

 무료 전자 책

배우기

github

Free unaffiliated eBook created from
Stack Overflow contributors.

#github

.....	1
1: github	2
.....	2
Examples.....	2
.....	2
.....	2
.....	2
.....	2
.....	2
.....	2
README	2
README -	2
.....	2
.....	2
.....	3
.....	3
.....	3
.....	3
.....	3
.....	3
.....	3
GitHub	4
.....	4
.....	5
.....	5
.....	5
.....	6
.....	7
.....	7
.....	7
.....	7
.....	7

.....	8
.....	8
SHA	8
.....	8
2: Gitflow	10
.....	10
.....	10
.....	10
Examples.....	10
5	10
3: GitHub	12
.....	12
Examples.....	12
.....	12
.....	12
.....	13
(:).....	13
4: GitHub	15
Examples.....	15
.....	15
5: GitHub	16
.....	16
.....	16
Examples.....	16
.....	16
.....	18
6: GitHub	21
Examples.....	21
.....	21
URL	21

7: GitHub	22
Examples	22
.....	22
Git	22
GitHub URL	22
.....	22
8: GitHub	24
.....	24
Examples	24
.....	24
9:	26
Examples	26
.....	26
.....	26
h21	27
10: GitHub ?	29
Examples	29
GitHub !	29
11:	31
Examples	31
.....	31
12:	32
.....	32
.....	32
Examples	32
.....	32
.....	32
13: GitHub /	33
Examples	33
GitHub /	33
14:	35
.....	

35

..... 35

Examples.....35

-35

BFG Repo Cleaner35

..... 35

..... 35

15:**37**

..... 37

Examples.....37

().....37

.....**38**

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

1: github

github , .

github . github .

Examples

GitHub Git . , GitHub .

-
- GitHub
 - !

Git / GitHub . GitHub GUI . [GitHub Desktop](#) .

. . .

1. .
2. " .
3. "New" .

1. [git](#) ().
2. GUI .
3. [github](#) .
4. + .
5. .
6. :

NB .

README

README.md GitHub README.rdoc . README.md rdoc .

README -

. , , (: ,) .

(). .

. , , , .

, gif , .

