38</b>
Examples .....	38
.....	38
<b>15:</b> .....	<b>39</b>
.....	39
.....	39
Examples .....	39
filter-branch .....	39
BFG .....	39
.....	40
.....	40
.....	<b>41</b>

---

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [github](#)

It is an unofficial and free github ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official github.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# 1: githubをいめる

このセクションでは、githubのと、がそれをするについてをします。

githubのきなテーマについてもれ、するトピックにリンクしてください。githubのドキュメントはしくなっているので、それらのトピックのバージョンをするがあります。

## Examples

インストールまたはセットアップ

GitHubはGitリポジトリのなコレクションです。つまり、GitHubをくのプロジェクトのコレクションとえることができます。

---

## アカウントの

- GitHubでのメインページをごください [ここで](#)
- ユーザーをし、メールアドレスをして、なパスワードをすると、がいました。

---

## なツール

Git / GitHubのにとって、バージョンのみをすることは、はらわしいかもしれせん。あなたがダウンロードしてうことができるGUIバージョンのGitHubがします。 [GitHub Desktop](#)はそのツールです。

---

## のリポジトリをする

リポジトリはプロジェクトとえることができます。リポジトリは、オンラインまたはオフラインでできます。のにいます。

### オンライン

1. にログインしてあなたのプロフィールにきます。
2. ページにある[リポジトリ]タブにします
3. の "New" ボタンをすと、うなずくがいました

### オフライン

1. [git](#)をダウンロードしてインストールしますしているオペレーティングシステムをしてください
2. ダウンロードとインストールの、コマンドラインツールをするか、GUIクライアントをダウンロードできます。
3. インストール、[github](#)でアカウントをする
4. の+をクリックし、しいリポジトリをするか、のリポジトリをインポートするかをします。
5. しいリポジトリをするは、リポジトリをし、またはのいずれかをします。
6. 「リポジトリの」をクリックします。

