

신입사원 Git 교육

개인 환경 만들어 실습하기

Git 도움받기

\$ git help <명령어>

\$ git <명령어> --help

The image shows a terminal window on the left and a web browser on the right. The terminal window displays the command `$ git remote --help` and the output `$ |`. The web browser shows the `git-remote(1) Manual Page` from the Git documentation. The page includes the `NAME` section, which states `git-remote - Manage set of tracked repositories`, and the `SYNOPSIS` section, which lists various `git remote` commands and their options.

git-remote(1) Manual Page

NAME

git-remote - Manage set of tracked repositories

SYNOPSIS

`git remote [-v] [--verbose]`
`git remote add [-t <branch>] [-m <master>] [-f] [--no-]tags] [--mirror=<fetch | push>]`
`git remote rename <old> <new>`
`git remote remove <name>`
`git remote set-head <name> (-a | --auto | -d | --delete | <branch>)`
`git remote set-branches [--add] <name> <branch>...`
`git remote set-url [--push] [--all] <name>`
`git remote set-url [--push] <name> <newurl> [<oldurl>]`
`git remote set-url --add [--push] <name> <newurl>`
`git remote set-url --delete [--push] <name> <url>`

* html의 경로

```
$ git help eunsil
C:/Program Files/Git/mingw64/share/doc/git-doc/giteunsil.html
```

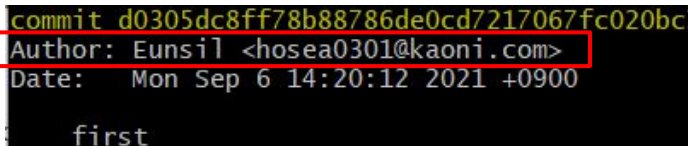
초기 환경설정

\$ git config

First-Time Git Setup

• Your Identity

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```



```
commit d0305dc8ff78b88786de0cd7217067fc020bc
Author: Eunsil <hosea0301@kaoni.com>
Date: Mon Sep 6 14:20:12 2021 +0900

first
```

• Checking Your Settings

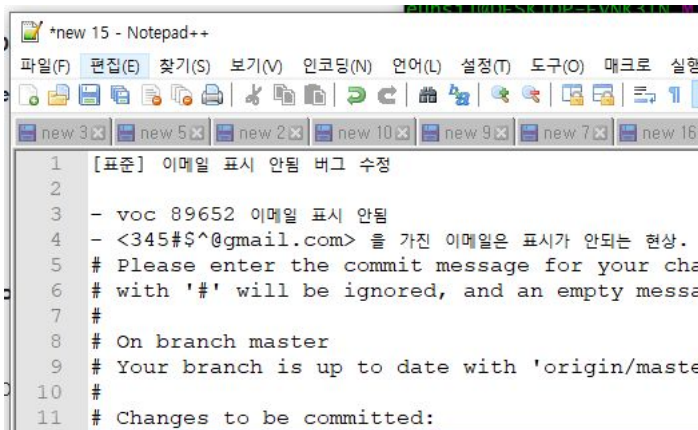
```
$ git config --global --list
user.name=Scott Chacon
user.email=schacon@gmail.com
...
```

```
$ git config --global user.name
Scott Chacon
```

• Git for Windows

```
$ git config --global core.editor
'''/c/Program Files (x86)/Notepad++/notepad++.exe' -multiInst
-notabbar -nosession -noPlugin'''
```

* 노트패드 설정



```
$ git config --global core.autocrlf true
$ git config --global fetch.prune true
```

* 줄바꿈 설정 : lf → crlf (Linux, git : lf / Windows : crlf)
* 원격 저장소에서 브랜치가 지워지면 → 로컬에 자동으로 반영하는 옵션.

\$ git config (alias / --unset)

```
$ git config --global alias.au "log --author=Eunsil"
```

```
$ git au --grep=12
commit 960c8af69148b7d30797150b7
Author: Eunsil <hosea0301@kaoni.com>
Date: Tue Sep 7 11:54:42 2021

12th
```

```
$ git config --global alias.lg log --graph --abbrev-commit --decorate --format=format:
```

위 명령어를 실행한 결과 ~/.gitconfig 파일에는 다음과 같은 문자열이 추가된다.

```
[alias]
  lg = log --graph --abbrev-commit --decorate --fo
```

```
$ git config --global --list
user.name=Eunsil
user.email=hosea0301@kaoni.com
core.autocrlf=true
core.editor=vim
fetch.prune=true
color.ui=auto
alias.au=log --author=Eunsil
```

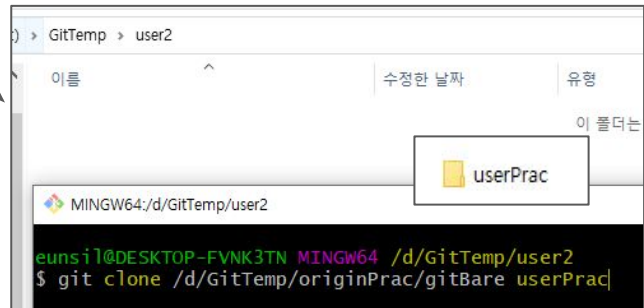
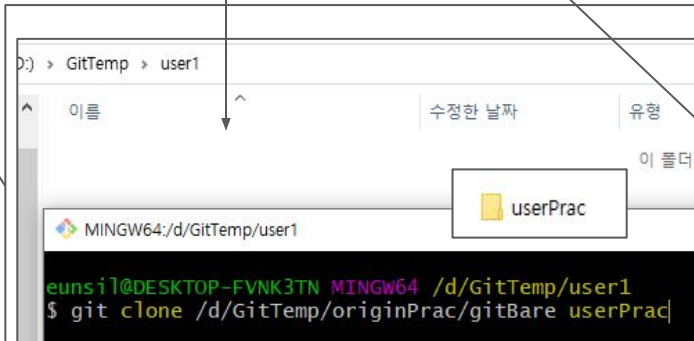
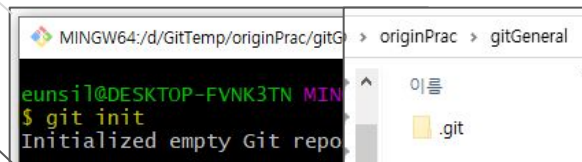
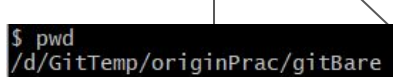
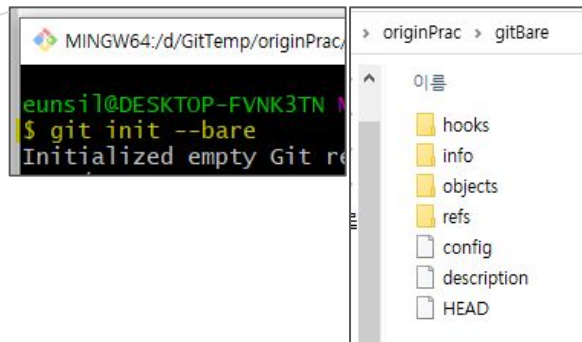
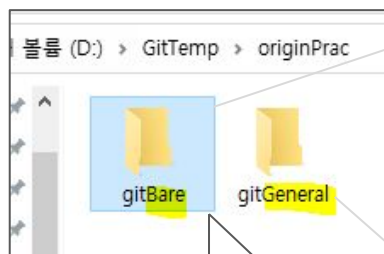
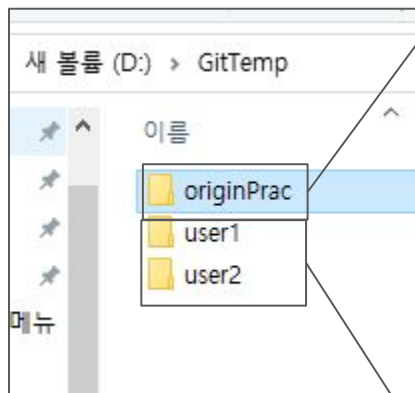
```
$ git config --global --unset alias.au

eunsil@DESKTOP-FVNK3TN MINGW64 /d/GitTe
$ git config --global --list
user.name=Eunsil
user.email=hosea0301@kaoni.com
core.autocrlf=true
core.editor=vim
fetch.prune=true
color.ui=auto
```