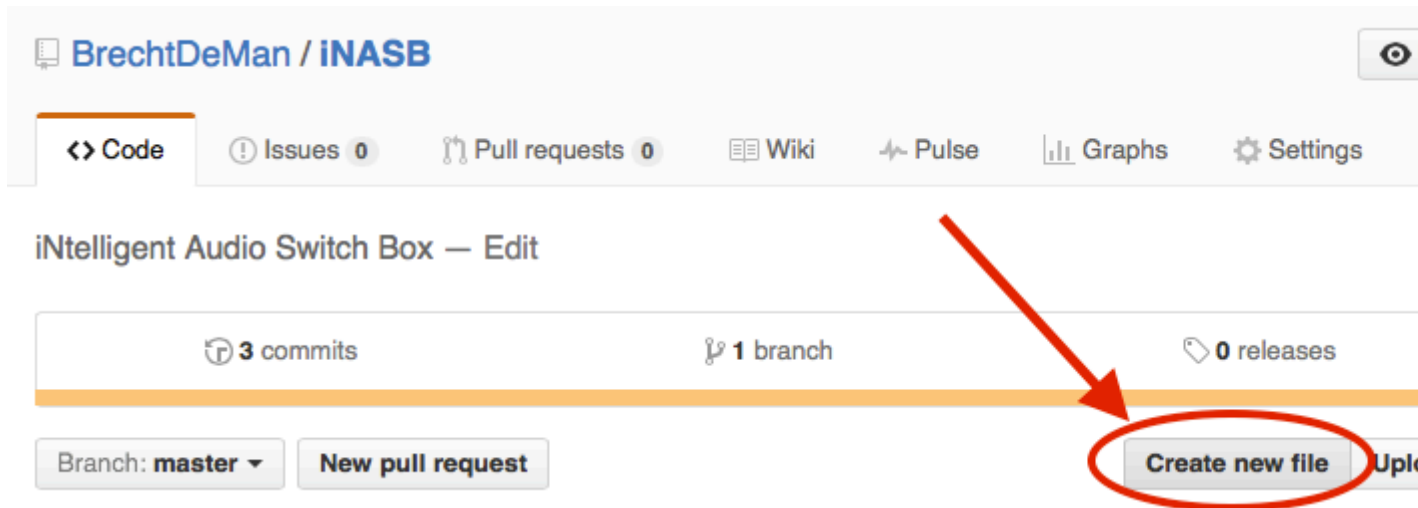
,

(: , ,) .

. . .

GitHub / .

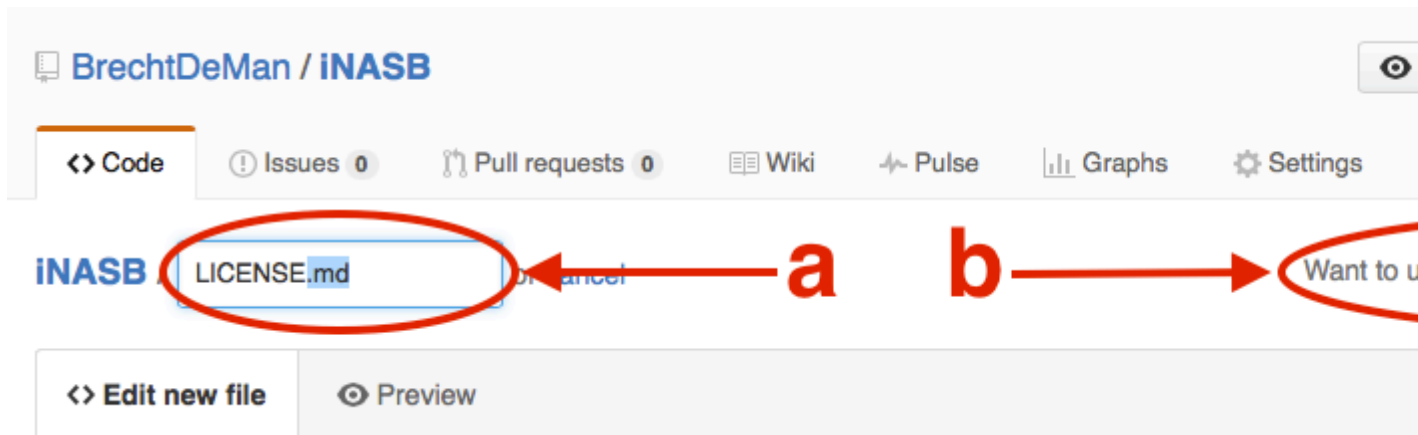
1. ' ' .