NBプライベートレポジトリはのユーザーにはできません。

## READMEファイル

プロジェクトにREADME.mdがない、GitHubはをするためにREADME.rdocをします。があるは、README.mdをして、rdocをとてします。

---

# READMEファイルには、

## プロジェクト

あなたのプロジェクトについてにしてください。また、プロジェクトのウェブサイトリンク、バッジ、コミュニティEメール、ソーシャルサイトなどをすることもできます。

## ダウンロード

ファイルファイルまたはファイルまたはインストールファイルのリンク。のバージョンへのリンクもあります。

## インストール

どのようにあなたののすることができます。、サードパーティライブラリ、などがまれます。

## デモンストレーション

コードサンプル、gifファイル、ビデオリンク、またはスクリーンショットがまれているがあります。

、など



プロジェクトをしてけられインスパイアされたやコミュニティのリスト

する

プロジェクトへの、バグの、パッチの。ドキュメントリンクもまれます。

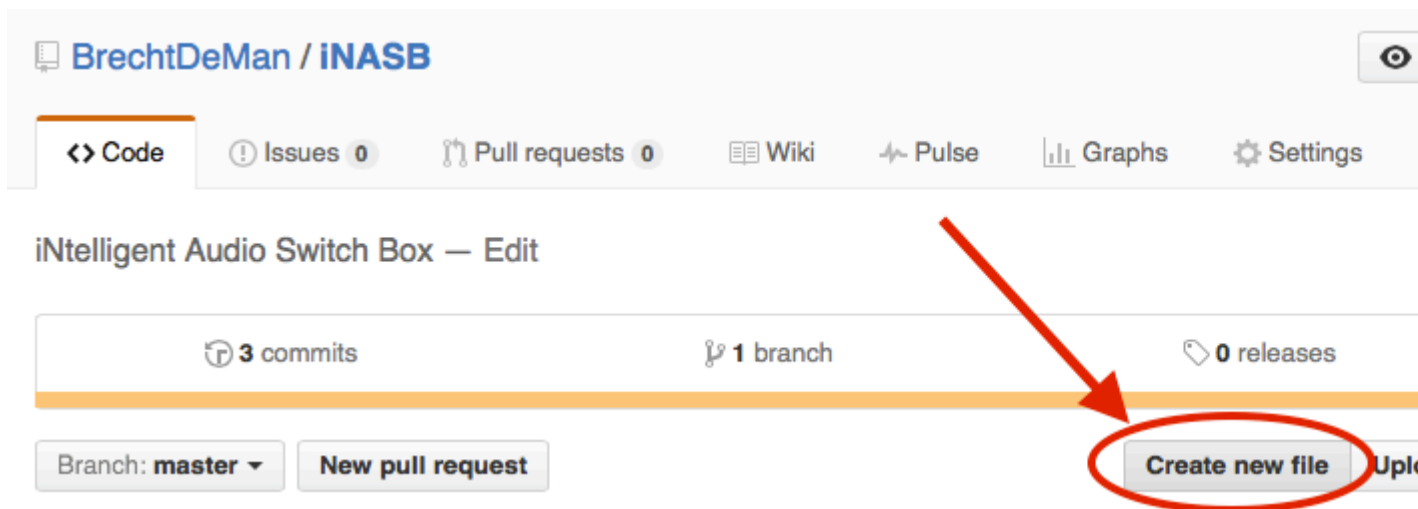
ライセンス

あなたのライセンスをしてください。ライセンスサイトへのリンクもできます。

## LICENSE ファイル

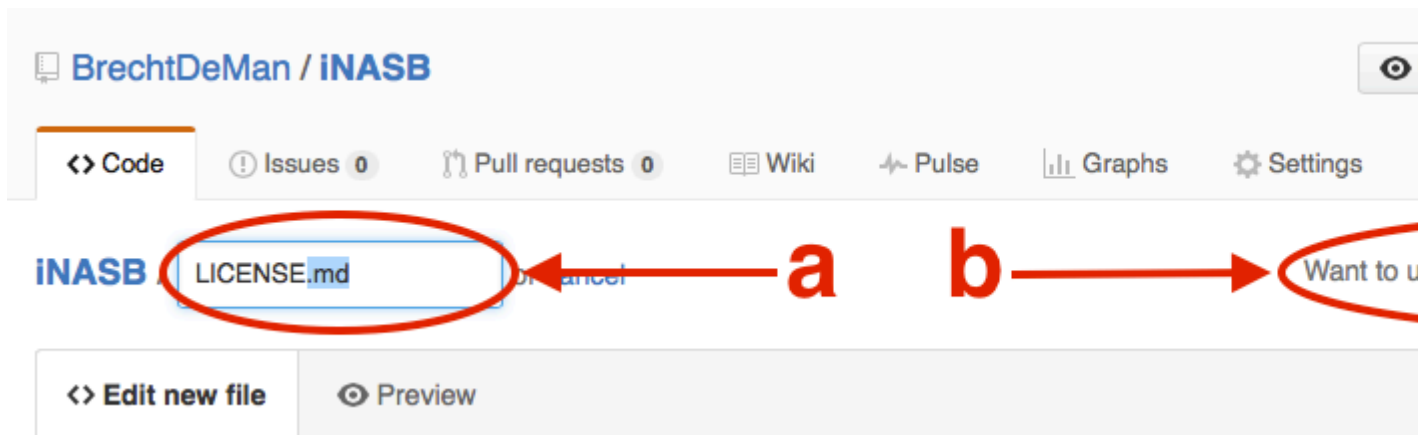
GitHubは、のテキスト/マークダウンファイルをするわりに、リポジトリにライセンスをすばやくするのにちます。

1. あなたのリポジトリで、[しいファイルを]をクリックします。



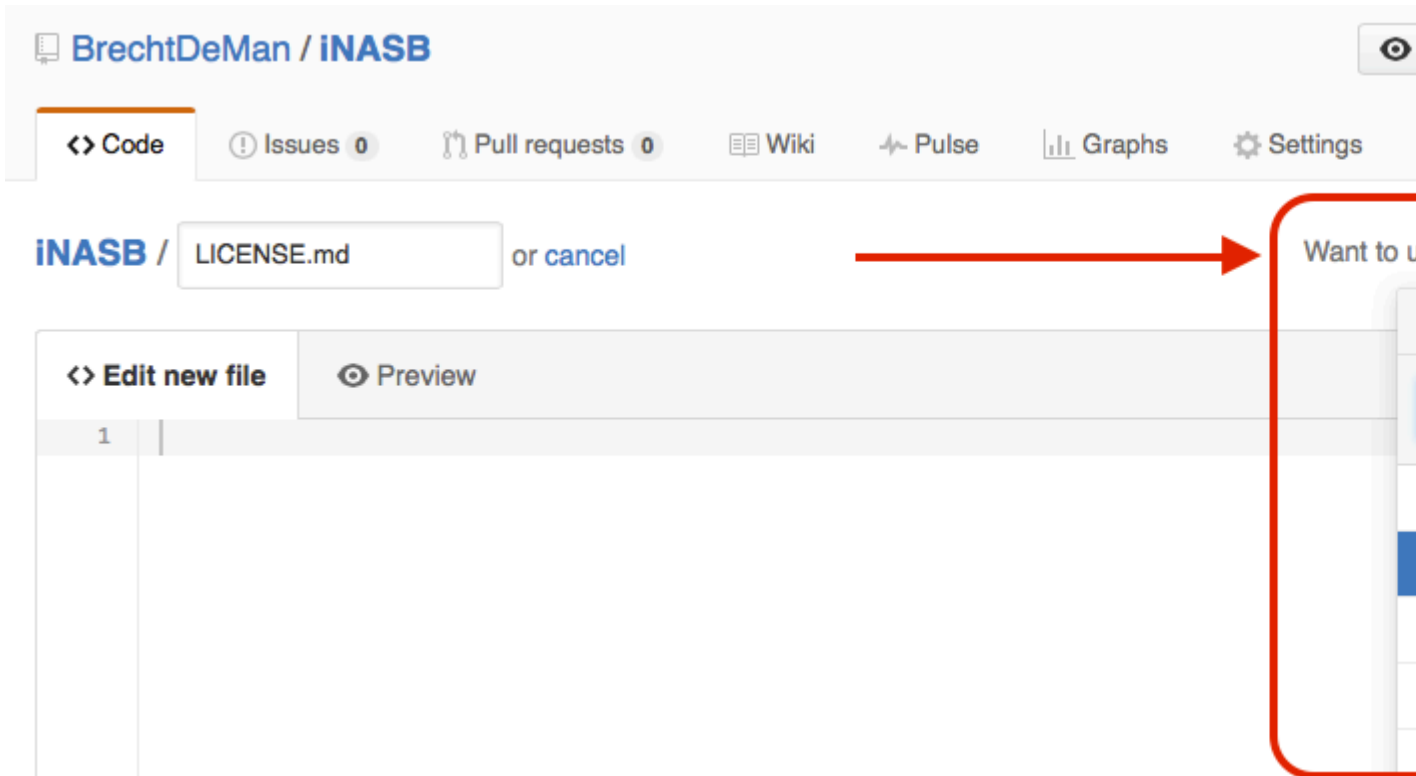
2. のページ

1. しいファイルのファイルとして `LICENSE.md` または `LICENSE.txt` をします。
2. しいテンプレートをしたいですかダイアログがされます。

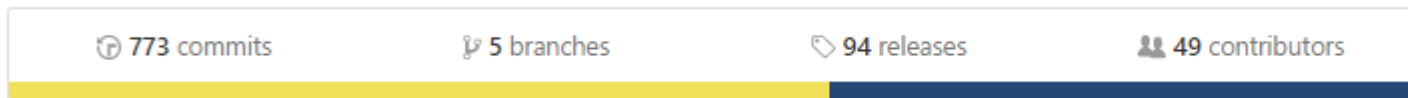


- 3.

