

# 깃허브 치트 시트

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

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

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

## 목록

- GitHub
  - Ignore Whitespace
  - Adjust Tab Space
  - Commit History by Author
  - Cloning a Repository
  - Branch
    - Compare all Branches to Another Branch
    - Comparing Branches
    - Compare Branches across Forked Repositories
  - Gists
  - Git.io
  - Keyboard Shortcuts
  - Line Highlighting in Repositories
  - Closing Issues via Commit Messages
  - Cross-Link Issues
  - Locking Conversations
  - CI Status on Pull Requests
  - Syntax Highlighting in Markdown Files
  - Emojis
  - Images/GIFs
    - Embedding Images in GitHub Wiki
  - Quick Quoting
  - Pasting Clipboard Image to Comments
  - Quick Licensing
  - Task Lists
    - Task Lists in Markdown Documents
  - Relative Links
  - Metadata and Plugin Support for GitHub Pages
  - Viewing YAML Metadata in your Documents
  - Rendering Tabular Data
  - Rendering PDF
  - Revert a Pull Request
  - Diffs
    - Rendered Prose Diffs
    - Diffable Maps

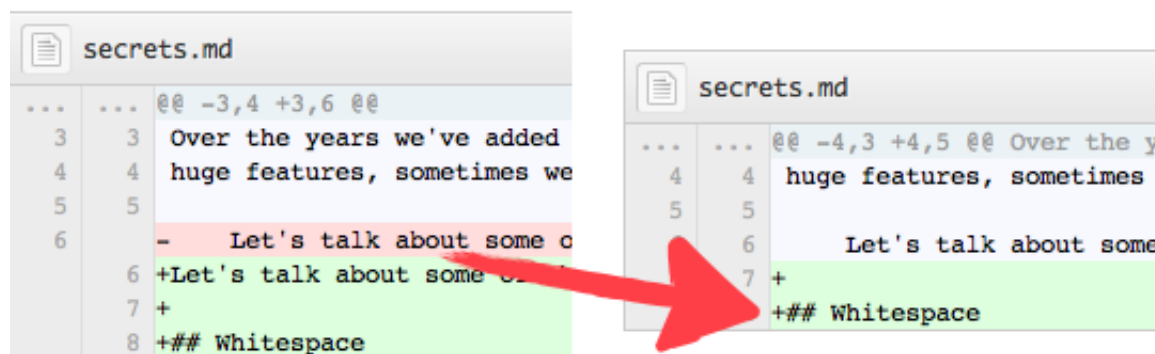
- Expanding Context in Diffs
  - Diff or Patch of Pull Request
  - Rendering and diffing images
- Hub
- Contributing Guidelines
- Octicons
- GitHub Resources
  - GitHub Talks
- Git
  - Remove All Deleted Files from the Working Tree
  - Previous Branch
  - Stripspace
  - SSH keys
  - Checking out Pull Requests
  - Empty Commits
  - Styled Git Status
  - Styled Git Log
  - Git Query
  - Git Grep
  - Merged Branches
  - Fixup and Autosquash
  - Web Server for Browsing Local Repositories
  - Git Configurations
    - Aliases
    - Auto-Correct
    - Color
  - Git Resources
    - Git Books

## GitHub

---

### Ignore Whitespace

?w=1 를 diff URL에 추가하면 변경된 코드만 볼 수 있도록, 공백만 바뀐 수정을 제거할 수 있습니다.



깃허브의 비밀을 더 읽어 보세요.

### Adjust Tab Space

diff나 파일 URL에 ?ts=4 를 덧붙이면 탭 문자의 크기를 기본값인 8대신 4공백으로 보여줍니다. ts 뒤의 숫자는 설정에 맞게 변경

하실 수 있습니다. gist나 raw파일 보기에는 적용 되지 않습니다만, 크롬 익스텐션이나 오페라 익스텐션으로 자동화 할 수 있습니다.

여기에 있는 Go 소스 파일은 ?ts=4 를 붙이기 전에는 이렇습니다.

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 }
11
```

그리고 ?ts=4 를 붙인 다음에는 이렇게 됩니다. —allow-empty

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 }
```

## Commit History by Author

특정 커미터가 한 모든 커밋을 보고 싶다면 URL에 ?author={user} 를 추가하세요.

<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

<>

커밋 뷰간의 차이에 대해 더 읽어보세요.

## Cloning a Repository

저장소를 클론할 때, 맨 뒤의 `.git` 은 생략할 수 있습니다.

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

깃 `clone` 커맨드에 대해 더 읽어 보세요.

## Branch


### Compare all Branches to Another Branch

저장소의 [브랜치](#) 페이지는, 커밋 버튼의 뒤에 있습니다.

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

... 메인 브랜치에 머지되지 않은 모든 브랜치의 리스트를 볼 수 있습니다.

여기에서 버튼 클릭으로 비교 페이지나 브랜치 삭제를 하실 수 있습니다.

 rails / rails

Watch 1,732

Unstar 24,618

Fork 9,521

OverviewActiveStaleAll branches

Search branches...

Default branch

master

Updated an hour ago by fxn

✓

Default

Active branches

<div>4-2-stable</div> Updated 2 hours ago by sgrif	✗	<div>1147269</div>	<div>Compare</div>
<div>4-1-stable</div> Updated 4 days ago by rafaelfranca	✓	<div>6892672</div>	<div>Compare</div>
<div>fix_nested_transactions_for_re...</div> Updated 8 days ago by chancancode	✗	<div>1493</div>	<div>Compare</div>
<div>3-2-stable</div> Updated 12 days ago by rafaelfranca	✗	<div>224781757</div>	<div>Compare</div>
<div>4-0-stable</div> Updated 12 days ago by rafaelfranca	✓	<div>119741081</div>	<div>Compare</div>

View more active branches >

### Comparing Branches

깃허브에서 브랜치 비교를 하시려면, URL을 이런 식으로 바꾸세요.

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

{range} 는 `master...4-1-stable` 식으로 적습니다.

예를 들어

```
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} 는 이렇게 적을 수도 있습니다.

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

날짜는 YYYY-MM-DD 형식으로 적습니다

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
------------	---	-----------

브랜치는 diff 나 patch 뷰로도 확인할 수 있습니다.

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

시간으로 커밋을 비교하는 법에 대해 더 읽어 보세요.

## Compare branches across forked repositories

포크된 저장소간의 브랜치를 비교하려면 URL을 이렇게 변경하세요.

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

예를 들면

```
https://github.com/rails/rails/compare/byroot:idempotent-counter-caches...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

	PostgreSQL, Support for materialized views. [Dave Lee & Yves Senn] ...	✔ def6071
	We can conditional define the tests depending on the adapter or ...	✖ ee36af1
	Merge pull request #14565 from rajcybage/conditional_test_cases ...	✔ c82483a
	DRY AS::SafeBuffer a bit using existing helper	✔ 8482895
	Fixed small documentation typo ...	✖ 8ae3f24
	Merge pull request #14568 from alex88/patch-1 ...	✔ 3bcc51a

## Gists

Gists는 본격적인 저장소를 만들지 않고, 작은 코드 조각과 일할 수 있는 쉬운 방법입니다.