Git 만들기

\$ git init (--bare)

(원격 저장소) **bare**-repository: 워킹 트리가 없고 변경사항만 추적하는 저장소
(일반 저장소) : 원격 저장소가 따로 존재하는 로컬 저장소



Git 복제하기

\$ git clone <파일 경로>

Git on the Server

- The Protocols

- : Local Protocol

This is often used if everyone on your team has access to a shared filesystem such as an NFS mount

```
eunsil@DESKTOP-EVVK3TN MINGW64 /d/GitTemp/user1
$ git clone /d/GitTemp/originPrac/gitBare
```

- : The SSH Protocol

SSH is the only network-based protocol that you can easily read from and write to. The other two network protocols (HTTP and Git) are generally read-only.

```
eunsil@DESKTOP-EVVK3TN MINGW64 /d/GitTemp/user1/user1
$ git clone jmocha@10.0.102.8:/git/ezFlow4Java.git
```

- : The Git Protocol

provides a service similar to the SSH protocol, but with absolutely no authentication.

```
eunsil@DESKTOP-EVVK3TN MINGW64 /d/GitTemp/user1/user1
$ git clone git://github.com/tswicegood/mysite.git
```

- : The HTTP/S Protocol

The beauty of the HTTP or HTTPS protocol is the simplicity of setting it up. It's possible to make Git push over HTTP as well, although that technique isn't as widely used and requires you to set up complex WebDAV requirements.

```
$ git clone https://github.com/libgit2/libgit2
```