ライセンスをしてください。



#### 4. リポジトリのにされるライセンス



[QAより](#) - のGithubプロジェクトにライセンスをする

## GitHub Flavored Markdown

GitHubは[Markdown](#)をし、しいなをします。

ヘッダ

```
# Header1
## Header2
### Header3
#### Header4
##### Header5
##### Header6
H1
===
H2
---
```

# Header1

---

## Header2

---

### Header3

#### Header4

##### Header5

###### Header6

# H1

---

## H2

---

---

```
*Italic1* _Italic2_  
**Bold1** __Bold2__  
***Bold_Italic***  
~~Strikethrough~~
```

*Italic1 Italic2*  
**Bold1 Bold2**  
***Bold\_Italic***  
~~Strikethrough~~

---

```
---  
***  
---
```

---

---

---

# リスト

unordered list:

```
* item-1
  * sub-item-1
  * sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
+ item-3
  + sub-item-5
  + sub-item-6
```

ordered list:

```
1. item-1
  1. sub-item-1
  2. sub-item-2
2. item-2
  1. sub-item-3
  2. sub-item-4
3. item-3
```

unordered list:

- item-1
  - sub-item-1
  - sub-item-2
- item-2
  - sub-item-3
  - sub-item-4
- item-3
  - sub-item-5
  - sub-item-6

ordered list:

1. item-1
  - i. sub-item-1
  - ii. sub-item-2
2. item-2
  - i. sub-item-3
  - ii. sub-item-4
3. item-3

```
Table Header-1 | Table Header-2 | Table Header-3
:--- | :---: | ---:
Table Data-1 | Table Data-2 | Table Data-3
TD-4 | Td-5 | TD-6
Table Data-7 | Table Data-8 | Table Data-9
```

Table Header-1	Table Header-2	Table Header-3
Table Data-1	Table Data-2	Table Data-3
TD-4	Td-5	TD-6
Table Data-7	Table Data-8	Table Data-9

## コード

```
inline code- `int i=0`

block code-
``` C
for(int i=0; i<10; i++){
    printf("Hallow World! \n");
}
...

```

inline code- `int i=0`

block code-

```
for(int i=0; i<10; i++){
    printf("Hallow World! \n");
}
```

## もり

```
> Stay hungry; stay foolish.
>> Quality is better than quantity.
>>> Life is not fair; get used to it.
```

Stay hungry; stay foolish.

Quality is better than quantity.

Life is not fair; get used to it.

## リンク

```
https://github.com
[GitHub] (https://github.com)
[GitHub] (https://github.com "github website")
[GitHub] [1]

[1]: https://github.com
```

<https://github.com>

[GitHub](#)

[GitHub](#)

[GitHub](#)

```
![GitHub Logo] (https://assets-cdn.github.com/images/icons/emoji/octocat.png "GitHub")
```



## タスクリスト

```
- [x] completed item
- [ ] incomplete item
```

completed item

incomplete item

```
:octocat: :+1: :book: :ghost: :bulb: :imp:
```



すべてのGitHubのemojies - [Emojiチートシート](#)。

## SHA リファレンス

コミットのSHA1ハッシュへののは、GitHubのコミットへのリンクにされます。

```
e7909ea4fbb162db3f7f543d43c30684a3fb745f
```

Write

Preview

e7909ea

## プルリクエストとの

プルリクエストまたはへののは、そのプルリクエストまたはににリンクされます。

これは、issue / Pull Requestのに#をけることでできます。

オンラインでgithubをいめるをむ <https://riptutorial.com/ja/github/topic/1214/github>をいめる

## 2: Gistをう

き

あなたのをかちうらしいです。のファイル、ファイルの、またはなアプリケーションをできます。あなたは<https://gist.github.com>でにアクセスできます。

すべてのはGitリポジトリです。つまり、forkしてすることができます。エディタはCodeMirrorをしています。

2つのタイプのがありますのとの。

さらに、gitをするときにGitHubにログインしていなければ、それはのになります。

あなたのをかちうらしいです。のファイル、ファイルの、またはなアプリケーションをできます。

2つのタイプのがありますのとの。さらに、gitをするときにGitHubにログインしていなければ、それはのになります。

の

パブリックジッパがディスカバーにされます。ディスカッションでは、されたしいをブラウズできます。なので、のにあなたのをつけてたいにもできます。

の

のはにれず、ではありません。それらをして、のであなたにたアイデアをきめたり、リストをしたり、とかちうができていないコードやをしたりしてください。

あなたはきなだけくののをすることができます。

の

GitHubにログインせずにをすると、それはのになります。のはまたはにすることができます。GitHub.comまたはGitHub Enterpriseでのをするには、それぞれGitHubサポートまたはサイトにしてください。したいのURLをしてください。

## Examples

の

のは、ほとんどでもかまいません。

Javascriptのな



```
function randomInt(min, max) {
  return Math.floor((max - min + 1) * Math.random()) + min;
}
```

の

GitHubにたくないものについては、のをするがあります。はにされないようにしたいや、をに  
したくないにできます。

のにしたJSONコードのな

```
{
  "id": AKIAIOSFODNN7EXAMPLE,
  "secret": wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY
}
```

オンラインでGistをうをむ <https://riptutorial.com/ja/github/topic/7978/gist>をう

## 3: Gitflow をってする

- git flow <サブコマンド>
- git flow init
- git flow [[リリース|プログラム] []]

### パラメーター

サブコマンド	
そのに	ブランチモデルをサポートするしいgit repoをします。
	フィーチャーブランチをします。
	リリースブランチをします。
ホットフィックス	プログラムのブランチをします。

- [からのgitflowのコンセプト](#)
- [ブランチモデル](#)

## Examples

ローカルに5つのブランチをする

Gitflowのもの1つ

### 1. レポをしてブランチをする

```
$ git flow init
# if you use default setup, you'll define six types of branches:
#
# main branches (lives forever)
#
# 1. master:  for production releases
# 2. develop: for "next release" development
#
# supporting branches
#
# 3. feature:  for a product feature
# 4. release:  for preparation of a new production release
# 5. hotfix:   for resolving critical bug of production version
# 6. support
#
# also, two main branches are created: master, develop
```

### 2. フィーチャのと

```
$ git flow feature start my_feature
  # create branch 'feature/my_feature' based on the 'develop'

  # made development and commits...

$ git flow feature finish my_feature
  # merge 'feature/my_feature' back to the 'develop'
  # delete 'feature/my_feature'
```

### 3. リリースのと

```
$ git flow release start my_release
  # create branch 'release/my_release' based on the 'develop'

  # made bug fixes...

$ git flow release finish my_release
  # merge branch 'release/my_release' to the 'master' and add tag
  # merge branch 'release/my_release' back to the 'develop'
  # delete 'release/my_release'
```

### 4. プログラムのと

```
$ git flow hotfix start my_hotfix
  # create branch 'hotfix/my_hotfix' based on the 'master'

