









2016-04-18

softcontext@gmail.com

# Git

동일한 정보에 대한 여러 버전을 관리하는 분산 버전관리 시스템이다. master 저장소와 참여하는 모든 개발자가 개별적으로 갖고 있는 로컬 저장소로 구성된다. 2005년 탄생했으며 리눅스 커널 개발자인 리누스 토르발즈가 기존 버전관리 프로그램들의 상용화에 반대하여 2주만에 만든 것으로 유명하다.

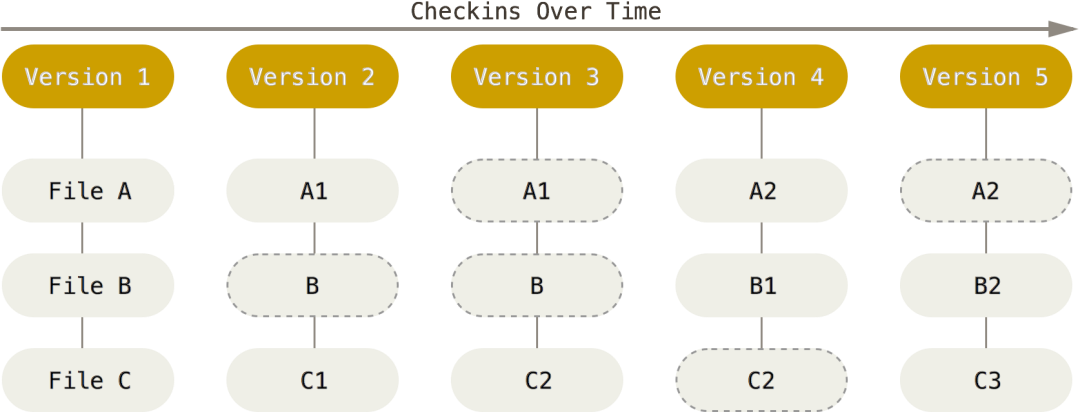
## 버전관리의 여러 명칭

* Version Control System
* Revision Control System
* Source Control System
* Source Code Management
* Distributed Version Control System

## 작동방식

Git의 데이터는 파일들의 Snapshot이다. Git은 파일을 저장하거나 커밋하는 시점에서 변화를 감지한다.

파일이 달라지지 않았다면 Git은 성능을 위해서 파일을 저장하지 않고 링크만 저장한다. 다음 화면에서 점선으로 표시되는 파일은 링크만 저장한 것이다.



## 체크섬

Git은 체크섬으로 데이터를 관리한다. 체크섬은 원자적인(Atomic) 데이터 단위다.

Git은 SHA-1 해시를 사용하여 체크섬을 만든다. 체크섬은 40자 길이의 16진수 문자열이다.

SHA-1은 다음처럼 생겼다.

24b9da6552252987aa493b52f8696cd6d3b00373

## 파일상태

* untracked

깃이 관리하지 않는 상태다. 새로 파일을 만들면 기본적으로 비추적 상태가 된다.

* unmodified

add를 하면 깃은 파일의 변화를 추적한다. 추적은 하지만 변화는 감지되지 않은 상태다.

* modified

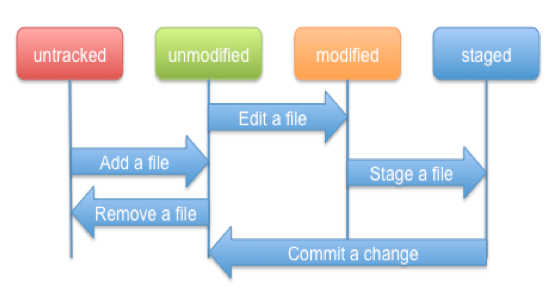
깃이 추적하는 파일을 변경하면 수정상태가 된다.

* staged

대상 파일을 커밋대상으로 지정하기 위해서 add를 하면 스테이지에 올린 상태가 된다.

* commited

로컬 저장소에 저장을 하면 대상 파일은 비수정 상태로 돌아간다.



## 세가지 영역

* Working Directory

프로젝트 파일들이 있는 디렉토리다.

Working Directory의 모든 파일은 Tracked와 Untracked상태로 나뉜다.

Tracked 파일은Unmodified, Modified, Staged 상태 중 하나의 상태를 갖는다.

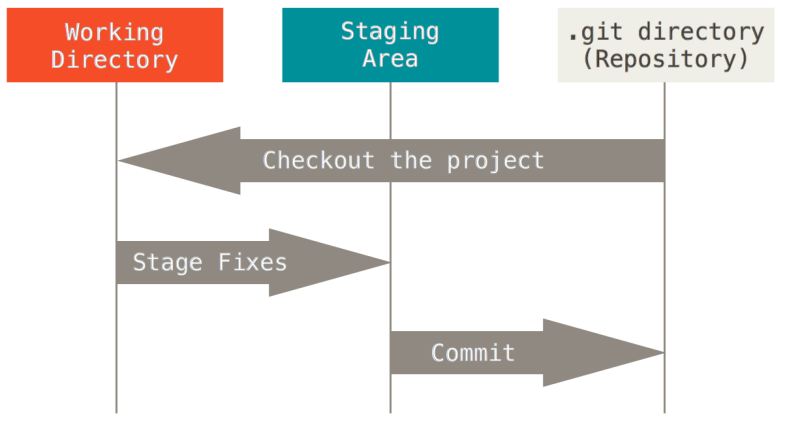
Tracked 파일이 아닌 모든 파일은 Untracked 상태다.

* Staging Area

Git Directory에 있다. 단순한 파일이고 곧 commit할 파일에 대한 정보를 저장한다.

* Git Directory

Git이 프로젝트의 메타데이터와 객체 데이터베이스를 저장하는 곳을 말한다. (.git)



## HEAD

깃은 HEAD라는 특수한 포인터를 갖고 있다.

이 포인터는 마지막 커밋을 가리킨다. 현재 작업하고 있는 브랜치를 가리키는 용도로도 사용한다.

## ORIG\_HEAD

HEAD가 이전에 가리키던 커밋을 참조한다.

커밋의 구성을 조작(rebase)할 때 이전 상태로 되돌리기 위한 참조용도로 사용한다.

# 설치

OS에 맞는 버전을 다운로드 후 기본 설정으로 설치한다.

http://git-scm.com/

## 계정등록

$ git config --global user.name "softcontext"

$ git config --global user.email "softcontext@gmail.com"

## 설정정보 확인

$ git config --global --list

user.name=softcontext

user.email=softcontext@gmail.com

filter.lfs.clean=git-lfs clean %f

filter.lfs.smudge=git-lfs smudge %f

filter.lfs.required=true

## 도움말 보기

$ git help

$ git help config

$ git checkout --help

# Git 사용법

깃을 설치할 때 자동적으로 Git Bash도 같이 설치가 된다.

바탕화면에 있는 아이콘을 클릭하여 깃 배쉬를 실행한 후, 다음 테스트 시나리오를 따라하면서 깃 사용법을 익혀 보자.

## 1. 저장소 생성

### 1.1 프로젝트 폴더 생성

적당한 루트 작업 폴더를 만들고 그 위치에서 다음 명령을 실행해 보자.

|  |
| --- |
| $ mkdir git\_tutorial  $ cd git\_tutorial |

### 1.2 저장소 초기화(생성)

프로젝트 루트 폴더에서 다음 명령을 실행하면 깃 로컬 저장소가 생성된다.

.git 이라는 폴더가 init 명령의 수행결과로 만들어지는 폴더이면 바로 로컬 저장소가 된다.

|  |
| --- |
| $ git init  Initialized empty Git repository in C:/Users/Chris/**git\_tutorial/.git/**  $ git add \*  fatal: pathspec '\*' did not match any files |

add 명령으로 현재 폴더 위치에 있는 모든(\*) 파일들을 깃이 추적하는 대상으로 지정한다.

파일이 아무것도 없으면 작업은 이루어지지 않는다.

## 2. 프로그램 작성

### 2.1 파이썬 설치

프로그램 작성 툴로 파이썬을 사용할 것이다. 다음 사이트에 접속하여 다운로드 후 설치한다.

https://www.python.org/downloads/

### 2.2 파이썬 프로그램 작성

프로그램 코딩 에디터로 vim을 사용할 것이다.

### 2.3 vim 에디터의 세 가지 모드

##### 1. 일반 모드

파일 진입 시, 첫 모드는 일반모드다. 입력모드, 명령모드에서 esc 키를 누르면 일반모드로 전환된다.

##### 2. 입력 모드

일반모드에서 i, o, a 키 중에 하나를 누르면 입력모드로 전환된다.

##### 3. 명령 모드

일반모드에서 : 키를 누르면 명령모드로 전환된다.

w: 저장 명령

q: 종료 명령

##### 자주 사용하는 복사 및 붙여넣기 단축 키

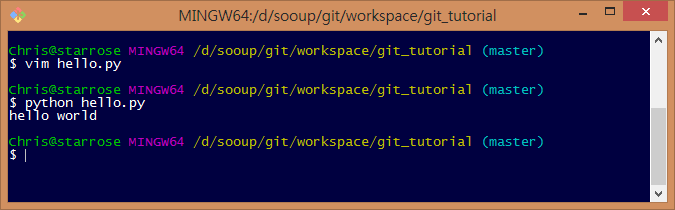
|  |
| --- |
| 한 줄 복사: 일반 모드 > 복사 대상 라인으로 커서이동 > yy  붙여넣기: 일반 모드 > 붙여넣기 위치로 커서이동 > p  한 줄 삭제: 일반 모드 > dd |

### 2.4 작성

|  |  |
| --- | --- |
| vim hello.py > enter  i   |  | | --- | | print("hello world") |   esc > :wq > enter |

### 2.5 확인

|  |
| --- |
| $ python hello.py  hello world |



## 3. master 브랜치: 변경사항을 저장소에 커밋

### 3.1 저장소 상태 확인

|  |
| --- |
| $ git status  On branch master  Initial commit  Untracked files:  (use "git add <file>..." to include in what will be committed)  hello.py  nothing added to commit but untracked files present (use "git add" to track) |

**Untracked files**

추적대상이 아닌 파일이 존재한다.

**$ git status -s**

상태 정보를 short 하게 조회한다. --short 옵션을 사용해도 된다.

Chris@starrose MINGW64 /d/sooup/git/workspace/git\_tutorial (master)

$ git status -s

?? hello.py

### 3.2 파일을 추적대상으로 지정

add 명령으로 원하는 파일을 추적 대상으로 추가한다.

|  |
| --- |
| $ git add hello.py  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory. |

### 3.3 확인

|  |
| --- |
| $ git status  On branch master  Initial commit  Changes to be committed:  (use "git rm --cached <file>..." to unstage)  new file: hello.py |

**Changes to be committed:**

커밋을 해야 할 변화가 감지되었다. 새로운 파일 하나가 커밋 대상으로 스테이지(커밋대상을 관리하는 캐시)에 존재한다는 것을 알 수 있다.

### 3.4 커밋

스테이지 상태에 파일을 커밋하여 깃 로컬 저장소에 저장한다.

최초로 커밋할 때 master라는 이름의 브랜치를 자동으로 만든다.

|  |
| --- |
| $ git commit  [master (root-commit) 085cf42] 'create "hello world" program'  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 hello.py |

커밋 로그를 남기기 위해서 vim 화면으로 자동 전환된다. 간단하게 커밋 메시지를 작성하고 저장하면 커밋이 수행된다.

### 3.5 확인

|  |
| --- |
| $ git status  On branch master  nothing to commit, working directory clean |

커밋을 수행할 대상이 존재하지 않는 상태가 됐음을 알 수 있다.

## 4. hotfix 브랜치: 새로운 브랜치 생성

### 4.1 브랜치 목록 확인

초기에는 하나의 master 브랜치만 존재한다.

|  |
| --- |
| $ git branch  \* master |

### 4.2 브랜치 생성

새로운 브랜치 hotfix를 만든다.

|  |
| --- |
| $ git branch hotfix |

git branch -m <old\_branch명> <new\_branch명>

브랜치명을 변경한다.

git branch -d <branch명>

브랜치를 삭제한다.

### 4.3 브랜치 목록 재 확인

\* 기호가 있는 것이 현재 작업 중인 브랜치다.

엉뚱한 브랜치에서 작업하지 않도록 주의가 필요하다.

|  |
| --- |
| $ git branch  hotfix  \* master |

### 4.4 작업대상 브랜치로 이동

|  |
| --- |
| $ git checkout hotfix  Switched to branch 'hotfix' |

### 4.5 작업 브랜치 이동 확인

|  |
| --- |
| $ git branch  \* hotfix  master |

### 4.6 새 브랜치에 파일 존재 확인

브랜치를 새로 만들면 master 브랜치에 있는 파일들이 자동으로 복제되어 존재한다.

|  |
| --- |
| $ ls  hello.py |

## 5. hotfix브랜치: 프로그램 수정

### 5.1 프로그램 수정

hotfix브랜치에서 프로그램을 수정한다. 여기서는 새로운 print 메소드를 추가할 것이다.

|  |  |
| --- | --- |
| vim hello.py > enter  i   |  | | --- | | print("hello world")  **print("hello world hotfix")** |   esc > :wq > enter |

### 5.2 수정 확인 및 실행

|  |
| --- |
| $ cat hello.py  print("hello world")  print("hello world hotfix")  $ python hello.py  hello world  hello world hotfix |

이제 master 브랜치에 hello.py 파일과 hotfix 브랜치에 hello.py 파일은 코드가 다른 상태가 되었다.

## 6. hotfix브랜치: 변경사항을 저장소에 커밋

hotfix 브랜치에서 수정작업이 완료되었고 이를 master 브랜치에 병합하여 반영하고자 한다.

### 6.1 추적 대상 확인

현재 작업 브랜치는 hotfix다.

|  |
| --- |
| $ git status  On branch hotfix  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: hello.py  no changes added to commit (use "git add" and/or "git commit -a") |

**Changes not staged for commit:**

커밋을 위해 스테이지에 올려야 하는 변화가 존재한다.

**no changes added to commit (use "git add" and/or "git commit -a")**

커밋을 수행하기 위해서 위 명령 중 하나를 사용하라.

### 6.2 변경된 모든 파일을 커밋

매번 파일 하나씩 add 명령으로 스테이지에 올려 놓고 커밋하는 작업이 번거로울 수 있다.

commit 명령 다음 옵션 -a를 사용하면 변경된 모든 파일을 대상으로 바로 커밋을 수행한다.

명령을 수행하면 커밋 메시지를 작성하기 위한 vim 에디터가 자동으로 기동한다.

커밋 메시지로 "hello.py modified on hotfix branch"라고 입력하고 저장한다.

|  |
| --- |
| $ git commit -a  [8d52190] added output "hello world hotfix"  1 file changed, 1 insertion(+) |

### 6.3 커밋 확인

master 브랜치에 병합하기 전에 해당 브랜치의 변경 파일들을 먼저 커밋해야 한다.

|  |
| --- |
| $ git status  On branch hotfix  nothing to commit, working directory clean |

hotfix 브랜치에 변경 내역은 모두 커밋되었음을 알 수 있다.

## 7. hotfix브랜치를 master 브랜치에 병합

### 7.1 master 브랜치로 이동

|  |
| --- |
| $ git checkout master  Switched to branch 'master' |

### 7.2 master 브랜치 점검

마스터 브랜치의 대상 파일은 이전에 커밋된 상태 그대로 유지되고 있음을 확인해 보자.

|  |
| --- |
| $ git branch  hotfix  \* master  $ ls  hello.py  $ cat hello.py  print("hello world")  $ python hello.py  hello world |

### 7.3 브랜치 병합

hotfix 브랜치를 현재 작업대상 브랜치(master)에 병합한다.

|  |
| --- |
| $ git merge hotfix  Updating 085cf42..8d52190  Fast-forward  hello.py | 1 +  1 file changed, 1 insertion(+) |

### 7.4 병합 확인

hotfix 브랜치에서 작업한 결과가 master 브랜치에 있는 대상 파일에 반영되었음을 알 수 있다.

|  |
| --- |
| $ ls  hello.py  $ cat hello.py  print("hello world")  **print("hello world hotfix")**  $ git status  On branch master  nothing to commit, working directory clean |

## 8. 충돌 시나리오 1단계

앞선 작업에서는 master 브랜치에서 작업한 파일을 hotfix 브랜치를 만들면서 복제하였고 hotfix 브랜치에서 코드를 추가하였다. hotfix 브랜치에 변경내역을 커밋한 후 master 브랜치로 이동하였다. master 브랜치에서 hotfix 브랜치에 변경내역을 merge 명령을 통해 master 브랜치에 반영하였다. 두 개의 branch를 병합하는 작업이 아무 문제없이 수행되었다. 문제가 발생하지 않은 이유는 병합 작업 시 코드 충돌이 발생하지 않았기 때문이다. 그러나 여러 브랜치에서 같은 파일을 수정한 다음 병합하고자 할 때 충돌이 발생하여 병합작업이 자동으로 완료되지 못 한다.

이제부터 충돌이 발생하는 경우에 처리하는 방법을 학습해 보자. 충돌 시나리오는 다음과 같다.

- master 브랜치에서 hello.py 코드를 수정한다.

- hotfix 브랜치에서 hello.py 코드를 수정한다.

위 작업이 수행된 상태에서 두 브랜치를 병합하는 작업을 수행하면 충돌이 발생하게 되어 자동으로 병합작업이 완료될 수 없다. 개발자가 충돌 지점을 확인하고 직접 충돌을 해결한 다음 커밋을 진행해야 한다.

### 8.1 master 브랜치에서 프로그램 수정

print 메소드를 하나 더 추가한다.

|  |  |
| --- | --- |
| vim hello.py > enter  i   |  | | --- | | print("hello world")  print("hello world hotfix")  **print("hello world master")** |   esc > :wq > enter |

### 8.2 master 브랜치에서 커밋

커밋 명령 사용 시 -m 옵션을 사용하면 바로 뒤에 붙인 문자열을 커밋 메시지로 사용한다. 따라서 vim은 기동하지 않는다. -a 옵션으로 스테이지에 올리는 작업과 커밋하는 작업을 한 번에 수행한다.

|  |
| --- |
| $ git commit -am 'added output "hello world master"'  [master 3adaac6] added output "hello world master"  1 file changed, 1 insertion(+) |

## 9. 충돌 시나리오 2단계

master 브랜치에서 수정한 다음 커밋한 파일과 같은 이름의 파일을 대상으로 이번에는 hotfix 브랜치에서 수정하고 커밋한다.

### 9.1 hotfix 브랜치로 이동

|  |
| --- |
| $ git checkout hotfix  Switched to branch 'hotfix' |

### 9.2 hotfix 브랜치에서 프로그램 수정

|  |  |
| --- | --- |
| vim hello.py > enter  i   |  | | --- | | print("hello world")  print("hello world hotfix")  **print("hello world 2nd hotfix")** |   esc > :wq > enter |

### 9.3 커밋

|  |
| --- |
| $ git commit -am 'added output "hello world 2nd hotfix"'  [hotfix 66c5d15] added output "hello world 2nd hotfix"  1 file changed, 1 insertion(+) |

## 10. 충돌 시나리오 3단계

1. 두 브랜치를 대상으로 병합을 시도한다.

2. 충돌이 발생하면 충돌 지점을 확인한다.

3. 충돌 부분을 해결한 후 커밋을 수행하여 병합작업을 완료한다.

### 10.1 master 트렁크와 hotfix 브랜치 상태 확인

master 브랜치를 간단하게 트렁크라고 부르기도 한다.

|  |
| --- |
| $ git checkout master  Switched to branch 'master'  $ cat hello.py  print("hello world")  print("hello world hotfix")  **print("hello world master")**  $ git checkout hotfix  Switched to branch 'hotfix'  $ cat hello.py  print("hello world")  print("hello world hotfix")  **print("hello world 2nd hotfix")** |

두 브랜치에 각각 존재하는 hello.py 파일에 상이한 코드가 존재한다. 이 경우 merge 작업 시 자동으로 병합작업이 완료되지 못 한다.

### 10.2 병합 시도 시 충돌 발생 확인

|  |
| --- |
| $ git checkout master  Switched to branch 'master'  $ git status  On branch master  nothing to commit, working directory clean  $ git branch  hotfix  \* master  $ git merge hotfix  Auto-merging hello.py  **CONFLICT (content): Merge conflict in hello.py**  **Automatic merge failed; fix conflicts and then commit the result.**  $ cat hello.py  print("hello world")  print("hello world hotfix")  **<<<<<<< HEAD**  **print("hello world master")**  =======  **print("hello world 2nd hotfix")**  **>>>>>>> hotfix** |

병합작업 중 충돌이 발생하는 지점에 master 브랜치에서 수정된 코드와 hotfix 브랜치에서 수정된 코드를 같이 표시하고 있다. 현재 작업 브랜치는 master 이므로 HEAD는 master 브랜치에서 마지막으로 커밋된 내용을 가리키고 있는 것이다.

### 10.3 충돌 해결

개발자가 직접 코드를 정리하여 충돌을 해결해야 한다. 두 개의 print 메소드 모두 반영해 보자.

vim 일반모드에서 dd키를 연타하여 코드가 아닌 병합 충돌메시지 부분을 한 줄씩 삭제한다.

|  |  |
| --- | --- |
| vim hello.py > enter   |  | | --- | | print("hello world")  print("hello world hotfix")  **print("hello world master")**  **print("hello world 2nd hotfix")** |   esc > :wq > enter |

### 10.4 충돌 해결 후 커밋

|  |
| --- |
| $ git status  On branch master  You have unmerged paths.  (fix conflicts and run "git commit")  Unmerged paths:  (use "git add <file>..." to mark resolution)  both modified: hello.py  no changes added to commit (use "git add" and/or "git commit -a")  Chris@starrose MINGW64 ~/git\_tutorial (master|MERGING)  $ git commit -am 'conflict resolved'  [master 87e271f] conflict resolved  Chris@starrose MINGW64 ~/git\_tutorial (master)  $ |

**master|MERGING**

마스터 브랜치에서 병합을 시도했는데 병합작업이 완료되지 않았고 진행상태임을 나타낸다.

즉, 충돌이 발생하여 병합을 완료하지 못하고 있는 상태를 나타낸다.

충돌된 코드를 정리하여 해결한 후 커밋을 수행하면 master|MERGING은 master로 바뀐다.

## 11. 로그 분석

### 11.1 graph

콘솔 모드에서 커밋정보를 그래프를 사용하여 보여준다.

|  |
| --- |
| $ git log --graph  \* commit 87e271ff70f8b51f31d029dacc2b5345a85d2a37  |\ Merge: 3adaac6 66c5d15  | | Author: softcontext <softcontext@gmail.com>  | | Date: Mon Apr 18 15:55:53 2016 +0900  | |  | | conflict resolved  | |  | \* commit 66c5d159d21f12929e87f90e2cd6de75f53453df  | | Author: softcontext <softcontext@gmail.com>  | | Date: Mon Apr 18 15:46:07 2016 +0900  | |  | | added output "hello world 2nd hotfix"  | |  \* | commit 3adaac6e2ef736e4680249d3208670da7b9da04c  |/ Author: softcontext <softcontext@gmail.com>  | Date: Mon Apr 18 15:39:35 2016 +0900  |  | added output "hello world master"  |  \* commit 8d521901684e4c5445a60376e578de64cc61f096  | Author: softcontext <softcontext@gmail.com>  | Date: Mon Apr 18 15:27:11 2016 +0900  |  | added output "hello world hotfix"  |  \* commit 085cf42d1f136132e76e2358f1816a868bfb9c8a  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:03:43 2016 +0900  'create "hello world" program' |

### 11.2 stat

콘솔 모드에서 커밋의 상세정보를 보여준다.

|  |
| --- |
| $ git log --stat  commit 87e271ff70f8b51f31d029dacc2b5345a85d2a37  Merge: 3adaac6 66c5d15  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:55:53 2016 +0900  conflict resolved  commit 66c5d159d21f12929e87f90e2cd6de75f53453df  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:46:07 2016 +0900  added output "hello world 2nd hotfix"  hello.py | 1 +  1 file changed, 1 insertion(+)  commit 3adaac6e2ef736e4680249d3208670da7b9da04c  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:39:35 2016 +0900  added output "hello world master"  hello.py | 1 +  1 file changed, 1 insertion(+)  commit 8d521901684e4c5445a60376e578de64cc61f096  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:27:11 2016 +0900  added output "hello world hotfix"  hello.py | 1 +  1 file changed, 1 insertion(+)  commit 085cf42d1f136132e76e2358f1816a868bfb9c8a  Author: softcontext <softcontext@gmail.com>  Date: Mon Apr 18 15:03:43 2016 +0900  'create "hello world" program'  hello.py | 1 +  1 file changed, 1 insertion(+) |

표시 정보가 한 화면을 넘어가면 화살키나 엔터키로 다음 정보를 볼 수 있다. 쉘 프롬프트를 바로 보고자 할 때는 ctrl+c키를 누른다.