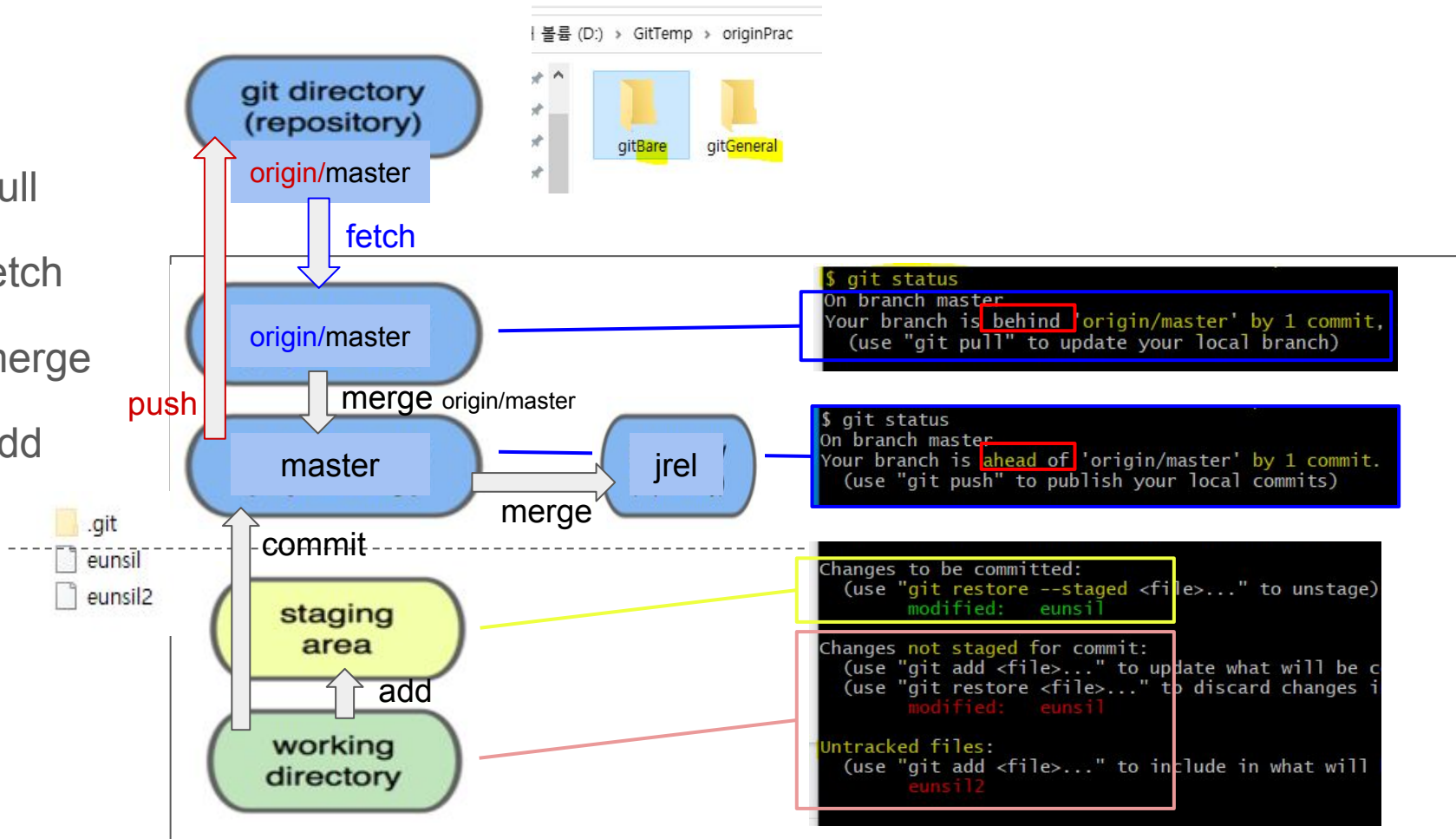
영역

\$ git pull

\$ git fetch

\$ git merge

\$ git add



영역

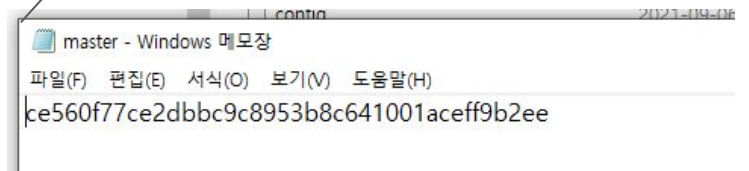
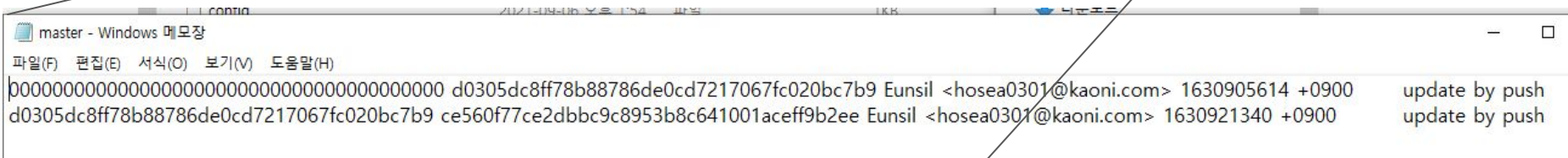
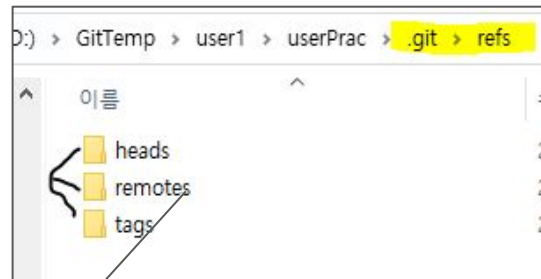
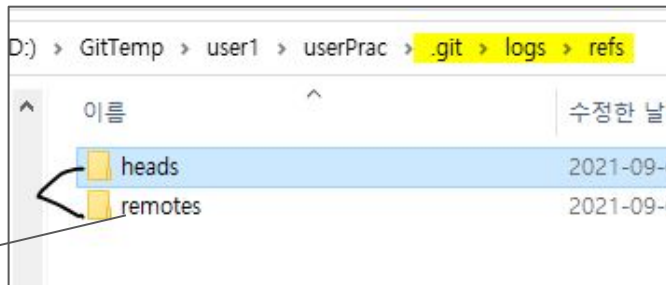


origin/master

origin

master

HEAD



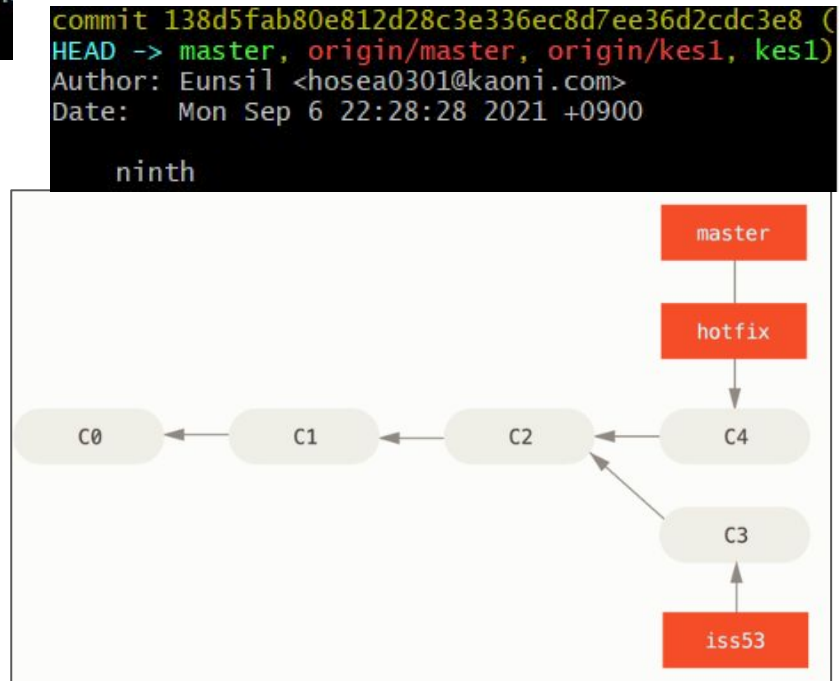
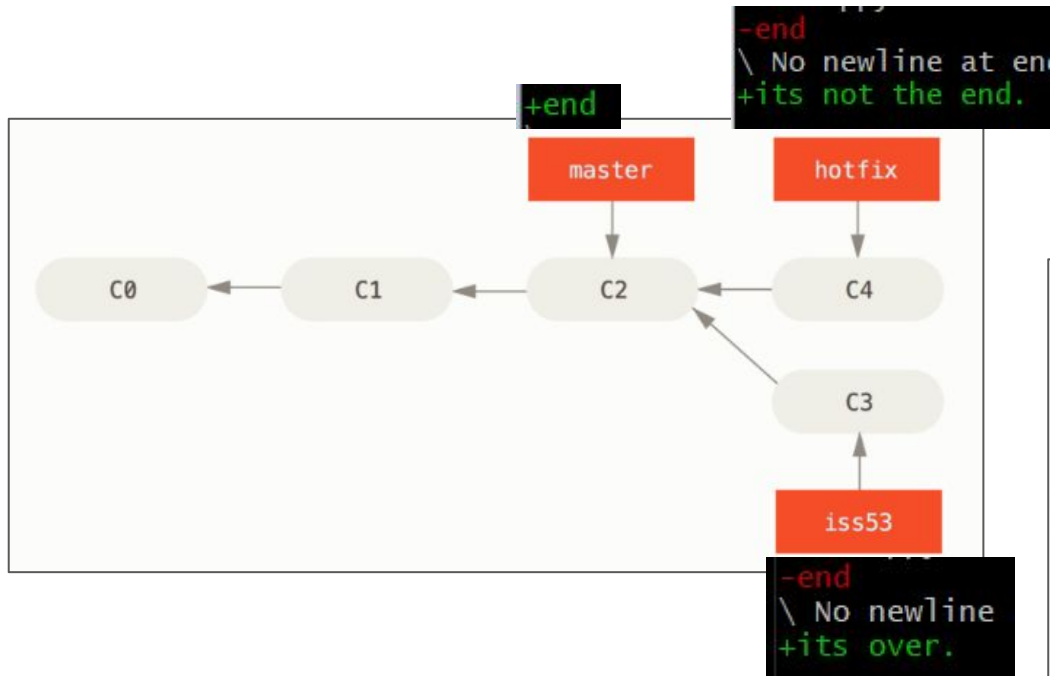
커밋 개체 / 브랜치

```
commit 138d5fab80e812d28c3e336ec8d7ee36d2cdc3e8 (HEAD -> kes1, origin/kes1)
Author: Eunsil <hosea0301@kaoni.com>
Date: Mon Sep 6 22:28:28 2021 +0900

ninth

commit fcb60cb9f8fd3cab175aa73f61687cf9d621846 (origin/master, master)
Author: Eunsil <hosea0301@kaoni.com>
Date: Mon Sep 6 21:40:00 2021 +0900

eighth
```



