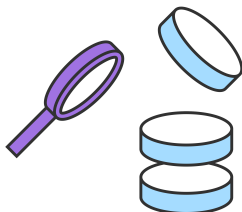


git blame

- 작동 원리
- 일반적인 옵션
- Git Blame vs Git Log



git blame 명령은 광범위한 사용법 옵션이있는 다목적 문제 해결 유틸리티입니다. git blame의 높은 수준의 기능은 파일의 특정 커밋 된 행에 첨부 된 작성자 메타 데이터의 표시입니다. 이는 파일 히스토리의 특정 지점을 검사하고 최종 작성자가 행을 수정 한 사람에 대한 컨텍스트를 얻는 데 사용됩니다. 이 코드는 특정 코드의 내역을 탐색하고 코드가 저장소에 추가 된 대상, 방법 및 이유를 알아내는 데 사용됩니다.

Git blame은 종종 GUI 디스플레이와 함께 사용됩니다. Bitbucket과 같은 온라인 Git 호스팅 사이트는 저장소에 대한 blame 뷰를 제공합니다. 이러한 뷰는 pull request 및 커밋에 대한 협업에서 참조됩니다. 또한, Git 통합이있는 대부분의 IDE에는 동적 인 blame 기능을 제공합니다.

작동 원리

git blame을 증명하기 위해서 우리는 약간의 역사가있는 저장소가 필요하다. 우리는 오픈 소스 프로젝트 인 git-blame-example을 사용할 것입니다. 이 오픈 소스 프로젝트는 다른 저자의 커밋이있는 [README.md](#) 파일이 들어있는 간단한 저장소입니다. git blame 사용 예제의 첫 번째 단계는 예제 저장소를 복제하는 것입니다.

```
git clone https://kevenzettler@bitbucket.org/kevenzettler/git-blame-example.
git && cd git-blame-example
```

이제 예제 코드의 복사본을 얻었으므로 git blame으로 살펴보기 시작할 수 있다. 예제 repo의 상태는 git log를 사용하여 검사 할 수 있습니다. 커밋 내역은 다음과 같아야합니다.

Git 가이드

1. Git 시작하기

- Git 저장소
 - git init
 - git clone
 - git config
- 변경 저장하기
 - git add
 - git commit
 - git diff
 - git stash
 - .gitignore
- 저장소 점검하기
 - git status
 - git log
 - git tag
 - git blame
- 변경 취소하기
 - git 실행 취소
 - git clean
 - git revert
 - git reset
 - git rm
- Rewriting history
 - git commit --amend
 - git rebase
 - git reflog

2. Git 협업하기

- 동기화하기
 - git remote
 - git fetch
 - git pull
 - git push
- 브랜치 사용하기
 - git branch
 - git checkout
 - git merge
 - 병합 충돌 해결하기 (Merge conflicts)
 - 병합 전략 (Merge strategies)
- Pull request 만들기

```
$ git log
commit 548dabed82e4e5f3734c219d5a742b1c259926b2
Author: Juni Mukherjee <jmukherjee@atlassian.com>
Date: Thu Mar 1 19:55:15 2018 +0000

Another commit to help git blame track the who, the what, and the when

commit eb06faedb1fdd159d62e4438fc8dbe9c9fe0728b
Author: Juni Mukherjee <jmukherjee@atlassian.com>
Date: Thu Mar 1 19:53:23 2018 +0000

Creating the third commit, along with Kev and Albert, so that Kev can get
git blame docs.

commit 990c2b6a84464fee153253dbf02e845a4db372bb
Merge: 82496ea 89feb84
Author: Albert So <aso@atlassian.com>
Date: Thu Mar 1 05:33:01 2018 +0000

Merged in albert-so/git-blame-example/albert-so/readmemd-edited-online-
with-bitbucket-1519865641474 (pull request #2)

README.md edited online with Bitbucket

commit 89feb84d885fe33d1182f2112885c2a64a4206ec
Author: Albert So <aso@atlassian.com>
Date: Thu Mar 1 00:54:03 2018 +0000

README.md edited online with Bitbucket
```

git blame은 개별 파일에서만 작동합니다. 유용한 출력을 위해서는 파일 경로가 필요합니다. git blame의 기본 실행은 명령 도움말 메뉴를 출력합니다. 이 예에서는 README.MD 파일을 사용합니다. 프로젝트의 문서 소스로 git 저장소의 루트에 README 파일을 포함시키는 것은 일반적인 오픈 소스 소프트웨어 관행입니다.

```
git blame README.MD
```