커밋을 자주 수행하면 나중에 구분하기 쉽고 원하는 커밋지점을 바로 찾을 수 있다. 자주 그리고 주기적으로 커밋하는 것을 권장한다.

# 커밋 대상에서 제외할 파일 정보 만들기

프로젝트에는 버전관리가 필요한 파일과 그렇지 않은 파일들이 같이 존재한다.

개발 시 의존하는 외부 라이브러리 파일들은 변하지 않기 때문에 깃 저장소에 저장할 필요가 없다.

불필요한 파일이 깃 저장소에 추가되는 것을 방지하기 위해서 .gitignore 파일을 사용한다.

파일 내용으로는 커밋 대상에서 제외하고 싶은 파일들에 매핑정보를 설정한다.

점으로 시작하는 파일은 히든 파일이므로 일부 IDE툴에서 기본적으로 표시하지 않을 수 있다.

설정을 변경하면 볼 수 있으며 EGit을 사용할 때 살펴보기로 한다.

## 1. 관리파일 생성

|  |
| --- |
| $ touch .gitignore  $ ls  hello.py  $ ls **-al**  total 25  drwxr-xr-x 1 Chris 197121 0 4월 18 16:02 ./  drwxr-xr-x 1 Chris 197121 0 4월 18 15:54 ../  drwxr-xr-x 1 Chris 197121 0 4월 18 15:55 .git/  -rw-r--r-- 1 Chris 197121 0 4월 18 16:02 .gitignore  -rw-r--r-- 1 Chris 197121 113 4월 18 15:54 hello.py |

콘솔에서 파일 목록 확인 시 히든파일도 함께 보고 싶다면 -al 옵션을 사용한다.

## 2. 설정정보를 생성해 주는 웹 사용하기

### 2.1 설정 정보 자동 생성 지원 사이트

설정 작업에 필요한 문법을 외우는 것은 내키지 않는 일이다. 설정정보를 자동으로 생성해 주는 사이트를 이용해 보자.

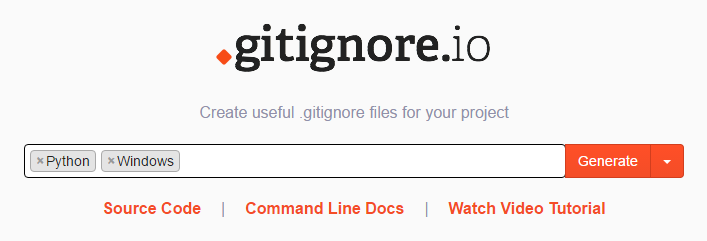
|  |
| --- |
| https://www.gitignore.io/ |

### 2.2 선택 키워드

검색창에 다음 단어를 각각 입력하고 엔터키를 치자.

|  |
| --- |
| windows  python |

generate 버튼을 클릭하여 코드를 생성한다. 결과를 복사해서 .gitignore 파일에 붙여넣는다.



## 3. 관리파일 작업결과

##### .gitignore

|  |
| --- |
| # Created by https://www.gitignore.io/api/windows,python  ### Windows ###  # Windows image file caches  Thumbs.db  ehthumbs.db  # Folder config file  Desktop.ini  # Recycle Bin used on file shares  $RECYCLE.BIN/  # Windows Installer files  \*.cab  \*.msi  \*.msm  \*.msp  # Windows shortcuts  \*.lnk  ### Python ###  # Byte-compiled / optimized / DLL files  \_\_pycache\_\_/  \*.py[cod]  \*$py.class  # C extensions  \*.so  # Distribution / packaging  .Python  env/  build/  develop-eggs/  dist/  downloads/  eggs/  .eggs/  lib/  lib64/  parts/  sdist/  var/  \*.egg-info/  .installed.cfg  \*.egg  # PyInstaller  # Usually these files are written by a python script from a template  # before PyInstaller builds the exe, so as to inject date/other infos into it.  \*.manifest  \*.spec  # Installer logs  pip-log.txt  pip-delete-this-directory.txt  # Unit test / coverage reports  htmlcov/  .tox/  .coverage  .coverage.\*  .cache  nosetests.xml  coverage.xml  \*,cover  .hypothesis/  # Translations  \*.mo  \*.pot  # Django stuff:  \*.log  local\_settings.py  # Flask instance folder  instance/  # Scrapy stuff:  .scrapy  # Sphinx documentation  docs/\_build/  # PyBuilder  target/  # IPython Notebook  .ipynb\_checkpoints  # pyenv  .python-version  # celery beat schedule file  celerybeat-schedule  # dotenv  .env  # virtualenv  venv/  ENV/  # Spyder project settings  .spyderproject  # Rope project settings  .ropeproject |

## 4. 관리파일 커밋

.gitignore 파일도 버전관리가 필요하다.

|  |
| --- |
| $ git status  On branch master  Untracked files:  (use "git add <file>..." to include in what will be committed)  .gitignore  nothing added to commit but untracked files present (use "git add" to track)  $ git add .gitignore  $ git commit -m 'added ".gitignore" file"'  [master 276f064] added ".gitignore" file"  1 file changed, 113 insertions(+)  create mode 100644 .gitignore  $ git status  On branch master  nothing to commit, working directory clean |

## 5. 깃 관리대상 제어

더 이상 깃이 파일을 관리하지 않기를 바라는 경우 다음 명령으로 처리할 수 있다.

**git rm <filename>**

파일을 삭제한다. Working Directory의 파일도 삭제되니 주의가 필요하다.

**git rm --cached <filename>**

해당 파일을 untracked 상태로 변경한다.

Working Directory의 파일은 삭제하지 않는다.

주로 .gitignore에 무시대상으로 설정하는 것을 빼먹은 경우 사용한다.

--cached 옵션 대신 --staged 옵션을 사용할 수 있다.

**git mv <origin\_file> <target\_file>**

히스토리를 유지한 상태로 파일이름을 변경한다. 해당 변경사항은 staged 상태가 된다.

다음 명령들을 수행한 것과 동일하다.

mv <origin\_file> <target\_file>

git rm <origin\_file>

git add <target\_file>

# GitHub

깃허브는 깃을 사용하는 프로젝트를 지원하는 웹 기반의 호스팅 서비스다.

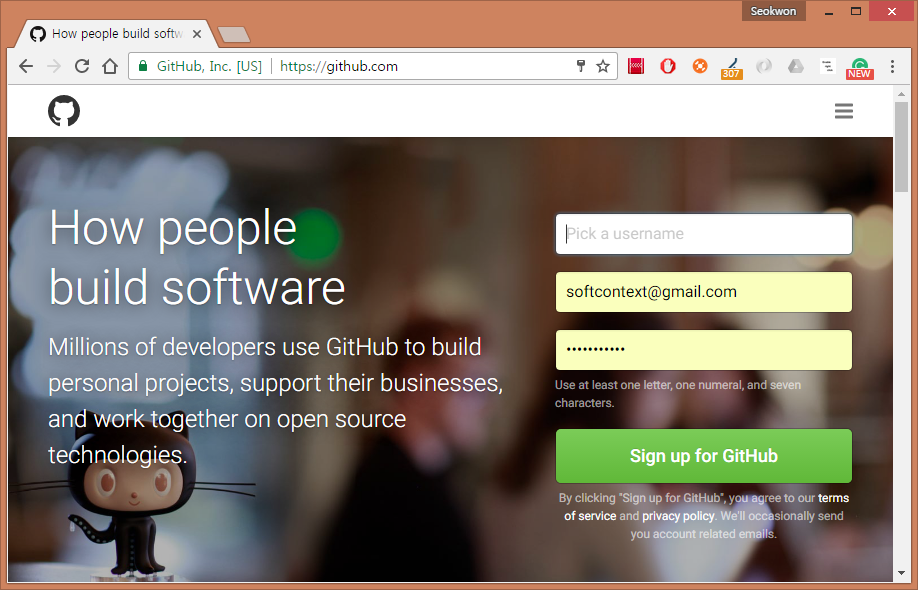
GitHub는 영리적인 서비스와 오픈소스를 위한 무상 서비스를 모두 제공한다.

가장 인기있는 오픈 소스 코드 저장소로 꼽히면 오픈 소스 프로젝트의 성지가 되었다.

## 1. 회원가입

원격 저장소인 깃허브는 무료로 사용할 수 있다. 대신 누구나 소스에 접근할 수 있는 오픈소스 정책을 따라야 한다. 폐쇄 프로젝트를 깃허브를 통해 운영하고자 할 경우에는 별도로 요금을 내야 한다.

|  |
| --- |
| https://github.com |



"Sign up for GitHub" 버튼을 클릭하여 회원가입을 한다.

# 복제: 원격 저장소 🡪 로컬 저장소

여러 개발자가 협업을 하기 위해서는 언제든지 접근할 수 있는 서버가 필요하다.

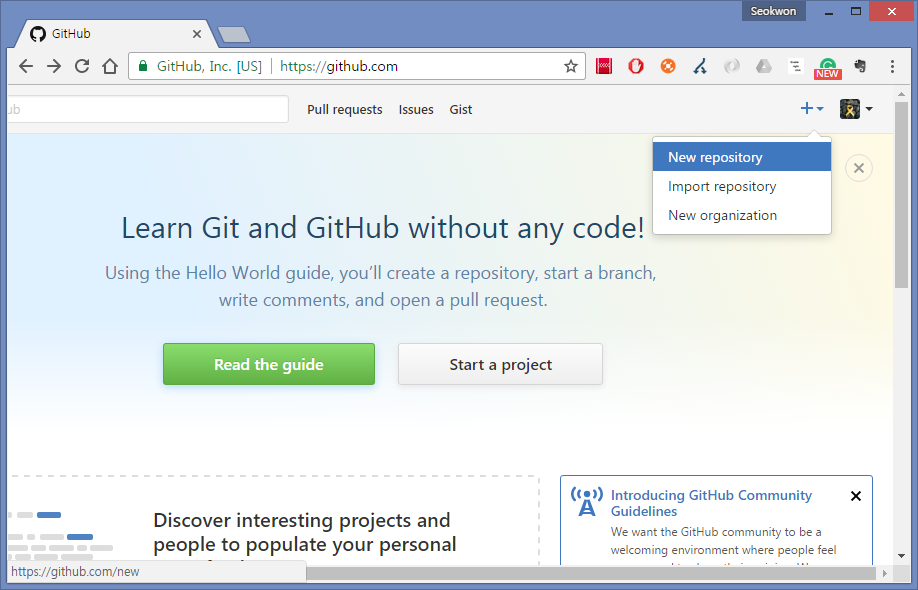
깃허브를 버전관리의 원격 서버로 사용한다.

로컬 저장소와 구분하기 위해서 깃허브를 원격(remote 또는 origin) 저장소라 부른다.

원격 저장소(깃허브)에 프로젝트를 로컬(내 컴퓨터) 저장소에 복제해 보자.

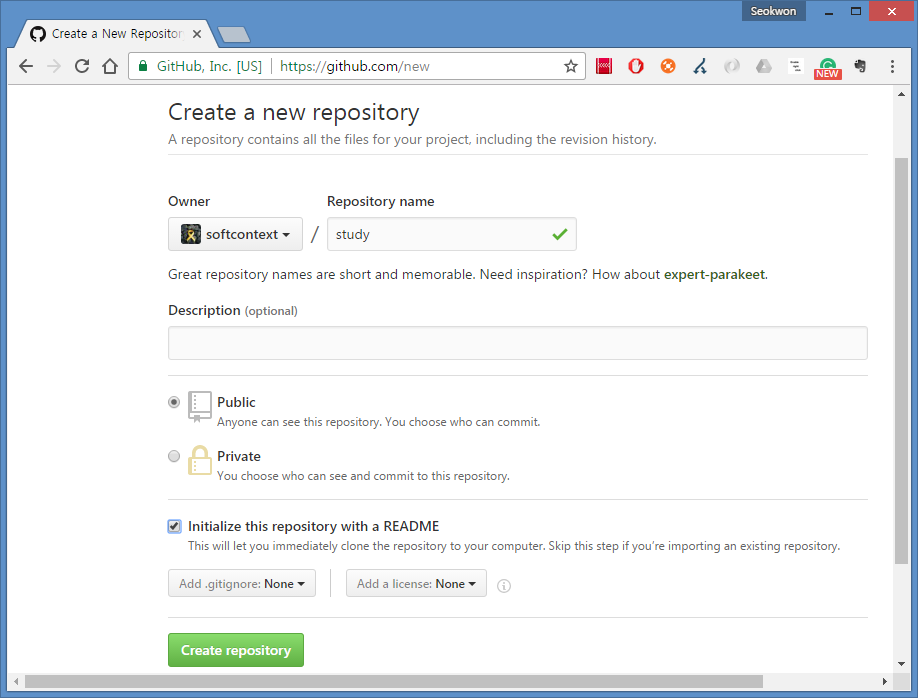
## 1. 원격 저장소 만들기

"New repository"를 클릭한다.

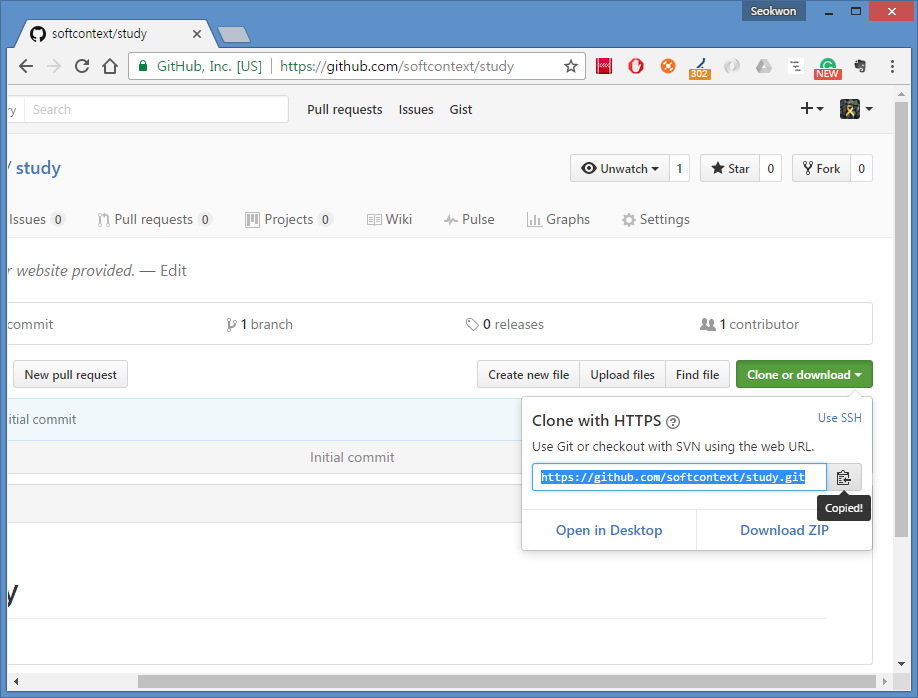


1. "Repository name" 입력란에 "study" 라고 저장소 이름을 입력한다.

2. "Initialize this repository with a README" 체크박스를 체크하여 README.md 파일을 생성한다.



3. 저장소가 만들어진 후 https 접근주소를 복사한다.



## 2. 복제(Clone)

원격저장소의 프로젝트를 그대로 로컬저장소에 복제해 보자.

|  |
| --- |
| $ mkdir github\_tutorial  $ cd github\_tutorial  $ **git clone https://github.com/softcontext/study.git**  Cloning into 'study'...  remote: Counting objects: 3, done.  remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0  Unpacking objects: 100% (3/3), done.  Checking connectivity... done.  $ ls  **study/**  $ cd study  $ ls  README.md  $ git status  On branch master  Your branch is up-to-date with '**origin**/master'.  nothing to commit, working directory clean |

origin은 원격(remote) 저장소의 별칭이다.

# 연결: 로컬 저장소 🡪 원격 저장소

여러명의 개발팀원들이 협업하는 프로젝트의 공유 시나리오를 살펴보자.

1. 개발팀장이 먼저 로컬 저장소에 프로젝트의 스캐폴딩을 만든다.

이 때 프로젝트에 필요한 기본 환경설정을 모두 마친다.

2. 개발팀장은 다른 개발 협업자에게 프로젝트를 공유해 주기 위해서 원격 저장소에 올린다.

3. 협업 개발자들은 원격저장소에 프로젝트를 자신의 로컬저장소에 복제한 후 개발을 진행한다.

## 1. 로컬 프로젝트 만들기

앞서서 만든 git\_tutorial 프로젝트를 그대로 사용한다.

## 2. 로컬 저장소에 커밋하기

앞서서 만든 git\_tutorial 프로젝트의 로컬 저장소를 그대로 사용한다. 이미 커밋이 되어 있다.

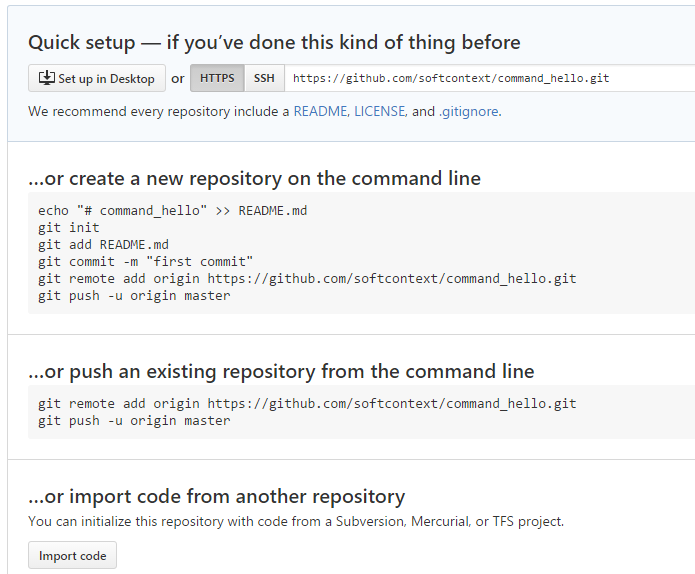
## 3. 원격 저장소생성

원격 저장소의 이름을 command\_hello라고 만든다.

**주의 : README.md 파일은 생성하지 않는다.**

로컬 저장소에 프로젝트를 그대로 원격 저장소에 올리고자 할 때 깃 저장소 내역이 다르면 깃허브가 거부하므로 처음에는 README.md 파일을 생성하지 않도록 한다.

README.md 파일을 만들지 않고 저장소를 생성하면 다음 화면을 보여준다. 파일을 로컬 저장소에서 만들고 원격저장소에 반영하는 작업과정을 알려주고 있다. 다음 과정을 학습한 후에 다시와서 살펴보면 어떤 작업과정을 통해서 처리하는지 알게 될 것이다.



## 4. Remote: 연결

원격 저장소에 연결해 보자.

|  |
| --- |
| Chris@starrose MINGW64 ~/git\_tutorial (master)  $ ls  hello.py  Chris@starrose MINGW64 ~/git\_tutorial (master)  $ **git remote add origin https://github.com/softcontext/command\_hello.git**  Chris@starrose MINGW64 ~/git\_tutorial (master)  $ git remote -v  origin https://github.com/softcontext/command\_hello.git (fetch)  origin https://github.com/softcontext/command\_hello.git (push) |

**add origin**

원격 저장소를 등록한다. 등록 시 원격 저장소의 별칭을 origin으로 설정한다.

**https://github.com/softcontext/command\_hello.git**

연결하고자 하는 원격 저장소의 주소를 지정한다.

## 5. Push: 로컬 작업 내역을 원격 저장소에 올리기

|  |
| --- |
| $ **git push origin --all**  Counting objects: 18, done.  Delta compression using up to 4 threads.  Compressing objects: 100% (11/11), done.  Writing objects: 100% (18/18), 2.09 KiB | 0 bytes/s, done.  Total 18 (delta 3), reused 0 (delta 0)  To https://github.com/softcontext/command\_hello.git  \* [new branch] hotfix -> hotfix  \* [new branch] master -> master |

**push origin --all**

origin에 현재 브랜치의 모든 파일을 push 한다. 사용자 계정정보를 환경설정으로 등록하지 않았다면 계정을 물어보는 창이 뜰 것이다. 깃허브에 회원가입 시 사용한 아이디와 패스워드를 사용하면 된다.

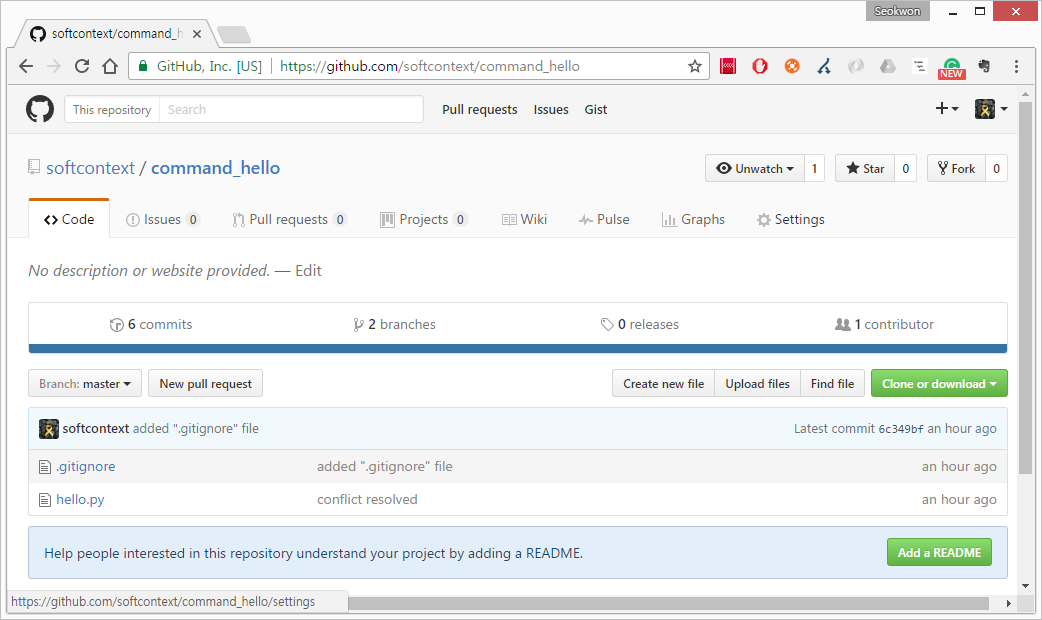
푸시작업을 진행할 때, 원격 저장소에 이미, 로컬 저장소의 브랜치와 같은 이름의 브랜치가 있다면 해당 브랜치를 작업대상으로 삼는다.

