

git rm

- Overview
- 사용법
- git rm 취소하기
- git rm 의 영향
- rm 대신 git rm 을 사용해야하는 이유
- 예제
- 더이상 파일시스템에 존재하지 않는 파일 제거하기

Git을 시작할 때 가장 많이하는 질문은 "Git이 파일을 더 이상 추적하지 않게하려면 어떻게해야합니까?" git rm 명령은 Git 저장소에서 파일을 제거하는 데 사용됩니다. 그것은 git add 명령의 역으로 생각할 수 있습니다.

Overview

git rm 명령은 개별 파일 또는 파일 모음을 제거하는 데 사용할 수 있습니다. git rm의 주요 기능은 Git 색인에서 추적 된 파일을 제거하는 것입니다. 또한 git rm을 사용하여 스테이징 색인과 작업 디렉토리에서 파일을 제거 할 수 있습니다. 작업 디렉토리에서만 파일을 제거하는 옵션은 없습니다. 조각중인 파일은 현재 HEAD의 파일과 동일해야 합니다. 파일의 HEAD 버전과 스테이징 색인 또는 작업 트리 버전간에 불일치가 있으면 Git이 제거를 차단합니다. 이 블록은 진행중인 변경 사항을 제거하지 못하도록하는 안전 메커니즘입니다.

git rm은 브랜치를 제거하지 않는다.

사용법

```
...
```

제거 할 대상 파일을 지정합니다. 옵션 값은 개별 파일, 파일로 구분 된 file1 file2 file3 또는 와일드 카드 파일 glob (~./ directory / *) 일 수 있습니다.

```
-f
--force
```

-f 옵션은 HEAD의 파일이 스테이징 색인 및 작업 디렉토리의 현재 내용과 일치하도록하기 위해 힘내 기가 만든 안전성 검사를 무시하는 데 사용됩니다.

```
-n
--dry-run
```

"dry run"옵션은 git rm 명령을 실행하지만 실제로 파일을 삭제하지 않는 보호 장치입니다. 대신 제거 된 파일을 출력합니다.

```
-r
```

-r 옵션은 'recursive'의 줄임말입니다. 재귀 적 모드에서 작동 할 때 rm은 대상 디렉토리 그 디렉토리의 모든 내용을 제거합니다.

```
--
```

separator 옵션은 파일 이름 목록과 git rm으로 전달되는 인수를 명시 적으로 구별하는 데 사용됩니다. 일부 파일 이름에 다른 옵션과 혼동 될 수있는 구문이있는 경우 유용합니다.

```
--cached
```

캐시 옵션은 스테이징 색인에서만 제거가 수행되도록 지정합니다. 작업 디렉토리 파일은 그대로 남습니다.

```
--ignore-unmatch
```

이로 인해 파일이 일치하지 않아도 명령이 0 sigterm 상태로 종료됩니다. 이것은 유닉스 수준의 상태 코드입니다. 코드 0은 명령의 성공적인 호출을 나타냅니다. --ignore-unmatch 옵션은 정상적으로 실패해야하는 더 큰 셸 스크립트의 일부로 git rm을 사용할 때 도움이 될 수 있습니다.

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

```
-q
--quiet
```

quiet 옵션은 git rm 명령의 출력을 숨깁니다. 이 명령은 일반적으로 제거된 각 파일에 대해 한 줄씩 출력합니다.

git rm 취소하기

git rm을 실행하는 것은 영구적인 업데이트가 아닙니다. 이 명령은 스테이징 색인과 작업 디렉토리를 업데이트합니다. 이러한 변경 사항은 새 확약이 작성되고 변경 사항이 확약 내역에 추가될 때까지 유지되지 않습니다. 즉, 여기있는 변경 사항은 일반적인 Git 명령을 사용하여 "실행 취소"할 수 있음을 의미합니다.

```
git reset HEAD
```

리셋하면 현재 스테이징 색인과 작업 디렉토리가 HEAD 커밋으로 되돌아갑니다. 이렇게하면 git rm이 실행 취소됩니다.

```
git checkout .
```

체크 아웃은 동일한 효과를 가지며 HEAD에서 최신 버전의 파일을 복원합니다.

git rm이 실행되었고 제거를 지속하는 새로운 커밋이 생성된 경우 git reflog를 사용하여 git rm 실행 전의 참조를 찾을 수 있습니다. git reflog 사용에 대해 자세히 알아보십시오.

git rm 의 영향

git rm 명령은 현재 분기에서만 작동합니다. 제거 이벤트는 작업 디렉토리 W 스테이징 색인 트리에만 적용됩니다. 파일 h 제는 새 확약이 작성될 때까지 저장소 실행 기록에 보존되지 않습니다.

rm 대신 git rm 을 사용해야 하는 이유

Git 저장소는 추적 중인 파일에서 일반 셸 rm 명령이 실행된 시점을 인식합니다. 제거를 반영하도록 작업 디렉토리를 업데이트합니다. 제거로 스테이징 색인을 업데이트하지 않습니다. 스테이징 색인에 변경 사항을 추가하려면 제거된 파일 경로에서 추가 git add 명령을 실행해야 합니다. git rm 명령은 작업 디렉토리와 스테이징 색인을 제거와 함께 업데이트한다는 점에서 바로 가기를 수행합니다.

예제

```
git rm Documentation/*.txt
```

이 예에서는 와일드 카드 파일 glob을 사용하여 Documentation 디렉토리의 하위 디렉토리 및 해당 하위 디렉토리의 모든 *.txt 파일을 제거합니다.

이 예에서는 별표 (*)가 슬래시로 이스케이프됩니다. 이것은 셸이 와일드 카드를 확장하지 못하게 하는 가드입니다. 그런 다음 와일드 카드는 Documentation / 디렉토리 아래에 있는 파일 및 하위 디렉토리의 경로 이름을 확장합니다.

```
git rm -f git-*.sh
```

더이상 파일시스템에 존재하지 않는 파일 제거하기

"rm 대신 git rm을 사용하는 이유"에서 설명한 것처럼 실제로 git rm은 표준 셸 rm과 git add를 결합하여 작업 디렉토리에서 파일을 제거하고 해당 제거를 스테이징 색인으로 승격시키는 편의 명령입니다. 표준 셸 rm 명령만 사용하여 여러 파일이 제거된 경우 저장소가 성가신 상태가 될 수 있습니다.

의도적으로 제거된 모든 파일을 다음 커밋의 일부로 기록하려는 경우 git commit -a는 다음 커밋 준비를 위해 모든 제거 이벤트를 스테이징 인덱스에 추가합니다.

그러나 셸 rm을 사용하여 제거된 파일을 지속적으로 제거하려는 경우 다음 명령을 사용하십시오.

```
git diff --name-only --diff-filter=D -z | xargs -0 git rm --cached
```

이 명령은 작업 디렉토리에서 제거 된 파일 목록을 생성하고 그 목록을 git rm --cached에 저장하여 스테이징 색인을 업데이트합니다.