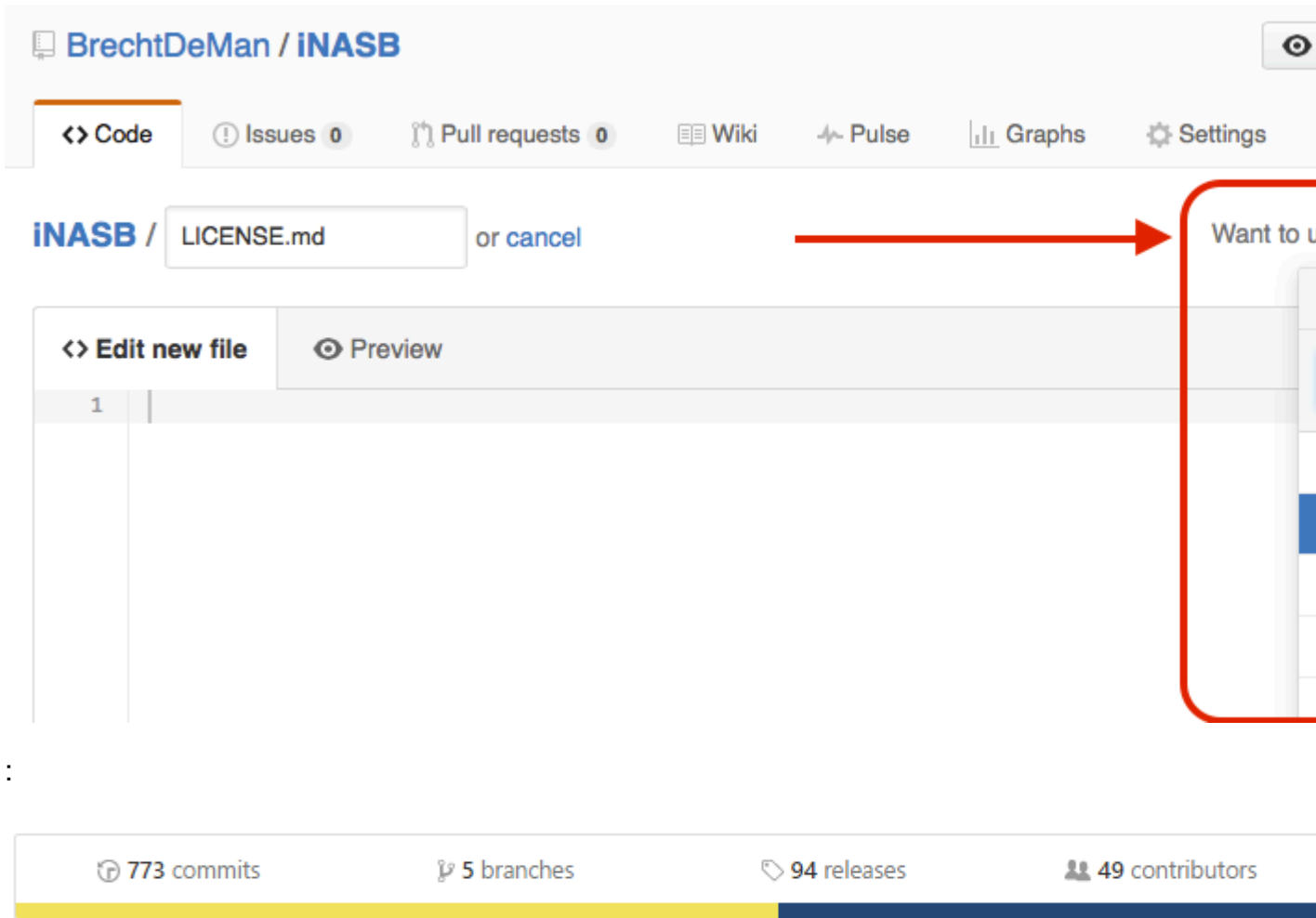
2. :

1. LICENSE.md LICENSE.txt .

2. ? .



3. .



4. :

Q & A - Github

GitHub

GitHub [Markdown](#) .

```
# 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~~

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

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

```
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");
}
```

```
> 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.

<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")
```



- [x] completed item
- [] incomplete item

completed item

incomplete item

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



GitHub emojis [Emoji Cheat Sheet](#) .

SHA

SHA1 GitHub .

```
e7909ea4fbb162db3f7f543d43c30684a3fb745f
```

Write

Preview

e7909ea

.

/ # .

github : <https://riptutorial.com/ko/github/topic/1214/github->

2: Gitflow

- `git flow <subcommand>`
-
- `git flow [| |][|]`



- `gitflow`
-

Examples

5

Gitflow

1.

```
$ 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.

```
$ 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.

```
$ 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.

```
$ 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'
```

Gitflow : <https://riptutorial.com/ko/github/topic/6231/gitflow->

3: GitHub

GitHub Desktop ?

GitHub Desktop , GitHub Desktop (), Git Windows MacOS .

git (GitHub) . .