The screenshot shows a GitHub Gist page. At the top, the GitHub Gist logo is on the left, a search bar in the center, and 'Discover Gists' on the right. Below the header, the gist is titled 'tiimgreen / app.rb' and is marked as 'PUBLIC'. It was 'Created 4 days ago'. On the right side of the header, there are buttons for 'Settings', 'Star' (0), and 'Fork' (0). The main content area shows a simple Ruby program: `puts 'Hello World'`. To the left of the code, there are tabs for 'Gist Detail' and 'Revisions' (1). Below the code, there are buttons for 'Download Gist', 'Clone this gist' (with a URL: `https://gist.github.com`), 'Embed this gist' (with an embed script), and 'Link to this gist' (with a URL: `https://gist.github.com`). On the right side of the main content, there is a comment section. A comment by 'rafalchmiel' says 'Wow, dat is some engineering.' and was posted '2 days ago'. Below the comment, there is a 'Write' tab and a 'Preview' tab. The 'Write' tab is active, showing a text area with the placeholder 'Leave a comment'. To the right of the text area, it says 'Comments are parsed with GitHub Flavored Markdown'. At the bottom right of the comment section, there is a green 'Add Comment' button.

이렇게 Gist URL뒤에 `.pibb` 를 넣으면 다른 사이트에 넣을수 있는 *HTML* *온리* 버전을 만들 수 있습니다.

Gists는 저장소처럼 취급할 수 있고 클론도 됩니다.

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

The screenshot shows a terminal window titled 'tim — bash — 80x17'. The terminal output is as follows:  
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\$

이는 Gists에서도 수정하고 업데이트를 푸쉬할 수 있다는 의미입니다.

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



하지만, Gists는 디렉토리를 지원하지 않습니다. 모든 파일은 저장소의 루트에 넣을 필요가 있습니다. *Gist를 만드는 법에 대해 더 읽어 보세요.*

## Git.io

Git.io는 깃허브를 위한 간단한 URL 단축기입니다.



순수 HTTP에서 Curl으로 사용하실 수도 있습니다.

```
$ 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/...
```

*Git.io에 대해 더 읽어 보세요.*

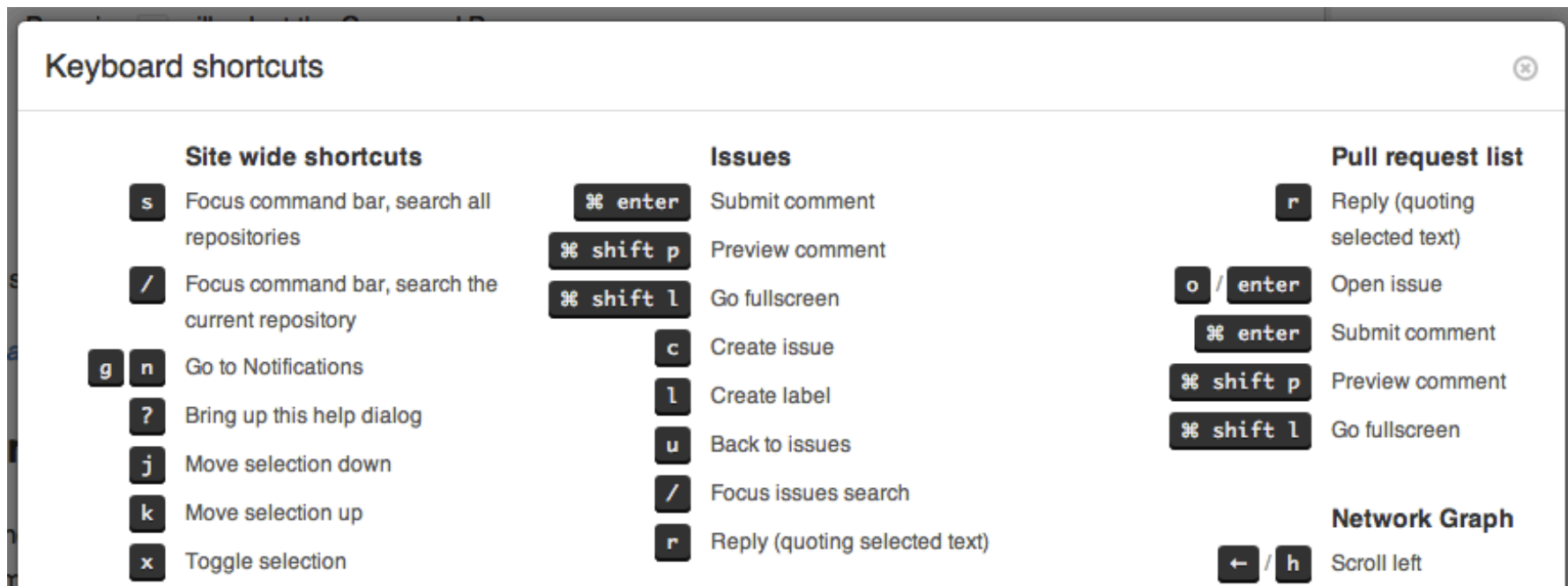
## Keyboard Shortcuts

저장소 페이지에서 단축키를 사용하면 쉽게 이동할 수 있습니다.

- **t** 를 누르면 파일 탐색 페이지로 이동합니다.
- **w** 를 누르면 브랜치 선택 페이지로 이동합니다.
- **s** 를 누르면 지금 저장소의 검색창이 포커스됩니다. 백스페이스를 누르면 채워져 있는 “This repository”를 지워 GitHub전체에서 검색할 수 있게 됩니다.
- **l** 를 누르면 있는 이슈의 라벨을 수정할 수 있습니다.
- **파일을 보고 있을 때** (예를 들어 `https://github.com/tiimgreen/github-cheat-sheet/blob/master/README.md`) **y** 를 누르면 URL을 지금 보고 있는 페이지를 사실상 고정하도록 합니다. 코드가 바뀐다고 해도 이번에 본 내용을 다시 볼 수 있습니다.

지금 페이지에서 쓸 수 있는 모든 단축키를 보시려면 **?** 를 누르세요.





사용할 수 있는 검색 문법에 대해 더 읽어 보세요.

## Line Highlighting in Repos

코드 파일 URL의 끝에 #L52 를 추가하거나 줄 번호를 클릭하면 그 줄 번호를 하이라이트합니다.

#L53-L60 처럼 범위지정도 가능합니다. 범위를 지정하려면 shift 키를 누른 채 두 라인을 클릭하면 됩니다.

