되돌리기

일을 하다보면 모든 단계에서 어떤 것은 되돌리고 싶을 때가 있다. 이번에는 우리가 한 일을 되돌리는 방법을 살펴볼 것이다**. 한 번 되돌리면 복구할 수 없어서 주의해야 한다. Git 을 사용하면 우리가 한 실수를 복구하지 못할 것은 거의 없지만 되돌리기는 복구할 수 없다.**

커밋 수정하기

종종 완료한 커밋을 수정해야 할 때가 있다. 너무 일찍 커밋했거나 어떤 파일을 뺴먹었을 때 그리고 커밋 메시지를 잘못 적었을 때 하게 된다. 다시 커밋하고 싶으면 –amend 옵션을 사용한다.

$ git commit --amend

이 명령은 Staging Area를 사용하여 커밋한다. 만약 마지막으로 커밋하고 나서 수정한 것이 없다면 (커밋 하자마자 바로 이 명령을 실행하는 경우) 조금 전에 한 커밋과 모든 것이 같다. 이때는 커밋 메시지만 수정한다.

편집기가 실행되면 이전 커밋 메시지가 자동으로 포함된다. 메시지를 수정하지 않고 그대로 커밋해도 기존의 커밋을 덮어쓴다.

커밋을 했는데 Stage 하는 것을 깜빡하고 빠트린 파일이 있으면 아래와 같이 고칠 수 있다.

$ git commit -m 'initial commit'

$ git add forgotten\_file

$ git commit --amend

여기서 실행한 명령어 3개는 모두 하나의 커밋으로 기록된다. 두 번째 커밋은 첫 번째 커밋을 덮어쓴다.

파일 상태를 Unstage로 변경하기

다음은 Staging Area와 워킹 디렉토리 사이를 넘나드는 방법을 설명한다. 두 영역의 상태를 확인할 때마다 변경된 상태를 되돌리는 방법을 알려주기 때문에 매우 편리하다. 예를 들어 파일을 두 개 수정하고서 따로따로 커밋하려고 했지만, 실수로 git add \* 라고 실행해 버렸다. 두 파일 모두 Staging Area에 들어 있다. 이제 둘 중 하나를 어떻게 꺼낼까? 우선 git status 명령으로 확인해보자.

$ git add .

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: README.txt

modified: benchmarks.rb

**Changes to be commited 밑에 git reset HEAD <file>… 이라는 문장을 볼 수 있다. 이 명령으로 Unstage 상태로 변경할 수 있다**. Benchmarks.rb 파일을 Unstage 상태로 변경해보자.

$ git reset HEAD benchmarks.rb

Unstaged changes after reset:

M benchmarks.rb

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: README.txt

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: benchmarks.rb

명령어가 낯설게 느껴질 수도 있지만 잘 작동한다. benchmarks..rb 파일은 Unstage 상태가 됐다.

Modified 파일 되돌리기

어떻게 해야 benchmarks.rb 파일을 수정하고 나서 다시 되돌릴 수 있을까? 그러니까 최근 커밋된 버전으로 아니면 처음 Clone 했을 떄처럼 워킹 디렉토리에 처음 Checkout 한 그 내용으로 되돌리는 방법이 뭘까?

Git status 명령이 친절하게 알려준다. 바로 위에 있는 예제에서 Unstaged 부분을 보자.

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: benchmarks.rb

위의 메시지는 수정한 파일을 되돌리는 방법을 꽤 정확하게 알려준다. (적어도 Git.1.6.1이후 버전부터는 그렇다. 만약 예전 것을 아직 사용하고 있으면 업그레이드하는 것이 좋다. 편의성이 많이 개선됐다.) 알려주는 대로 한 번 해보자.

$ git checkout -- benchmarks.rb

$ git status

On branch master

Changes to be committed:

(use "git reset HEAD <file>..." to unstage)

modified: README.txt

정상적으로 복원된 것을 알 수 있다. 하지만 이 명령은 꽤 위험한 명령이라는 것을 알아야 한다. 수정 이전의 파일로 덮어썼기 때문에 수정했던 내용은 전부 사라진다. 수정한 내용이 진짜 마음에 들지 않을 때에만 사용하자. 정말 이렇게 삭제해야 한다면 Stash 와 Branch를 사용하자. 다음 장에서 다루는 이 방법들이 훨씬 낫다.

**Git 으로 커밋한 모든 것은 언제나 복구할 수 있다. 삭제한 브랜치에 있었던 것도 –amend 옵션으로 다시 커밋한 것도 복구할 수 있다**.(자세한 것은 9장에서 다룬다.) 하지만, **커밋하지도 않고 잃어버린 것은 절대로 되돌릴 수 없다.**