단, 내역이 다르다면 푸시를 거부한다. 충돌이 발생하는 것이다. 이를 막기 위해서 원격 저장소 생성 시 README.md 파일을 만들지 않았다. 원격 저장소에 아무것도 없는 상태라면 충돌이 발생하지 않기 때문이다.

로컬 저장소의 브랜치와 같은 이름의 브랜치가 원격 저장소에 없다면 브랜치를 만든 후 푸시한다.

## 6. 푸시 확인

로컬 저장소에 컨텐츠가 원격 저장소에 잘 복제(push) 되었는지 깃허브에 접속하여 확인한다.



## 7. 로컬에서 파일 생성 후 커밋

로컬 저장소에서 README.md 파일을 만들고 원격 저장소에 추가해 보자.

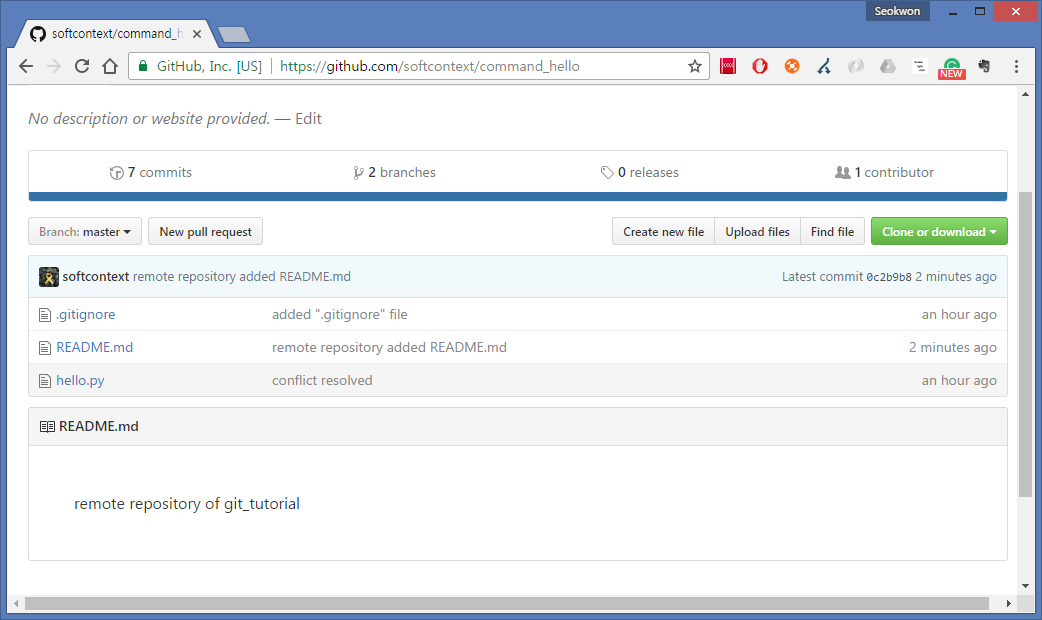
|  |
| --- |
| $ cat >> README.md  remote repository of git\_tutorial  [여기서 ctrl+c 키를 누른다.]  $ git add README.md  $ git commit -am 'remote repository added README.md' |

주의 : 원격 저장소에 올리기 전에 항상 커밋을 먼저 해야 함을 상기하자.

## 8. 동기화: 로컬 저장소에 있는 파일을 원격 저장소에 올리기

|  |
| --- |
| $ **git push origin master**  Counting objects: 3, done.  Delta compression using up to 4 threads.  Compressing objects: 100% (2/2), done.  Writing objects: 100% (3/3), 362 bytes | 0 bytes/s, done.  Total 3 (delta 0), reused 0 (delta 0)  To https://github.com/softcontext/command\_hello.git  276f064..220aa07 master -> master |

origin(원격저장소)에 master(로컬저장소의 브랜치)의 컨텐츠를 push한다.



# 동기화: 원격 저장소 🡨🡪 로컬 저장소

프로젝트를 공유하여 개발이 시작되었다고 가정하자. 여러 개발자들이 작업하는 폴더 및 파일을 구분하여 작업을 수행하면 충돌이 발생하지는 않는다. 하지만 때에 따라서는 같은 파일을 여러 개발자가 동시에 수정해야 할 필요가 있을 수도 있다. 이런 경우 원격 저장소와 로컬 저장소간의 병합작업 시 충돌이 발생할 수 있다.

## 1. 원격 저장소와 로컬 저장소의 충돌 시나리오

협업으로 프로젝트를 진행할 때, 다른 협업자가 먼저 프로그램을 수정한 후 원격 저장소에 커밋했다고 가정하자. 그 후, 나도 프로그램을 수정한 후 로컬저장소에 커밋하고 원격 저장소에 푸시하고자 한다.

이 때, 푸시를 시도하면 충돌이 발생하게 되어 나의 푸시는 허용되지 않는다.

## 2. 충돌 해결 순서

1. 먼저 내가 원격 저장소의 내용을 로컬 저장소로 가져온다.

2. 충돌을 해결하고 로컬 저장소에 커밋한다.

3. 다음으로 로컬 저장소의 내용을 원격 저장소에 푸시한다.

## 3. 원격 저장소의 내용을 로컬 저장소로 가져오는 2가지 방법

### fetch

원격 저장소의 커밋들을 로컬 저장소로 가져온다.

**개발자가 직접 로컬 저장소의 커밋과 가져온 원격 저장소의 커밋을 병합해야 한다.**

충돌이 발생한다면 개발자가 직접 충돌을 해결하고 커밋해야 한다.

### pull

원격 저장소의 커밋들을 로컬 저장소로 가져온다.

이 때, **자동으로 로컬 브랜치에 병합하는 작업이 이루어 진다.**

충돌이 발생해도 알려주지 않는다. 따라서, 어떤 내용이 병합되었는지 파악하기 힘들다.

보통은 권장하지 않는 방법이다.

충돌이 확실히 없다고 판단할 수 있는 경우 사용한다.

나는 작업한 내용이 없고 다른 개발자가 변경한 내용이 있다고 판단될 때 사용한다.

# 원격과 로컬의 충돌 시나리오 테스트

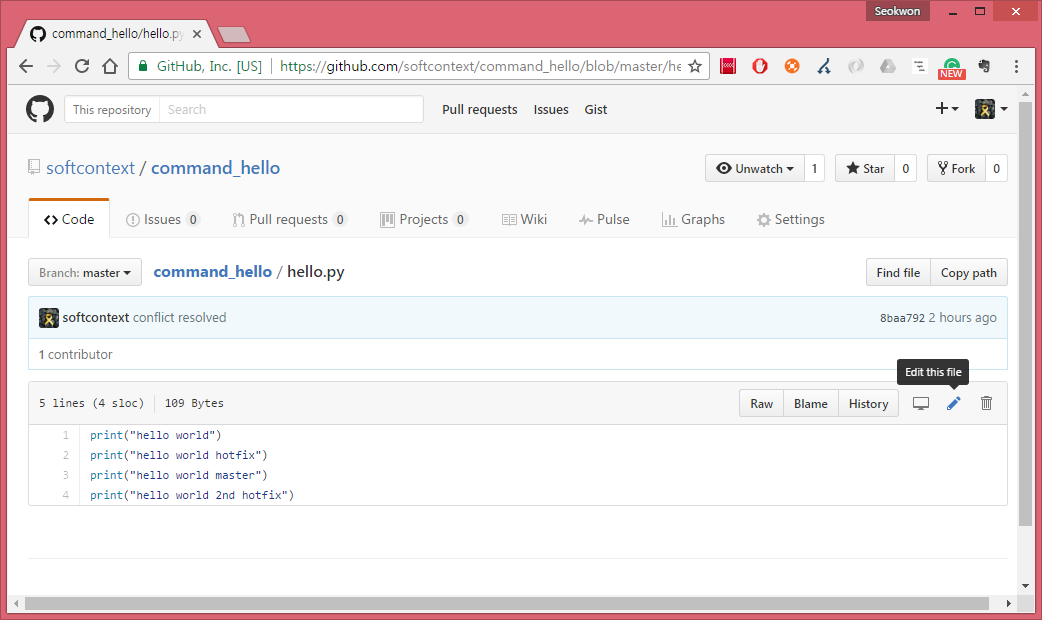
## 1. 원격 저장소 연결확인

|  |
| --- |
| $ git remote -v  origin https://github.com/softcontext/command\_hello.git (fetch)  origin https://github.com/softcontext/command\_hello.git (push) |

## 2. github에서 직접 파일 수정 후 커밋

다른 개발자가 먼저 파일을 수정 후 커밋한 상태로 만들어 놓기 위한 작업을 진행한다.

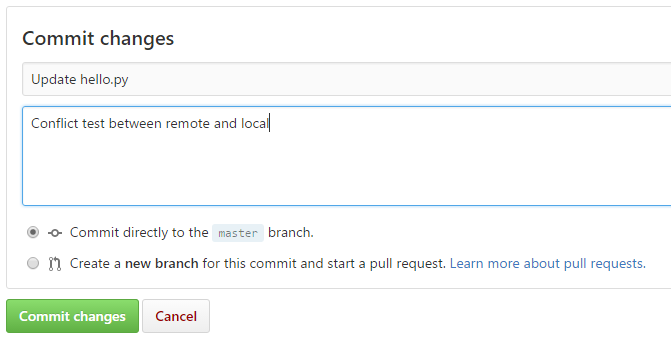
깃허브에 접속하여 작업한다.



다음처럼 간단히 주석을 한 줄 추가한다.

hello.py

|  |
| --- |
| **// github modification**  print("hello world")  print("hello world hotfix")  print("hello world master")  print("hello world 2nd hotfix") |



커밋 메시지를 작성하고 "Commit changes" 버튼을 클릭한다.

## 3. 로컬에서 파일 수정 후 커밋

이번에는 로컬에서 대상 파일을 수정한 상태로 만들기 위해서 주석을 한 줄 추가한다.

hello.py

|  |
| --- |
| **// local modification**  print("hello world")  print("hello world hotfix")  print("hello world master")  print("hello world 2nd hotfix") |

로컬 저장소에 커밋하기

|  |
| --- |
| $ git commit -a -m 'hello.py modified on local'  [master f5a01ae] hello.py modified on local  1 file changed, 1 insertion(+) |

## 4. Push 시도, 충돌 확인

푸시가 거부되면 거부 사유를 알려준다.

|  |
| --- |
| $ **git push origin master**  To https://github.com/softcontext/command\_hello.git  ! [**rejected**] master -> master (fetch first)  error: failed to push some refs to 'https://github.com/softcontext/command\_hello.git'  hint: Updates were rejected because the remote contains work that you do  hint: not have locally. This is usually **caused by another repository pushing**  hint: **to the same ref**. You may want to first integrate the remote changes  hint: (e.g., 'git pull ...') before pushing again.  hint: See the 'Note about fast-forwards' in 'git push --help' for details. |

## 5. 충돌해결 1단계: Fetch

이미 원격 저장소에 연결되어 있으므로 간단히 fetch라는 명령으로 가져올 수 있다.

|  |
| --- |
| $ git fetch  remote: Counting objects: 3, done.  remote: Compressing objects: 100% (3/3), done.  remote: Total 3 (delta 2), reused 0 (delta 0), pack-reused 0  Unpacking objects: 100% (3/3), done.  From https://github.com/softcontext/command\_hello  220aa07..00259a0 master -> origin/master |

## 6. 충돌해결 2단계: origin/master 🡪 master 병합

원격 저장소의 master 브랜치를 로컬 저장소의 master 브랜치로 병합한다. 이 때 충돌이 발생한다.

|  |
| --- |
| $ git branch -a  hotfix  \* master  remotes/origin/hotfix  remotes/origin/master  $ **git merge origin/master**  Auto-merging hello.py  **CONFLICT (content): Merge conflict in hello.py**  Automatic merge failed; fix conflicts and then commit the result. |

## 7. 충돌 확인

|  |
| --- |
| $ git diff  **diff --cc hello.py**  **index 908580b,de7e791..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,4 -1,4 +1,8 @@@  ++<<<<<<< HEAD  +// local modification  ++=======  + // github modification  ++>>>>>>> origin/master  print("hello world")  print("hello world hotfix")  print("hello world master") |

**++<<<<<<< HEAD**

**+// local modification**

현재 HEAD는 로컬 master 브랜치를 가리키고 있다. 따라서 로컬 저장소에 코드내역에 해당한다.

**++=======**

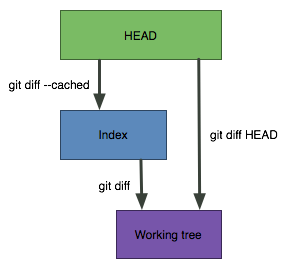
구분선

**+ // github modification**

**++>>>>>>> origin/master**

origin/master가 갖고 있는 파일의 코드내역에 해당한다.

## diff 명령 옵션 살펴보기



**git diff**

Working Directory와 staged 영역인 Index의 차이를 비교한다.

**git diff --cached**

staged 영역과 커밋 간의 차이를 비교한다.

git diff --staged 명령과 동일하다.

**git diff HEAD**

커밋(인덱스 포함)과 Working Directory 사이의 차이를 비교한다.

## 8. 충돌 해결 후 커밋

hello.py 파일을 열어 코드가 아닌 부분을 제거한다.

|  |
| --- |
| // local modification  // github modification  print("hello world")  print("hello world hotfix")  print("hello world master")  print("hello world 2nd hotfix") |

로컬에서 커밋을 수행한다.

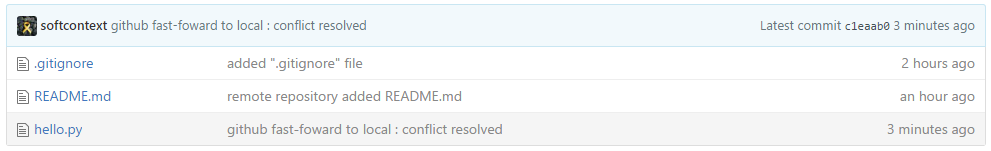
|  |
| --- |
| $ git commit -a -m 'github fast-forward 2 loocal: conflict resolved'  [master 1474bd0] github fast-forward 2 loocal: conflict resolved |

깃허브의 원격 저장소가 로컬 저장소보다 앞섰고 충돌이 발생한 것을 해결하였다고 기록하였다.

## 9. 푸시 및 확인

다시 로컬 저장소에서 원격 저장소로 푸시를 시도한다. 성공적으로 푸시 되었음을 확인하기 위해서 깃허브에 접속하여 확인한다.

|  |
| --- |
| $ git push origin master  Counting objects: 6, done.  Delta compression using up to 4 threads.  Compressing objects: 100% (6/6), done.  Writing objects: 100% (6/6), 581 bytes | 0 bytes/s, done.  Total 6 (delta 4), reused 0 (delta 0)  To https://github.com/softcontext/command\_hello.git  00259a0..1474bd0 master -> master |



# EGit

EGit은 깃을 이클립스에서 사용할 수 있도록 이클립스 개발팀이 지원하는 툴이다.

## 설치

Spring STS를 설치하면 Git 플러그인이 이미 설치되어 있는 상태이므로 바로 사용할 수 있다.

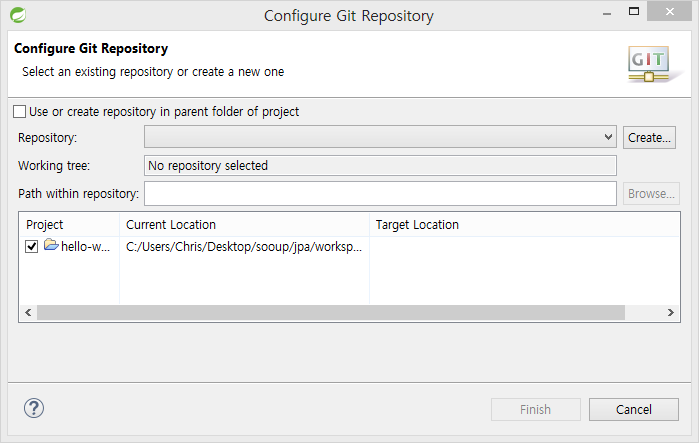
Git 퍼스펙티브를 추가한다.

## 프로젝트 생성

hello-world-git 이름으로 자바 프로젝트를 만든다.

## 저장소 생성

프로젝트를 선택 > 마우스 오른쪽 클릭 > Team > Share Project 클릭



create 버튼을 클릭한다. **hello-world-git** 이름으로 폴더를 추가한다.

프로젝트 파일의 위치와 로컬 저장소에 위치를 구분할 것이다. 같은 위치를 사용해도 무방하다.

C:\Users\Chris\git\**hello-world-git**

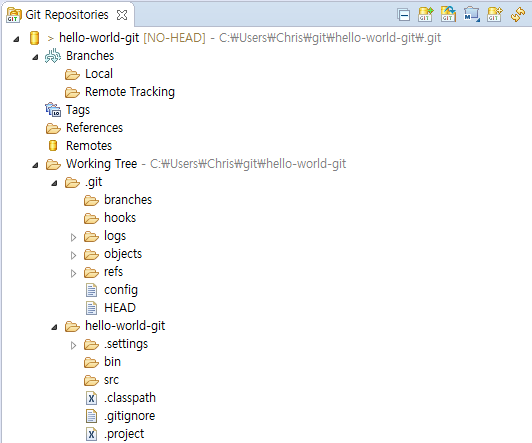
Finish 버튼을 클릭한다.



프로젝트 이름 옆에 저장소 정보가 표시된다. 아직 커밋 정보는 존재하지 않는다.

## Git 관련 정보 확인

Git 퍼스펙티브로 이동한다. 이클립스 화면 왼쪽에서 깃 리파지토리 뷰를 살펴본다.



Working Tree 하위를 살펴보자.

.git : 프로젝트의 로컬 저장소

hello-world-git : 프로젝트 폴더

### 버전관리가 필요 없는 파일 및 폴더를 설정하기

프로젝트 루트에 .gitignore 파일을 연다.

.gitignore

|  |
| --- |
| /bin/ |

bin 폴더가 기본으로 설정되어 있어 컴파일하면 생성되는 바이너리 파일들은 저장소에 포함되지 않도록 설정되어 있음을 알 수 있다.

https://www.gitignore.io 에 접속하여 다음 키워드를 사용하여 설정내용을 구한다.

|  |
| --- |
| windows  eclipse  java |

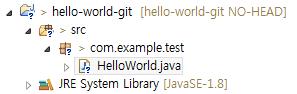
# Egit 사용법

## 프로그램 파일 작성

HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  System.out.println("hello world");  }  } |

파일을 저장하면 물음표가 표시된다. 물음표는 파일이 추적중이 아닌 상태라고 알려주는 표시다.



물음표가 붙은 파일을 추적 상태로 만들고 깃 로컬 저장소에 커밋해야 한다.

## 커밋 대상을 추가하면서 커밋하기

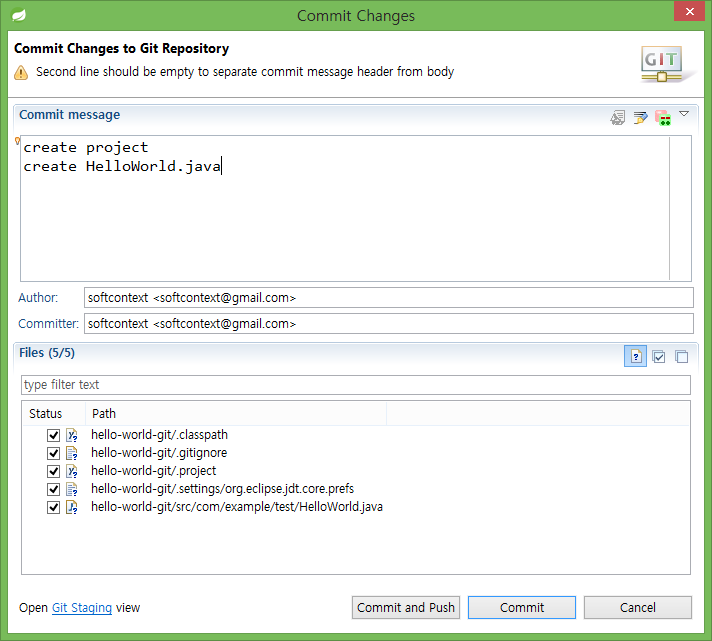
이깃은 대상 파일을 추적 대상으로 add 하면서 바로 커밋을 수행할 수 있는 기능을 제공한다.

프로젝트 선택 > 마우스 오른쪽 버튼 클릭 > Team > Commit 클릭

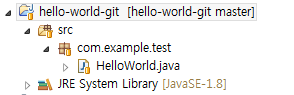
1. 커밋 메시지를 작성한다.

2. 커밋 대상 파일을 선택한다.

3. Commit 버튼을 클릭한다.



파일이 추적 중 이라는 의미로 노란 원통 아이콘이 표시된다.



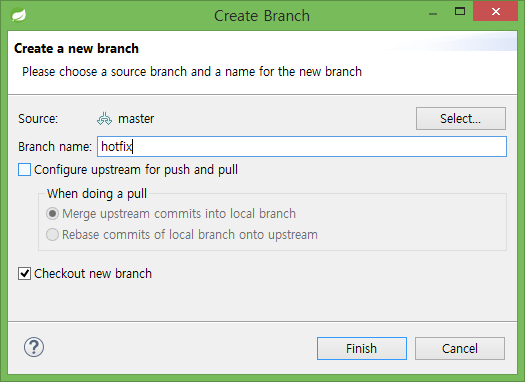
## 새로운 브랜치를 생성하면서 체크아웃하기

브랜치를 새로 만들어 보자.

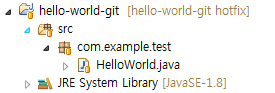
Team > Switch To > New Branch 클릭

1. 브랜치 이름은 hotfix로 한다.

2. Finish 버튼을 클릭한다.



새로 만든 브랜치로 체크아웃이 된 상태가 된다. 프로젝트 이름 다음에 표시되는 정보를 확인하자.



## 새 브랜치에서 코드 수정 후 커밋하기

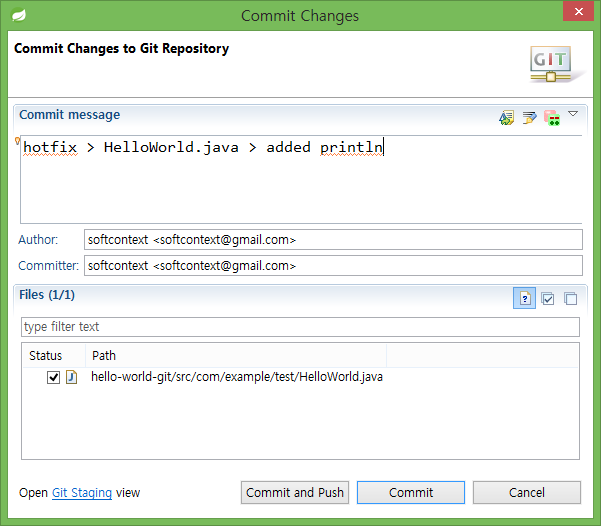
HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  System.out.println("hello world");  **System.out.println("hello hotfix!");**  }  } |

Team > Commit 클릭

1. 커밋 메시지를 작성한다.

2. Commit 버튼을 클릭한다.



## master 브랜치에 hotfix 브랜치를 병합하기

### 작업 브랜치 변경

마스터 브랜치로 체크아웃한다.