[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    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
```


## Closing Issues with Commits


어떤 커밋이 이슈를 고쳤다면, fix/fixes/fixed , close/closes/closed , resolve/resolves/resolved 를 키워드로 해당 이슈가 마스터 브랜치에 커밋 될 때 닫을 수 있습니다.

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

이렇게 하면 이슈를 닫고 클로징 커멘트로 참조하게 합니다.

 **tiimgreen** closed this issue from a commit 9 days ago

 **Fix screwup, fixes #12** 36f4317

 **tiimgreen** closed this in **36f4317** 9 days ago

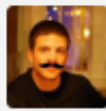
커밋 메시지로 이슈 닫기에 대해 더 읽어 보세요.


## Cross-Link Issues

같은 저장소의 다른 이슈를 링크하기 원한다면, # 뒤에 이슈 번호만 입력하시면 자동으로 링크됩니다.

다른 저장소의 이슈를 링크하고 싶다면 {user}/{repo}#이슈\_번호 로 할 수 있습니다. (예 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**

## Locking Conversations


풀 리퀘스트와 이슈는 레포지터리의 주인이나 협업자에 의해 잠글 수 있습니다.

 **Parsed as Markdown**

 **Edit in fullscreen**


 **Lock issue**

이는 프로젝트의 협업자가 아닌 사람은 더 이상 커멘트를 달 수 없게된다는 말입니다.



Write

Preview



This conversation has been locked and limited to collaborators.

대화 잠그기에 대해 더 읽어 보세요.

## CI Status on Pull Requests

정확히 설정되었다면, 풀 리퀘스트를 받거나 풀 리퀘스트에 새로운 커밋을 할 때 마다, [Travis CI](#)가 풀 리퀘스트를 빌드 할 것입니다.

Travis CI 시작하기를 좀 더 읽어보세요.

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

커밋 상태 API에 대해 좀 더 읽어보세요.

## Syntax Highlighting in Markdown Files

마크다운에서 예를 들어, 루비 코드를 선택스 하이라이트 하려면 이렇게 합니다.

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

이렇게 보입니다.

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

깃허브는 [Linguist](#)를 사용해 언어를 감지하고 선택스를 하이라이트합니다. 언어 YAML 파일을 정독하시면 어떤 키워드가 유효한지 확인하실 수 있습니다.

*깃허브 Flavored 마크다운에 대해 더 읽어 보세요.*

## Emojis

에모지는 풀 리퀘스트, 이슈, 커밋 메시지, 저장소 설명등에 :에모지의\_이름: 으로 넣을 수 있습니다.

깃허브에서 사용 가능한 에모지의 전 목록은 [emoji-cheat-sheet.com](#) 나 [scotch-io/All-Github-Emoji-Icons](#)에서 확인하실 수 있

습니다.

깃허브에서 많이 사용하는 에모지 탑 5위는 이렇습니다.

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

## Images/GIFs

커맨트나 README등등에 이미지와 GIF를 넣을 수 있습니다.

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

저장소에 있는 이미지는 직접 불러서 사용할 수 있습니다.

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



모든 이미지는 GitHub에서 케쉬합니다. 그래서 호스트가 죽어도 이미지는 여전히 남습니다.

## Embedding Images in GitHub Wiki

이미지를 위키 페이지에넣는 방법은 여럿 있습니다. 위에 보이는 일반 마크다운 문법도 있지만, 이미지에 높이와 넓이를 지정할 수 있는 문법도 있습니다.

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

이렇게 보입니다.

## Home (Preview)



### README not found

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

## Quick Quoting

스레드에 답글들 때 전에 누가 말한 어떤 내용을 인용하고 싶다면, 텍스트를 하이라이트하고 `r` 을 누르세요. 이렇게 하면 내용을 텍스트 박스에 인용 포맷으로 복사해 줍니다.



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

빠른 인용에 대해 더 읽어 보세요.

## Pasting Clipboard Image to Comments

(크롬에서만 동작합니다)

스크린샷을 찍고 클립보드에 있는 경우 (mac: `cmd-ctrl-shift-4` ), 간단히 이미지를 커멘트 섹션에 붙여넣기( `cmd-v` / `ctrl-v` )할 수 있고 이는 자동으로 깃허브에 업로드 됩니다.

Issues

Pull requests

Labels

Milestones

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



Clipboard image paste

Write

Preview

Markdown supported

I think this is a huge protip. Capture screenshot (region) to clipboard. Then cmd-v it.



*issue attachments에 대해 더 읽어 보세요.*

## Quick Licensing

저장소를 만들 때, 깃허브는 만들어진 저작권을 추가할 수 있는 옵션을 제공합니다.

### ☐ 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

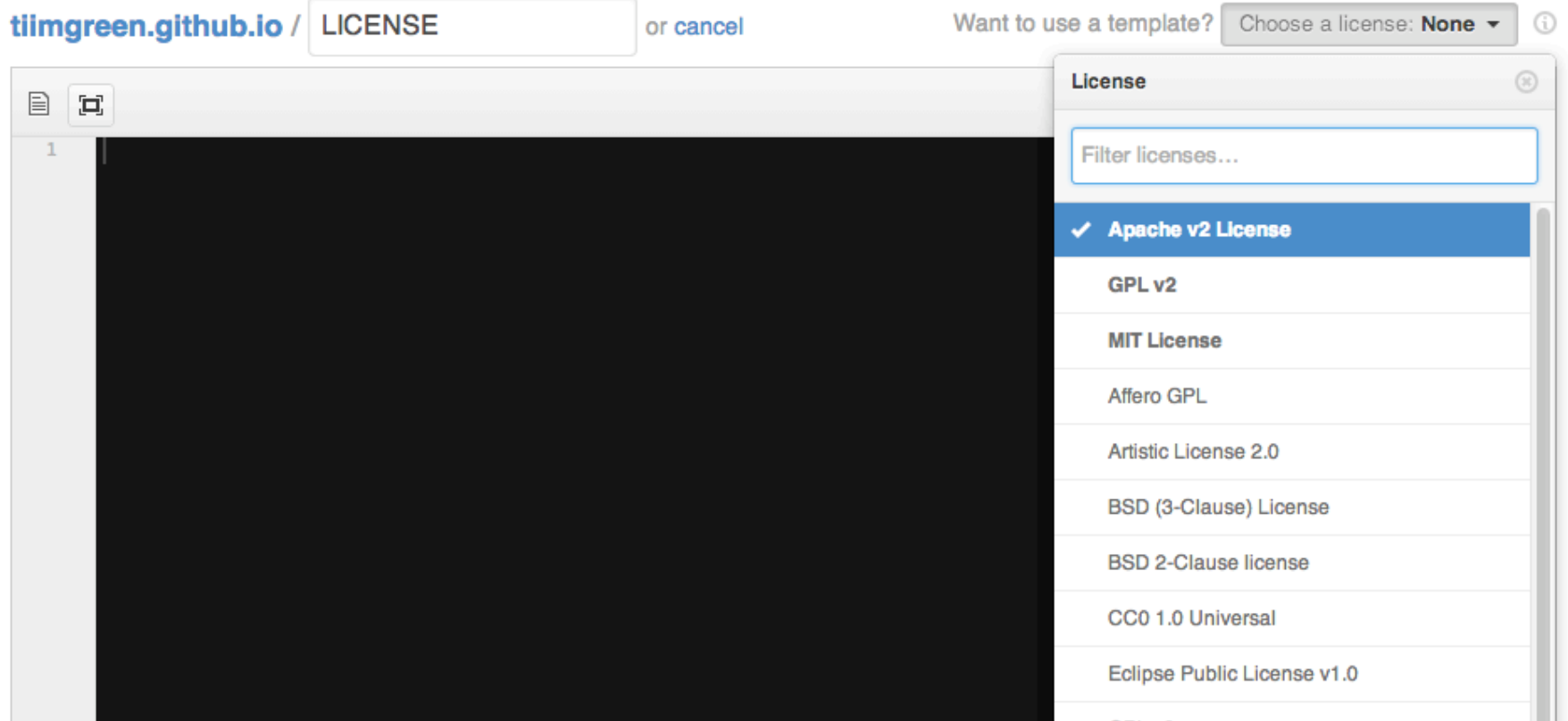
✓ MIT License

Afero GPL

[Terms](#) [Privacy](#) [Security](#) [Contact](#)

이미 있는 저장소에도 웹 인터페이스에서 새 파일을 만들어 추가할 수 있습니다. 이름에 `LICENSE` 를 입력하면 옵션으로 템플릿을 사용할 수 있습니다.





.gitignore 에도 똑같이 적용됩니다.