\$ git status

add

```
Changes to be committed:  
(use "git restore --staged <file>..." to unstage)  
modified:   eunsil
```

```
Changes not staged for commit:  
(use "git add <file>..." to update what will be committed)  
(use "git restore <file>..." to discard changes in working directory)  
modified:   eunsil
```

```
Untracked files:  
(use "git add <file>..." to include in what will be committed)  
eunsil2
```

\$ git restore --staged <파일>

파일 되돌리기

\$ git restore <파일명>

\$ git checkout <파일명>

\$ git reset (--hard) HEAD^

\$ git diff (--staged,cached <파일> / --stat <기간> <파일>)

```
$ git status
On branch master
Your branch is up to date with
the remote branch master.

Changes not staged for commit:
  (use "git add <file>..." to
  (use "git restore <file>..."
  modified:   eunsil
```

```
no changes added to commit (use
```

```
eunsil@DESKTOP-FVNK3TN MINGW64
```

```
$ git diff
```

```
diff --git a/eunsil b/eunsil
index 6989ef5..a8cce12 100644
```

```
--- a/eunsil
+++ b/eunsil
@@ -1,3 +1,3 @@
```

```
hi
i am eunsil
-oh no
+be happy!
```

```
$ git status
On branch master
Your branch is up to date with
the remote branch master.

Changes to be committed:
  (use "git restore --staged <
  modified:   eunsil
```

```
eunsil@DESKTOP-FVNK3TN MINGW64
$ git diff
```

```
eunsil@DESKTOP-FVNK3TN MINGW64
```

```
$ git diff --staged
```

```
$ git diff --cached
```

```
diff --git a/eunsil b/eunsil
index 6989ef5..a8cce12 100644
```

```
--- a/eunsil
+++ b/eunsil
@@ -1,3 +1,3 @@
hi
i am eunsil
-oh no
+be happy!
```

```
$ git diff --stat HEAD~5
.gitignore | 1 +
eunsil     | 2 ++
2 files changed, 3 insertions(+)
```

```
$ git diff --stat HEAD~5 eunsil
eunsil | 2 ++
1 file changed, 2 insertions(+)
```

- o git diff <branch명> <다른 branch명> : 로컬의 브랜치간 비교
- o git diff <branch명> <origin/branch명> : 로컬 브랜치와 원격 브랜치간의 비교
- o git diff <commit해시마이다> <commit해시마이다> : 커밋끼리 비교
- o git diff <비교대상 1>..<비교대상 2> : 비교대상1과 비교대상2의 차이점 비교 .. 이 들어감

\$ git commit (-a / -m <메세지>)

-a

-m <msg>

--all

--message=<msg>

```
$ git status
On branch master
Your branch is up to date with 'origin/master'

Changes to be committed:
  (use "git restore --staged" to unstage)
        modified:   eunsil
```

```
$ git commit
```

```
$ git status
On branch master
Your branch is up to date with 'origin/master'

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   eunsil

no changes added to commit (use "git add" to upload)
```

```
$ git commit -a
```

```
$ git commit -a -m "eighth"
[master 94fd8b2] eighth
1 file changed, 1 insertion(+),
```

\$ git commit (--amend / -c / -C)

amend 미국·영국 [ə'mend]

동사 (법 등을) 개정[수정]하다

-C <commit>

--reuse-message=<commit>

Take an existing commit object when creating the commit.

-c <commit>

--reedit-message=<commit>

-C da867d23f9459

```
$ git commit -a
[master 3cafe70] eighth
1 file changed, 1 insertion(+)
```

```
commit 7c7fca5c608423dad9d9
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:22:44

eighth
```

```
$ git commit --amend
```

```
$ git commit -a --amend
```

```
$ git commit -c HEAD -a
[master 531a57b] eighth
Date: Mon Sep 6 21:29:14 2021 +0900
1 file changed, 1 insertion(+)
```

```
$ git commit -C HEAD -a
```

```
$ git commit -C HEAD -a --amend
```

```
$ git log
commit c79e013530eca7214b6fe
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:03:17 2
```

여덟

```
commit e14c19391496f44e811a2
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 20:06:42 2
```

seventh

```
commit 702a40323033748b4698
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:29:14
```

eighth_수정 1

```
commit e532741916f6812b0e57
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:29:14
```

eighth

```
commit f2b8224f1f44e6e88232
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:29:14
```

eighth

```
commit e532741916f6812b0e57
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:29:14
```

eighth

```
commit d1914fa4041e395a60e8d
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 21:29:14 2
```

eighth

```
commit e14c19391496f44e811a2
Author: Eunsil <hosea0301@k
Date: Mon Sep 6 20:06:42 2
```

seventh

\$ git log (-<개수> / --author=<등록자> / --grep=<검색어> / <파일>)

```
$ git log -1
commit fcb60cb9f8fd3c
Author: Eunsil <hosea0301@kaoni.com>
Date: Mon Sep 6 21:44:12 2021 +0900

    eighth
```

```
$ git log --author=Eunsil
commit 9d5dd4c2df95f29fb078
Author: Eunsil <hosea0301@kaoni.com>
Date: Thu Sep 2 16:06:24 2021 +0900
```

[카 이 스트] (mail) 메 인

- 120시 간 (5월)로 변 경

```
commit f7fc5bf51c107f188c96
Author: Eunsil <hosea0301@kaoni.com>
Date: Tue Aug 17 14:38:48 2021 +0900
```

[카 이 스트] (mail) 메 인

```
commit 8a555dccfd0fceedb53a
Author: Eunsil <hosea0301@kaoni.com>
Date: Fri Apr 9 17:59:16 2021 +0900
```

```
$ git log --grep=웹 폴 더
commit fa8c5c24dcc613812ee2cd444a0f4
Author: suasua <tndk19@kaoni.com>
Date: Wed Apr 21 17:24:10 2021 +0900
```

[웹 폴 더] 카 이 스트 개 발 건 표 준 적

- (#4150) 생 성 자 인 데 폴 더 삭 제 안
> 삭 제 하 려 고 하 는 폴 더 하 위 까 지
안 내 문 구 에 '하 위 포 함' 메 시 지 추 가
> 폴 더 와 파 일 의 생 성 자 를 चे क ह ळ

```
commit d3e7888c68c95d2bd8aeeb4108934
Author: jwseo99 <jwseo99@kaoni.com>
Date: Wed Apr 14 15:12:05 2021 +0900
```

[카 이 스트] 웹 폴 더 줄 겨 찾 기 에 서 구

```
commit b90e6618be72b2fbdeaa4c576aed2
Author: jwseo99 <jwseo99@kaoni.com>
Date: Wed Apr 14 15:09:02 2021 +0900
```