Examples

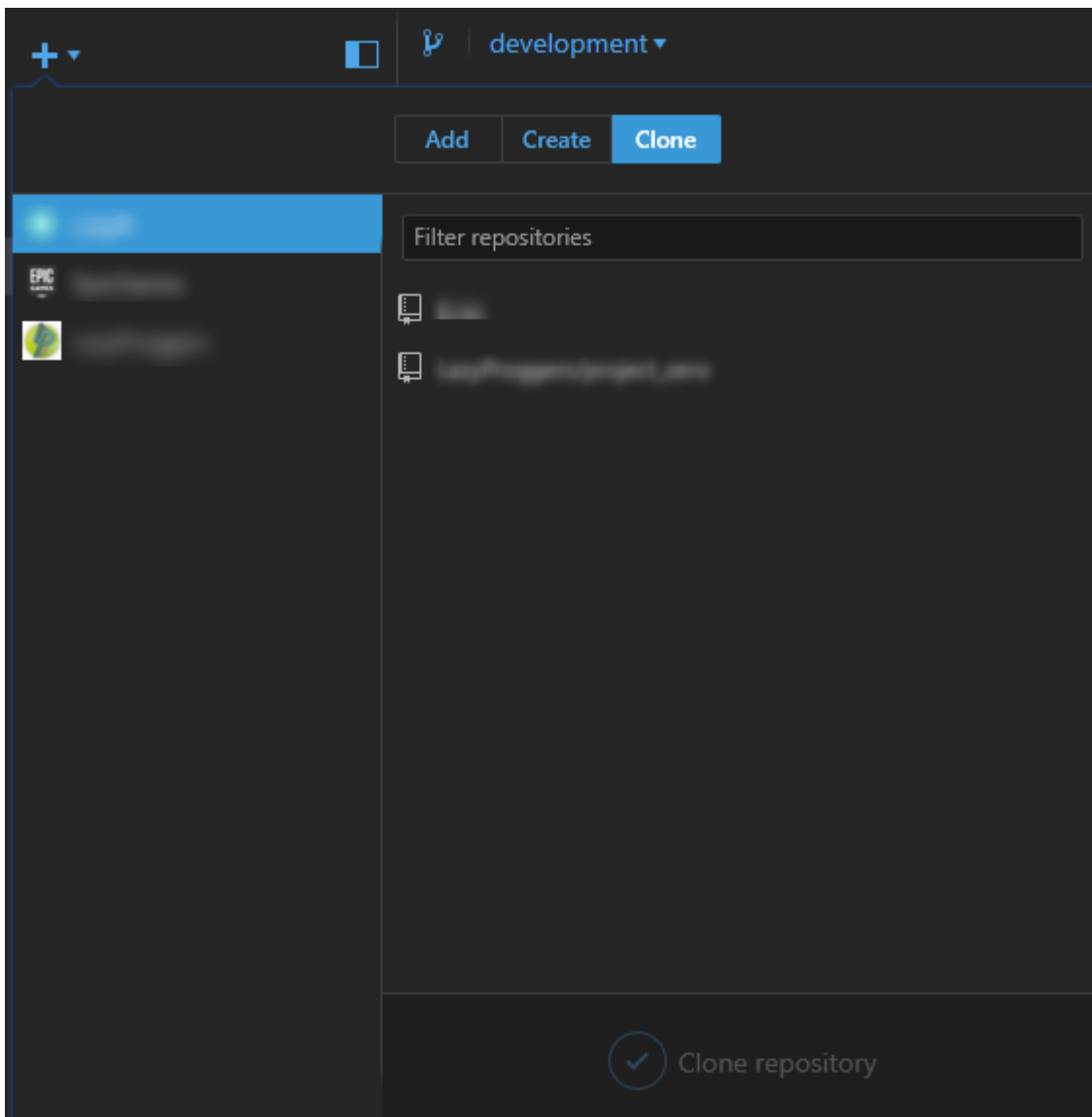
MacOS Windows . . .