*오픈 소스 저작권에 대해 더 읽어 보세요.*

## Task Lists

이슈와 풀 리퀘스트에서 밑의 문법으로 체크박스를 넣을 수 있습니다.(스페이스에 주의하세요.)

- [ ] 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

체크박스가 클릭 되면, 순수 마크다운에서 갱신이 이루어집니다.



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

*테스크 리스트에 대해 더 읽어 보세요.*

## Task Lists in Markdown Documents

이제 마크다운 문서에서 읽기 전용 체크리스트를 넣을 수 있습니다.

- [ ] Mercury
- [x] Venus
- [x] Earth
- [x] Mars
- [ ] Jupiter

- ☐ Mercury
- ☒ Venus
- ☒ Earth
- ☒ Mars
- ☐ Jupiter

*마크다운 문서에서의 테스트 리스트에 대해 더 읽어 보세요.*

## Relative Links

상대 경로 링크는 마크다운 파일이 내부 콘텐츠로 링크될 때 추천합니다.

[Link to a header](#awesome-section)

[Link to a file](docs/readme)

절대 경로 링크는 URL이 변경 될 때 마다 변경해야 합니다. (예를 들어 저장소 이름 변경, 유저이름 변경, 프로젝트 포크) 상대 경로 링크를 사용하면 문서를 쉽게 스스로 독립적으로 만들 수 있습니다.

*상대 경로 링크에 대해 더 읽어 보세요.*

## Metadata and Plugin Support for GitHub Pages

지킬로 작성된 페이지와 포스트에서, 저장소 정보는 예를 들어 `{{ site.github.project_title }}` 처럼 `site.github` 이름 공간으로 사용하고 표시할 수 있습니다.

Jemoji와 jekyll-mentions플러그인은 GitHub.com에서 처럼 `emoji`와 `@mentions`을 지킬 포스트와 페이지에서 사용하게 합니다.

*저장소 메타 데이터와 깃허브 페이지의 플러그인 지원에 대해 더 읽어 보세요.*

## Viewing YAML Metadata in your Documents

Jekyll이나 GitHub Pages같은 많은 블로그에서 포스트의 처음에 YAML 포맷의 메타데이터를 필요로 합니다. 깃허브는 이 메타 정보를 읽기 편하게 테이블로 표시해 줍니다.

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](#))

문서에서 YAML 메타데이터 보기에 대해 더 읽어 보세요.

## Rendering Tabular Data

깃허브는 .csv (comma-separated) 와 .tsv (tab-separated)파일에 대해 표(tabular) 데이터 보기를 지원합니다.

file | 869 lines (868 sloc) | 188.222 kb

EditRawBlameHistoryDelete

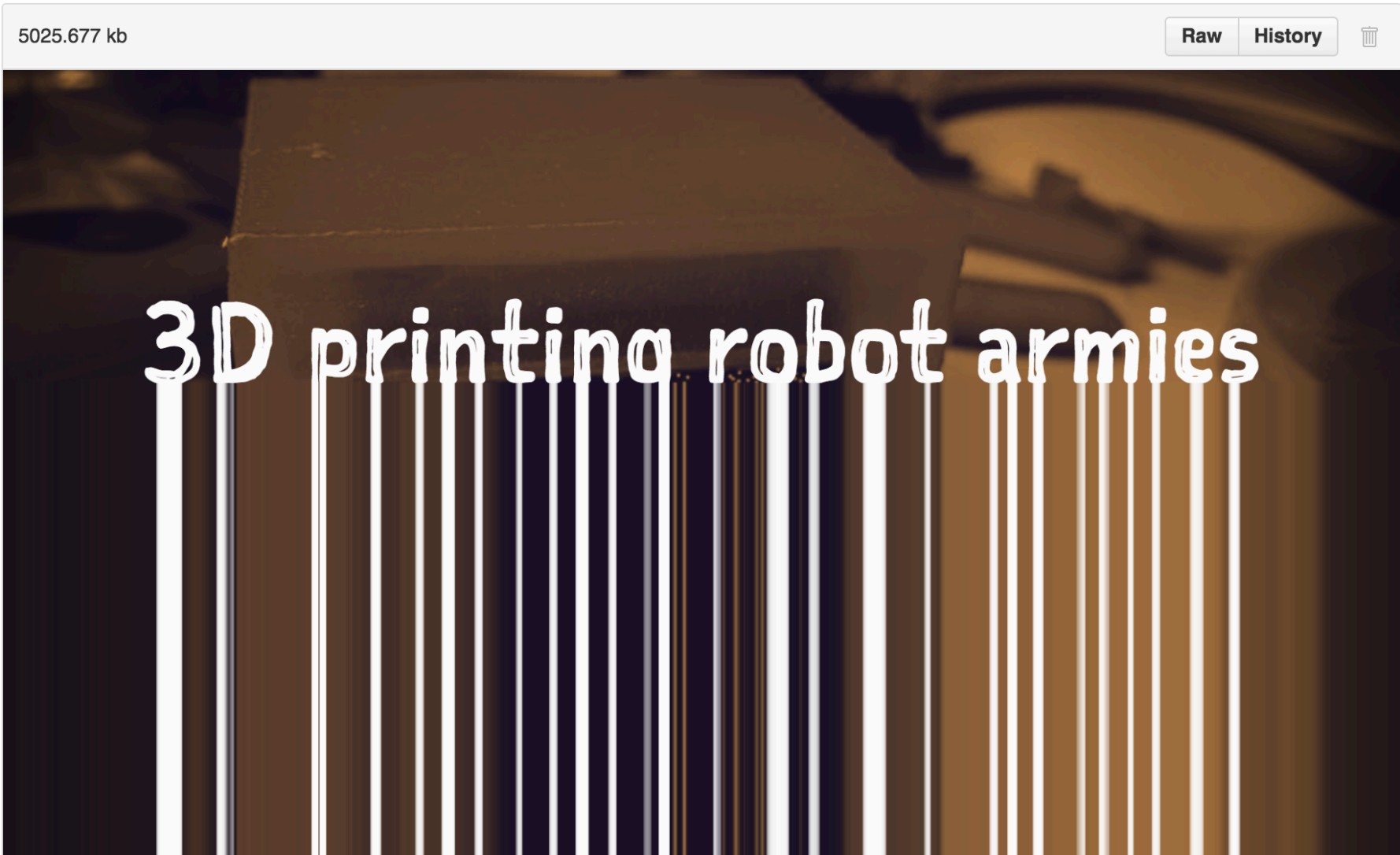
Search this file...

|    |                        |              |                                       |                                       |                              |
|----|------------------------|--------------|---------------------------------------|---------------------------------------|------------------------------|
| 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                 |

표 데이터 표시에 대해 더 읽어 보세요.

## Rendering PDF

깃허브에서 PDF를 렌더링할 수 있습니다.



*PDF 렌더링에 대해 더 읽어 보세요.*

## Revert a Pull Request

풀 리퀘스트가 머지된 후에, 아무 도움이 안되거나 머지가 잘못된 결정이었다는 걸 눈치 챌 때가 있습니다.

풀 리퀘스트 페이지의 커밋의 오른쪽에 있는 **Revert** 버튼을 클릭하면 이 풀리퀘스트를 되돌리는 풀리퀘스트를 만들어 되돌릴 수 있습니다.

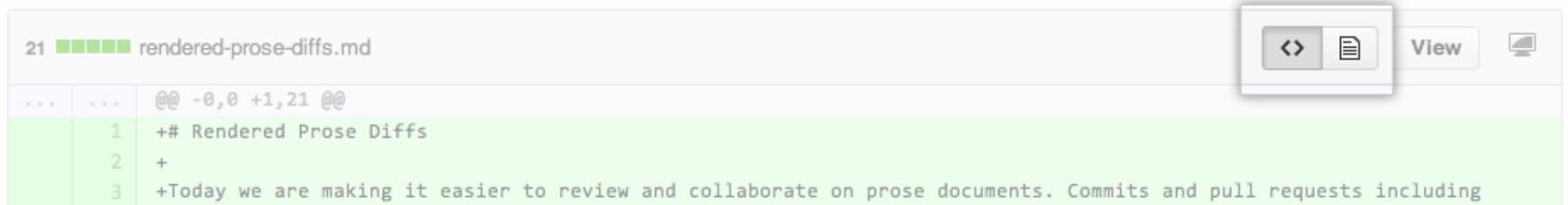


*풀 리퀘스트 되돌리기에 대해 더 읽어 보세요.*

## Diffs

### Rendered Prose Diffs

GitHub에서 지원하는 산문(prose) 파일(예를 들어 Markdown)이 있는 커밋과 풀 리퀘스트에서는 *source*와 *rendered* 뷰 기능을 사용할 수 있습니다.



“rendered” 버튼을 클릭하시면 렌더된 문서에서 변경을 확인하실 수 있습니다. 렌더된 산문(prose) 뷰는 문장을 추가, 삭제, 변경했을 때 유용합니다.

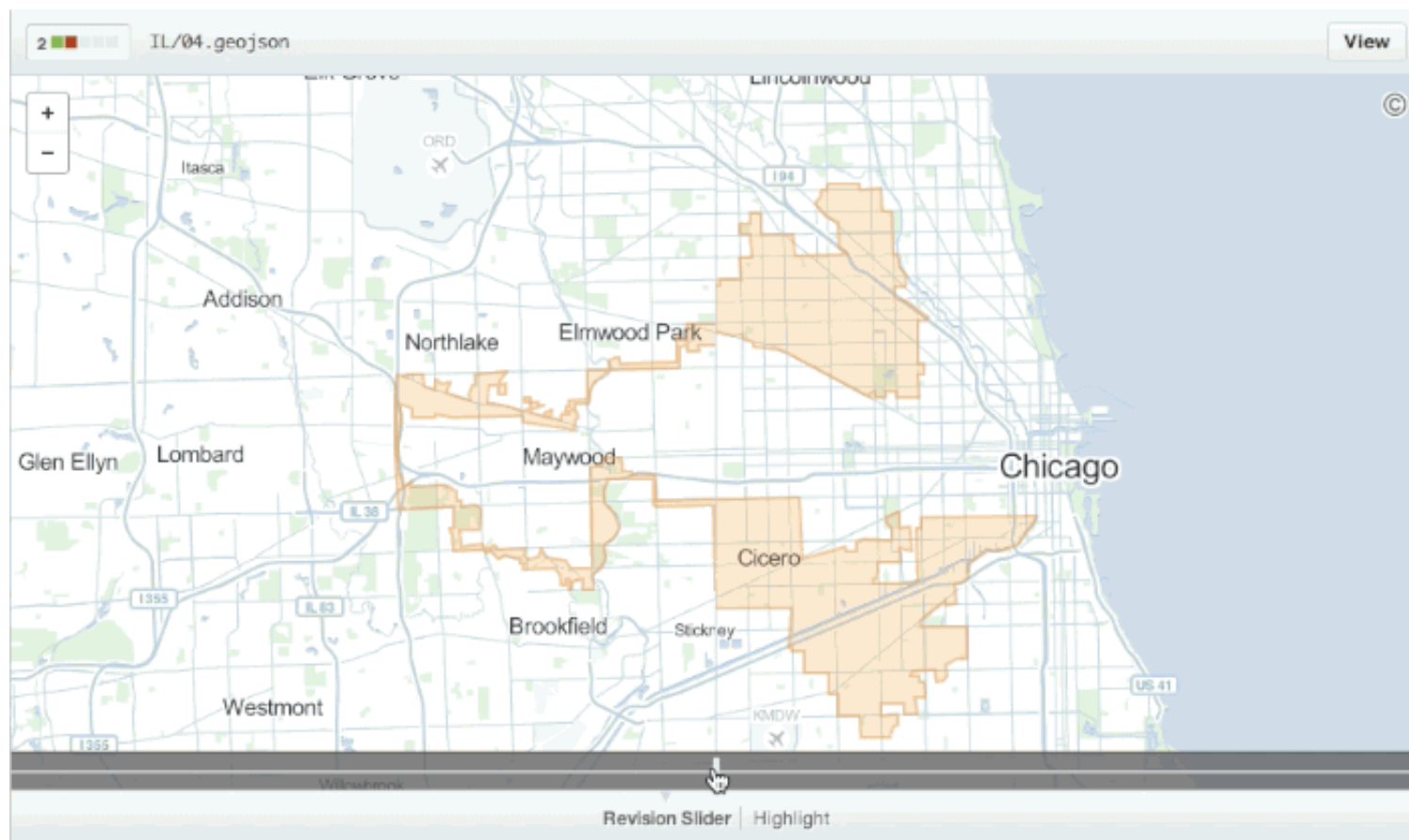
Obviously, you'll want to change `<your_client_id>` to match your actual Client ID.

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.

렌더된 산문(prose) diff에 대해 더 읽어 보세요.

## Diffable Maps

지오데이터가 포함된 커밋이나 풀 리퀘스트를 볼 때마다 깃허브는 무엇이 변경 되었는지 시각적으로 보여줍니다.



diff 가능한 맵에 대해 더 읽어 보세요.

## Expanding Context in Diffs

diff의 닫아둔 곳의 펼침 버튼을 클릭하면, 문맥을 더 볼 수 있습니다. 전체 파일을 표시할때까지 계속 펼칠 수 있으며, 깃허브에서 diff가 표시되는 장소라면 어디서든 사용가능합니다.

|     |     |  |
|-----|-----|--|
| 235 | 249 | - (id)initWithServer:(OCTServer *)server {   |
| 236 | 250 | NSParameterAssert(server != nil);  |
| 237 | 251 |  |
| 238 | 252 | self = [super initWithBaseURL:server.APIEndpoint];   |
| 239 | 253 | if (self == nil) return nil;   |
| 240 | 254 |  |
| 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;   |

*diff에서의 문맥 확장에 대해 더 읽어 보세요.*

## Diff or Patch of Pull Request

URL뒤에 .diff 나 .patch 를 붙이면 풀 리퀘스트의 diff를 얻을 수 있습니다. 예를 들면