  # made some hotfixes...

$ git flow hotfix finish my_hotfix
  # merge branch 'hotfix/my_hotfix' back to the 'master' and add tag
  # merge branch 'hotfix/my_hotfix' to the 'develop'
  # delete 'hotfix/my_hotfix'
```

オンラインでGitflowをとってするをむ <https://riptutorial.com/ja/github/topic/6231/gitflowをとってする>

## 4: GitHubからのリポジトリのクローン

- `git clone github.com/username/repository`

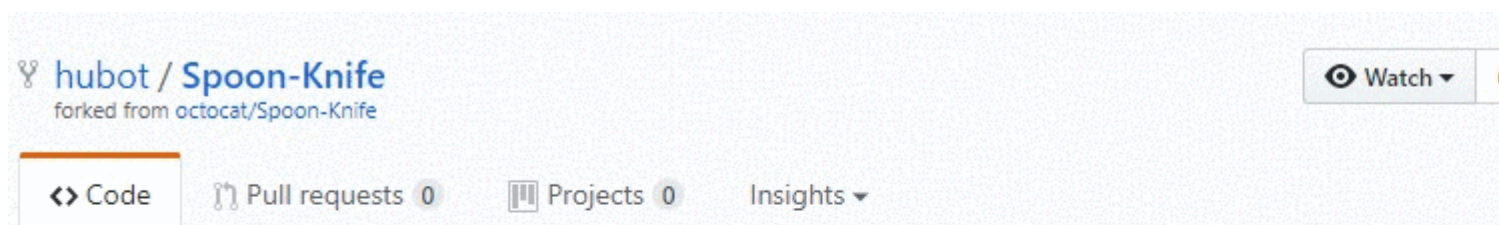
### Examples

リポジトリのクローン

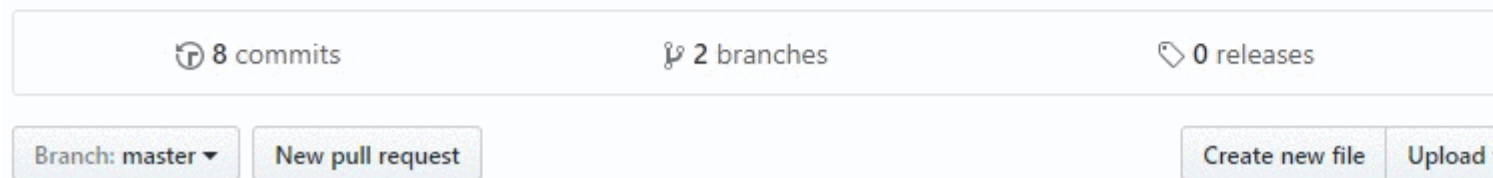
1. クローンをするリポジトリにします [https://github.com/ username / repo](https://github.com/username/repo)のようなもの



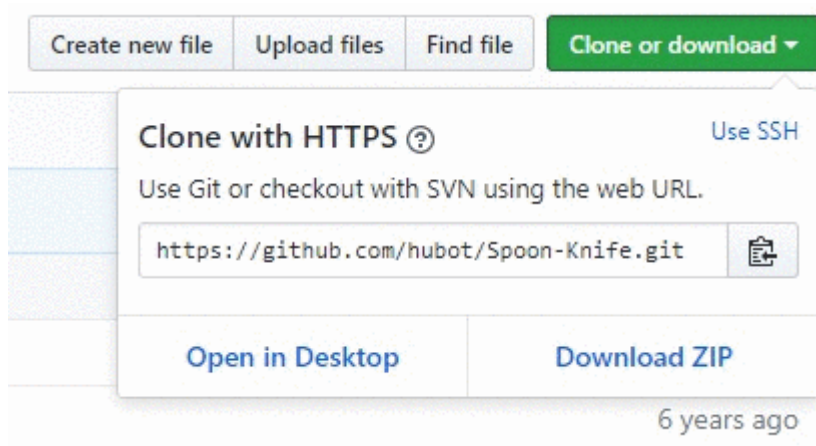
2. で、 `clone`というののボタンをクリックするか、ダウンロードしてください



This repo is for demonstration purposes only. Comments and issues may or may not be responded to.



3. さなウィンドウがされ、URL [https://github.com/ username / repo .git](https://github.com/username/repo.git)のようなものをコピーします。



4. そのプロジェクトをクローンしたいマシンのターミナルウィンドウをきます

5. コマンドラインからプロジェクトをすするにします。
6. のコマンドをします。 `git clone <copied_url_from_step_3>`
7. Enterをす
8. のようなものがされます

<repo\_name>へのクローン<repo\_name> ...

リモートオブジェクト10、。

リモートオブジェクトをすする1008/8、。

10デルタ1、 10デルタ1

アンパックオブジェクト10010/10、。

オンラインでGitHubからのリポジトリのクローンをむ

<https://riptutorial.com/ja/github/topic/3761/github>からのリポジトリのクローン

## 5: GitHub デスクトップ

き

GitHub Desktopをインストールしてするには

GitHub Desktopは、とじように、WindowsとMacOSのデスクトップで、クローン、プッシュ、プルGitHubデスクトップでの、マージなどのGitのなをんでいます...