Team > Switch To > master 클릭

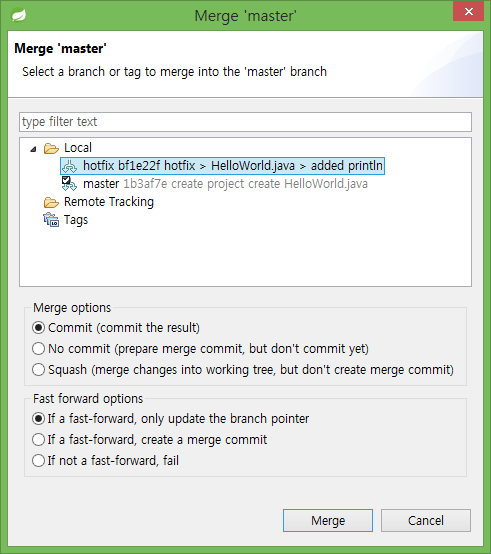
마스터 브랜치에 HelloWorld.java 파일의 내용은 마지막으로 커밋한 내용 그대로임을 확인한다.

### 병합: hotfix 🡪 master

Team > Merge 클릭

병합대상 브랜치 hotfix를 선택한다.

Merge 버튼을 클릭한다.

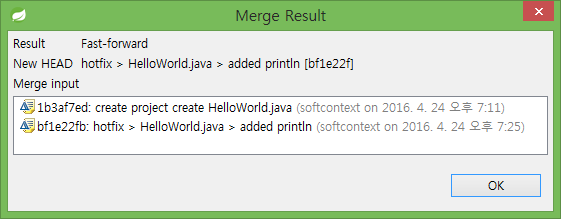


**If a fast-forward, only update the branch pointer**

브랜치 hotfix가 더 최신내용을 담고 있다면, 브랜치 포인터를 업데이트 한다.

브랜치 hotfix는 타겟 브랜치 master의 모든 내용을 포함하고 있어야 fast-forward 상태가 된다.

### 병합 결과 확인



### 병합 내용 확인

HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  System.out.println("hello world");  System.out.println("hello hotfix!");  }  } |

## 충돌 시나리오 테스트

### 1. master 브랜치에서 프로그램 수정 후 커밋

HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  **// master > added comment**  System.out.println("hello world");  System.out.println("hello hotfix!");  }  } |

### 2. hotfix 브랜치에서 프로그램 수정 후 커밋

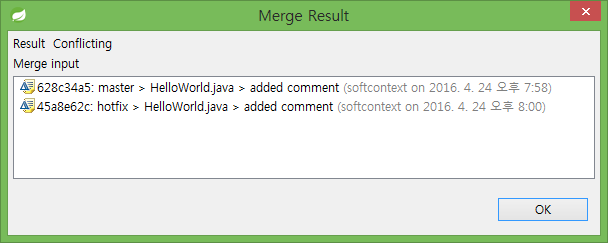
HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  **// hotfix > added comment**  System.out.println("hello world");  System.out.println("hello hotfix!");  }  } |

### 3. 병합 시도 및 충돌 확인

Team > Merge 클릭

충돌 메시지를 확인하고 OK 버튼을 클릭하여 창을 닫는다.



### 4. 충돌 해결

HelloWorld.java

|  |
| --- |
| package com.example.test;  public class HelloWorld {  public static void main(String[] args) {  **<<<<<<< HEAD**  // master > added comment  **=======**  // hotfix > added comment  **>>>>>>> refs/heads/hotfix**  System.out.println("hello world");  System.out.println("hello hotfix!");  }  } |

두 브랜치에서 추가한 내용을 모두 포함하는 형태로 충돌 부분을 해결한다.

### 5. 충돌 해결을 표시(통보)

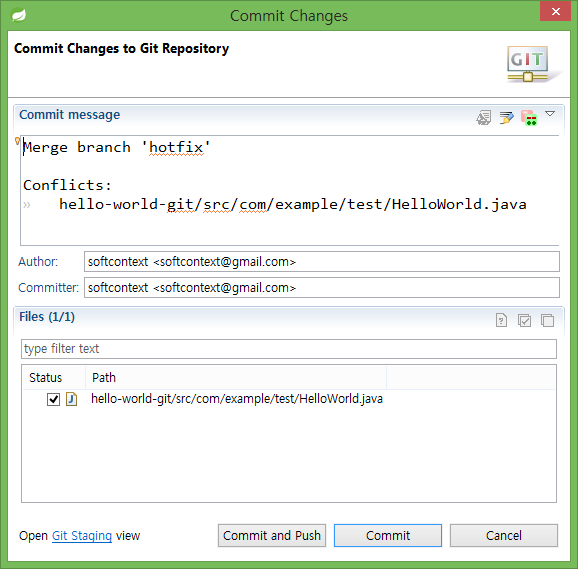
작업 대상 파일 선택 > 마우스 오른쪽 버튼 클릭 > Team > Add to Index 클릭

이제 패키지 익스플로러에 충돌 표시가 사라질 것이다.

### 6. 커밋

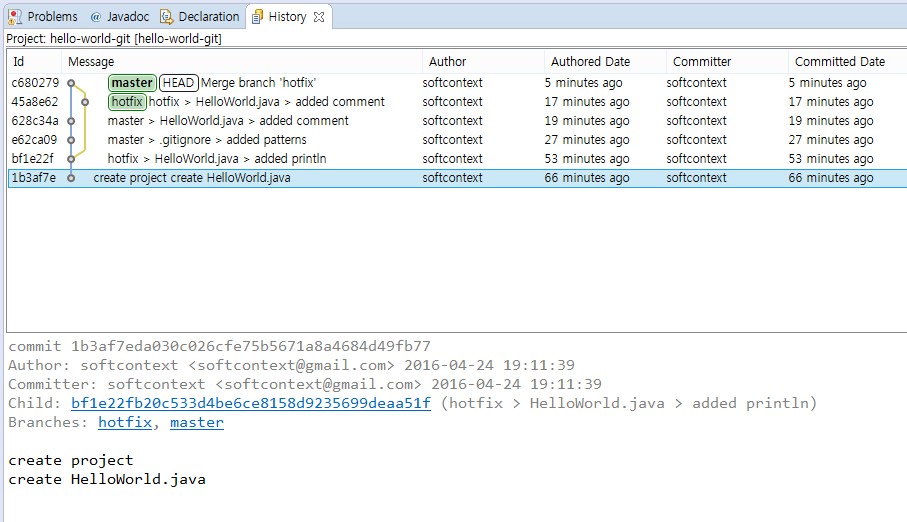
Team > Commit 을 선택하면 자동으로 작성된 커밋 메시지를 볼 수 있다.

필요한 메시지를 보강하고 Commit 버튼을 클릭한다.



## 작업 로그

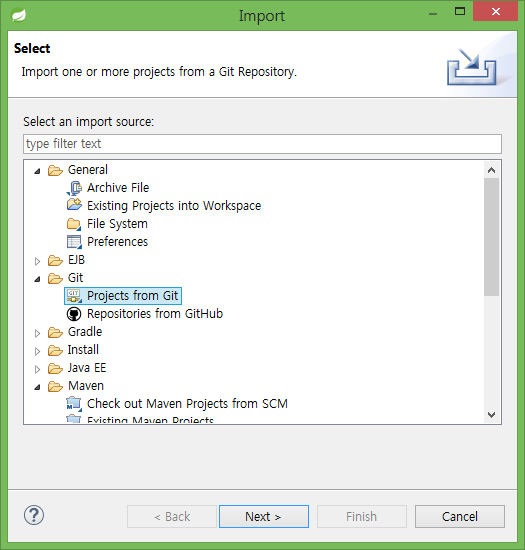
Team > Show in History 클릭



# 원격 저장소를 로컬 저장소에 클론(복제)하기

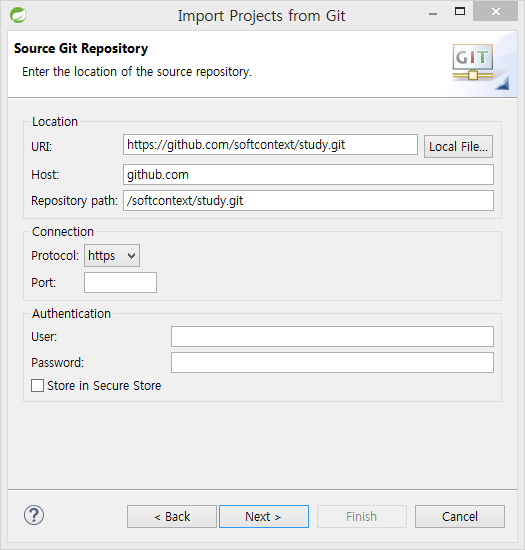
### 1. 프로젝트 Import

- File > Import > Git > Projects from Git



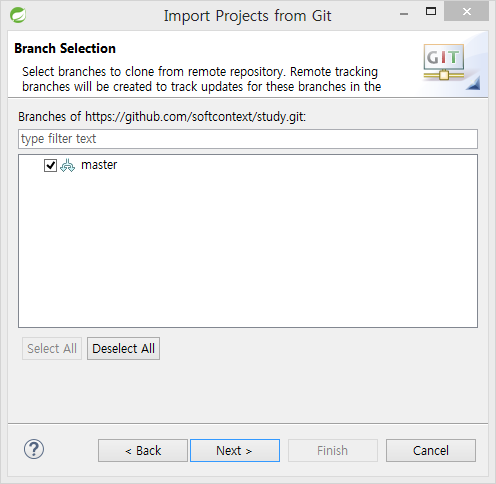
### 2. Next > Clone URI > Next

URI 정보는 깃허브에서 주소를 복사해서 사용한다.



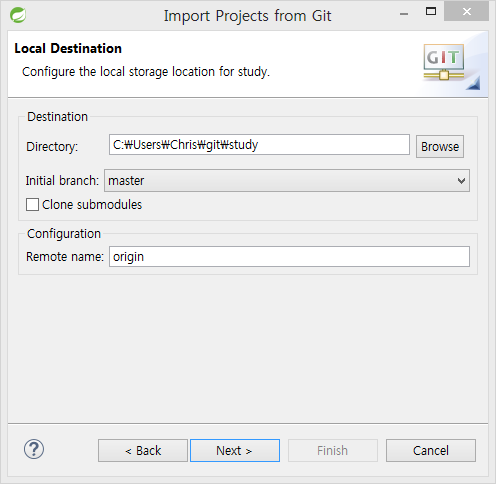
### 3. 브랜치 선택 창

깃허브에는 브랜치가 하나밖에 없어서 추가로 선택할 브랜치가 보이지 않고 있다.



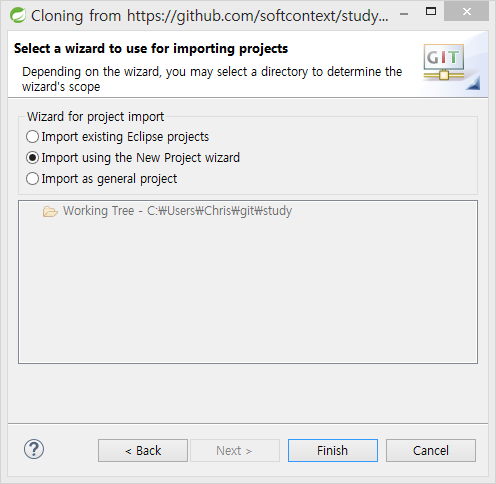
### 4. 로컬 저장소 위치 선택

Clone submodules: 해당 원격 저장소가 프로젝트에서 사용되는 라이브러일 경우 체크한다.

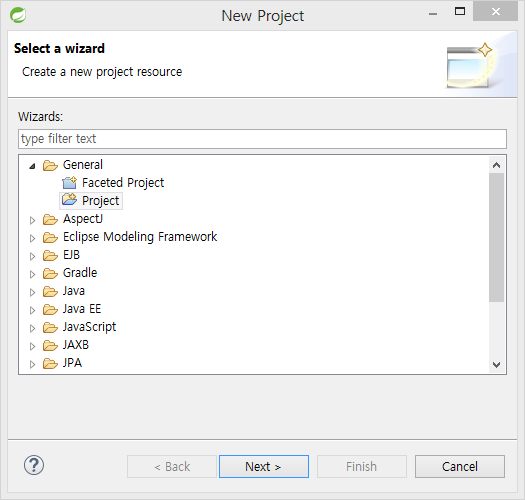


### 5. 새로운 프로젝트 생성 마법사 선택

앞선 작업으로 깃허브 로컬저장소 처리는 완료되었다. 이제 부터는 프로젝트 관련 설정이다.



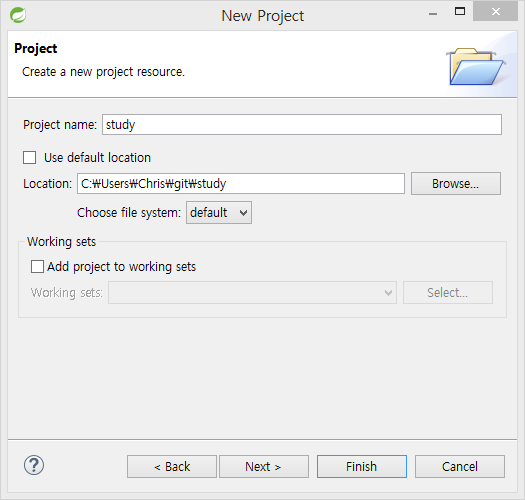
불러오는(복제해서 새로 만드는) 프로젝트에 알맞는 카테고리를 선택한다.



### 6. 새 프로젝트 디렉토리 지정

깃허브 저장소의 위치와 프로젝트 루트의 위치가 다르므로 Use default location 선택을 해제한다.

Browse 버튼을 클릭하여 깃 저장소 폴더를 선택한다.



### 7. 클론작업 완료확인

이클립스에서 새로 만들어진 프로젝트를 확인한다.



# 로컬 저장소와 원격 저장소를 연결하기

앞서서 만든 이클립스 프로젝트 hello-world-git을 대상으로 작업한다.

### 1. 깃허브에 저장소 만들기

eclipse\_hello라는 이름으로 생성한다.

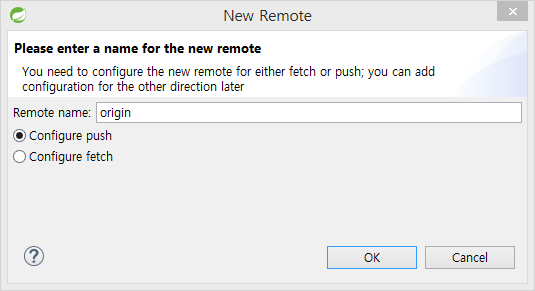
README.md 파일은 만들지 않는다. 로컬 저장소의 내역을 원격 저장소에 푸시할 것이다.

### 2. 원격 저장소 연결

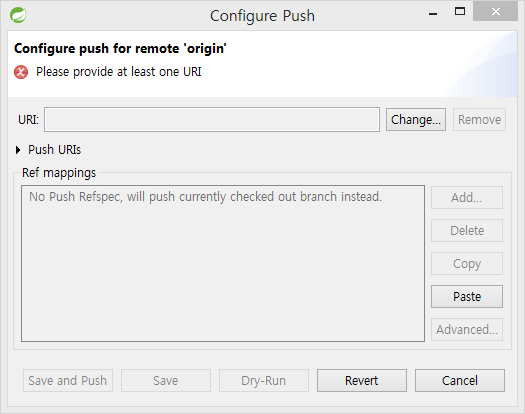
깃 퍼스펙티브로 이동한다.

hello-world-git 하부에 Remotes를 선택 > 마우스 오른쪽 버튼 클릭 > Create Remote 클릭

새로 뜨는 확인창에 정보를 확인하고 OK 버튼을 클릭한다.



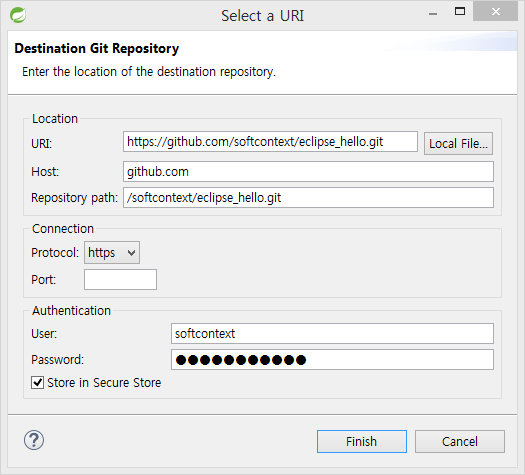
Change 버튼을 클릭한다.



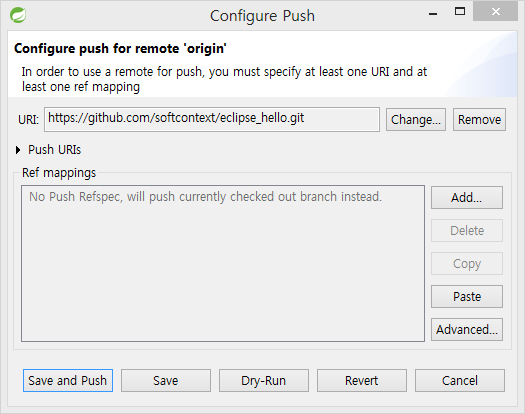
### 3. 깃허브 저장소 정보 설정

깃허브 저장소 주소를 복사해 URI 입력창에 붙여 넣는다.

깃허브 계정정보를 입력하고 Finish 버튼을 클릭한다.



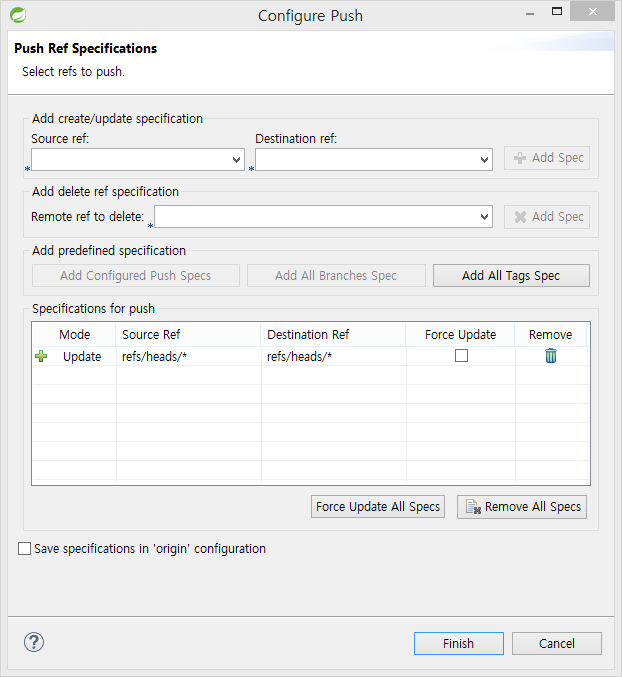
새로 뜨는 Configure Push 창에서 Advanced 버튼을 클릭한다.



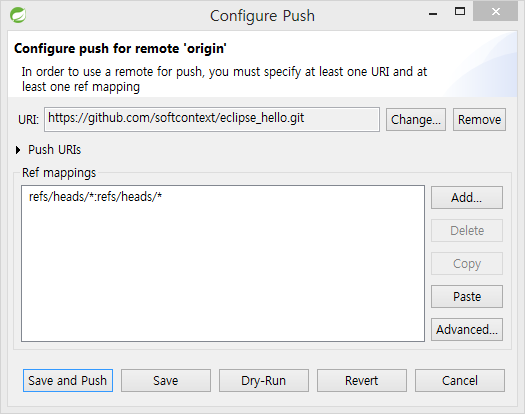
### 4. 로컬 저장소의 모든 작업 내역 선택하기

Add All Branches Spec 버튼을 클릭한다.

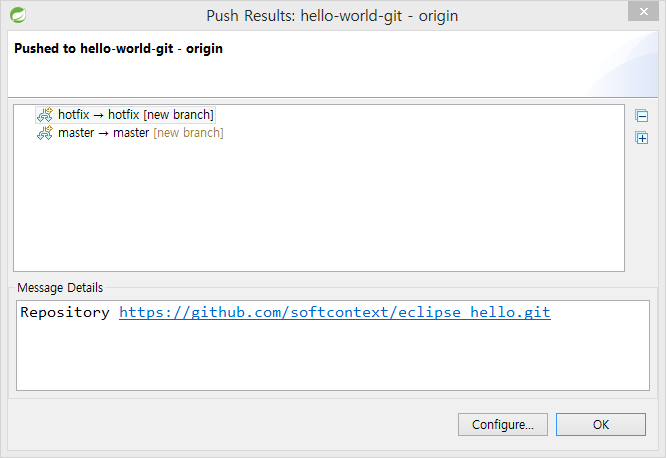
Finish 버튼을 클릭한다.



Save and Push 버튼을 클릭한다.



### 5. 푸시 결과 확인



깃허브에 접속해서 푸시 작업이 잘 이루어졌는지 확인한다.

# 로컬 작업 내역을 원격 저장소에 올리기

새로운 자바 파일을 만들고 이를 원격 저장소에 푸시해보자.

### 1. 파일 생성

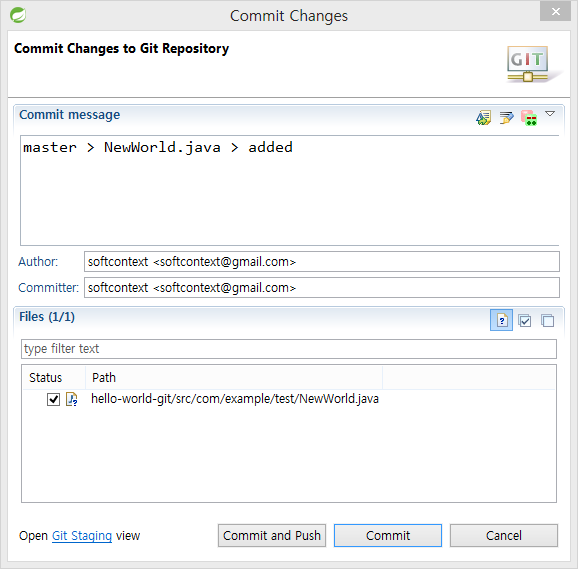
NewWorld.java

|  |
| --- |
| package com.example.test;  public class NewWorld {  public static void main(String[] args) {  System.out.println("new java file for test of push");  }  } |

### 2. 커밋

Team > Commit 클릭

커밋 메시지를 작성하고 Commit 버튼을 클릭한다.



### 3. 원격 저장소에 푸시

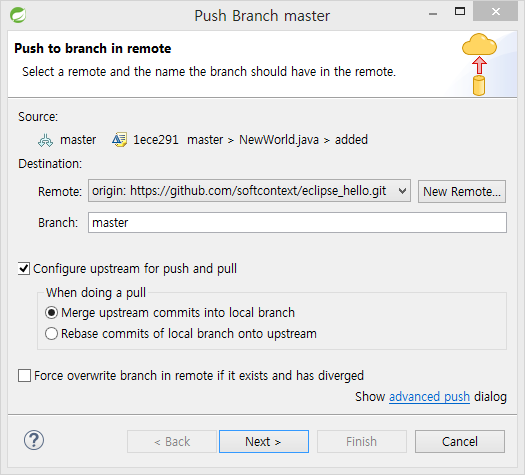
**- Push to Upstream**

모든 브랜치의 작업 내역을 푸시한다. 간편하지만 세분화 작업은 안 된다.

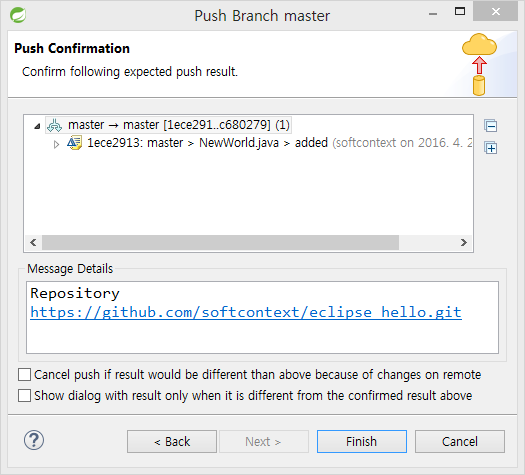
**- Push Branch 'master'**

작업 내역을 세분화해서 푸시한다.