[카 이 스트] 웹 폴 더 줄 겨 찾 기 에 서 구

```
commit 8a555dccfd0fceedb53a6c9e16bfe
Author: Eunsil <hosea0301@kaoni.com>
Date: Fri Apr 9 17:59:16 2021 +0900
```

[웹 폴 더] 구 성 원 을 가 저 오 는 데 ten

- EzWebFolderDAO.getFolderUsers

```
$ git log src/main/resources/egovframework/egovProps/config.properties
commit c6b95d4005202ab75781646acc6f3ba3tae0db70
Author: HongSeungBi <hsb1024@kaoni.com>
Date: Mon Feb 1 10:48:45 2021 +0900
```

* config의 오 타 수 정

```
commit 0c18fd89320df18df51734f0c7f76d882275c17a
Author: KimMinSeong <mins99@kaoni.com>
Date: Tue Dec 29 15:24:54 2020 +0900
```

전 자 결 제 > 문 서 유 통 관 련 config 파 일 수 정

```
commit d97478640ca7a3831217d0a0f2bedc0b93087c92
Merge: b30f6c0a5c 78fb61a22f
Author: shim8934 <shim8934@kaoni.com>
Date: Mon Dec 7 15:11:40 2020 +0900
```

Merge branch 'site_kaist' of 20.41.120.84:git/ezFlow4Java into site

```
commit b30f6c0a5c7106aa96160bf14f05391e262e1da4
Author: shim8934 <shim8934@kaoni.com>
Date: Mon Dec 7 15:10:01 2020 +0900
```

REQ129 - 기 록 를 존 재 하 는 기 록 철 만 복 사 할 수 있 는 옵션 추 가

화 면 UI는 계 획 서 에 없 지 만 , display:none 처 리 로 관 리 자 단 에 서 메 뉴 보

```
commit f77a1b684c7560ce10e01c1ec78d6d9e95078724
Author: kjh <kjh@kaoni.com>
Date: Mon Dec 7 10:25:34 2020 +0900
```

config.UserInfo_ApprovalG_VIEW 닷 넷 카 이 스트 설 정 으 로 변 경

```
commit 3d8eaaf51ca98b657c3e7930764e2437f28b20f0
Author: shim8934 <shim8934@kaoni.com>
```


\$ git log (--since,before,until,after / --pretty=<포맷> / --graph)

```
$ git log --since=1month
commit 9d5dd4c2df95f29fb078ecb98c7e22068
stWebfolderPortletCount, kes_kaistWebfol
Author: Eunsil <hosea0301@kaoni.com>
Date: Thu Sep 2 16:06:24 2021 +0900

[카 이 스트] (mail) 메 인 화 면 신 규 업 무
- 120시 간 (5월)로 변경

commit f7fc5bf51c107f188c96cb57a5c4f06ac
Author: Eunsil <hosea0301@kaoni.com>
Date: Tue Aug 17 14:38:48 2021 +0900

[카 이 스트] (mail) 메 인 화 면 신 규 업 무
```

--since, --after 명시한 날짜 이후의 커밋만 검색한다.

--until, --before 명시한 날짜 이전의 커밋만 조회한다.

```
$ git log --pretty=format:"%h %s" --graph
* 4da245265b Merge branch 'master' of jmoc
|
| * 5f9b6d4697 메 인 - 게 시 판 포 틀 릿
| * 5d604d2f2f [강 남 대] #76431 게 시 판 > 포 토
| * fc3c670838 [강 남 대] #76424 포 토 /썸 네 일 게
| * 354f307266 Merge branch 'master' of 10
|
| * 6bac2455bd Merge branch 'master' of
|
| * aecba1b065 * ezEKP SSL 적 용 방 법 설 명
| * 7d6b8d9220 [강 남 대] #76505 일 정 관 리
| * 시 깨 저 서 표 시
|
| * 16bfd63e94 Merge branch 'master' of
|
| * 431e0a7dd7 ezEKP 자 바 개 발 자 가 이 드
|
| * 7f09d3659b * ezMobile 관 련 문 서 정 리
| * 2eeecdd0c [강 남 대] #76418 '게 시 판 > 포
| * 할 시 '이 후 로 표 시 되 지 않 는 현 상
```

표 1. git log --pretty=format 에 쓸 몇가지 유용한 옵션

옵션	설명
%H	커밋 해시
%h	짧은 길이 커밋 해시
%T	트리 해시
%t	짧은 길이 트리 해시
%P	부모 해시
%p	짧은 길이 부모 해시
%an	저자 이름
%ae	저자 메일
%ad	저자 시각 (형식은 --date=옵션 참고)
%ar	저자 상대적 시각
%cn	커미터 이름
%ce	커미터 메일
%cd	커미터 시각
%cr	커미터 상대적 시각
%s	요약

\$ git show

da867d23f9459

\$ git show HEAD

commit 9d5dd4c2df95f29fb078ecb98c7e22068f4775ce (HEAD -> site_newkaist, origin/site_newkaist, origin/ksWebfolderPortletCount, kes_kaistWebfolderPortletCount)

Author: Eunsil <hosea0301@kaoni.com>

Date: Thu Sep 2 16:06:24 2021 +0900

[카 이 스트] (mail) 메 인 화 면 신 규 업 무 자 료 건 수 표 시 _수 정 1

- 120시 간 (5일)로 변 경

diff --git a/src/main/java/egovframework/ezEKP/ezNewPortal/service/EzNewPortalService.java b/src/main/java/egovframework/ezEKP/ezNewPortal/service/EzNewPortalService.java

index e3142a04a5..2ef422628b 100644

--- a/src/main/java/egovframework/ezEKP/ezNewPortal/service/EzNewPortalService.java

+++ b/src/main/java/egovframework/ezEKP/ezNewPortal/service/EzNewPortalService.java

@@ -210,7 +210,7 @@ public interface EzNewPortalService {

public List<OrganJobVO> getTitleList(String type, int tenantId, String companyId) throws Exception;

public List<OrganGroupVO> getGroupList(int tenantId, String companyId) throws Exception;

public List<FileVO> getWebFolderFileList(String folderId, int tenantId, String offset) throws

Exception;

- public int getWebFolderFileTodayCnt(String folderId, int tenantId, String offset) throws Exception;

+ public int getWebFolderFileTodayCnt(String folderId, int tenantId) throws Exception;

public void addPortalTenantConfig(int tenantId, String propertyName, String propertyValue, String

description, String configName, String configType) throws Exception;

범위

* 범위 : .. 와 ... 의 차이

<https://git-scm.com/docs/git-diff>

\$ git diff A..B

(= git diff A B)

- A 와 B 를 비교

\$ git diff A...B

(= git diff \$(git merge-base A B) B)