デスクトップクライアントのなは、gitとGitHubをってにできるようにすることです。バックグラウンドでは、ほとんどのユーザーがコマンドラインからするのとじコマンドをします。

### Examples

インストールとセットアップ

MacOSとWindowsマシンのインストーラが々にされているため、インストールはにです。、ベータとの2つのバージョンがダウンロードされています。

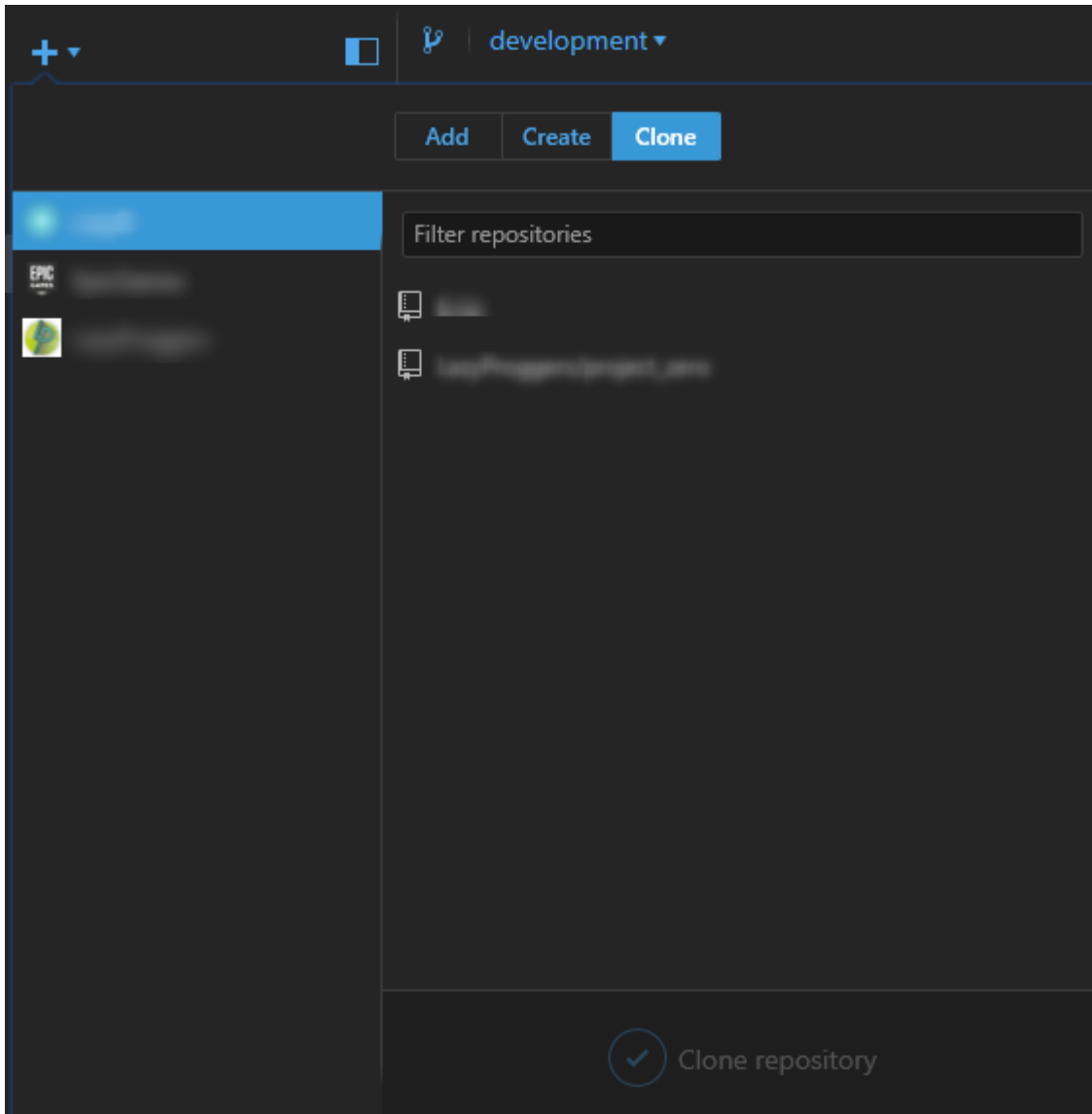
セットアップはプログラムをダウンロードしたにされ、GitHubでログインするがあります。それはにのステップです。それは、リポジトリをしたり、することができます。

インストールにGitHub DesktopだけでなくGitもインストールされます。だからあなたはそれを々にインストールするはありません。

リポジトリの

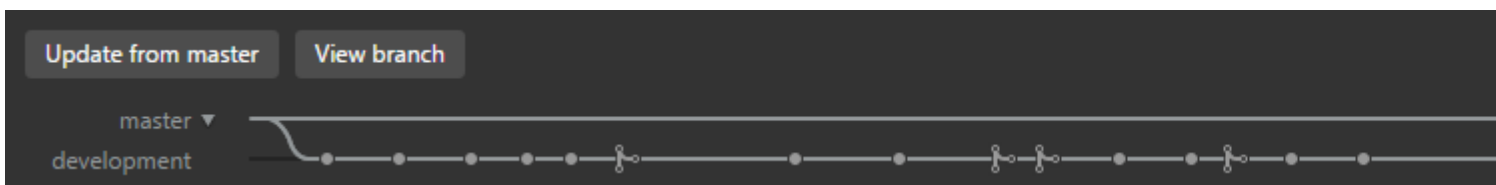
GitHub Desktopのように、ほとんどののはにシンプルです。「Clone a repository」ではのプラスをし、いくつかのリポジトリがあります。をします。あるいは、したいのリポジトリへのリンクをりけることもできます。

しいバージョンベータでは、ジェットではありません。



のことができます。のブランチをしたら、`git checkout BRANCHNAME`とじボタンをすがあります。

いバージョンでは、に2つのなるブランチをし、プッシュをすることができます。さらに、プロジェクトのタイムラインをすることもできますを



しいブランチをする

ブランチ・シンボルクライアントをクリックするか、「File --> New Branch」をクリックして、しいブランチをことができます。

ブランチをクリックすると、しいブランチがベースとしてするブランチをことができます。

プッシュとプルまたはボタン

プル

コマンドラインのように、しばらくのリポジトリののをするがあります。GitHubデスクトップでは、このプロセスはの`sync`ボタンによってびされます。

す

あなたがローカルでをえ、それらをしたいは、サマリーテキストボックスにかをくことによってコミットします。に、あなたの`Commit to YOURCURRENTBRANCH`「あなた」にします。はボタンをすがあり、プッシュがわれます。

テキストボックスから、のコミットやへの、やをすることができます。

ボタンは、`Push`、`Pull`または`Checkout`できます。

オンラインでGitHubデスクトップをむ <https://riptutorial.com/ja/github/topic/10023/githubデスクトップ>



---

## 6: GitHubのバックアップ

### Examples

ユーザーのすべてのリポジトリのクローン

のコマンドをし、ユーザをユーザにきえて、そのユーザのGitHubリポジトリをすべてカレントディレクトリにクローンします。