```
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
```

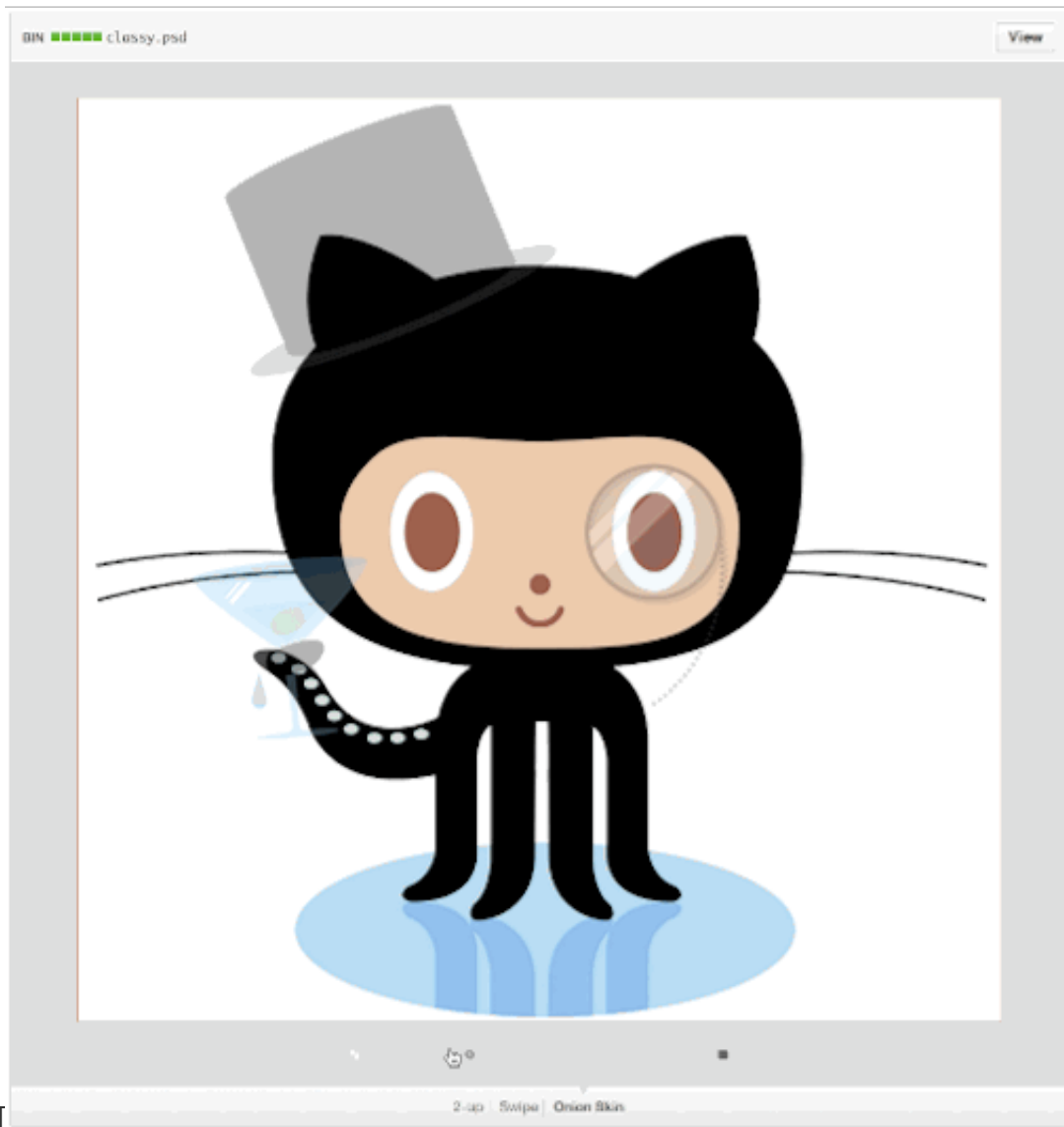
.diff 확장자는 이런 내용의 평범한(plain) 텍스트를 줍니다.