위의 명령을 실행하면 첫 번째 비난 출력 샘플을 얻을 수 있습니다. 다음 출력은 README의 전체 책임 출력의 하위 집합입니다. 또한 이 출력은 이 글을 쓰는 시점의 repo 상태를 반영합니다.

```
$ git blame README.md
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 1) # Git Blame example
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 2)
89feb84d (Albert So 2018-03-01 00:54:03 +0000 3) This repository is an
example of a project with multiple contributors making commits.
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 4)
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 5) The repo use used
elsewhere to demonstrate `git blame`
82496ea3 (kevzettler 2018-02-28 13:37:02 -0800 6)
89feb84d (Albert So 2018-03-01 00:54:03 +0000 7) Lorem ipsum dolor sit
amet, consectetur adipisicing elit, sed do eiusmod TEMPOR incididunt ut
labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis
aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu
fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident,
sunt in culpa qui officia deserunt mollit anim id est laborum
89feb84d (Albert So 2018-03-01 00:54:03 +0000 8)
eb06faed (Juni Mukherjee 2018-03-01 19:53:23 +0000 9) Annotates each line
in the given file with information from the revision which last modified
the line. Optionally, start annotating from the given revision.
eb06faed (Juni Mukherjee 2018-03-01 19:53:23 +0000 10)
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 11) Creating a line to
support documentation needs for git blame.
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 12)
548dabed (Juni Mukherjee 2018-03-01 19:55:15 +0000 13) Also, it is
important to have a few of these commits to clearly reflect the who, the
what and the when. This will help Kev get good screenshots when he runs
the git blame on this README.
```

이것은 [README.md](#) 파일의 처음 13 행의 샘플입니다. 이 출력을 더 잘 이해하기 위해 라인을 구분할 수 있습니다. 다음 표는 3 행의 내용을 표시하며 표의 열은 열의 내용을 나타냅니다.

Id	Author	Timestamp	Line Number	Line Content
89feb84d	Albert So	2018-03-01 00:54:03 +0000	3	This repository is an example of a project with multiple contributors making commits.

blame 출력 목록을 검토하면 몇 가지 관찰을 할 수 있습니다. 저자가 세 명 있습니다. 프로젝트의 유지 보수 담당자 인 Kev Zettler 외에도 Albert So, Juni Mukherjee도 나와 있습니다. 저자는 일반적으로 git blame 산출물의 가장 중요한 부분입니다. 타임 스탬프 열은 주로 유용합니다. 변경 내용은 줄 내용 열에 의해 표시됩니다.

일반적인 옵션

```
git blame -L 1,5 README.md
```

-L 옵션은 출력을 요청 된 행 범위로 제한합니다. 여기에서는 출력을 1-5 행으로 제한했습니다.

```
git blame -e README.md
```

-e 옵션은 사용자 이름 대신 작성자 전자 메일 주소를 표시합니다.

```
git blame -w README.md
```

-w 옵션은 공백 변경을 무시합니다. 이전 작성자가 탭에서 공백으로 바꾸거나 새 줄을 추가하여 파일의 간격을 수정 한 경우 불행히도 이러한 변경 사항을 표시하여 git blame의 결과를 가립니다.

```
git blame -M README.md
```

-M 옵션은 동일한 파일 내의 이동 또는 복사 된 행을 감지합니다. 이렇게하면 선을 이동하거나 복사 한 마지막 작성자 대신 선의 원래 작성자가 보고됩니다.

```
git blame -C README.md
```

-C 옵션은 다른 파일에서 이동되거나 복사 된 행을 감지합니다. 이렇게하면 선을 이동하거나 복사 한 마지막 작성자 대신 선의 원래 작성자가 보고됩니다.

Git Blame vs Git Log

634/5000

git blame은 라인을 수정 한 마지막 저자를 표시하지만 종종 원래 라인이 추가 된 시점을 알고 싶을 때가 있습니다. 이것은 git blame을 사용하여 성취하기가 번거로울 수 있습니다. -w, -C 및 -M 옵션을 조합해야 합니다. git log 명령을 사용하는 것이 훨씬 더 편리 할 수 있습니다.

특정 코드 조각이 추가되거나 수정 된 모든 원래 커밋을 나열하려면 -S 옵션을 사용하여 git log를 실행하십시오. 찾고있는 코드에 -S 옵션을 추가하십시오. 위의 README 출력에있는 행 중 하나를 예로 들어 보겠습니다. README 출력의 12 행에서 "CSS3D 및 WebGL 렌더러"텍스트를 가져와 봅시다.

```
$ git log -S"CSS3D and WebGL renderers." --pretty=format:'%h %an %ad %s'
e339d3c85 Mario Schuettel Tue Oct 13 16:51:06 2015 +0200 reverted README.
md to original content
509c2cc35 Daniel Tue Sep 8 13:56:14 2015 +0200 Updated README
cb20237cc Mr.doob Mon Dec 31 00:22:36 2012 +0100 Removed DOMRenderer. Now
with the CSS3DRenderer it has become irrelevant.
```

이 결과는 README의 내용이 3 명의 다른 저자에 의해 3 번 추가되거나 수정되었음을 보여줍니다. 그것은 원래 Mr.doob이 커밋 cb20237cc에 추가했습니다. 이 예제에서 git log 앞에 --pretty-format 옵션이 추가되었습니다. 이 옵션은 git log의 기본 출력 형식을 git log의 형식과 일치하는 것으로 변환합니다. 사용법과 설정 옵션에 대한 더 자세한 정보는 git log 페이지를 방문하십시오.