```
curl "https://api.github.com/users/username/repos?page=1&per_page=100" | grep -e 'git_url*' |  
cut -d \" -f 4 | xargs -L1 git clone
```

これは、の100のリポジトリのみをクローンします。

オンラインでGitHubのバックアップをむ <https://riptutorial.com/ja/github/topic/3760/githubのバックアップ>

## 7: GitHub ページ

### Examples

リポジトリのページツールの

1. GitHubのウェブサイトにく
2. あなたのリポジトリをく
3. をクリックします。
4. GitHub Pagesので、 "Launch Automatic Page Generator"をクリックします。
5. になってください

### Gitを使ってページをからする

1. しいリポジトリをするか、のリポジトリをします。
2. のない `gh-pages` というしいブランチをする

```
$ git checkout --orphan gh-pages  
  
# ensure you are in the correct directory then,  
# remove all files from the old working tree  
$ git rm -rf
```

3. リポジトリのルートに `index.html` ファイルをします。

```
$ echo "Hello World" > index.html  
$ git add index.html  
$ git commit -a -m "First pages commit"
```

4. Githubにプッシュ。

```
$ git push origin gh-pages
```

`http(s)://<username>.github.io/<projectname>`でしいGithub Pagesサイトをみむことができます。

### GitHub ページのカスタムURLをする

レジストラからドメインがになります。

プロジェクトリポジトリの `gh-pages` ブランチまたは `username.github.io` リポジトリのメインブランチで、 **のドメインである** `www.yourdomain.com` のをつCNAMEファイルをします。

レジストラのドメインページで、ドメインをGitHub Webサイトにけます。2つのCNAMEレコードをします1つはルート・エイペックス@、もう1つはwww。どちらも `username.github.io` または `username.github.io/repository` しているがあり `username.github.io/repository`。DNSプロバイダが

ルートエイペックス@のALIASレコードをサポートしていないは、に192.30.252.153と192.30.252.154をすAレコードをします。

---

## リソース

[カスタムドメインのGitHub](#)

[スタックオーバーフローQA「GitHubプロジェクトページのカスタムドメイン」](#)

[Audrey Watters - GitHubを使ってWebプロジェクトをパワーアップするとその](#)

[Alex Cican - のウェブサイトをDropboxとGitHubにさせる](#)

[Treehouse - GitHubページを使ってあなたのウェブサイトをホストする](#)

[オンラインでGitHubページをむ https://riptutorial.com/ja/github/topic/3759/githubページ](https://riptutorial.com/ja/github/topic/3759/githubページ)

## 8: GitHub ボタンの

き

GitHubボタンとはですか GitHubボタンは、きなりリポジトリにユーザーをリダイレクトするWebサイトにできるボタンです。

クレジット

- [Recordit](#)でされたGif
- Snipping Toolでした
- なチュートリアルでされたコードエディタは[codepen.io](#)でした

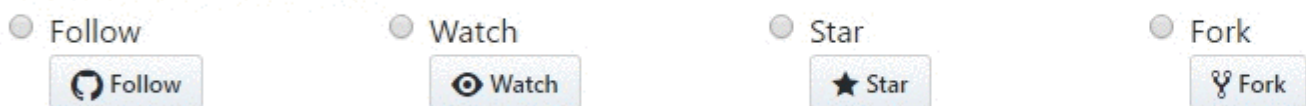
### Examples

フォローボタン

フォローボタンは、GitHubのユーザーページにリンクし、ユーザーにフォローするようにするボタンです。はのとおりで。

1. [githubにボタン](#)
2. 「フォロー」をクリック

### Choose a button



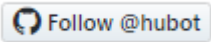
3. あなたのGitHubユーザーを "user" とかかれたボックスにきます。

### Button options

 / 

- Large button
- Show count
- Standard icon

4. 「きいボタン」、「」、および「アイコン」のボックスをしてボタンをカスタマイズします。



5. このコードを<head>か、コードの<body>にします。

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on GitHub">Follow @hubot</a>
```

6. カスタマイズされたボタンのレンダリングコードをコードにします。

browser tabs: buttons, A Pen by James Patrick K

address bar: Secure | https://buttons.github.io

taskbar: github:buttons, Material icons - Mater, karan/Projects: A list c, android-chrome-144, (137) Chill Study Beat

# GitHub:buttons

## Choose a button

- Follow
- Watch

アクションをするようにします。これらのボタンは、のようなをできます。

- リポジトリを
- リポジトリにスターをける
- リポジトリをフォークする
- リポジトリをダウンロードする
- リポジトリでをする

いくつかをするのはとおりです。

1. [githubにボタン](#)
2. するボタンタイプウォッチ、スター、フォーク、ダウンロード、またはをクリックします。


## Choose a button



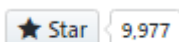
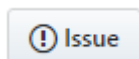
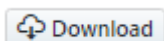
3. あなたのGitHubユーザを "user" とかかれたボックスにね、あなたのリポジトリを "repo"

## Button options

 / 

- Large button
- Show count 
- Standard icon

4. 「きいボタン」、「」、および「アイコン」のボックスをしてボタンをカスタマイズします



5. このコードを<head>か、コードの<body>にします。

```
<a class="github-button" href="https://github.com/hubot" aria-label="Follow @hubot on GitHub">Follow @hubot</a>
```

6. カスタマイズされたボタンのレンダリングコードをコードにします。

# GitHub:buttons

## Choose a button

- Follow
- Watch



## 9: GitHub リポジトリから1つのファイルをダウンロードする

### Examples

コマンドラインをしてパブリックリポジトリからファイルをリネームする

ここでは、GitHubのgitignoreリポジトリからNode.gitignoreファイルをし、のディレクトリにダウンロードして.gitignoreというにします。しいnode.jsプロジェクトをするかのなアクションです。

```
$ curl http://github.com/github/gitignore/raw/master/Node.gitignore -o .gitignore
```

あなたがダウンロードしたいファイルのURLをつける

1. リポジトリののファイルにする
2. [の]ボタンをクリックします
3. アドレスバーからURLをコピーする

GitHubのgitignoreリポジトリののをしてください <http://github.com/github/gitignore/raw/master/Node.gitignore>

あなたは、々のファイルをダウンロードするためにするURLをhtmlページをダウンロードすることからにすることができます。ブランチのに/ raw / rightというサブディレクトリをします。

オンラインでGitHubリポジトリから1つのファイルをダウンロードするをむ

<https://riptutorial.com/ja/github/topic/10898/github> リポジトリから1つのファイルをダウンロードする

# 10: あなたのウェブサイトにGitHubタイムライン/フィードをする

## Examples

あなたのウェブサイトにGitHubタイムライン/フィードをする

このドキュメントでは、GitHubのフィード/タイムラインをウェブサイトにするについてします。

はからです

<https://newtonjoshua.com>

### GitHubタイムライン

GitHubは、Atomのユーザーのタイムラインをします。

あなたのタイムラインはのことができます

[https://github.com/{{GitHub\\_username}}.atom](https://github.com/{{GitHub_username}}.atom)