- (A 와 B의 공통 조상)과 B를 비교

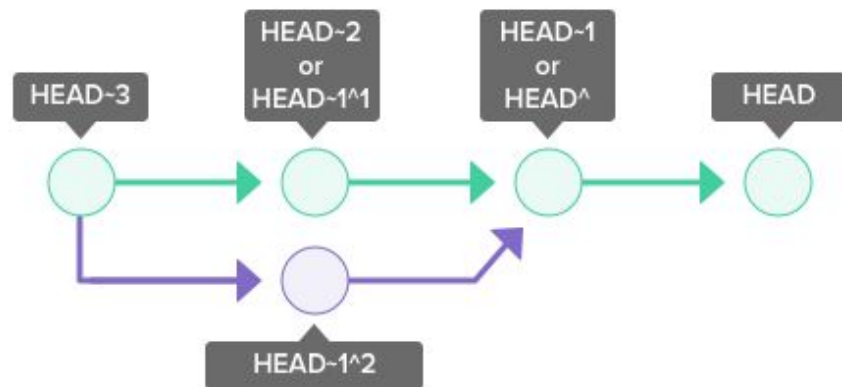
* [^ , ~ , @] 차이

~2 : 2단계위의 조상

^2 : 2번째 부모

@{2} : 2번째 이전 커밋 , reflog로 확인 가능

<https://backlog.com/git-tutorial/using-branches/git-switch-branches/>



\$ git remote (add / -v / show / rm,remove)

* 추가 \$ git remote add Dev jmochoa@10.0.102.8:git/ezFlow4Java.git |

* 조회 \$ git remote -v
Dev jmochoa@10.0.102.8:git/ezFlow4Java.git (fetch)
Dev jmochoa@10.0.102.8:git/ezFlow4Java.git (push)
origin jmochoa@10.0.102.8:sitegit/buhmwoo/ezFlow4Java.git (fetch)
origin jmochoa@10.0.102.8:sitegit/buhmwoo/ezFlow4Java.git (push)

\$ git remote show
Dev
origin

\$ git remote show Dev
* remote Dev
Fetch URL: jmochoa@10.0.102.8:git/ezFlow4Java.git
Push URL: jmochoa@10.0.102.8:git/ezFlow4Java.git
HEAD branch: master
Remote branches:
AddjobProxy tracked
ApprGuide tracked
ESmemodev tracked
JavaPMS tracked
NeoApprovalG tracked

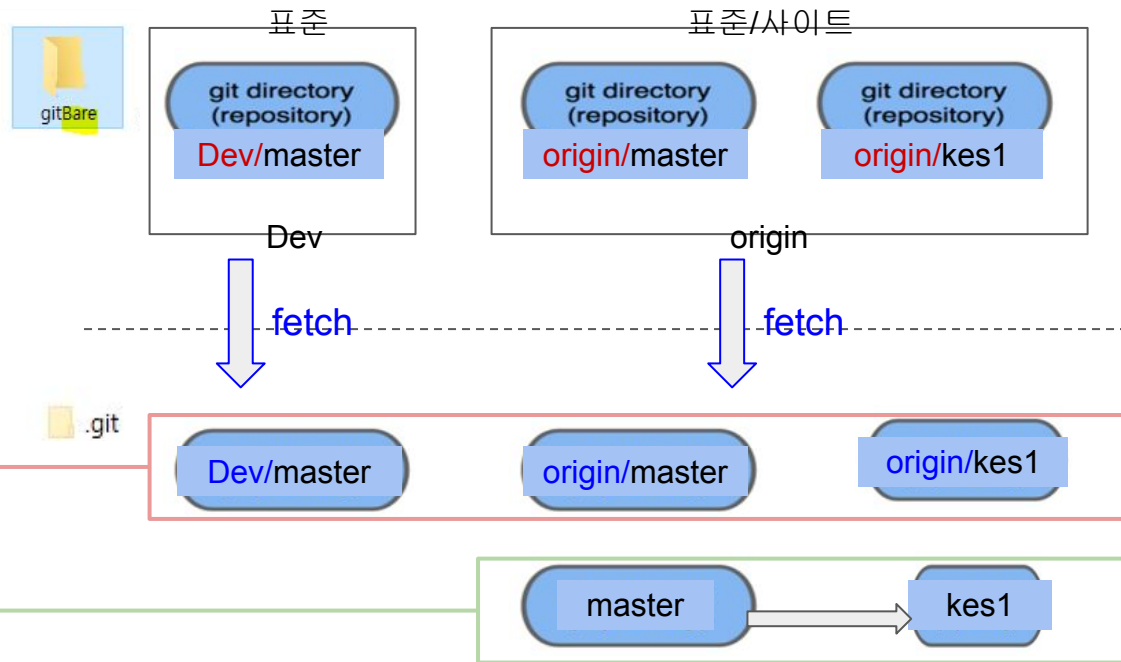
* 삭제 \$ git remote remove Dev |

\$ git branch (-r)

```
$ git branch -r
  Dev/webfolderJP
  Dev/webfolderOracle
  Dev/webfolderSorting
  origin/AddjobProxy
  origin/ApprGuide
  origin/ESmemodev
  origin/HEAD -> origin/master
  origin/NeoApprovalG
```

* 현재

```
$ git branch
* jrel
kes_originCompAndHeadIdManage
master
site_buhmwoo
site_buhmwoo_dev
```



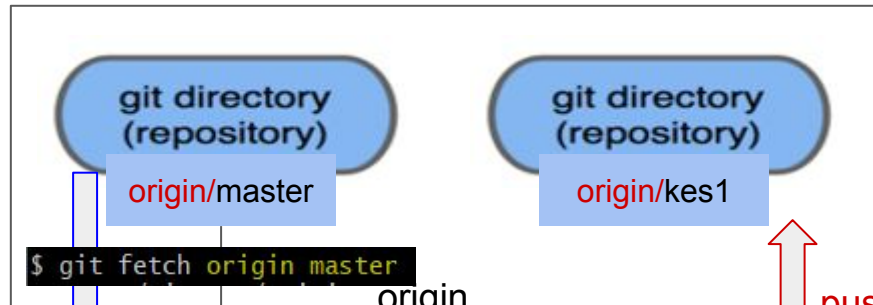
삭제: \$ git branch -d / -D (강제)

브랜치 및 원격 추가

\$ git fetch
\$ git pull
\$ git checkout (-b)
\$ git branch -u
(--set-upstream-to=)

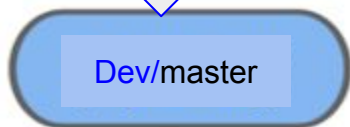
표준

표준/사이트



```
$ git fetch Dev
```

fetch



merge

```
$ git merge Dev/master  
Already up to date
```

pull



upstream

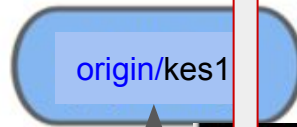
```
$ git fetch origin master
```

origin

fetch



upstream



push

```
$ git push origin kes1
```



```
$ git branch -u origin/kes1
```

```
$ git branch kes1
```

```
$ git checkout kes1
```

```
= $ git checkout -b kes2
```


\$ git push origin ~

생성: \$ git push origin new

푸시: \$ git push origin new:new

삭제: \$ git push origin :new

git-push(1) Manual Page

NAME

git-push - Update remote refs along with associated objects

SYNOPSIS

```
git push [--all | --mirror | --tags] [--follow-tags] [--atomic] [-n | --dry-run] [--receive-pack=<git-receive-pack>]
[--repo=<repository>] [-f | --force] [-d | --delete] [--prune] [-v | --verbose]
[-U | --set-upstream] [-o <string> | --push-option=<string>]
f: foo "cloned" | cloned=true/false if asked
```

`git push origin HEAD:master`

Push the current branch to the remote ref matching `master` in the `origin` repository. This form is convenient for pushing the current branch without thinking about its local name.