GitHub . . .

: GitHub Desktop Git . .

GitHub Desktop : "() (repos).) . .

: ()() .



. , git checkout BRANCHNAME sync () git checkout BRANCHNAME .

2 . ()



() File --> New Branch .

(:)

()

. GitHub Desktop sync .

. Commit to YOURCURRENTBRANCH . .

⋮, .

Sync Push, Pull Checkout .

GitHub : <https://riptutorial.com/ko/github/topic/10023/github->

4: GitHub

Examples

GitHub username .

```
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
```

100 .

GitHub : <https://riptutorial.com/ko/github/topic/3760/github->

5: GitHub

GitHub ? GitHub !

:

- [Recordit](#) Gif
-
- [codepen.io](#).

Examples

GitHub . . .

1. `github` : .
2. "" .

Choose a button

Follow Watch Star Fork

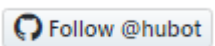
3. GitHub ": user"


Button options

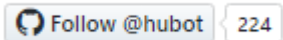
/

- Large button
- Show count
- Standard icon

4. "", " " " " " :







5. `<head>` `<body>` .

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

6. .

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

-
-
-
-

1. `github :`
2. `(, , ,)`.

Choose a button


Follow Watch Star Fork

Follow Watch Star Fork

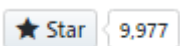
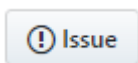
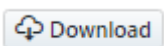
3. GitHub `": user"` `": repo"`.

Button options

`:user` / `:repo`

- Large button
- Show count 
- Standard icon

4. `"", " " " " "` :



5. `<head>` `<body>` .

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

6. .

GitHub:buttons

Choose a button

- Follow
- Watch

6: GitHub

Examples

GitHub gitignore Node.gitignore .gitignore . node.js .

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

URL

- 1.
2. "
3. URL

GitHub gitignore . <http://github.com/github/gitignore/raw/master/Node.gitignore>

URL html URL . / raw / right .

GitHub : <https://riptutorial.com/ko/github/topic/10898/github---->

7: GitHub

Examples

1. GitHub .
- 2.
3. .
4. GitHub Pages "Launch Automatic Page Generator" .
- 5.

Git

1. .
2. gh-pages

```
$ 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. index.html .

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

4. Github .

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

http(s)://<username>.github.io/<projectname> Github Pages
http(s)://<username>.github.io/<projectname>

GitHub URL

.