```
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
- [Merged Branches](#merged-branches)
- [Quick Licensing](#quick-licensing)
- [TODO Lists](#todo-lists)
+- [Relative Links](#relative-links)
- [.gitconfig Recommendations](#gitconfig-recommendations)
-   [Aliases](#aliases)
-   [Auto-correct](#auto-correct)
@@ -381,6 +382,19 @@ When they are clicked, they will be updated in the pure Markdown:
- [ ] Sleep

(...)
```

## Rendering and diffing images

깃허브는 PNG, JPG, GIF, PSD를 포함한 여러 일반 적인 이미지 형식을 표시할 수 있습니다. 추가적으로 이미지 형식의 diff를 표시 하는 여러 방법을 제공합니다.



](https://github.com/blog/1845-psd-viewing-diffing

이미지 diff에 대해 더 읽어 보세요.

## Hub

Hub는 깃허브를 좀 더 쉽게 사용할 수 있도록 추가 기능 및 명령을 제공하는 커맨드 라인 깃 래퍼입니다.

이렇게 할 수 있습니다.

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

Hub가 제공하는 더 멋진 기능들을 확인해 보세요.

## Contributing Guidelines

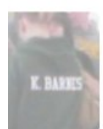
저장소의 제일 위에 CONTRIBUTING 파일을 넣어두면 기여자가 이슈를 만들거나 풀 리퀘스트를 만들 때 링크로 보여줍니다.

Browse Issues

Milestones

Search:

Issues &



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

Houston, we have a problem!

Write

Preview

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

Leave a comment



기여하기에 대해 더 읽어 보세요.

## Octicons

깃허브 아이콘(옥티콘)이 이제 오픈소스가 되었습니다.



깃 허브의 옥티콘에 대해 더 읽어보세요.

## GitHub Resources

| Title            | Link  |
|------------------|---|
| GitHub Explore   | <a href="https://github.com/explore">https://github.com/explore</a>       |
| GitHub Blog      | <a href="https://github.com/blog">https://github.com/blog</a>             |
| GitHub Help      | <a href="https://help.github.com/">https://help.github.com/</a>           |
| GitHub Training  | <a href="https://training.github.com/">https://training.github.com/</a>   |
| GitHub Developer | <a href="https://developer.github.com/">https://developer.github.com/</a> |

## GitHub Talks

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



# Git

---

## Remove All Deleted Files from the Working Tree

`/bin/rm` 을 사용해 대량의 파일을 지울 때, 하나씩 제거할 필요 없이 밑의 명령어를 사용해 작업 디렉터리와 인덱스에서 지울 수 있습니다.

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

예를 들어

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

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

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

## Previous Branch

커맨드 라인에서 이전 디렉터리로 이동하려면 이렇게 합니다.

```
$ cd -
```

비슷하게, 깃에서 마지막 브랜치로 돌아가는 건 이렇게 할 수 있습니다.

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

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

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

*깃 브랜치에 대해 더 읽어 보세요.*

## Stripspace

깃 Stripspace는 이런 일을 합니다.

- 줄 끝의 공백을 제거

- 빈줄을 줄임
- 파일 끝에 빈 줄을 추가

커맨드를 실행 할 때 파일을 반드시 이런식으로 넘겨줘야 합니다.

```
$ git stripsspace < README.md
```

깃 *stripsspace* 커맨드에 대해 더 읽어 보세요.

## SSH keys

밑의 주소를 방문하면 공개 ssh 키의 목록을 일반 텍스트 형식으로 볼 수 있습니다.

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

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

공개 ssh 키 접근에 대해 더 읽어 보세요.

## Checking out Pull Requests

풀 리퀘스트는 깃허브 저장소에서 사용하는 특별한 브랜치로 여러 방법으로 로컬로 가져 올수 있습니다.

빠르게 diff 나 merge 를 하기위해 특정 풀 리퀘스트를 임시로 FETCH\_HEAD 로 가져오려면 이렇게 합니다.

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

모든 풀 리퀘스트 브랜치를 refspec에 의한 로컬 리모트 브랜치로 받을 수도 있습니다.

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

밑의 줄들을 저장소의 .git/config 에 추가하여, 리모트의 풀 리퀘스트를 자동으로 가져오게도 할 수 있습니다.

```
[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/*
```

포크 기반의 풀 리퀘스트 기여를 위해, 풀 리퀘스트의 리모트 브랜치를 checkout 해 로컬 브랜치를 만드는 것은 유용합니다.

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

여러 저장소를 다루어야 한다면, 전역 깃 설정에서 풀 리퀘스트를 받을(fetching) 수 있도록 설정 할 수 도 있습니다.

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

이렇게 하면 모든 저장소에서 짧은 명령어를 사용 할 수 있습니다.