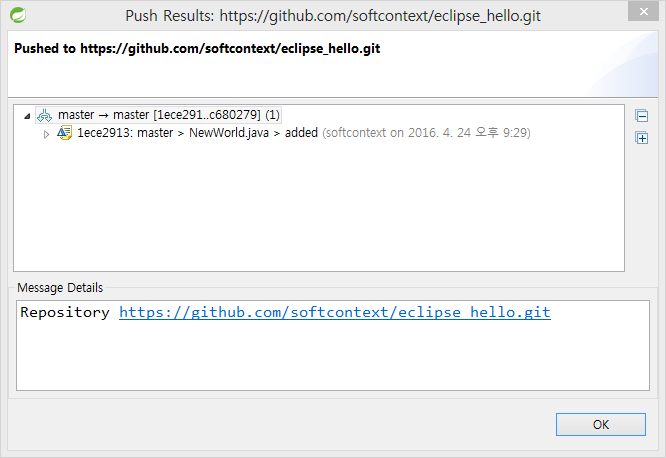
Team > Push Branch 'master' 클릭 > 항목을 확인하고 Next 버튼을 클릭한다.



정보를 확인하고 Finish 버튼을 클릭한다.



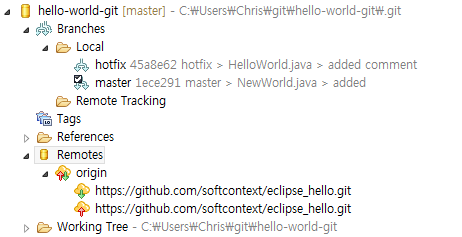
### 4. 푸시 결과 확인



깃허브에 접속해서 푸시가 잘 되었는지 확인한다.

# 원격 저장소와 로컬 저장소의 충돌 해결

깃 퍼스펙티브로 이동한다.



**hello-world-git > Branches > Local**

로컬 저장소의 브랜치가 표시된다.

**hello-world-git > Remotes > origin**

원격 저장소 연결여부가 표시된다.

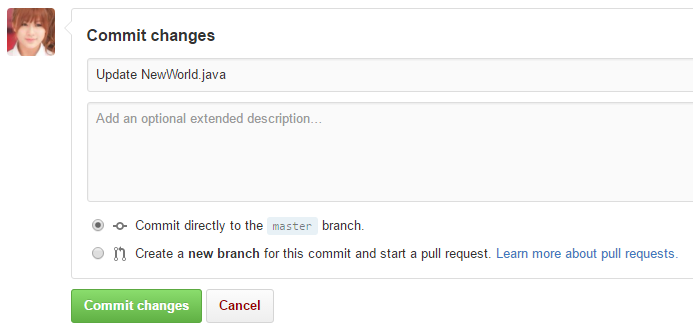
### 1. 원격 저장소에서 파일 수정

NewWorld.java

|  |
| --- |
| package com.example.test;  public class NewWorld {  public static void main(String[] args) {  **// origin > added comment**  System.out.println("new java file for test of push");  }  } |

### 2. 원격 저장소에서 커밋

커밋 메시지를 작성하고 Commit changes 버튼을 클릭한다.



### 3. 로컬 저장소에서 파일 수정

NewWorld.java

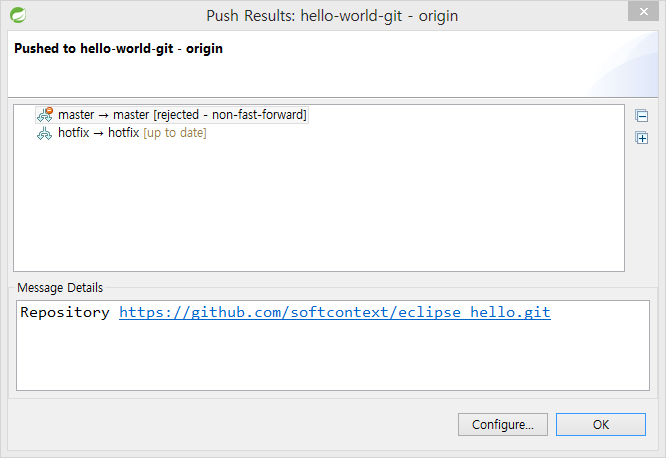
|  |
| --- |
| package com.example.test;  public class NewWorld {  public static void main(String[] args) {  **// local > added comment**  System.out.println("new java file for test of push");  }  } |

### 4. 로컬 저장소에서 커밋

Team > Commit 클릭 > 커밋 메시지를 작성하고 완료한다.

### 5. 로컬 저장소에 변경 내역을 원격 저장소에 푸시하기

Team > Push to Upstream 클릭



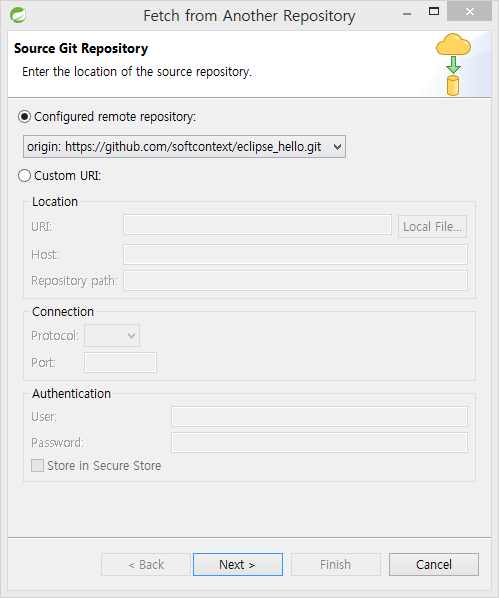
푸시가 실패한다. fast-forward가 아니기 때문에 거부되었다.

### 6. 원격 저장소에 내역을 로컬 저장소로 가져오기

깃 퍼스펙티브로 이동한다.

프로젝트를 선택 > 마우스 오른쪽 버튼 클릭 > Remote > Fetch 클릭

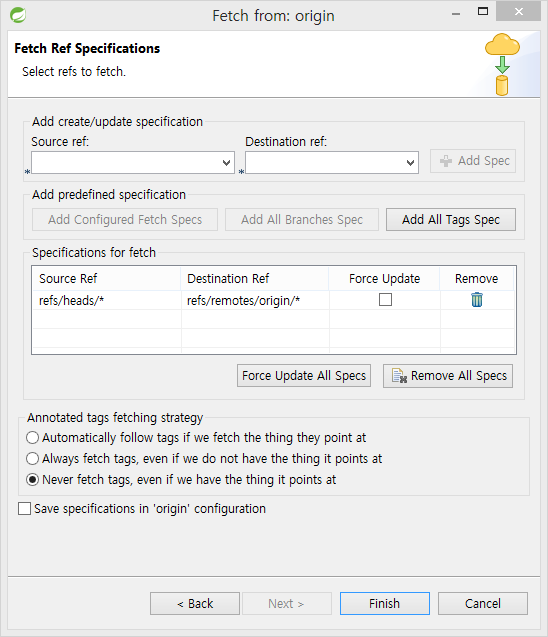
새로 뜨는 창에서 Next 버튼을 클릭한다.



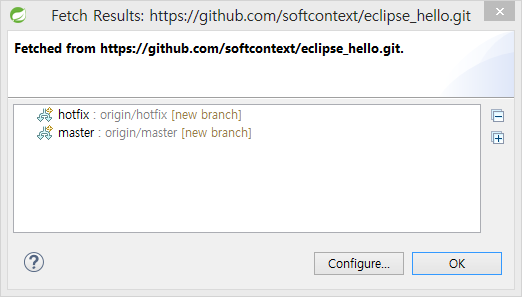
페치 대상을 선택하는 창이 뜨면 Add All Branches Spec 버튼을 클릭한다.

태그를 붙이지 않기 위해서 옵션을 Never fetch tags…로 변경한다.

Finish 버튼을 클릭한다.



페치작업 결과를 확인하고 OK 버튼을 클릭하여 창을 닫는다.

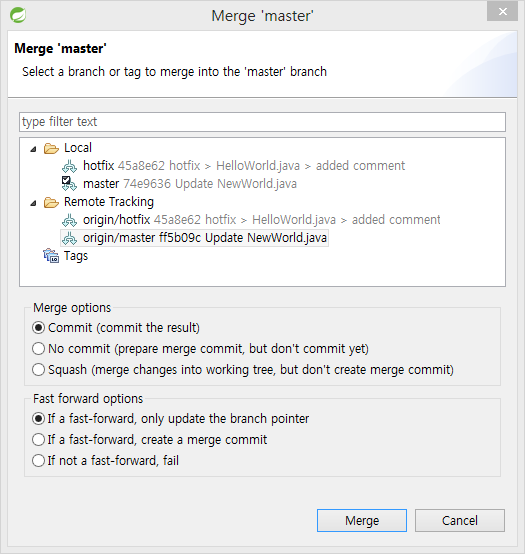


### 7. 원격 저장소 내역을 로컬 저장소에 병합하기

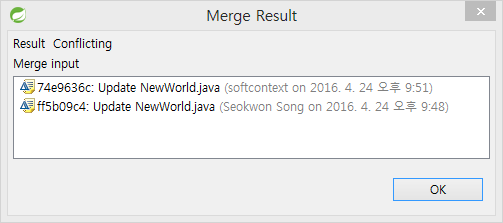
깃 퍼스펙티브로 이동한다.

프로젝트 선택 > 마우스 오른쪽 버튼 클릭 > Merge 클릭

Merge 'master' 창이 뜨면 origin/master를 선택하고 Merge 버튼을 클릭한다.



충돌 정보를 확인한다.



### 8. 충돌 해결

NewWorld.java

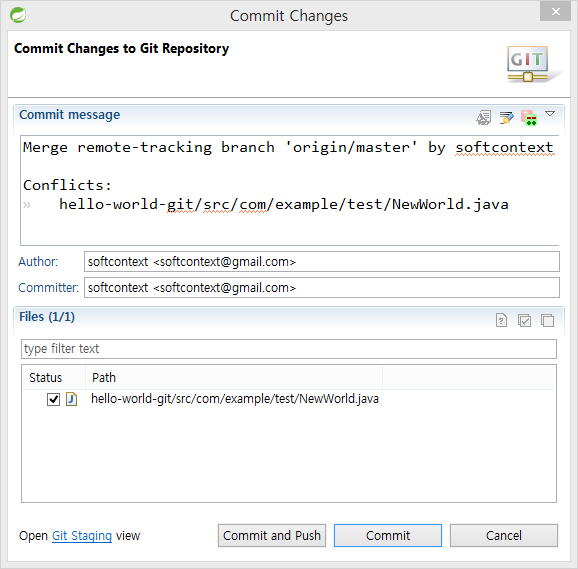
|  |
| --- |
| package com.example.test;  public class NewWorld {  public static void main(String[] args) {  **<<<<<<< HEAD**  **// local > added comment**  **=======**  **// origin > added comment**  **>>>>>>> refs/remotes/origin/master**  System.out.println("new java file for test of push");  }  } |

적절히 코드를 수정해 충돌을 해결한다.

패키지 익스플로러로 이동한다.

프로젝트를 선택 > Team > Add to Index 클릭하여 충돌이 해결되었음을 알린다.

Team > Commit 클릭하여 커밋을 수행한다.

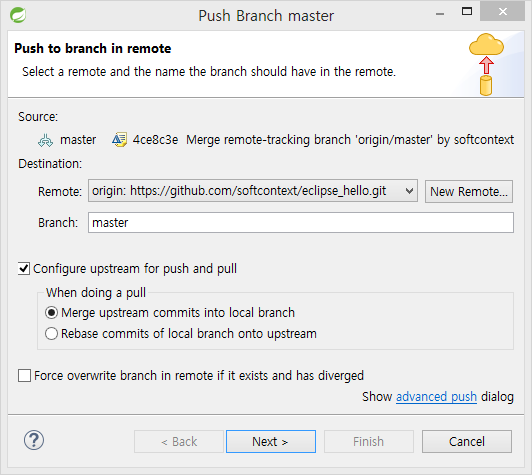


### 9. 로컬 저장소에 변경 내역을 원격 저장소에 푸시하기

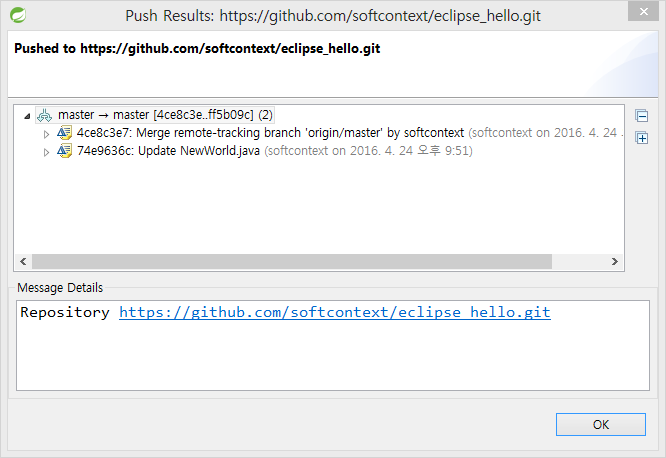
깃 퍼스펙티브로 이동한다.

프로젝트 선택 > 마우스 오른쪽 버튼 클릭 > Push Branch 'master' 클릭

새로 뜨는 창에서 설정 정보를 확인하고 Next 버튼을 클릭한다.



푸시 결과 확인



깃허브에 접속해서 푸시결과를 확인한다.

# 원격 저장소의 내역을 로컬 저장소로 Pull 하기

원격 저장소의 내역을 로컬 저장소로 가져오면서 자동으로 병합할 때 pull 명령을 사용한다.

### 1. 깃허브에서 파일 수정 및 커밋하기

|  |
| --- |
| package com.example.test;  public class NewWorld {  public static void main(String[] args) {  // local > added comment  // origin > added comment  System.out.println("new java file for test of push");  **System.out.println("origin > added println");**  }  } |

### 2. 원격 저장소의 내역을 로컬 저장소로 풀 하기

깃허브 퍼스펙티브로 이동한다.

프로젝트 선택 > 마우스 오른쪽 버튼 클릭 > Pull 클릭

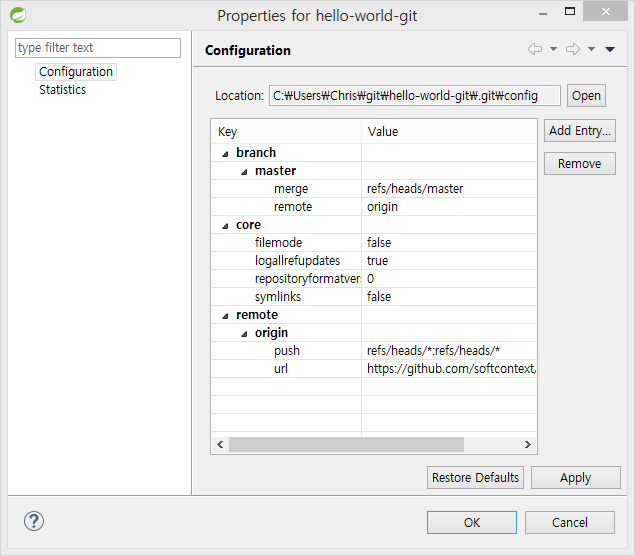
### Nothing to fetch 에러 해결 방법

이깃을 사용할 때 설정이 미흡하면 'Nothing to fetch' 에러가 발생할 수 있다.

깃허브 퍼스펙티브로 이동한다.

프로젝트 선택 > 마우스 오른쪽 버튼 클릭 > Properties 클릭

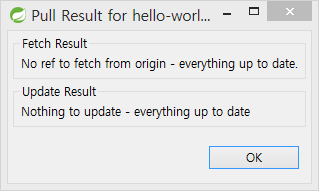
새로 뜨는 Configuration 창에서 Open 버튼을 클릭한다.



.git/config 파일이 열리면 다음 fetch 설정을 추가한다.

|  |
| --- |
| [core]  symlinks = false  repositoryformatversion = 0  filemode = false  logallrefupdates = true  [remote "origin"]  url = https://github.com/softcontext/eclipse\_hello.git  push = refs/heads/\*:refs/heads/\*  **fetch = +refs/heads/\*:refs/remotes/origin/\***  [branch "master"]  remote = origin  merge = refs/heads/master |

다시 pull 명령을 실행해 보자. 잘 처리된다.



# Git 활용

이전에 학습했던 내용을 복습하고 실전적인 깃 활용방법을 연습해 보자.

다음 내용을 보고 새로운 프로젝트를 만든 후 따라서 작업을 진행한다.

## 실습

|  |
| --- |
| D 드라이브 밑에 작업 폴더를 만든다.  ------------------------  D:\projects\git\flask  flask 폴더에서 Git Bash로 작업을 진행한다.  ------------------------  mkdir git\_lesson  cd git\_lesson  깃 저장소를 만든다.  ------------------------  ls -al  git init  ls -al  파이썬 파일을 작성한다.  ------------------------  vim hello.py  print("hello world")  cat hello.py  python hello.py  깃 저장소 동기화 상태를 확인한다.  ------------------------  git status  git add hello.py  git status  깃 저장소에 커밋한다.  ------------------------  git commit -m 'project created. hello.py created.'  git status  git log  새로운 브랜치를 만든다.  자동으로 master 브랜치에 내역이 새로운 브랜치에 그대로 복사된다.  ------------------------  git branch  git branch hotfix1  git branch  git checkout hotfix1  git branch  ls -al  hotfix1 브랜치에서 파일을 수정한다.  ------------------------  vim hello.py  print("added on hotfix1")  cat hello.py  python hello.py  커밋되지 않은 파일을 최종 커밋 시점으로 되돌린다.  수정 내역이 사라졌음을 확인한다.  ------------------------  git status  **git checkout HEAD -- hello.py**  cat hello.py  git status  다시 hotfix1 브랜치에서 파일을 수정한다.  ------------------------  vim hello.py  print("added on hotfix1")  cat hello.py  python hello.py  hotfix1 브랜치에서 커밋한다.  ------------------------  git status  git commit -am 'hello.py changed on hotfix1'  git status  git log  master 브랜치로 이동한다.  ------------------------  git branch  git checkout master  git branch  hotfix1 브랜치를 master 브랜치로 합병한다.  ------------------------  ls -al  cat hello.py  git merge hotfix1  cat hello.py  git status  master 브랜치에서 파일을 수정한다.  간단히 print 메소드 하나를 추가하자.  ------------------------  vim hello.py  print("added on master")  cat hello.py  git status  master 브랜치에서 커밋한다.  ------------------------  git log  git commit -am 'hello.py updated on master'  git status  git log  hotfix1 브랜치로 이동한다.  ------------------------  git branch  git checkout hotfix1  git branch  ls -al  cat hello.py  master 브랜치에 hello.py 파일과 hotfix1 브랜치에 hello.py 파일은 내역이 다른 상태다.  **master 브랜치의 내역을 그대로 hotfix1 브랜치로 가져오기 위해서 병합작업을 진행한다.**  합병 시 merge commit 기록을 남기기 위해서 --no-ff 옵션을 사용한다.  ------------------------  **git merge master --no-ff**  Merge branch 'master' into hotfix1 confirmed by Kris  git status  ls -al  cat hello.py  합병을 취소(reset)하고자 할 때 사용할 수 있는 가장 간단한 방법을 살펴보겠다.  --merge:  이 옵션은 합병 작업과 아무 관련이 없다.  --hard 와 같은 의미지만 커밋되지 않은 내역을 건드리지 않는 보다 안전한 방법이다.  ORIG\_HEAD:  합병 커밋이 아닌 이전에 수행된 오리지널 커밋을 가리킨다.  **reset --merge ORIG\_HEAD**:  reset 명령을 실행했을 때 지운 오리지널 커밋 내역을 보관하도록 조치한다.  결과적으로 로그 상에 3번 째, 4번 째 커밋이 지워진다.  4번 째 커밋은 내역이 없는 단지 합병 커밋으로 오리지널 커밋이 아니다.  3번 째 커밋은 오리지널 커밋으로 지워졌지만 커밋 시 적용되었던 변경 내역은 유지되고 있다.  ------------------------  **git reset --merge ORIG\_HEAD**  git log  git status  ls -al  cat hello.py  master 브랜치로 이동해서 지원진 3번 째 커밋의 변경 내역이 유지되고 있음을 확인한다.  ------------------------  git branch  git checkout master  git branch  la -al  cat hello.py  hotfix1 브랜치로 이동한다.  hotfix1 브랜치에서 파일을 수정한다. print 메소드 하나를 추가할 것이다.  ------------------------  git branch  git checkout hotfix1  ls -al  cat hello.py  vim hello.py  print("added new print method on hotfix1 without knowing change on master branch")  hotfix1 브랜치에서 커밋한다.  ------------------------  git status  git log  git commit -am 'hello.py changed on hotfix1 for conflict'  git status  git log  hotfix1 브랜치에서 파일 내역을 확인한다.  ------------------------  hello.py on hotfix1 branch  print("hello world")  print("added on hotfix1")  print("added new print method on hotfix1 without knowing change on master branch")  master 브랜치에서 파일 내역을 확인한다.  두 브랜치에서 파일 내역이 다르므로 합병을 시도할 때 충돌이 발생할 것이다.  ------------------------  hello.py on master branch  print("hello world")  print("added on hotfix1")  print("added on master")  hotfix1 브랜치를 master 브랜치로 합병한다.  충돌이 발생할 것이다.  ------------------------  git merge hotfix1  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  Automatic merge failed; fix conflicts and then commit the result.  충돌 지점을 확인한다.  ------------------------  cat hello.py  print("hello world")  print("added on hotfix1")  **<<<<<<< HEAD**  **print("added on master")**  **=======**  **print("added new print method on hotfix1 without knowing change on master branch")**  **>>>>>>> hotfix1**  충돌을 해결한다.  ------------------------  vim hello.py  print("hello world")  print("added on hotfix1")  print("resolved: added on master")  print("resolved: added new print method on hotfix1 without knowing change on master branch")  충돌을 해결한 후 커밋해서 merging 상태를 끝낸다.  ------------------------  git commit -am 'conflict resolved: hotfix1 into master'  git status  로그를 확인한다.  ------------------------  git log  #5 commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  #4 commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  #3 commit b9f55dc0cc84393c1fb846ec6d33958c389ca13d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:17:27 2016 +0900  hello.py updated on master    deleted #4 merge history master into hotfix1  deleted #3 commit hello.py updated on master    #2 commit 1289a990106188154ad311fbdcf75d7d0c8b2e1e  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 18:57:44 2016 +0900  hello.py changed on hotfix1  #1 commit 668697b1b35a878eaf5b23b6fabdfad20323b97c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 18:32:01 2016 +0900  project created. hello.py created.  로그를 그래프로 조회한다.  ------------------------  $ git log --graph  \* commit 6f036ad4f8166630465340ba304cb8be36cebffa  |\ Merge: b9f55dc d5dfb02  | | Author: softcontext <softcontext@gmail.com>  | | Date: Tue May 10 20:07:39 2016 +0900  | |  | | conflict resolved: hotfix1 into master  | |  | \* commit d5dfb0266616849f16b86398db7a3d392957ae53  | | Author: softcontext <softcontext@gmail.com>  | | Date: Tue May 10 19:56:16 2016 +0900  | |  | | hello.py changed on hotfix1 for conflict  | |  \* | commit b9f55dc0cc84393c1fb846ec6d33958c389ca13d  |/ Author: softcontext <softcontext@gmail.com>  | Date: Tue May 10 19:17:27 2016 +0900  |  | hello.py updated on master  |  \* commit 1289a990106188154ad311fbdcf75d7d0c8b2e1e  | Author: softcontext <softcontext@gmail.com>  | Date: Tue May 10 18:57:44 2016 +0900  |  | hello.py changed on hotfix1  |  \* commit 668697b1b35a878eaf5b23b6fabdfad20323b97c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 18:32:01 2016 +0900  project created. hello.py created.  master 브랜치에서 파일을 한번 더 수정한다.  ------------------------  git checkout master  vim hello.py  **print("-------------------")**  print("hello world")  print("added on hotfix1")  print("resolved: added on master")  print("resolved: added new print method on hotfix1 without knowing change on master branch")  master 브랜치에서 커밋한다.  ------------------------  git status  git commit -am 'hello.py refactored on master'  git status  git log --graph |

이전에 배운 내용을 복습 해 보았다. 추가로, 취소하는 명령도 살펴보았다.