<https://developer.github.com/v3/activity/feeds>をしてください。

### GoogleフィードAPI

フィードAPIをすると、Atom、RSS、MediaのいずれかのRSSフィードをJavaScriptのみでダウンロードできるため、わずかのJavaScriptでコンテンツやそのAPIとフィードをマッシュアップできます。これにより、ウェブサイトのにフィードをすることができます。

をしてください <https://developers.google.com/feed/v1/devguide>

JavaScript APIのみみフィードAPIのをするには、ウェブページのヘッダーにのスク립トをめます。

```
<script type="text/javascript" src="https://www.google.com/jsapi"></script>
```

に、google.loadモジュール、バージョン、パッケージをしてフィードAPIをみます。

```
<script type="text/javascript">
  google.load("feeds", "1");
</script>
```

フィードURLののようにgoogle.feeds.Feedをびすことができます

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
```

フィードをみむ `.load` コールバックは、Googleのサーバーからコンストラクタでされたフィードをダウンロードし、ダウンロードがしたらのコールバックをびします。

```
<script type="text/javascript">

function initialize() {
  feed.load(function(result) {
    if (!result.error) {
      var container = document.getElementById("feed");
      result.feed.entries.forEach(function (feed) {
        var feedTitle= feed.title;
        var feedLink = feed.link;
        var feedDate = formatDate(feed.publishedDate);
        var feedContent = formatContent(feed.content);

        // display the feed in your website
      });
    }
  });
}
google.setOnLoadCallback(initialize);

</script>
```

**onLoad**ハンドラのびし `setOnLoadCallback` コールバックは、このびしをむページがロードされるとびされるされたハンドラをするです。 `callback`は、されているドキュメントがロードされ、APIがな

```
<script type="text/javascript">
  google.setOnLoadCallback(initialize);
</script>
```

フィードエントリの `.setNumEntriesnum`は、このフィードによってみまれたフィードエントリのを `num`にします。デフォルトでは、`Feed`クラスは4つのエントリをみみます。

```
var feed = new google.feeds.Feed("https://github.com/{{GitHub_UserName}}.atom");
feed.setNumEntries(500);
```

これで、GitHubのフィード/タイムラインをフォーマットしてWebサイトにすることができます。

オンラインであなたのウェブサイトにはGitHubタイムライン/フィードをするをむ

<https://riptutorial.com/ja/github/topic/7479/あなたのウェブサイトにはgithubタイムライン-フィードをする>

# 11: カスタムGitHubラベルをするには

## Examples

カスタムGitHubラベルをする

できるだけプロセスをするためのなGIFがあります。



### Ahmad Awais

ahmadawais

Full Stack WordPress Dev — Front-end Fanatic — WP Core Contributor — TEDx Speaker — Open Sourcerer! <sup>100</sup>

Edit profile

Developer Program Member

@WPTie / WordPress

WP-Admin, TRAC; CORE

Overview

Repositories 262

Stars 1.1k

Followers 187

### Pinned repositories

#### WP Gulp

<sup>100</sup> % <sup>W</sup> → Use Gulp with WordPress. An advanced but portable Gulp front end and build workflow for you WordPress plugins and themes.

★ 203 ● JavaScript

#### WPCustomize

WP Customize component related boilerplate theme and features implementation.

★ 25 ● PHP

#### WP-API/WP-API

WP REST API - a JSON-based REST API for WordPress.

★ 3,565 ● PHP

#### \_child

\_child is a WordPr

★ 32 ● PHP

#### Sublime-WP-C

Sublime Package e

★ 22 ● PHP

#### WordPress/tw

Twenty Sixteen is a WordPress layout right sidebar that w has custom color c

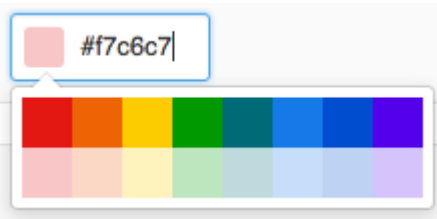
★ 340 ● CSS

ラベルは、カテゴリ、またはのなをすために、にしたり、をきしたりすることができます。

GitHubで、リポジトリのメインページにします。

1. リポジトリのにある[Issues]または[Pull requests]をクリックします。
2. Labelsボタンをクリックします。フィールドのに、Labelsをクリックします。
3. [しいラベル]をクリックしてしいラベルをするか、[]をクリックしてのラベルをします。

4. テキストボックスに、しいラベルをします。
5. カラーバーからラベルのをします。カラーバーのにある16をすることで、このをカスタマイズできます。



6. [ラベルの]をクリックして、しいラベルをします。

はそれがけてしいそれをアップアップしてください。

オンラインでカスタムGitHubラベルをするにはをむ <https://riptutorial.com/ja/github/topic/7159/カスタムgithubラベルをするには>

## 12: フォークされたリポジトリをする

- [GitHub Help](#) リモートにフォークをする
- [GitHub Help](#) フォークの
- [StackOverFlow](#) でのあるans

### Examples

フォークのリモートをし、フォークマスタブランチをさせます。

#### 1. フォークのリモートをする

```
$ cd my_local_repo
$ git remote add upstream https://github.com/ORIGINAL_OWNER/ORIGINAL_REPOSITORY.git
  # Specify a new remote upstream repository that will be synced with the fork
$ git remote -v
  # Verify the new upstream repository specified for my fork
```

#### 2. ローカルでフォークをする

```
$ cd my_local_repo
$ git fetch upstream
  # Fetch the branches and their respective commits from the upstream repository
  # Commits to master will be stored in a local branch, upstream/master
$ git checkout master
$ git merge upstream/master
  # Merge the changes from upstream/master into your local master branch
  # brings your fork's master branch into sync with the upstream repo
```

#### 3. のフォークをGithubにさせる