`git push origin master:refs/heads/experimental`

Create the branch `experimental` in the `origin` repository by copying the current `master` branch. This form is needed to create a new branch or tag in the remote repository when the local name and the remote name are different; otherwise, the ref name on its own will work.

`git push origin :experimental`

Find a ref that matches `experimental` in the `origin` repository (e.g. `refs/heads/experimental`), and delete it.

`git push origin +dev:master`

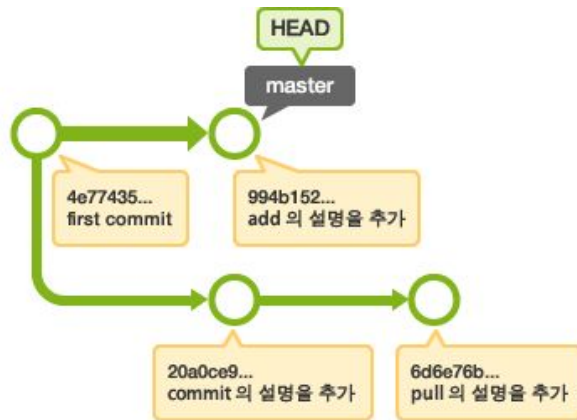
Update the origin repository's master branch with the dev branch, allowing non-fast-forward updates. **This can create unreferenced commits dangling in the origin repository.** Consider the following situation, where a fast-forward update is not possible:

```
o---o---o---A---B  origin/master
      \
      X---Y---Z  dev
```

The above command would change the origin repository to

```
A---B  (unnamed branch)
      /
```

\$ git merge (--squash)



--squash: 브랜치 상의 모든 커밋을 하나로 병합하여 merge.

squash 미국식 [skwa:j, skwo:j] < >

1. 동사 짓누르다, 으깨다, 찌부러뜨리다
2. 동사 (좁은 곳에) 밀어 넣다[쑤셔 박다].
3. 명사 스퀘시(사면이 벽으로 된 코트에서



```
$ git merge --squash kes1
Updating 138d5fa..5e7f904
Fast-forward
Squash commit -- not updating HEAD
eunsil | 2 ++
1 file changed, 2 insertions(+)

eunsil@DESKTOP-FVNK3TN MINGW64 /d/GitTemp/user1/
userPrac (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    modified:   eunsil
```

```
$ git log -3
commit 5e7f90483a0a5162fc51c2349
(HEAD -> kes1, origin/kes1)
Author: Eunsil <hosea0301@kaoni>
Date: Tue Sep 7 10:47:04 2021
```

11th

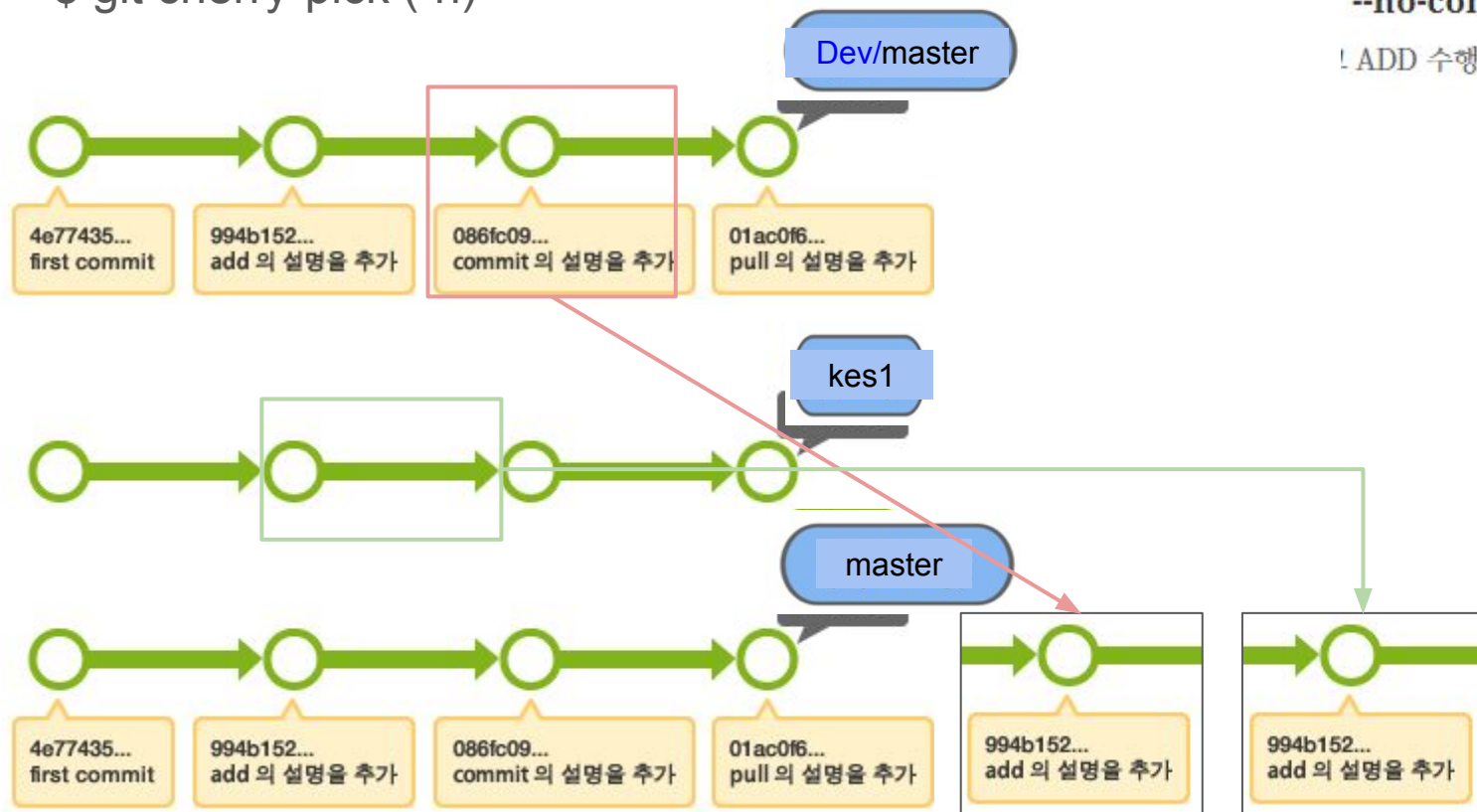
```
commit a8da04247f457f4fe84b91e69
Author: Eunsil <hosea0301@kaoni>
Date: Tue Sep 7 10:46:36 2021
```

10th

```
commit 138d5fab80e812d28c3e336e0
(origin/master, master)
Author: Eunsil <hosea0301@kaoni>
Date: Mon Sep 6 22:28:28 2021
```

ninth

\$ git cherry-pick (-n)



-n

--no-commit

! ADD 수행하고 Commit은 안함.