이제 깃 활용 학습을 위한 설정을 진행해 보자.

1. 새로운 폴더를 '1'이라는 이름으로 만든다.

2. 위에서 만든 프로젝트 git\_lesson을 복사해서 1 폴더 밑에 복사해 넣는다.

git\_lesson 폴더는 백업용으로 남겨두고 앞으로는 1 폴더 밑에 git\_lesson 을 대상으로 연습할 것이다.

백업을 했기 때문에 언제든지 처음부터 다시 연습하고자 할 때 편리하다.

## git tag: 커밋을 참조하는 태그 붙이기

커밋을 참조하기 위해서 태그를 붙일 수 있다.

연습을 하던 중 무언가 엉킨다면 백업 폴더를 다시 복사해서 작업하도록 한다.

### 1. light weight tag

커밋에 간단한 태그를 붙인다. 체크섬 옵션정보를 생략하면 최근 커밋을 대상으로 태그를 붙인다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git tag 1.0  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log --decorate -2  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d (HEAD -> master, tag: 1.0)  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git show-ref --tags  203980f1c2b6b35193ef752f4c6e09a928bb708d refs/tags/1.0 |

**git tag 1.0**

가장 최근 커밋에 light weight 태그를 붙인다.

**git log --decorate -2**

최근 2개의 커밋을 예쁘게 조회한다.

(HEAD -> master, tag: 1.0)

HEAD 포인터는 master의 커밋을 참조하고 있고 커밋에 '1.0'이라는 태그가 붙어 있음을 알 수 있다.

**git show-ref --tags**

태그와 체크섬 값을 같이 조회한다.

### 2. 특정 커밋에 태그 붙이기

체크섬 값의 앞 네자리를 사용하여 특정 커밋에 태그를 붙일 수 있다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log -2  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git tag 0.9 6f03  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git show-ref --tags  6f036ad4f8166630465340ba304cb8be36cebffa refs/tags/0.9  203980f1c2b6b35193ef752f4c6e09a928bb708d refs/tags/1.0 |

### 3. annotated tag

-a 옵션을 사용하여 태그를 추가하면 태그 정보를 편집할 수 있는 vim 에디터가 기동한다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ clear  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log -3  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git tag -a 0.8 d5df |

vim 편집기에서 다음 정보를 입력한다.

|  |
| --- |
| ver 0.8 by Kris |

0.8 태그의 정보를 확인해 보자.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git show-ref --tags  65ea26bd6e43eacfe604a2a353f71c407fe06b5c refs/tags/0.8  6f036ad4f8166630465340ba304cb8be36cebffa refs/tags/0.9  203980f1c2b6b35193ef752f4c6e09a928bb708d refs/tags/1.0  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git show 0.8  tag 0.8  Tagger: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:53:45 2016 +0900  ver 0.8 by Kris  commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  **diff --git a/hello.py b/hello.py**  **index 699c506..9b11044 100644**  **--- a/hello.py**  **+++ b/hello.py**  @@ -1,2 +1,3 @@  print("hello world")  print("added on hotfix1")  +print("added new print method on hotfix1 without knowing change on master branch") |

### 4. 태그 삭제

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log --decorate -3  commit 18749abb443d0eee024cf3d4050b3009408de091 (HEAD -> master, tag: 1.1)  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 22:16:24 2016 +0900  README.md added on master  commit 2930c33a43a399a8ad01008aac7a43474d13af86  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 21:52:39 2016 +0900  hello.py changed for test and comment added  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d (tag: 1.0)  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git tag -d 1.1  Deleted tag '1.1' (was 18749ab)  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log --decorate -3  commit 18749abb443d0eee024cf3d4050b3009408de091 (HEAD -> master)  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 22:16:24 2016 +0900  README.md added on master  commit 2930c33a43a399a8ad01008aac7a43474d13af86  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 21:52:39 2016 +0900  hello.py changed for test and comment added  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d (tag: 1.0)  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master |

## git commit --amend: 마지막 커밋 수정하기

--amend 옵션을 주면 최종 커밋을 대체하는 새로운 커밋을 만들어 덮어쓴다.

명령을 실행하면 마지막 커밋과 커밋하지 않은 상태에 있는 변경 내역이 서로 합쳐진 새 커밋을 만든다.

만약 아무런 변경 내역이 없다면 커밋 메시지만 변경하는 효과가 적용된다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 10 20:19 ./  drwxr-xr-x 1 Chris 197121 0 5월 10 20:19 ../  drwxr-xr-x 1 Chris 197121 0 5월 10 21:45 .git/  -rw-r--r-- 1 Chris 197121 209 5월 10 20:14 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ cat hello.py  print("new hello world")  print("-------------------")  print("hello world")  print("added on hotfix1")  print("resolved: added on master")  print("resolved: added new print method on hotfix1 without knowing change on master branch")  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch 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: hello.py  no changes added to commit (use "git add" and/or "git commit -a")  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git commit -am 'hello.py changed for test'  [master 98e6f41] hello.py changed for test  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch master  nothing to commit, working directory clean  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ cat hello.py  // comment added  print("new hello world")  print("-------------------")  print("hello world")  print("added on hotfix1")  print("resolved: added on master")  print("resolved: added new print method on hotfix1 without knowing change on master branch")  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch 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: hello.py  no changes added to commit (use "git add" and/or "git commit -a")  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git commit -a --amend  [master 2930c33] hello.py changed for test and comment added  Date: Tue May 10 21:52:39 2016 +0900  1 file changed, 2 insertions(+)  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch master  nothing to commit, working directory clean  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git log -2  commit 2930c33a43a399a8ad01008aac7a43474d13af86  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 21:52:39 2016 +0900  hello.py changed for test and comment added  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master |

## git checkout HEAD -- filename

커밋하지 않은 변경 내역을 취소한다.

filename 대상 파일의 내용이 최종 커밋 시점으로 되돌아간다.

-- 옵션은 checkout 명령에 뒤따라 오는 것이 파일이라는 것을 통보한다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ cat README.md  metadata: first information  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch master  Untracked files:  (use "git add <file>..." to include in what will be committed)  README.md  nothing added to commit but untracked files present (use "git add" to track)  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git add README.md  warning: LF will be replaced by CRLF in README.md.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch master  Changes to be committed:  (use "git reset HEAD <file>..." to unstage)  new file: README.md  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git commit -am 'README.md added on master'  [master 18749ab] README.md added on master  warning: LF will be replaced by CRLF in README.md.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 README.md  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ cat README.md  metadata: first information  metadata: second information  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git status  On branch 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")  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git add README.md  warning: LF will be replaced by CRLF in README.md.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ git checkout HEAD -- README.md  Chris@starrose MINGW64 /d/projects/git/flask/1/git\_lesson (master)  $ cat README.md  metadata: first information |

## git revert: 공개된 커밋의 변경 내역을 되돌리기

커밋을 되돌린다. 이 때, 지운 커밋내역을 남긴다.

이미 다른 개발자들과 공유된 커밋 내역을 수정하는 것은 위험하다.

대신, 커밋으로 발생한 변경 내역의 반대 커밋을 수행하여 변경 내역을 되돌리는 것이 안전하다.

즉, 추가한 코드를 빼거나 지운 코드를 다시 추가하는 새로운 커밋을 수행하는 것이다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ vim say.py  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ cat say.py  print("what's up")  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git add --all  warning: LF will be replaced by CRLF in say.py.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git commit -m 'say.py added on master'  [master 8024c2e] say.py added on master  warning: LF will be replaced by CRLF in say.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 say.py  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log --decorate -2  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23 (HEAD -> master)  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ vim say.py  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ cat say.py  print("what's up")  print("bye bye")  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git commit -am 'say.py changed but it is useless'  warning: LF will be replaced by CRLF in say.py.  The file will have its original line endings in your working directory.  [master warning: LF will be replaced by CRLF in say.py.  The file will have its original line endings in your working directory.  e5fbcd6] say.py changed but it is useless  warning: LF will be replaced by CRLF in say.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git status  On branch master  nothing to commit, working directory clean  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log --decorte -3  fatal: unrecognized argument: --decorte  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log --decorate -3  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187 (HEAD -> master)  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git revert e5fb  [master 95f3ad0] Revert "say.py changed but it is useless"  1 file changed, 1 deletion(-)  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log --decorate -3  commit 95f3ad09909b332a81a1714280a1f046720db4f3 (HEAD -> master)  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:30:06 2016 +0900  Revert "say.py changed but it is useless"  This reverts commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187.  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ cat say.py  print("what's up") |

작업대상 say.py 파일을 확인해보면 추가한 코드가 사라졌음을 알 수 있다.

## git reset: 변경 내역을 취소하기

커밋을 리셋한다. revert 명령과 비슷해 보인다. 차이점은 지운 커밋내역을 남기지 않는다는 것이다.

### git reset 명령모드

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 명령모드 | 설명 | HEAD 위치 | 인덱스(스테이지) | 저장소 |
| hard | 전부 되돌림 | 되돌림 | 되돌림 | 되돌림 |
| mixed | **기본값**  인덱스, 커밋을 되돌림 | 되돌림 | 되돌림 | 유지 |
| soft | 커밋만 되돌림 | 되돌림 | 유지 | 유지 |

### git reset 명령옵션

|  |  |
| --- | --- |
| 명령옵션 | 설명 |
| ^ or ~ | ~은 하나의 커밋을 의미한다. 표시한 수 만큼 커밋을 되돌린다.  ~~~ 표시와 ~3 은 같은 의미다. |
| ORIG\_HEAD | 지운 커밋 내역을 보관한다. 지운 커밋을 되돌릴 때 사용한다. |

### git reset --soft HEAD~~~

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -5  commit 95f3ad09909b332a81a1714280a1f046720db4f3  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:30:06 2016 +0900  Revert "say.py changed but it is useless"  This reverts commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187.  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git reset --soft HEAD~~~  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -3  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ ls -al  total 6  drwxr-xr-x 1 Chris 197121 0 5월 11 19:30 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 19:21 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 19:41 .git/  -rw-r--r-- 1 Chris 197121 209 5월 10 20:14 hello.py  -rw-r--r-- 1 Chris 197121 20 5월 11 19:30 say.py  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ cat say.py  print("what's up") |

최근 커밋이 바뀐 것을 확인할 수 있다.

파일은 그대로 존재하며 파일 내용이 바뀐 것이 아님을 알 수 있다.

### git reset --soft ORIG\_HEAD

이번에는 원래 상태대로 돌려놓는 작업을 해 보자.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -4  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  commit b9f55dc0cc84393c1fb846ec6d33958c389ca13d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:17:27 2016 +0900  hello.py updated on master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git reset --soft ORIG\_HEAD  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -4  commit 95f3ad09909b332a81a1714280a1f046720db4f3  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:30:06 2016 +0900  Revert "say.py changed but it is useless"  This reverts commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187.  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master |

### git reset --hard HEAD~~~

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -5  commit 95f3ad09909b332a81a1714280a1f046720db4f3  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:30:06 2016 +0900  Revert "say.py changed but it is useless"  This reverts commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187.  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git reset --hard HEAD~3  HEAD is now at 203980f hello.py refactored on master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -5  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  commit d5dfb0266616849f16b86398db7a3d392957ae53  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:56:16 2016 +0900  hello.py changed on hotfix1 for conflict  commit b9f55dc0cc84393c1fb846ec6d33958c389ca13d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 19:17:27 2016 +0900  hello.py updated on master  commit 1289a990106188154ad311fbdcf75d7d0c8b2e1e  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 18:57:44 2016 +0900  hello.py changed on hotfix1  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 11 19:57 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 19:21 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 19:57 .git/  -rw-r--r-- 1 Chris 197121 209 5월 10 20:14 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git reset --hard ORIG\_HEAD  HEAD is now at 95f3ad0 Revert "say.py changed but it is useless"  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ git log -5  commit 95f3ad09909b332a81a1714280a1f046720db4f3  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:30:06 2016 +0900  Revert "say.py changed but it is useless"  This reverts commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187.  commit e5fbcd6a4f1966bd70a58bf13b2a85fd55a24187  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:28:47 2016 +0900  say.py changed but it is useless  commit 8024c2e6197fb8bfe0eac61b10c91c06868edd23  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 19:26:35 2016 +0900  say.py added on master  commit 203980f1c2b6b35193ef752f4c6e09a928bb708d  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:17:36 2016 +0900  hello.py refactored on master  commit 6f036ad4f8166630465340ba304cb8be36cebffa  Merge: b9f55dc d5dfb02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 10 20:07:39 2016 +0900  conflict resolved: hotfix1 into master  Chris@starrose MINGW64 /d/projects/git/flask/2/git\_lesson (master)  $ ls -al  total 6  drwxr-xr-x 1 Chris 197121 0 5월 11 19:59 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 19:21 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 19:59 .git/  -rw-r--r-- 1 Chris 197121 209 5월 10 20:14 hello.py  -rw-r--r-- 1 Chris 197121 20 5월 11 19:59 say.py |

checkout과 reset명령은 되돌린다는 점에서 비슷해 보이지만 결정적인 차이점이 있다.

**git checkout 체크섬값**

해당 브랜치 이후에 변경 내역이 사라진다.

**git reset --soft 체크섬값 / git reset --mixed 체크섬값**

해당 브랜치 이후에 변경 내역이 유지된다.

## git rebase

깃은 커밋 내역을 수정할 수 있다. 하지만 원격 저장소에 푸시가 끝난 커밋을 수정하는 것은

권장하지 않는다. 로컬 저장소에서 커밋을 깔끔하게 정리한 다음 원격 저장소에 푸시하는 것이 좋다.

병합을 하기전에 rebase를 진행하면 브랜치를 하나의 줄기로 관리할 수 있다.

브랜치들의 이력을 확인하면서 커밋들을 정리해 보자.

git\_rebase라는 새로운 프로젝트 폴더를 만든 후 작업을 따라한다.

### 1. master commit

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase  $ git init  Initialized empty Git repository in D:/projects/git/flask/git\_rebase/.git/  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ cat hello.py  print("1: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git add --all  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git commit -am '1: hello.py created on master'  [master (root-commit) 90ce34f] 1: hello.py created on master  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 2. hotfix1 commit

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git branch hotfix1  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git checkout hotfix1  Switched to branch 'hotfix1'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ git branch  \* hotfix1  master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 11 21:56 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 21:41 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 21:59 .git/  -rw-r--r-- 1 Chris 197121 28 5월 11 21:56 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ git log  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ cat hello.py  print("1: added on master")  print("2: added on hotfix1")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ git commit -am '2: hello.py updated on hotfix1'  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  [hotfix1 warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  82f22b5] 2: hello.py updated on hotfix1  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ git log  commit 82f22b5bb398c5e40743156ca4b17ed400218015  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 3. master commit

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix1)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ cat hello.py  print("1: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ cat hello.py  print("1: added on master")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git commit -am '3: hello.py updated on master'  [master 72fa7ad] 3: hello.py updated on master  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 4. hotfix2 commit

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git branch  hotfix1  \* master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git branch hotfix2  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git checkout hotfix2  Switched to branch 'hotfix2'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 11 22:04 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 21:41 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 22:08 .git/  -rw-r--r-- 1 Chris 197121 58 5월 11 22:04 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ cat hello.py  print("1: added on master")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ cat hello.py  print("1: added on master")  print("3: added on master")  print("4: added on hotfix2")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ git commit -am '4: hello.py updated on hotfix2'  [hotfix2 cee97ce] 4: hello.py updated on hotfix2  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ git log  commit cee97ce0a78ccce6946f9e7ecabbbe834dafea32  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 5. hotfix3 commit

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix2)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git branch  hotfix1  hotfix2  \* master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git checkout -b hotfix3  Switched to a new branch 'hotfix3'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ git branch  hotfix1  hotfix2  \* hotfix3  master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 11 22:12 ./  drwxr-xr-x 1 Chris 197121 0 5월 11 21:41 ../  drwxr-xr-x 1 Chris 197121 0 5월 11 22:12 .git/  -rw-r--r-- 1 Chris 197121 58 5월 11 22:12 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ cat hello.py  print("1: added on master")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ cat hello.py  print("1: added on master")  print("3: added on master")  print("5: added on hotfix3")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ git commit -am '5: hello.py updated on hotfix3'  [hotfix3 dea6eb8] 5: hello.py updated on hotfix3  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ git log  commit dea6eb8899665715e0c5829b986ba57e4d7f53e9  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 6. hotfix1 🡪 master

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (hotfix3)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git branch  hotfix1  hotfix2  hotfix3  \* master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ cat hello.py  print("1: added on master")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git merge hotfix1  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  Automatic merge failed; fix conflicts and then commit the result.  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git diff  **diff --cc hello.py**  **index f8686c2,aa8d893..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,2 -1,2 +1,6 @@@  print("1: added on master")  ++<<<<<<< HEAD  +print("3: added on master")  ++=======  + print("2: added on hotfix1")  ++>>>>>>> hotfix1  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ cat hello.py  print("1: added on master")  print("2: added on hotfix1")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git commit -am 'merge conflict resolved. hotfix1 > master'  [master 4d95579] merge conflict resolved. hotfix1 > master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log  commit 4d955799399e6699c1a8f5fd05bd4f72f3f3ecbc  Merge: 72fa7ad 82f22b5  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 16:57:19 2016 +0900  merge conflict resolved. hotfix1 > master  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 82f22b5bb398c5e40743156ca4b17ed400218015  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 7. hotfix2 🡪 master

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git merge hotfix2  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  Automatic merge failed; fix conflicts and then commit the result.  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git diff  **diff --cc hello.py**  **index 78f4e36,b5e97aa..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,4 -1,3 +1,8 @@@  print("1: added on master")  +print("2: added on hotfix1")  print("3: added on master")  ++<<<<<<< HEAD  +  ++=======  + print("4: added on hotfix2")  ++>>>>>>> hotfix2  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ cat hello.py  print("1: added on master")  print("2: added on hotfix1")  print("3: added on master")  print("4: added on hotfix2")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git commit -am 'merge conflict resolved. hotfix2 > master'  [master d5a7f5c] merge conflict resolved. hotfix2 > master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log  commit d5a7f5c203d104f06757ca8a46c647eb08f4a5b6  Merge: 4d95579 cee97ce  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 17:04:20 2016 +0900  merge conflict resolved. hotfix2 > master  commit 4d955799399e6699c1a8f5fd05bd4f72f3f3ecbc  Merge: 72fa7ad 82f22b5  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 16:57:19 2016 +0900  merge conflict resolved. hotfix1 > master  commit cee97ce0a78ccce6946f9e7ecabbbe834dafea32  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 82f22b5bb398c5e40743156ca4b17ed400218015  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 8. hotfix3 🡪 master

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git merge hotfix3  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  Automatic merge failed; fix conflicts and then commit the result.  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git diff  **diff --cc hello.py**  **index 73c60b0,a39c91d..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,4 -1,3 +1,8 @@@  print("1: added on master")  +print("2: added on hotfix1")  print("3: added on master")  ++<<<<<<< HEAD  +print("4: added on hotfix2")  ++=======  + print("5: added on hotfix3")  ++>>>>>>> hotfix3  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ cat hello.py  print("1: added on master")  print("2: added on hotfix1")  print("3: added on master")  print("4: added on hotfix2")  print("5: added on hotfix3")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master|MERGING)  $ git commit -am 'merge conflict resolved. hotfix3 > master'  [master 0f79fb6] merge conflict resolved. hotfix3 > master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase (master)  $ git log --graph  \* commit 0f79fb693023b2f0f37c67e0fbb428dfac05067c  |\ Merge: d5a7f5c dea6eb8  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 17:07:17 2016 +0900  | |  | | merge conflict resolved. hotfix3 > master  | |  | \* commit dea6eb8899665715e0c5829b986ba57e4d7f53e9  | | Author: softcontext <softcontext@gmail.com>  | | Date: Wed May 11 22:14:26 2016 +0900  | |  | | 5: hello.py updated on hotfix3  | |  \* | commit d5a7f5c203d104f06757ca8a46c647eb08f4a5b6  |\ \ Merge: 4d95579 cee97ce  | | | Author: softcontext <softcontext@gmail.com>  | | | Date: Thu May 12 17:04:20 2016 +0900  | | |  | | | merge conflict resolved. hotfix2 > master  | | |  | \* | commit cee97ce0a78ccce6946f9e7ecabbbe834dafea32  | |/ Author: softcontext <softcontext@gmail.com>  | | Date: Wed May 11 22:09:33 2016 +0900  | |  | | 4: hello.py updated on hotfix2  | |  \* | commit 4d955799399e6699c1a8f5fd05bd4f72f3f3ecbc  |\ \ Merge: 72fa7ad 82f22b5  | |/ Author: softcontext <softcontext@gmail.com>  |/| Date: Thu May 12 16:57:19 2016 +0900  | |  | | merge conflict resolved. hotfix1 > master  | |  | \* commit 82f22b5bb398c5e40743156ca4b17ed400218015  | | Author: softcontext <softcontext@gmail.com>  | | Date: Wed May 11 22:00:49 2016 +0900  | |  | | 2: hello.py updated on hotfix1  | |  \* | commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 9. rebase hotfix1

연습을 하다가 무언가 꼬이면 다시할 수 있도록 앞서 작업한 내용은 백업으로 남겨둔다.

새로운 폴더 git\_rebase\_quiz 폴더를 만든 후 git\_rebase 폴더를 복사해 붙여 넣는다.

merge 전 상태로 되돌기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log  commit 0f79fb693023b2f0f37c67e0fbb428dfac05067c  Merge: d5a7f5c dea6eb8  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 17:07:17 2016 +0900  merge conflict resolved. hotfix3 > master  commit d5a7f5c203d104f06757ca8a46c647eb08f4a5b6  Merge: 4d95579 cee97ce  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 17:04:20 2016 +0900  merge conflict resolved. hotfix2 > master  commit 4d955799399e6699c1a8f5fd05bd4f72f3f3ecbc  Merge: 72fa7ad 82f22b5  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 16:57:19 2016 +0900  merge conflict resolved. hotfix1 > master  commit dea6eb8899665715e0c5829b986ba57e4d7f53e9  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit cee97ce0a78ccce6946f9e7ecabbbe834dafea32  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 82f22b5bb398c5e40743156ca4b17ed400218015  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  merge conflict resolved. hotfix3 > master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git reset --hard HEAD~3  HEAD is now at 72fa7ad 3: hello.py updated on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git branch  hotfix1  hotfix2  hotfix3  \* master |

merge 대신 rebase 사용하기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git checkout hotfix1  Switched to branch 'hotfix1'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1)  $ git rebase master  First, rewinding head to replay your work on top of it...  Applying: 2: hello.py updated on hotfix1  Using index info to reconstruct a base tree...  M hello.py  Falling back to patching base and 3-way merge...  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  error: Failed to merge in the changes.  Patch failed at 0001 2: hello.py updated on hotfix1  The copy of the patch that failed is found in: .git/rebase-apply/patch  When you have resolved this problem, run "git rebase --continue".  If you prefer to skip this patch, run "git rebase --skip" instead.  To check out the original branch and stop rebasing, run "git rebase --abort". |

### rebase 시 발생하는 충돌 해결방법

**git rebase --continue**

충돌을 해결한 후 rebase 작업을 계속 진행하여 완료한다.