```
git fetch origin
```

```
git checkout pr/42
```

*풀 리퀘스트를 로컬로 체크아웃 하는 방법에 대해 더 읽어 보세요.*

## Empty Commits

--allow-empty 를 추가하시면 코드의 변경 없이 커밋을 넣을 수 있습니다.

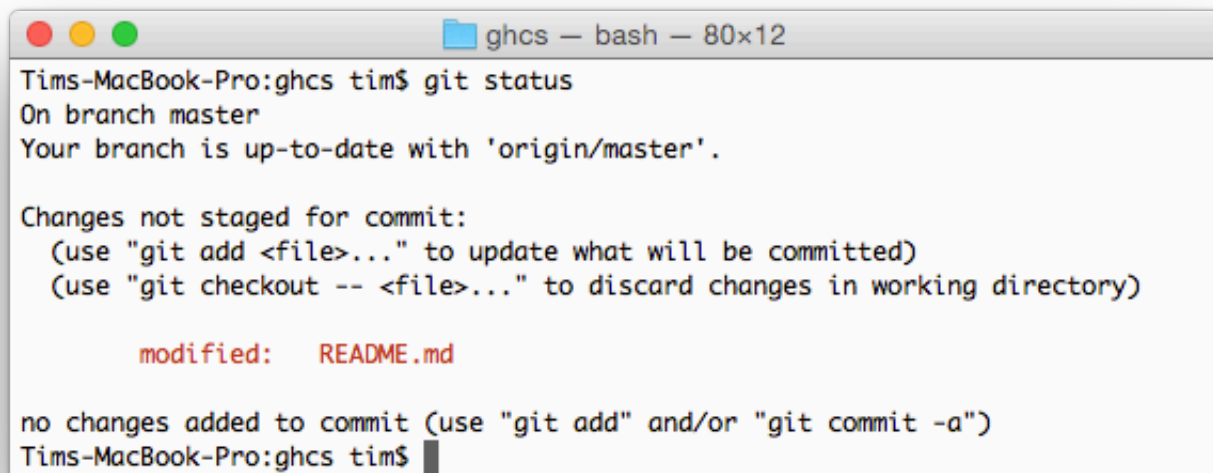
```
$ git commit -m "Big-ass commit" --allow-empty
```

있을 법한 사용법에는 이런 것들이 있습니다.

- 한 물치의 일을 시작하기 전에 주석으로 사용
- 코드와 관련없는 프로젝트 수정을 할때 주석으로 사용
- 저장소를 사용하는 사람과의 대화
- 나중에 리베이스하지 않을 저장소의 첫 커밋으로 사용 `git commit -m "init repo" --allow-empty` .

## Styled Git Status

```
$ git status
```

A terminal window titled 'ghcs - bash - 80x12' showing the output of 'git status'. The output is styled with colors: 'On branch master' and 'Your branch is up-to-date with 'origin/master'.' are in blue. 'Changes not staged for commit:' is in red, followed by instructions in blue. 'modified: README.md' is in red. 'no changes added to commit' is in blue. The prompt 'Tims-MacBook-Pro:ghcs tim\$' is 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$
```

이렇게 바뀝니다.

```
$ git status -sb
```

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

깃 `status` 커맨드에 대해 더 읽어 보세요.

## Styled Git Log

이 명령을 실행하면

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

이렇게 보입니다.

```
ghcs — less — 155x33
Tims-MacBook-Pro:ghcs tim$ git log --all --graph --pretty=format:'%Cred%h%Creset -%C(auto)%d%Creset %s %Cgreen(%cr) %C(bold blue)<an>%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 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 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>
| \
| * 1eafd88 - Update README.md (6 months ago) <ChelseaStats>
```

Palesz님 고맙습니다.

NOTE: 이 명령을 알리아스 (단축 명령)으로 넣을 수 있습니다. 여기의 소개를 보세요.

깃 `log` 커맨드에 대해 더 읽어 보세요.

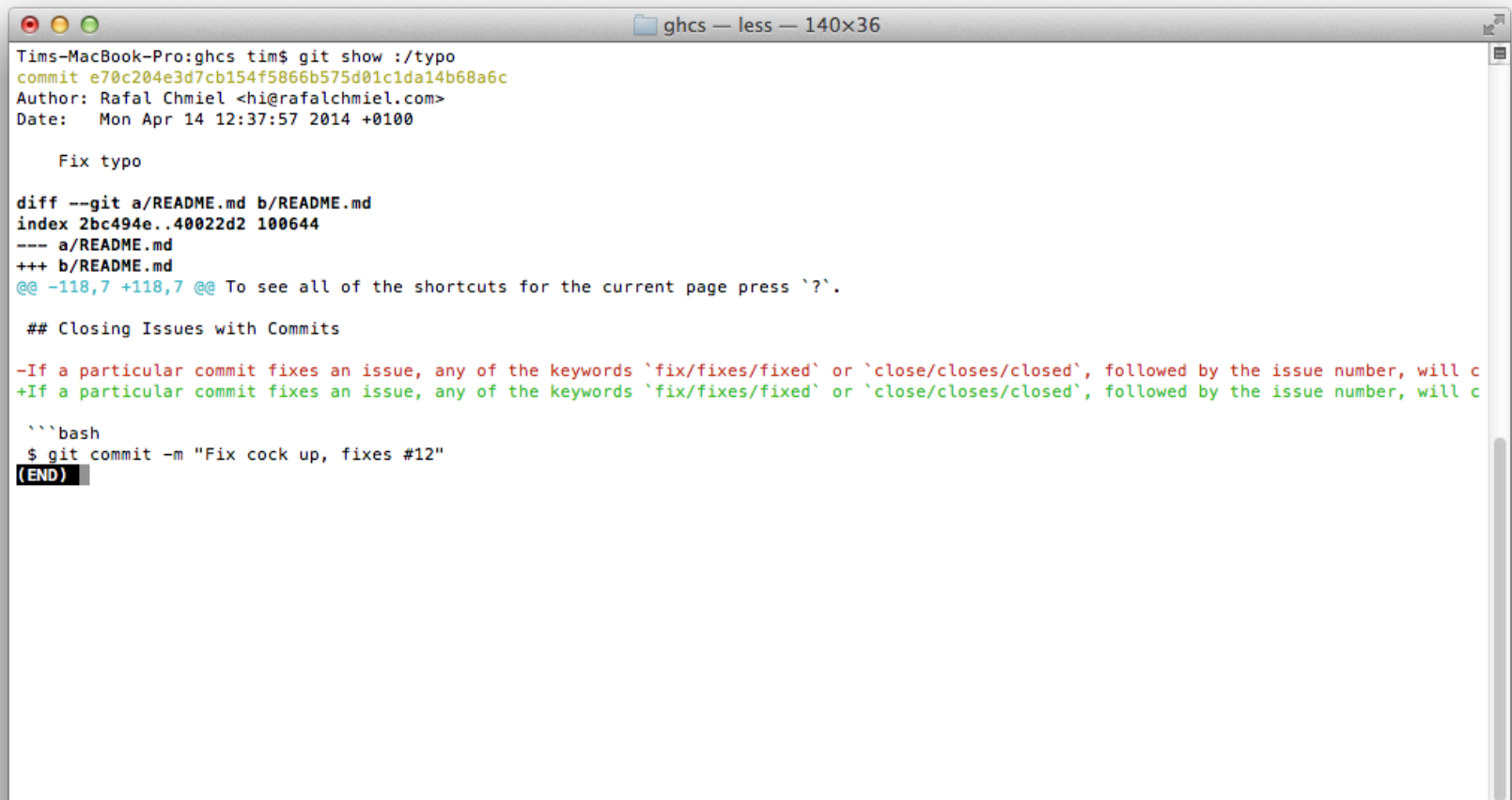
# Git Query

깃 쿼리는 모든 이전 커밋 메시지에서 검색해 가장 최근의 쿼리에 맞는 커밋을 찾아줍니다.

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

query 에는 검색하고 싶은 말을 (대소문자를 구분해) 넣으세요. 주어진 검색어가 있는 가장 마지막 커밋을 표시하고 변경된 라인을 보여줍니다.