```
gh-pages username.github.io www.yourdomain.com CNAME .
```

```
GitHub . CNAME ( (@) www). username.github.io username.github.io/repository . DNS  
(@) ALIAS 192.30.252.153 192.30.252.154 A .
```

[GitHub](#)

[Q & A : "GitHub"](#)

[Audrey Watters - GitHub](#) :

[Alex Cican - Dropbox](#) [GitHub](#)

[Treehouse - GitHub](#)

[GitHub](#) : <https://riptutorial.com/ko/github/topic/3759/github->

8: GitHub

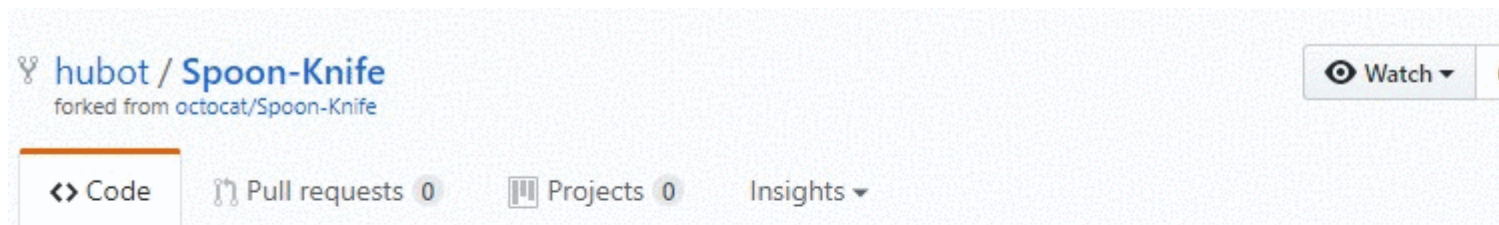
- `git clone github.com/username/repository`

Examples

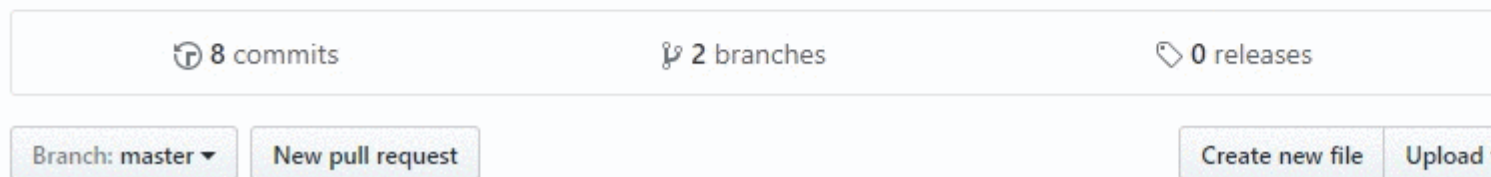
1. (: <https://github.com/username/repo>).



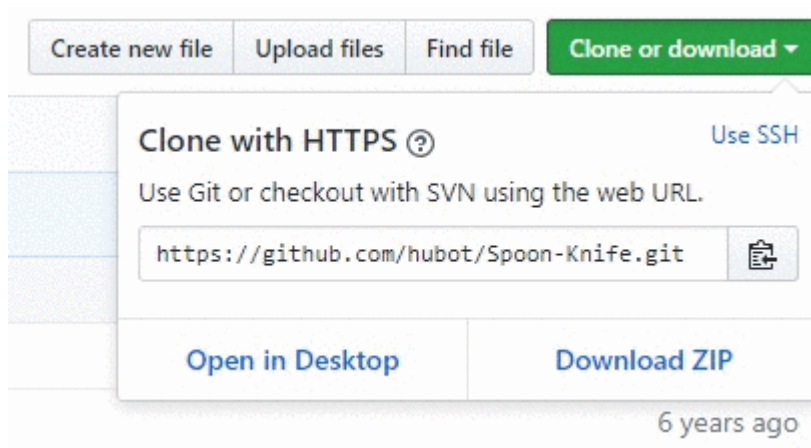
2. `clone` .



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



3. URL (<https://github.com/username/repo.git>).



4. .
5. .
6. : `git clone <copied_url_from_step_3>`
7. .

8. .

<repo_name> ...

: : 10, .

: : 100 % (8/8), .

: 10 (1), 10 (1)

: 100 % (10/10) .

GitHub : <https://riptutorial.com/ko/github/topic/3761/github-->

9:

Examples

([New Pull Request](#)), GitHub ([New Pull Request](#)).

New Pull Request .



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**



Update README.md

Write

Preview

Leave a comment

Attach files by dragging & dropping, [selecting](#)

M↓ Styling with Markdown is supported

: <https://riptutorial.com/ko/github/topic/5761/>

10: GitHub ?

Examples

GitHub !

GIF .



Ahmad Awais

ahmadawais

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

[Edit profile](#)

Developer Program Member

@WPTie / WordPress

WP-Admin, TRAC; CORE

Overview Repositories **262** Stars **1.1k** Followers **187**

Pinned repositories

≡ **WPGulp**
¹⁰⁰ % → 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 WordPress
★ 32 ● PHP

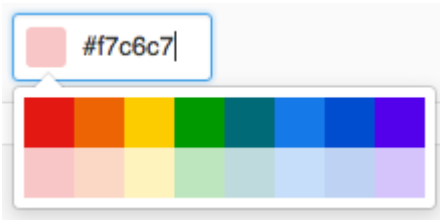
≡ **Sublime-WP-C**
Sublime Package
★ 22 ● PHP

≡ **WordPress/tw**
Twenty Sixteen is a WordPress layout right sidebar that w has custom color c
★ 340 ● CSS

GitHub .

1. Issues Pull requests .
2. .
3. .
4. .

5. . 16 .



6. .

. upvote.

GitHub ? : <https://riptutorial.com/ko/github/topic/7159/-github----->

11:

Examples

1. GitHub .
2. .
3. .
4. .
5. (,)
6. : .
7. .

: <https://riptutorial.com/ko/github/topic/3757/>

12:

Gists . , . <https://gist.github.com> .

(Git repository). . CodeMirror .

:

GitHub .

Gists . , .

: . GitHub .

Discover . . .

. , , .

GitHub . . GitHub.com GitHub Enterprise GitHub . URL .

Examples

.
:

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

GitHub . .

JSON :

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

: [https://riptutorial.com/ko/github/topic/7978/-](https://riptutorial.com/ko/github/topic/7978/)

13: GitHub /

Examples

GitHub /

GitHub / .

: :

<https://newtonjoshua.com>

GitHub :

GitHub Atom .

.

https://github.com/{{GitHub_username}}.atom

: <https://developer.github.com/v3/activity/feeds>

Google API :

API JavaScript Atom, RSS Media RSS JavaScript API . .

: <https://developers.google.com/feed/v1/devguide>

JavaScript API : API .

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

google.load(, ,) API.

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

URL : google.feeds.Feed() .

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

: .load() Google .