**git rebase --skip**

병합 대상(master) 브랜치의 내용이 그대로 복사된다. 다시 git rebase 명령을 실행할 수 없다.

권장하지 않는다.

**git rebase --abort**

명령을 취소한다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ git diff  **diff --cc hello.py**  **index f8686c2,aa8d893..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,2 -1,2 +1,6 @@@  print("1: added on master")  ++<<<<<<< 72fa7ada0017abc4f27d7f351ba264b01c913fe5  +print("3: added on master")  ++=======  + print("2: added on hotfix1")  ++>>>>>>> 2: hello.py updated on hotfix1  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ cat hello.py  print("1: added on master")  print("2: added on hotfix1")  print("3: added on master")  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ git rebase --continue  hello.py: needs merge  You must edit all merge conflicts and then  mark them as resolved using git add  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ git add hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1|REBASE 1/1)  $ git rebase --continue  Applying: 2: hello.py updated on hotfix1 |

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1)  $ git log  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix1)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log --graph  \* commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  | Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

여기까지만이라면 단순하게 hotfix1과 master 브랜치가 따로따로 있는 것에 불과하다.

hotfix1의 base를 다시(re) 설정한 것과 같은 효과다.

추가로 병합작업을 해야만 비로소 master 브랜치에 hotfix1 브랜치가 반영된다.

병합한 흔적을 남기고자 할 때 --no-ff 라는 옵션을 추가하자.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git merge hotfix1 --no-ff  Merge made by the 'recursive' strategy.  hello.py | 1 +  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log  commit eeed82290490a011b339e00b83aa671c566de69f  Merge: 72fa7ad 1394424  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:00:35 2016 +0900  Merge branch 'hotfix1'  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log --graph  \* commit eeed82290490a011b339e00b83aa671c566de69f  |\ Merge: 72fa7ad 1394424  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:00:35 2016 +0900  | |  | | Merge branch 'hotfix1'  | |  | \* commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:00:49 2016 +0900  |  | 2: hello.py updated on hotfix1  |  \* commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  | Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 10. rebase hotfix2

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git checkout hotfix2  Switched to branch 'hotfix2'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix2)  $ git log  commit cee97ce0a78ccce6946f9e7ecabbbe834dafea32  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix2)  $ git rebase master  First, rewinding head to replay your work on top of it...  Applying: 4: hello.py updated on hotfix2  Using index info to reconstruct a base tree...  M hello.py  Falling back to patching base and 3-way merge...  Auto-merging hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix2)  $ git log  commit edd229b510ac44a912ffa9c19f59806d628eff46  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit eeed82290490a011b339e00b83aa671c566de69f  Merge: 72fa7ad 1394424  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:00:35 2016 +0900  Merge branch 'hotfix1'  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix2)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git merge hotfix2 --no-ff  Merge made by the 'recursive' strategy.  hello.py | 1 +  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log --graph  \* commit 230e44ad9c119826638674e2d85db9a2b9c78c99  |\ Merge: eeed822 edd229b  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:08:19 2016 +0900  | |  | | Merge branch 'hotfix2'  | |  | \* commit edd229b510ac44a912ffa9c19f59806d628eff46  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:09:33 2016 +0900  |  | 4: hello.py updated on hotfix2  |  \* commit eeed82290490a011b339e00b83aa671c566de69f  |\ Merge: 72fa7ad 1394424  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:00:35 2016 +0900  | |  | | Merge branch 'hotfix1'  | |  | \* commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:00:49 2016 +0900  |  | 2: hello.py updated on hotfix1  |  \* commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  | Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 11. rebase hotfix3

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git checkout hotfix3  Switched to branch 'hotfix3'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3)  $ git log  commit dea6eb8899665715e0c5829b986ba57e4d7f53e9  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3)  $ git rebase master  First, rewinding head to replay your work on top of it...  Applying: 5: hello.py updated on hotfix3  Using index info to reconstruct a base tree...  M hello.py  Falling back to patching base and 3-way merge...  Auto-merging hello.py  CONFLICT (content): Merge conflict in hello.py  error: Failed to merge in the changes.  Patch failed at 0001 5: hello.py updated on hotfix3  The copy of the patch that failed is found in: .git/rebase-apply/patch  When you have resolved this problem, run "git rebase --continue".  If you prefer to skip this patch, run "git rebase --skip" instead.  To check out the original branch and stop rebasing, run "git rebase --abort".  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3|REBASE 1/1)  $ git diff  **diff --cc hello.py**  **index 73c60b0,a39c91d..0000000**  **--- a/hello.py**  **+++ b/hello.py**  @@@ -1,4 -1,3 +1,8 @@@  print("1: added on master")  +print("2: added on hotfix1")  print("3: added on master")  ++<<<<<<< 230e44ad9c119826638674e2d85db9a2b9c78c99  +print("4: added on hotfix2")  ++=======  + print("5: added on hotfix3")  ++>>>>>>> 5: hello.py updated on hotfix3  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3|REBASE 1/1)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3|REBASE 1/1)  $ git add hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3|REBASE 1/1)  $ git rebase --continue  Applying: 5: hello.py updated on hotfix3  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3)  $ git log  commit 429eb33242b3cd56a67ef22f4b7b3c23943c0603  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit 230e44ad9c119826638674e2d85db9a2b9c78c99  Merge: eeed822 edd229b  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:08:19 2016 +0900  Merge branch 'hotfix2'  commit edd229b510ac44a912ffa9c19f59806d628eff46  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit eeed82290490a011b339e00b83aa671c566de69f  Merge: 72fa7ad 1394424  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:00:35 2016 +0900  Merge branch 'hotfix1'  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (hotfix3)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log --graph  \* commit 230e44ad9c119826638674e2d85db9a2b9c78c99  |\ Merge: eeed822 edd229b  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:08:19 2016 +0900  | |  | | Merge branch 'hotfix2'  | |  | \* commit edd229b510ac44a912ffa9c19f59806d628eff46  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:09:33 2016 +0900  |  | 4: hello.py updated on hotfix2  |  \* commit eeed82290490a011b339e00b83aa671c566de69f  |\ Merge: 72fa7ad 1394424  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:00:35 2016 +0900  | |  | | Merge branch 'hotfix1'  | |  | \* commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:00:49 2016 +0900  |  | 2: hello.py updated on hotfix1  |  \* commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  | Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git merge hotfix3 --no-ff  Merge made by the 'recursive' strategy.  hello.py | 1 +  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log --graph  \* commit a54701a1251e590daf37a02c0fe1adbc63e89b2f  |\ Merge: 230e44a 429eb33  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:13:52 2016 +0900  | |  | | Merge branch 'hotfix3'  | |  | \* commit 429eb33242b3cd56a67ef22f4b7b3c23943c0603  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:14:26 2016 +0900  |  | 5: hello.py updated on hotfix3  |  \* commit 230e44ad9c119826638674e2d85db9a2b9c78c99  |\ Merge: eeed822 edd229b  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:08:19 2016 +0900  | |  | | Merge branch 'hotfix2'  | |  | \* commit edd229b510ac44a912ffa9c19f59806d628eff46  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:09:33 2016 +0900  |  | 4: hello.py updated on hotfix2  |  \* commit eeed82290490a011b339e00b83aa671c566de69f  |\ Merge: 72fa7ad 1394424  | | Author: softcontext <softcontext@gmail.com>  | | Date: Thu May 12 18:00:35 2016 +0900  | |  | | Merge branch 'hotfix1'  | |  | \* commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  |/ Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:00:49 2016 +0900  |  | 2: hello.py updated on hotfix1  |  \* commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  | Author: softcontext <softcontext@gmail.com>  | Date: Wed May 11 22:04:53 2016 +0900  |  | 3: hello.py updated on master  |  \* commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

## git rebase -i

두 개의 커밋 내역을 병합한다. i는 interactive의 약자다.

- 남기는 커밋 메시지 앞에는 접두어 pick을 붙인다.

- 없애는 커밋 메시지 앞에는 접두어 fixup을 붙인다.

체크섬 값은 필수값이고 기존 커밋 메시지를 변경할 수 없다.

앞서서 작업한 git\_rebase 프로젝트 폴더의 최근 커밋 내역(merge 커밋 제외) 두 개를 합쳐보자.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log -6  commit a54701a1251e590daf37a02c0fe1adbc63e89b2f  Merge: 230e44a 429eb33  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:13:52 2016 +0900  Merge branch 'hotfix3'  commit 429eb33242b3cd56a67ef22f4b7b3c23943c0603  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit 230e44ad9c119826638674e2d85db9a2b9c78c99  Merge: eeed822 edd229b  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:08:19 2016 +0900  Merge branch 'hotfix2'  commit edd229b510ac44a912ffa9c19f59806d628eff46  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit eeed82290490a011b339e00b83aa671c566de69f  Merge: 72fa7ad 1394424  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:00:35 2016 +0900  Merge branch 'hotfix1'  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git rebase -i HEAD~~  [detached HEAD 6ff8cb8] 4: hello.py updated on hotfix2  Date: Wed May 11 22:09:33 2016 +0900  1 file changed, 2 insertions(+)  Successfully rebased and updated refs/heads/master.  Chris@starrose MINGW64 /d/projects/git/flask/git\_rebase\_quiz/git\_rebase (master)  $ git log -6  commit 6ff8cb82173ab9c55a1afe4d561dd38f81715325  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2  commit eeed82290490a011b339e00b83aa671c566de69f  Merge: 72fa7ad 1394424  Author: softcontext <softcontext@gmail.com>  Date: Thu May 12 18:00:35 2016 +0900  Merge branch 'hotfix1'  commit 1394424b15c1cbaefb6915dcd8e4a8a2391171ba  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:00:49 2016 +0900  2: hello.py updated on hotfix1  commit 72fa7ada0017abc4f27d7f351ba264b01c913fe5  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:04:53 2016 +0900  3: hello.py updated on master  commit 90ce34f3428c42f56c1d2b09257b1c720f7c2f48  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 21:57:07 2016 +0900  1: hello.py created on master |

### 작업대상 커밋

|  |
| --- |
| commit 429eb33242b3cd56a67ef22f4b7b3c23943c0603  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:14:26 2016 +0900  5: hello.py updated on hotfix3  commit edd229b510ac44a912ffa9c19f59806d628eff46  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2 |

### 작업결과 커밋

|  |
| --- |
| commit 6ff8cb82173ab9c55a1afe4d561dd38f81715325  Author: softcontext <softcontext@gmail.com>  Date: Wed May 11 22:09:33 2016 +0900  4: hello.py updated on hotfix2 |

작업대상 커밋과 그 사이에 있던 병합 커밋 내역이 모두 새로 만들어지는 커밋에 포함되었다.

## cherry-pick

히스토리를 하나의 가지로 관리하려고 master브랜치에 병합하기 전 master 브랜치를 기반으로

Rebase 해 보았다. rebase는 master 브랜치를 Fast-forward시켜 평평한 히스토리를 유지할 수

있는 방법이다.

한 브랜치에서 다른 브랜치로 작업한 내용을 옮기는 또 다른 방식으로 Cherry-pick이란 것도 있다.

Cherry-pick은 커밋 하나만 Rebase하는 것이다. 커밋 하나로 Patch 내용을 만들어 master 브랜치에

적용을 하는 것이다. hotfix1 브랜치에 있는 커밋중에서 하나만 고르거나 hotfix1 브랜치에 커밋이

하나밖에 없을 때 Rebase보다 유용하다.

### 작업 프로젝트 만들기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask  $ mkdir git\_cherry  Chris@starrose MINGW64 /d/projects/git/flask  $ cd git\_cherry  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry  $ git init  Initialized empty Git repository in D:/projects/git/flask/git\_cherry/.git/  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ touch memo.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git status  On branch master  Initial commit  Untracked files:  (use "git add <file>..." to include in what will be committed)  memo.txt  nothing added to commit but untracked files present (use "git add" to track)  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git add --all  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git commit -am 'project created'  [master (root-commit) 07980f8] project created  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 memo.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git log  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created |

### 연습 환경 만들기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git branch  \* master  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git checkout -b hotfix  Switched to a new branch 'hotfix'  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git branch  \* hotfix  master  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ touch hotfix.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git add \*  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git commit -am 'hotfix: add: hotfix.txt'  [hotfix 60b8984] hotfix: add: hotfix.txt  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 hotfix.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git log  commit 60b8984b031a1c2ffbed69b2b3698abe6abe6df3  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:11:46 2016 +0900  hotfix: add: hotfix.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ ls  hotfix.txt memo.txt |

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ ls  memo.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ touch master.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ ls  master.txt memo.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git add --all  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git commit -am 'master: add: master.txt'  [master 8dcf12e] master: add: master.txt  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 master.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git log  commit 8dcf12e87f1ac1c153b076758f195a8fd8d1c431  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:14:07 2016 +0900  master: add: master.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created |

### hotfix 브랜치에 커밋 하나를 따서(새로 만들어서) master 브랜치에 추가하기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git checkout hotfix  Switched to branch 'hotfix'  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git log  commit 60b8984b031a1c2ffbed69b2b3698abe6abe6df3  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:11:46 2016 +0900  hotfix: add: hotfix.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (hotfix)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git log  commit 8dcf12e87f1ac1c153b076758f195a8fd8d1c431  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:14:07 2016 +0900  master: add: master.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git cherry-pick 60b8  [master 33e2a83] hotfix: add: hotfix.txt  Date: Tue May 17 20:11:46 2016 +0900  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 hotfix.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git log  commit 33e2a837cac8cc643efc808da29f8da383c30c37  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:11:46 2016 +0900  hotfix: add: hotfix.txt  commit 8dcf12e87f1ac1c153b076758f195a8fd8d1c431  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:14:07 2016 +0900  master: add: master.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ ls  hotfix.txt master.txt memo.txt |

Rebase나 Cherry-pick 방식으로 브랜치를 합치고 나면 필요없는 브랜치나 커밋은 삭제하는 것이 좋다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git branch  hotfix  \* master  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git branch -d hotfix  error: The branch 'hotfix' is not fully merged.  If you are sure you want to delete it, run 'git branch -D hotfix'.  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git branch -D hotfix  Deleted branch hotfix (was 60b8984).  Chris@starrose MINGW64 /d/projects/git/flask/git\_cherry (master)  $ git log  commit 33e2a837cac8cc643efc808da29f8da383c30c37  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:11:46 2016 +0900  hotfix: add: hotfix.txt  commit 8dcf12e87f1ac1c153b076758f195a8fd8d1c431  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:14:07 2016 +0900  master: add: master.txt  commit 07980f84eec904fd52b9503310583cb70ad0010c  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:06:22 2016 +0900  project created |

번거로울 수 있는 브랜치 정리 작업을 피하거나 여러 브랜치의 이력을 남기고자 한다면

master 브랜치는 새로 파일을 만들거나 수정하지말고 오직 병합작업만 하는 것이 좋다.

## merge --squash

병합 작업 시 가져오는 브랜치에 모든 커밋을 하나의 커밋으로 뭉개서 작업 대상 브랜치에 적용한다.

작업 프로젝트 생성

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask  $ mkdir git\_squash  Chris@starrose MINGW64 /d/projects/git/flask  $ cd git\_squash  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash  $ touch who.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash  $ git init  Initialized empty Git repository in D:/projects/git/flask/git\_squash/.git/  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git add --all  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git commit -am 'project created'  [master (root-commit) c4f1877] project created  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 who.txt  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git log  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created |

master 브랜치에서 hello.py 파일 생성

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ cat hello.py  print("hello")  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git add --all  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git commit -am 'master: add: hello.py'  [master 8b041b3] master: add: hello.py  warning: LF will be replaced by CRLF in hello.py.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git log  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created |

hotfix 브랜치 만들기, hello.py 수정

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git checkout -b hotfix  Switched to a new branch 'hotfix'  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git log  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ cat hello.py  print("hello")  print("hi")  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git commit -am 'hotfix: update: hello.py: add: hi'  [hotfix 8909c80] hotfix: update: hello.py: add: hi  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git log  commit 8909c80746f6234aa22d1ec500f1ef7341985c02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 21:13:01 2016 +0900  hotfix: update: hello.py: add: hi  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created |

한번 더 hello.py 수정

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ vim hello.py  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ cat hello.py  print("hello")  print("hi")  print("good to see u")  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git commit -am 'hotfix: update: hello.py: add: good to see u'  [hotfix a56a9dd] hotfix: update: hello.py: add: good to see u  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git log  commit a56a9dd39e99f17ae0c581c6ff8abcfe61ba87cb  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 21:16:45 2016 +0900  hotfix: update: hello.py: add: good to see u  commit 8909c80746f6234aa22d1ec500f1ef7341985c02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 21:13:01 2016 +0900  hotfix: update: hello.py: add: hi  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created |

병합 작업 시 squash 적용하기

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ cat hello.py  print("hello")  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git merge --squash hotfix  Updating 8b041b3..a56a9dd  Fast-forward  Squash commit -- not updating HEAD  hello.py | 2 ++  1 file changed, 2 insertions(+)  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git log  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ cat hello.py  print("hello")  print("hi")  print("good to see u")  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (master)  $ git checkout hotfix  Switched to branch 'hotfix'  Chris@starrose MINGW64 /d/projects/git/flask/git\_squash (hotfix)  $ git log  commit a56a9dd39e99f17ae0c581c6ff8abcfe61ba87cb  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 21:16:45 2016 +0900  hotfix: update: hello.py: add: good to see u  commit 8909c80746f6234aa22d1ec500f1ef7341985c02  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 21:13:01 2016 +0900  hotfix: update: hello.py: add: hi  commit 8b041b35959109979c104eebf217c84cf6a1df73  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:49:03 2016 +0900  master: add: hello.py  commit c4f1877ebeb3df2fb40a98a5c51c864444cb25cd  Author: softcontext <softcontext@gmail.com>  Date: Tue May 17 20:38:58 2016 +0900  project created |

# SourceTree

소스트리는 애틀래시안에서 만든 깃과 머큐리얼을 위한 GUI 관리 도구다.

## 다양한 깃 GUI 도구 소개

http://git-scm.com/downloads/guis

## 설치

https://www.sourcetreeapp.com/download/

## 원격 저장소 복제

미리 만들어 본 eclipse\_hello.git을 대상으로 작업한다.

## 로컬 저장소 추가

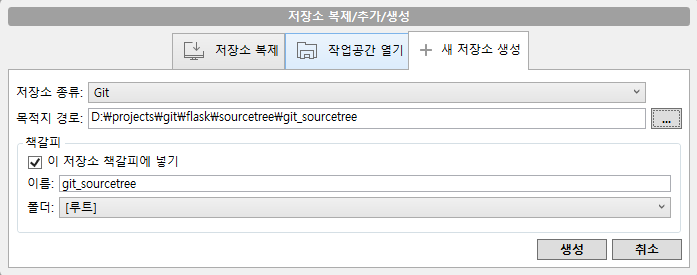
미리 만들어 본 git\_rebase 프로젝트를 대상으로 작업한다.

## 새로운 저장소 생성

### 1. 새 저장소 생성

소스트리 > 복제/생성 아이콘 > 새 저장소 생성 > 폴더 지정 > 생성

적당한 위치에 git\_sourcetree 라고 폴더를 새로 만들어 작업한다.



### 2. 파일 생성

소스트리는 텍스트 편집기를 지원하지 않으므로 깃 배쉬에서 작업한다.

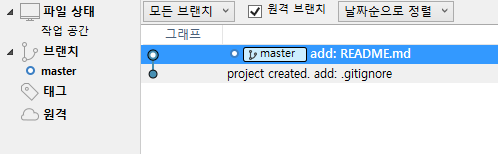
.gitignore 파일을 만들고 첫 번째 커밋을 수행한다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ ls -al  total 4  drwxr-xr-x 1 Chris 197121 0 5월 13 17:42 ./  drwxr-xr-x 1 Chris 197121 0 5월 13 17:40 ../  drwxr-xr-x 1 Chris 197121 0 5월 13 17:45 .git/  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ touch .gitignore  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ ls -al  total 4  drwxr-xr-x 1 Chris 197121 0 5월 13 17:46 ./  drwxr-xr-x 1 Chris 197121 0 5월 13 17:40 ../  drwxr-xr-x 1 Chris 197121 0 5월 13 17:45 .git/  -rw-r--r-- 1 Chris 197121 0 5월 13 17:46 .gitignore  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git status  On branch master  Initial commit  Untracked files:  (use "git add <file>..." to include in what will be committed)  .gitignore  nothing added to commit but untracked files present (use "git add" to track)  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git add --all  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git commit -am 'project created. add: .gitignore'  [master (root-commit) 6535146] project created. add: .gitignore  1 file changed, 0 insertions(+), 0 deletions(-)  create mode 100644 .gitignore  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git log  commit 6535146eb7360b16beabe2e0adc457c27d870f65  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:47:54 2016 +0900  project created. add: .gitignore |

README.md 파일을 추가로 생성한다.

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ cat README.md  This is a test repository for SourceTree.  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git add \*  warning: LF will be replaced by CRLF in README.md.  The file will have its original line endings in your working directory.  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git commit -am 'add: README.md'  [master ecb7bfb] add: README.md  warning: LF will be replaced by CRLF in README.md.  The file will have its original line endings in your working directory.  1 file changed, 1 insertion(+)  create mode 100644 README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git log  commit ecb7bfba41dedffe43ca1fc239c1cac06d04f9ab  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:52:40 2016 +0900  add: README.md  commit 6535146eb7360b16beabe2e0adc457c27d870f65  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:47:54 2016 +0900  project created. add: .gitignore |

### 3. 소스트리에서 커밋 히스토리 확인



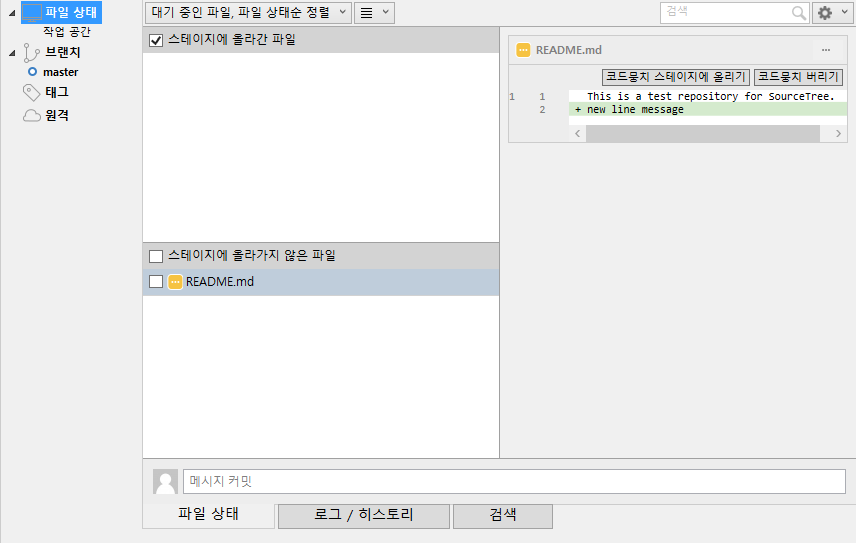
### 4. 파일 수정 후 소스트리에서 처리

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ cat README.md  This is a test repository for SourceTree.  new line message |

#### 소스트리로 스테이지에 추가

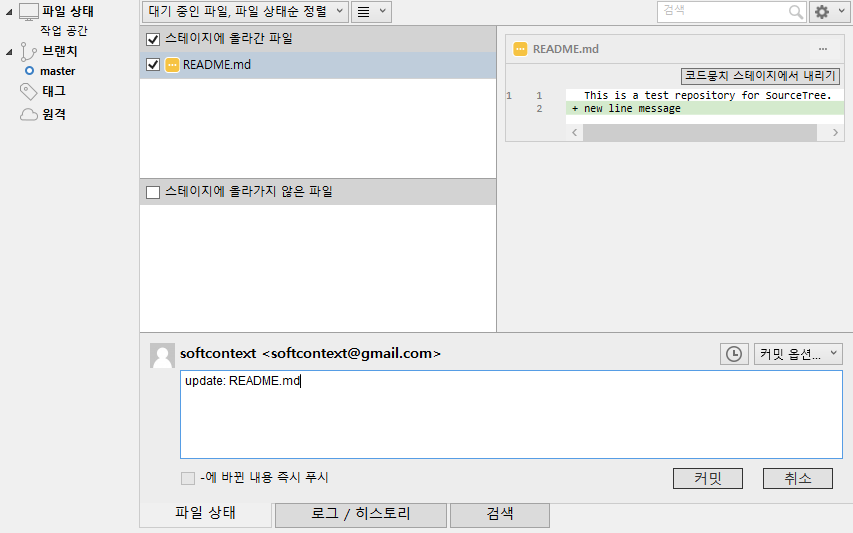
소스트리 > 파일상태 > 스테이지에 올라가지 않은 파일 확인 후 클릭

> 스테이지에 올라간 파일로 변경된다. 이 작업은 git add 명령을 실행한 것과 같다.