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

A screenshot of a macOS terminal window titled 'ghcs — less — 140x36'. The terminal shows the output of the command 'git show :/typo'. The output includes commit details: 'commit e70c204e3d7cb154f5866b575d01c1da14b68a6c', 'Author: Rafal Chmiel <hi@rafalchmiel.com>', and 'Date: Mon Apr 14 12:37:57 2014 +0100'. It then shows a diff for 'Fix typo' in 'README.md', indicating a change from 'a/README.md' to 'b/README.md'. The diff content includes a section titled '## Closing Issues with Commits' and a paragraph explaining that keywords like 'fix/fixes/fixed' or 'close/closes/closed' followed by an issue number will close the issue. The terminal ends with a bash prompt '\$ git commit -m "Fix cock up, fixes #12"' and a blacked-out '(END)' prompt.

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

    Fix typo

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

## Closing Issues with Commits

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

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

NOTE: 나오려면 q 를 누르세요.

# Git Grep

Git Grep은 패턴에 매치하는 줄의 목록을 반환합니다.

다음 명령을 실행해 보세요.

```
$ git grep aliases
```

aliases 문자열이 포함된 모든 파일을 보여줍니다.

```
Tims-MacBook-Pro:ghcs tim$ git grep aliases
README.ja.md: - [Aliases](#aliases)
README.ko.md: - [Aliases](#aliases)
README.ko.md: - [Aliases](#aliases)
README.ko.md:NOTE: 이 명령을 알리야스 (단축 명령)으로 넣을 수 있습니다. [여기](#aliases)의 소개를 보세요.
README.md: - [Aliases](#aliases)
README.md: - [Aliases](#aliases)
README.md:*This can be aliased using the instructions found [here](https://github.com/tiimgreen/github-cheat-sheet#aliases).*
README.md:$ git grep aliases
README.md:will show all the files containing the string *aliases*.
README.md:![git grep aliases](http://i.imgur.com/q3WkUgl.png?1)
README.md:Some useful aliases include:
README.zh-cn.md: - [Aliases](#aliases)
README.zh-tw.md: - [Aliases](#aliases)
Tims-MacBook-Pro:ghcs tim$ █
```

나오려면 *q* 를 누르세요.

여러 플러그를 이용해서 좀 더 복잡한 검색을 할 수 있습니다. 예를 들면,

- `-e` 다음 파라미터는 패턴 (e.g. regex)
- `--and` , `--or` , `--not` 으로 여러 패턴을 조합

이렇게 사용할 수 있습니다.

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

깃 *grep* 커맨드에 대해 더 읽어 보세요.

## Merged Branches

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

이 명령어는 현재 브랜치에 머지된 모든 브랜치의 목록을 보여줍니다.

거꾸로

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

이 명령어는 현재 브랜치에 머지되지 않은 브랜치의 목록을 보여줍니다.

깃 *branch* 커맨드에 대해 더 읽어 보세요.

## Fixup and Autosquash

이전(HEAD의 한개 이상 전의) 커밋에 잘못된 부분이 있다면, 예를들어 `abcde` 라면, 문제를 수정하고 밑의 커맨드를 입력해 고칠 수 있습니다.

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

깃 `commit` 커맨드에 대해 더 읽어 보세요. 깃 `rebase` 커맨드에 대해 더 읽어 보세요.

## Web Server for Browsing Local Repositories

깃 `instaweb` 커맨드를 사용하면, 즉석에서 `gitweb` 안에서 작업중인 저장소를 둘러 볼 수 있습니다. 이 커맨드는 `gitweb` 과 로컬 저장소를 브라우징 하기 위한 웹 서버를 설정하는 스크립트입니다.

```
$ git instaweb
```

Opens:

projects / [.git](#) / summary

summary | [shortlog](#) | [log](#) | [commit](#) | [commitdiff](#) | [tree](#)

commit  ? search:  ☐ re

description

Unnamed repository; edit this file 'description' to name the repository.

owner

Rafal Chmiel

last change

Mon, 14 Apr 2014 20:04:18 +0100 (20:04 +0100)

shortlog

2 min ago

Rafal Chmiel

Add use-cases for empty commits, thanks to @davej ...

master

origin/HEAD

origin/master

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

10 min ago

Tim Green

Revert title back to original

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

12 min ago

Rafal Chmiel

Add 'Git' to title

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

15 min ago

Rafal Chmiel

Update contributing guide to fit new organization

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

17 min ago

Rafal Chmiel

:star2: Organize sections (fixes #21) :star:

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

68 min ago

Rafal Chmiel

:star: Don't push, I'll be reorganizing the whole file

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

2 hours ago

Tim Green

Update intro paragraph

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

2 hours ago

Tim Green

Clarify Rendered Prose Diffs

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

3 hours ago

Rafal Chmiel

Add wiki images section under images/GIFs

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

3 hours ago

Rafal Chmiel

Add 'Embedding Images in GitHub Wiki' section thanks...

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

3 hours ago

Rafal Chmiel

Add info about '.pibb' thanks to @jballanc :heart:

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

3 hours ago

Rafal Chmiel

Make list of emojis prettier

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

3 hours ago

Rafal Chmiel

Just a little :trollface:

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

4 hours ago

Rafal Chmiel

:trollface: :trollface: :trollface:

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

4 hours ago

Rafal Chmiel

Add 'Contributing Guidelines' section :feelsgood:

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

4 hours ago

Rafal Chmiel

Search bar -> command bar

[commit](#)

[commitdiff](#)

[tree](#)

[snapshot](#)

...

heads

2 min ago

[master](#)

[shortlog](#) | [log](#) | [tree](#)

remotes

깃 `instaweb` 커맨드에 대해 더 읽어 보세요.

## Git Configurations

`.gitconfig` 는 모든 설정이 들어있는 파일입니다.

### Aliases

알리아스는 커스텀 깃 명령어를 등록할 수 있는 헬퍼입니다. 예를 들어 `git a` 로 `git add --all` 를 실행하게 할 수 있습니다.

알리아스를 추가하려면 `~/.gitconfig` 를 찾아 다음 포맷으로 입력하거나



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

커맨드 라인에서 다음 명령을 입력합니다.

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

예를들면

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

NOTE: 여러 함수를 알리아스하려면 따옴표를 이용하세요.

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

여기 몇 가지 권장 알리아스가 있습니다.

Alias	Current Command	What to Type		
git cm	git commit	git config --global alias.cm commit		
git co	git checkout	git config --global alias.co checkout		
git ac	git add . -A git commit	git config --global alias.ac '!git add -A && git commit'		
git st	git status -sb	git config --global alias.st 'status -sb'		
git tags	git tag -l	git config --global alias.tags 'tag -l'		
git branches	git branch -a	git config --global alias.branches 'branch -a'		
git cleanup	`git branch --merged	grep -v '*'	xargs git branch -d`	`git config --global alias.cleanup '!git branch --merged
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 --"		

일부 알리아스는 @mathiasbynens님의 dotfiles에서 가져왔습니다.  
(<https://github.com/mathiasbynens/dotfiles/blob/master/.gitconfig>)

## Auto-correct

현재 git comit 를 치면 이런 결과를 보실 수 있습니다.

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

# Did you mean this?
#      commit
```

comit 을 쳤을 때 commit 을 부르려면, 자동 교정을 켜주세요.

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

이제 이런 결과를 보실 수 있습니다.

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

## Color

깃 커맨드 라인에 색을 넣으려면 이렇게 하세요.

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

깃 *config* 커맨드에 대해 더 읽어 보세요.

## Git Resources

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

## Git Books

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