```
<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 : `setOnLoadCallback ()` . callback API .

```

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

```

: .setNumEntries (num) num . Feed .

```

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

```

GitHub / .

GitHub / : <https://riptutorial.com/ko/github/topic/7479/--github----->

14:

SSH Git . git filter-branch BFG Repo-Cleaner .

1. () (rebase) . .

2. git filter-branch (Git 1.8.5).

```
git for-each-ref --format = ' % (refname)'refs / original | git update-ref --stdin
```

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

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

Examples

-

```
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
```

.gitignore .

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

GitHub .

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

Git .

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

BFG Repo Cleaner

BFG Repo cleaner git filter-branch . . .

: [BFG Repo Cleaner](#)

Java Runtime Environment (Java 7 - BFG v1.12.3 Java 6). jar .

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


: <https://riptutorial.com/ko/github/topic/8170/----->

15:

- [GitHub Help](#) :
- [GitHub Help](#) :
- [StackOverFlow](#) ans

Examples

()

1.

```
$ 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.

```
$ 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. Github

```
$ git push origin master
```

: <https://riptutorial.com/ko/github/topic/3758/--->

| S. No | | Contributors |
|-------|----------|--|
| 1 | github | BadAllOff , BrechtDeMan , Community , H. Pauwelyn , Hamzawey , Hugo , intboolstring , Kronos , Mateusz Piotrowski , Minhas Kamal , Nicholas Qiao , rpadovani |
| 2 | Gitflow | Derek Liu |
| 3 | GitHub | creyD |
| 4 | GitHub | geek1011 |
| 5 | GitHub | James Kerrane |
| 6 | GitHub | ownsourcing dev training |
| 7 | GitHub | BrechtDeMan , geek1011 , Mono |
| 8 | GitHub | demonplus , geek1011 , Hamzawey , James Kerrane , Mateusz Piotrowski |
| 9 | | Maxcell |
| 10 | GitHub ? | Ahmad Awais |
| 11 | | geek1011 , Hamzawey , SuperBiasedMan |
| 12 | | Kronos , tehp |
| 13 | GitHub / | Hugo , Newton Joshua |
| 14 | | Gautam Krishna R , Kronos |
| 15 | | Derek Liu |