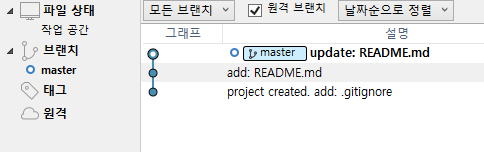


#### 소스트리로 커밋

소스트리 > 커밋 아이콘 > 커밋 메시지 창이 보인다. 메시지 작성 후 > 커밋 버튼 클릭

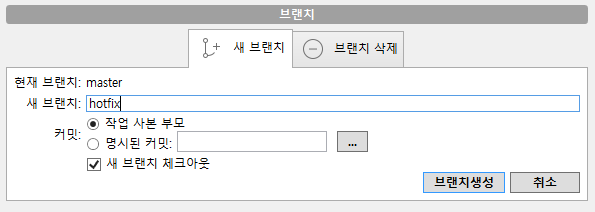


### 5. 소스트리에서 커밋 히스토리 확인



### 6. 새로운 브랜치 생성

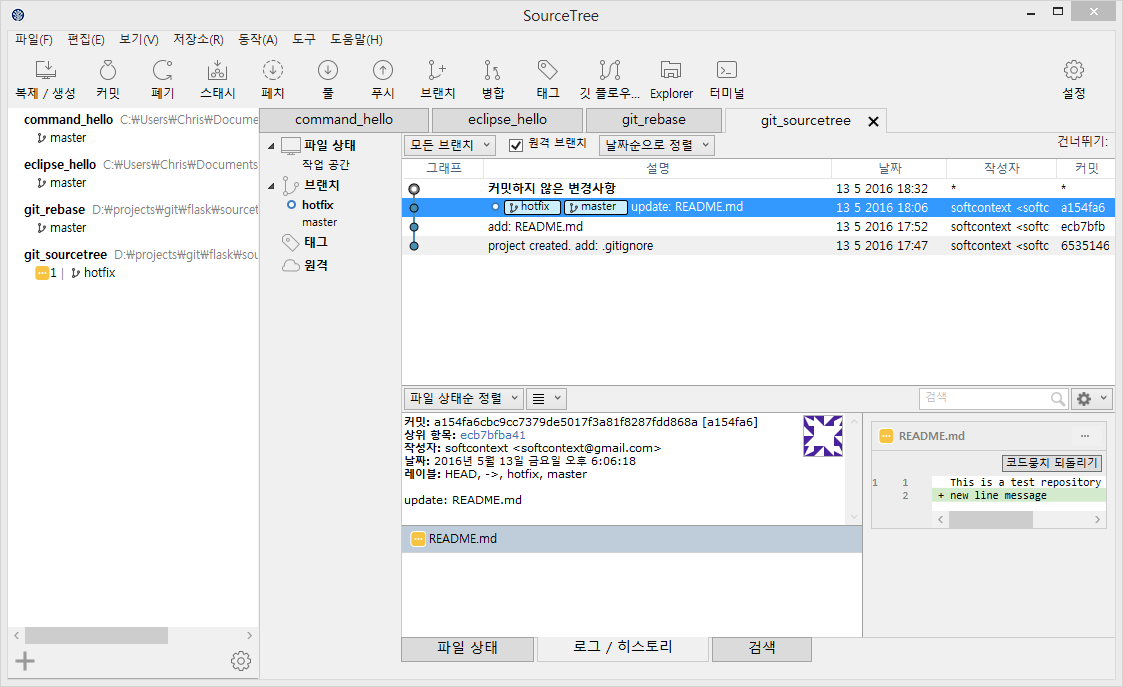
소스트리 > 브랜치 아이콘 > 브랜치 이름 입력 > 브랜치생성 클릭



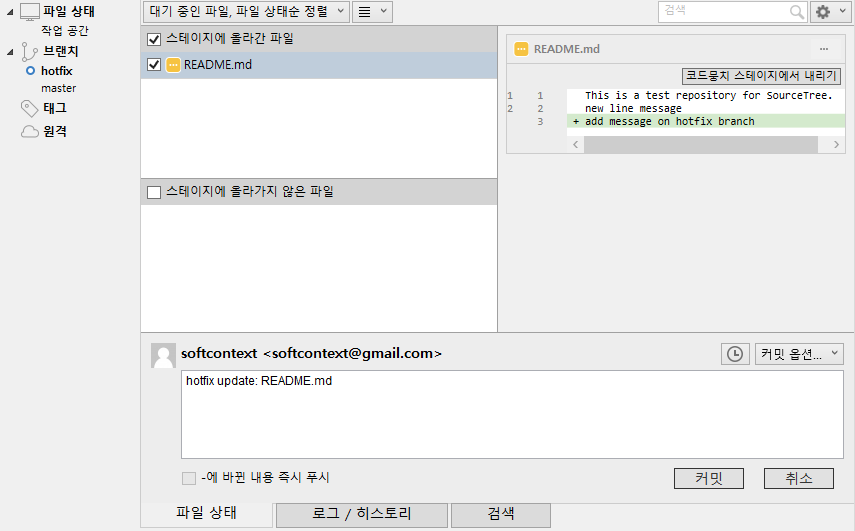
### 7. hotfix 브랜치에서 파일 수정

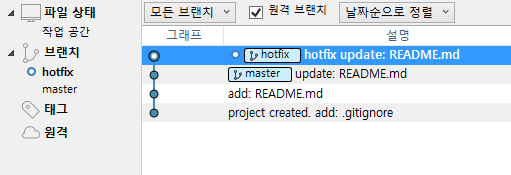
|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix)  $ cat README.md  This is a test repository for SourceTree.  new line message  add message on hotfix branch |

### 8. 소스트리에서 상태 확인

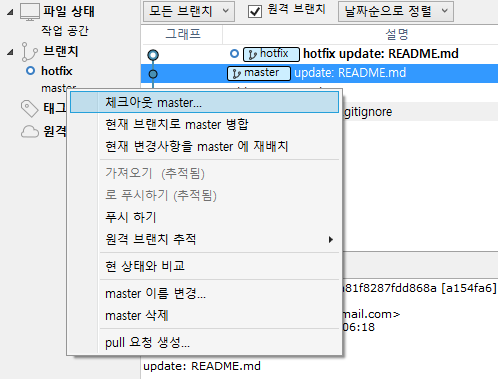


### 9. 소스트리에서 스테이지에 올린 후 커밋





### 10. master 브랜치로 이동



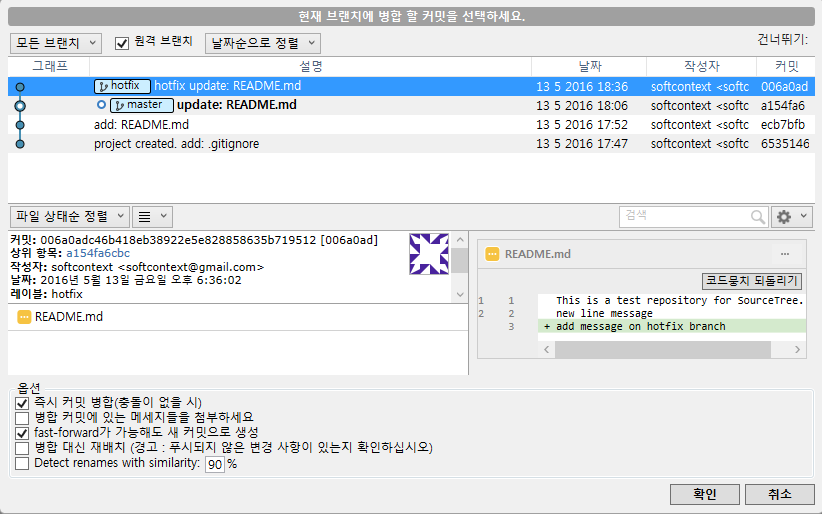
### 11. 병합

소스트리 > 병합 아이콘 > 아래 정보대로 선택 후 > 확인 버튼

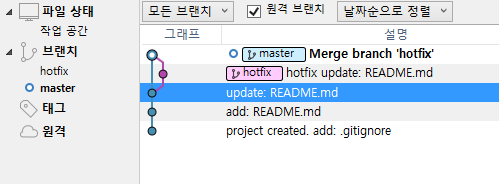
현재 작업 브랜치: master

병합 대상 브랜치: hotfix

병합 커밋을 기록으로 남기기 위해서 'fast-forward가 가능해도 새 커밋으로 생성' 체크박스를 선택

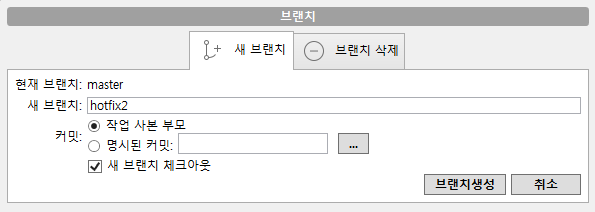


#### 작업결과 확인



## 충돌 해결

### 1. 새 브랜치 hotfix2를 만든다.



### 2. 파일 수정

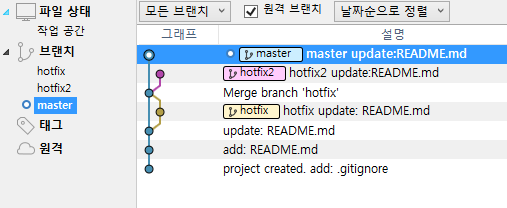
|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 13 18:43 ./  drwxr-xr-x 1 Chris 197121 0 5월 13 17:40 ../  drwxr-xr-x 1 Chris 197121 0 5월 13 18:56 .git/  -rw-r--r-- 1 Chris 197121 0 5월 13 17:46 .gitignore  -rw-r--r-- 1 Chris 197121 91 5월 13 18:43 README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ cat README.md  This is a test repository for SourceTree.  new line message  add message on hotfix branch  #add comment on hotfix2  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ git status  On branch hotfix2  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")  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ git commit -am 'hotfix2 update:README.md'  [hotfix2 20a688a] hotfix2 update:README.md  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ git log  commit 20a688a616bb1e24d6ba34512176a76a309e91b0  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:58:44 2016 +0900  hotfix2 update:README.md  commit 7813a18b04cb135ef56220428be368b13f95911c  Merge: a154fa6 006a0ad  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:43:45 2016 +0900  Merge branch 'hotfix'  commit 006a0adc46b418eb38922e5e828858635b719512  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:36:02 2016 +0900  hotfix update: README.md  commit a154fa6cbc9cc7379de5017f3a81f8287fdd868a  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:06:18 2016 +0900  update: README.md  commit ecb7bfba41dedffe43ca1fc239c1cac06d04f9ab  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:52:40 2016 +0900  add: README.md  commit 6535146eb7360b16beabe2e0adc457c27d870f65  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:47:54 2016 +0900  project created. add: .gitignore |

### 3. master 브랜치로 이동 후 파일 수정

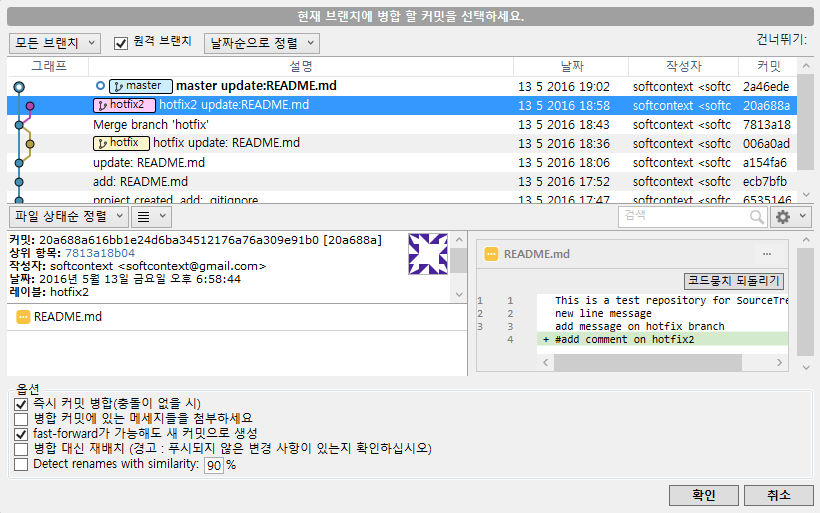
|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (hotfix2)  $ git checkout master  Switched to branch 'master'  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ ls -al  total 5  drwxr-xr-x 1 Chris 197121 0 5월 13 19:00 ./  drwxr-xr-x 1 Chris 197121 0 5월 13 17:40 ../  drwxr-xr-x 1 Chris 197121 0 5월 13 19:00 .git/  -rw-r--r-- 1 Chris 197121 0 5월 13 17:46 .gitignore  -rw-r--r-- 1 Chris 197121 91 5월 13 19:00 README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ cat README.md  This is a test repository for SourceTree.  new line message  add message on hotfix branch  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ cat README.md  This is a test repository for SourceTree.  new line message  add message on hotfix branch  #add comment on master  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git commit -am 'master update:README.md'  [master 2a46ede] master update:README.md  1 file changed, 1 insertion(+)  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git log  commit 2a46ede3751923813e597bd5a5d3a69df2eaa902  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 19:02:20 2016 +0900  master update:README.md  commit 7813a18b04cb135ef56220428be368b13f95911c  Merge: a154fa6 006a0ad  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:43:45 2016 +0900  Merge branch 'hotfix'  commit 006a0adc46b418eb38922e5e828858635b719512  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:36:02 2016 +0900  hotfix update: README.md  commit a154fa6cbc9cc7379de5017f3a81f8287fdd868a  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 18:06:18 2016 +0900  update: README.md  commit ecb7bfba41dedffe43ca1fc239c1cac06d04f9ab  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:52:40 2016 +0900  add: README.md  commit 6535146eb7360b16beabe2e0adc457c27d870f65  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:47:54 2016 +0900  project created. add: .gitignore |

### 4. 소스트리에서 병합 시도

#### 커밋 상태 확인

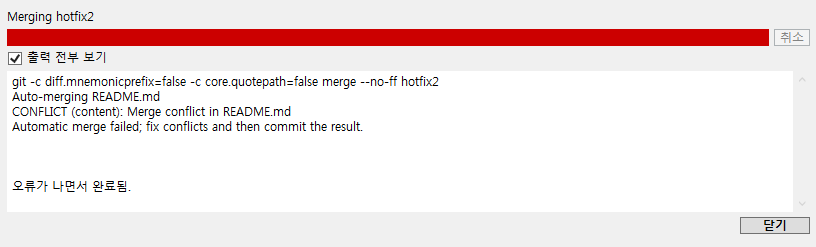


#### 병합 시도



#### 충돌 발생

메시지를 확인하고 닫기 버튼을 클릭한다.

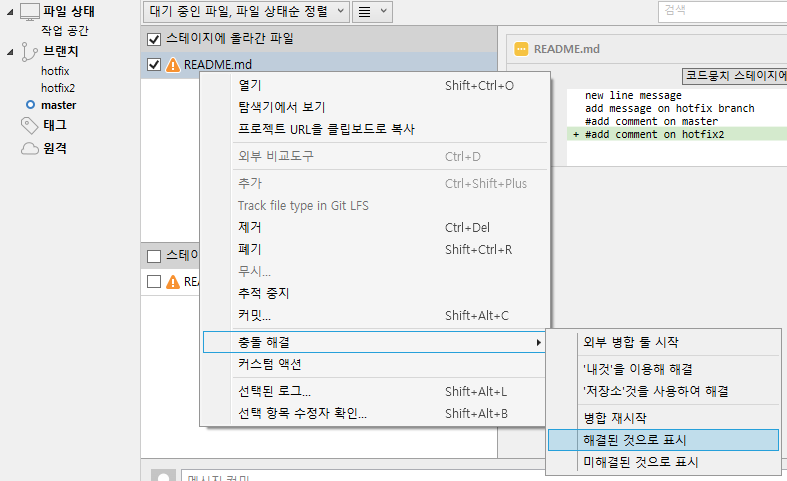


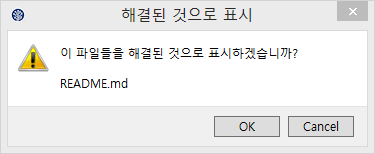
#### 충돌 해결

|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master|MERGING)  $ git diff  **diff --cc README.md**  **index 40892b2,51e3d9a..0000000**  **--- a/README.md**  **+++ b/README.md**  @@@ -1,4 -1,4 +1,8 @@@  This is a test repository for SourceTree.  new line message  add message on hotfix branch  ++<<<<<<< HEAD  +#add comment on master  ++=======  + #add comment on hotfix2  ++>>>>>>> hotfix2  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master|MERGING)  $ vim README.md  Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master|MERGING)  $ cat README.md  This is a test repository for SourceTree.  new line message  add message on hotfix branch  #add comment on master  #add comment on hotfix2 |

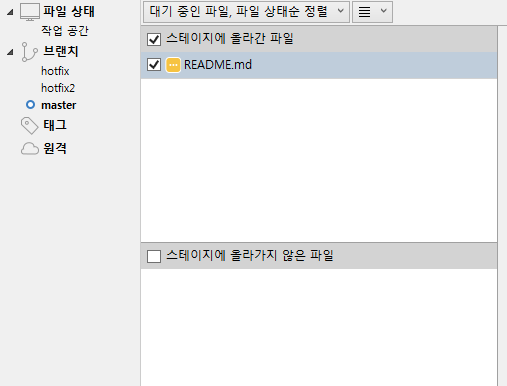
#### 충돌 해결 표시

병합 창을 닫고 > 하단 파일 상태 탭 선택 > 스테이지에 올라간 파일 영역에서 해결된 것으로 표시



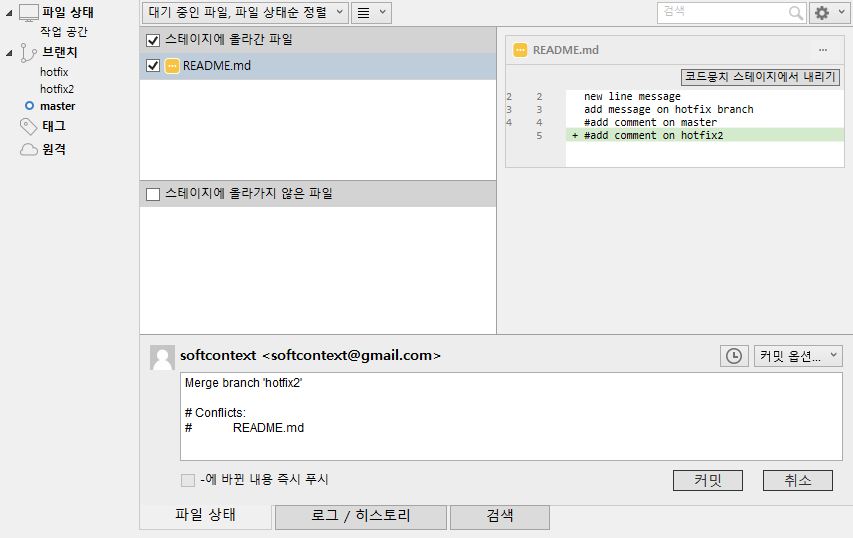


스테이지에 올라가지 않은 파일 부분에 파일 내역이 사라진다.

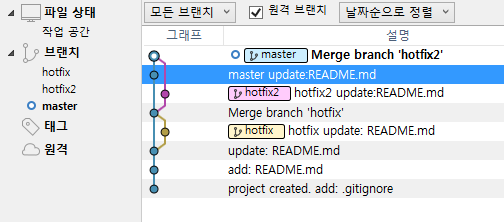


#### 커밋 실행

병합 충돌이 발생한 후의 커밋은 자동으로 준비된 메시지를 미리 넣어준다.



#### 로그 내역 확인



|  |
| --- |
| Chris@starrose MINGW64 /d/projects/git/flask/sourcetree/git\_sourcetree (master)  $ git log --graph  \* commit e9826e89805a5057d93df54b00ab495a93a0d264  |\ Merge: 2a46ede 20a688a  | | Author: softcontext <softcontext@gmail.com>  | | Date: Fri May 13 19:17:20 2016 +0900  | |  | | Merge branch 'hotfix2'  | |  | | # Conflicts:  | | # README.md  | |  | \* commit 20a688a616bb1e24d6ba34512176a76a309e91b0  | | Author: softcontext <softcontext@gmail.com>  | | Date: Fri May 13 18:58:44 2016 +0900  | |  | | hotfix2 update:README.md  | |  \* | commit 2a46ede3751923813e597bd5a5d3a69df2eaa902  |/ Author: softcontext <softcontext@gmail.com>  | Date: Fri May 13 19:02:20 2016 +0900  |  | master update:README.md  |  \* commit 7813a18b04cb135ef56220428be368b13f95911c  |\ Merge: a154fa6 006a0ad  | | Author: softcontext <softcontext@gmail.com>  | | Date: Fri May 13 18:43:45 2016 +0900  | |  | | Merge branch 'hotfix'  | |  | \* commit 006a0adc46b418eb38922e5e828858635b719512  |/ Author: softcontext <softcontext@gmail.com>  | Date: Fri May 13 18:36:02 2016 +0900  |  | hotfix update: README.md  |  \* commit a154fa6cbc9cc7379de5017f3a81f8287fdd868a  | Author: softcontext <softcontext@gmail.com>  | Date: Fri May 13 18:06:18 2016 +0900  |  | update: README.md  |  \* commit ecb7bfba41dedffe43ca1fc239c1cac06d04f9ab  | Author: softcontext <softcontext@gmail.com>  | Date: Fri May 13 17:52:40 2016 +0900  |  | add: README.md  |  \* commit 6535146eb7360b16beabe2e0adc457c27d870f65  Author: softcontext <softcontext@gmail.com>  Date: Fri May 13 17:47:54 2016 +0900  project created. add: .gitignore |

# GitLab

GitLab은 폐쇄적으로 이용할 수 있는 원격 저장소 서비스를 제공한다. 사용법은 깃허브와 비슷하므로 자세한 설명은 생략한다.