```
$ git push origin master
```

オンラインでフォークされたリポジトリをするをむ <https://riptutorial.com/ja/github/topic/3758/フォークされたリポジトリをする>

---

## 13: プルリクエスト

### Examples

プルリクエストをく

しいプルリクエスト

あなたがプルリクエストをしたいときはいつでもこれはのによって異なりますが、いつでもこれをうことができます、にんでGitHubにあなたのためにくのをさせることができますブランチにプッシュしたことをすアラートボックスの**Compare Pull Request**ボタン *CLONE*または*DOWNLOAD*としないでくださいをします。

それのは、ブランチのにある[ **New Pull Request** ]ボタンをします。



This repository

Search



maxcell / **example-so-documenta**

**<> Code**

! Issues **0**

🔗 Pull request

Repository used for Documentation in Sta

🔄 **2 commits**

Your recently pushed branches:

🔗 **example-branch** (less than a minute ago)

Branch: **example-branch** ▼

**New pull request**

This branch is 1 commit ahead of master.



**maxcell** committed on **GitHub** Update README

📄 **README.md**





するメッセージがされます。あなたは、あなたがマージしたいしいブランチをとっていることをかめたいだけです。BASEとは、どちらをえたいのか、それをするのかということです。したがって、このには、`master`にある`example-branch`と々はん`master`きまれました。


# Comparing changes



Choose two branches to see what's changed or to sta

 base: master ▾ ... compare: example-branch

 **Update README.md #1**  
No description available

 **1 commit**

 Commits on Aug 17, 2016

  **maxcell** Update README

 Showing **1 changed file** with **2 additions** and **0 de**

2    README.md

```
... @@ -1,2 +1,4 @@  
1 # example-so-documentation  
2 Repository used for Documentation in St
```

---

# 14:

## Examples

の

1. をするプロジェクトのGitHubページにします。
2. [**Issues**]をクリックします。
3. の[ ]をクリックします。
4. のタイトルをします。
5. のをしますログ、コードスニペットなどをむ
6. オプションするにをするには、プレビューをクリックします。
7. [しいをする]をクリックします。

オンラインでをむ <https://riptutorial.com/ja/github/topic/3757/>

## 15: データやきなファイルをする

### き

パスワードやSSHなどのデータをGitリポジトリにコミットするは、からすることができます。リポジトリのからなファイルをするには、`git filter-branch`コマンドまたはBFG Repo-Cleanerをできます。

1. あなたのいされたリポジトリからしたブランチは、にリベースして、マージしないようにしてください。1つのマージコミットは、されたのまたはをし、ページのをかけたばかりです。
2. しばらくして`git filter-branch`にしないがないとしていれば、のコマンドGit 1.8.5ををって、ローカルリポジトリのすべてのオブジェクトのをし、ガベージコレクションをできます。

```
git for-each-ref --format = 'refnameをする' refs / original | git update-ref --stdin
```

```
git reflog expire --expire = now --all
```

```
git gc --prune = now
```

## Examples

### filter-branchをする

```
git filter-branch --force --index-filter \  
'git rm --cached --ignore-unmatch PATH-TO-YOUR-FILE-WITH-SENSITIVE-DATA' \  
--prune-empty --tag-name-filter cat -- --all
```

なデータをむファイルを.gitignoreにして、ってコミットしないようにします。

```
echo "YOUR-FILE-WITH-SENSITIVE-DATA" >> .gitignore  
git add .gitignore  
git commit -m "Add YOUR-FILE-WITH-SENSITIVE-DATA to .gitignore"
```

### ローカルリポジトリをGitHubにプッシュする

```
git push origin --force --all
```

あなたのタグきリリースからファイルをするには、Gitタグをにすがあります

```
git push origin --force --tags
```

### BFGレポクリーナーの

BFG Repo Cleanerは、git filter-branchのわりです。これは、ソースからコンパイルされたバイナリのようにコミットされたデータやきなファイルをするためにできます。これはScalaでかれています。

プロジェクトのウェブサイト [BFG Repo Cleaner](#)

Java Runtime EnvironmentJava 7 - BFG v1.12.3はJava 6をサポートするのバージョン Scalaライブラリとのすべてののは、ダウンロードなjarファイルにりたたまれています。

## データをむファイルをする

```
bfg --delete-files YOUR-FILE-WITH-SENSITIVE-DATA
```

オンラインでデータやきなファイルをするをむ <https://riptutorial.com/ja/github/topic/8170/データやきなファイルをする>

## クレジット

S. No		Contributors
1	githubをいめる	<a href="#">BadAllOff</a> , <a href="#">BrechtDeMan</a> , <a href="#">Community</a> , <a href="#">H. Pauwelyn</a> , <a href="#">Hamzawey</a> , <a href="#">Hugo</a> , <a href="#">intboolstring</a> , <a href="#">Kronos</a> , <a href="#">Mateusz Piotrowski</a> , <a href="#">Minhas Kamal</a> , <a href="#">Nicholas Qiao</a> , <a href="#">rpadovani</a>
2	Gistをう	<a href="#">Kronos</a> , <a href="#">tehp</a>
3	Gitflowをとってする	<a href="#">Derek Liu</a>
4	GitHubからのリポジトリのクローン	<a href="#">demonplus</a> , <a href="#">geek1011</a> , <a href="#">Hamzawey</a> , <a href="#">James Kerrane</a> , <a href="#">Mateusz Piotrowski</a>
5	GitHubデスクトップ	<a href="#">creyD</a>
6	GitHubのバックアップ	<a href="#">geek1011</a>
7	GitHubページ	<a href="#">BrechtDeMan</a> , <a href="#">geek1011</a> , <a href="#">Mono</a>
8	GitHubボタンの	<a href="#">James Kerrane</a>
9	GitHubリポジトリから1つのファイルをダウンロードする	<a href="#">ownsourcing dev training</a>
10	あなたのウェブサイトにGitHubタイムライン/フィードをする	<a href="#">Hugo</a> , <a href="#">Newton Joshua</a>
11	カスタムGitHubラベルをするには	<a href="#">Ahmad Awais</a>
12	フォークされたリポジトリをする	<a href="#">Derek Liu</a>
13	プルリクエスト	<a href="#">Maxcell</a>
14		<a href="#">geek1011</a> , <a href="#">Hamzawey</a> , <a href="#">SuperBiasedMan</a>
15	データやきなファイルをする	<a href="#">Gautam Krishna R</a> , <a href="#">Kronos</a>