충돌 해결하기 CONFLICT

MERGING

```
$ git merge master
Auto-merging eunsil
CONFLICT (content): Merge conflict in
Automatic merge failed; fix conflicts
result.

eunsil@DESKTOP-FVNK3TN MINGW64 /d/GitT
(kes2) MERGING
$ git status
On branch kes2
Your branch is up to date with 'origin/
kes2'.

You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the
  merge)

Unmerged paths:
  (use "git add <file>..." to mark res
  olution)
    both modified:   eunsil
```

```
$ git merge --abort
```

CHERRY-PICKING

```
$ git cherry-pick 5e7f90483a0a5162fc51c23492487e2980b24
d
Auto-merging eunsil
CONFLICT (content): Merge conflict in eunsil
error: could not apply 5e7f904... 11th
hint: after resolving the conflicts, mark the corrected
paths
hint: with 'git add <paths>' or 'git rm <paths>'
hint: and commit the result with 'git commit'

eunsil@DESKTOP-FVNK3TN MINGW64 /d/GitTemp/user2/userPra
(kes2) CHERRY-PICKING
$ git status
On branch kes2
Your branch is up to date with 'origin/kes2'.

You are currently cherry-picking commit 5e7f904.
  (fix conflicts and run "git cherry-pick --continue")
  (use "git cherry-pick --skip" to skip this patch)
  (use "git cherry-pick --abort" to cancel the cherry-p
  k operation)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
    both modified:   eunsil
```

```
$ git cherry-pick --abort
```

```
$ git diff
diff --cc eunsil
index 791e1b6,0a06c65..0000000
--- a/eunsil
+++ b/eunsil
@@@ -1,4 -1,6 +1,10 @@@
    hi
    i am eunsil
    be happy!
++<<<<<< HEAD
+its over.
++=====
+ its not the end.
+ again
+ twice
++>>>>>> 5e7f904... 11th
```

```
$ git add .
eunsil@DESKTOP-FVNK3TN MINGW64
$ git cherry-pick --continue
```

```
$ git add .
eunsil@DESKTOP-FVNK3TN MINGW64
$ git commit
[kes2 983e753] Merge branch
```

책임 확인하기 blame

\$ git blame (-L) <파일>

```
$ git blame eunsil
ce560f77 (Eunsil      2021-09-06 18:41:50 +0900 1) hi
22c26458 (Eunsil      2021-09-06 19:01:46 +0900 2) i am eunsil
fcbc60cb (Eunsil      2021-09-06 21:40:00 +0900 3) be happy!
00000000 (Not Committed Yet 2021-09-07 11:19:34 +0900 4) <<<<<< HEAD
f532bce1 (Eunsil      2021-09-06 22:30:51 +0900 5) its over.
00000000 (Not Committed Yet 2021-09-07 11:19:34 +0900 6) =====
138d5fab (Eunsil      2021-09-06 22:28:28 +0900 7) its not the end.
a64e1490 (Eunsil      2021-09-07 10:51:42 +0900 8) again
a64e1490 (Eunsil      2021-09-07 10:51:42 +0900 9) twice
00000000 (Not Committed Yet 2021-09-07 11:19:34 +0900 10) >>>>>> master
```

`-L <n,m>` Process only line range n,m, counting from 1

```
$ git blame -L 4,10 eunsil
```

-L : line

```
4) <<<<<< HEAD
5) its over.
6) =====
7) its not the end.
8) again
9) twice
10) >>>>>> master
```


되돌리기

\$ git reset (--mixed(default) / --soft / --hard) <커밋위치>

```
$ git reset HEAD~3
Unstaged changes after reset:
M   $ git reset --mixed HEAD~3
Unstaged changes after reset:
M   eunsil

eunsil@DESKTOP-FVVK3TN MINGW64
$ git status
On branch kes2
Your branch is behind 'origin/kes2' (use "git pull" to update your local branch)

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   eunsil

Untracked files:
  (use "git add <file>..." to track)
        eunsil3
```

```
$ git reset --soft HEAD~3

eunsil@DESKTOP-FVVK3TN MINGW64
$ git status
On branch kes2
Your branch is behind 'origin/kes2' (use "git pull" to update your local branch)

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   eunsil
        new file:   eunsil3
```

```
$ git reset --hard HEAD~3
HEAD is now at 983e753 Merge
eunsil@DESKTOP-FVVK3TN MINGW64
$ git status
On branch kes2
Your branch is behind 'origin/kes2' (use "git pull" to update your local branch)

nothing to commit, working tree clean
```


\$ git revert (-n)

```
$ git revert HEAD
[kes2 05efd93] Revert "14th"
1 file changed, 1 deletion(-)
```

```
eunsil@DESKTOP-FVNK3TN MINGW64 /d/
$ git log -3
commit 05efd93e528dbdb1b105d79baf
Author: Eunsil <hosea0301@kaoni.co.kr>
Date: Tue Sep 7 13:28:58 2021 +0900
```

Revert "14th"

This reverts commit a48a813d7451e25149b541128a

```
commit a48a813d7451e25149b541128a
Author: Eunsil <hosea0301@kaoni.co.kr>
Date: Tue Sep 7 12:12:38 2021 +0900
```

14th

```
@@ -2,4 +2,3 @@ hi
i am eunsil
be happy!
the eunsil3 file
-hi
```

```
@@ -2,3 +2,4 @@ hi
i am eunsil
be happy!
the eunsil3 file
+hi
```

-n

--no-commit

(kes2|REVERTING)

```
$ git revert -n HEAD
```

```
eunsil@DESKTOP-FVNK3TN MINGW64
$ git status
On branch kes2
Your branch is up to date with
```

```
You are currently reverting
(all conflicts fixed: run
(use "git revert --skip" to
(use "git revert --abort" to
```

```
Changes to be committed:
(use "git restore --staged" to
modified: eunsil
```

```
@@ -2,4 +2,3 @@ hi
i am eunsil
be happy!
the eunsil3 file
-hi
```

\$ git rebase - 순서 변경

```
$ git rebase -i HEAD~3
```

```
pick 35d7d7e revert 2 :add 2  
pick 8ff029f revert 2 :add 4  
pick 6f80d81 revert 2 :add 3
```

```
Auto-merging eunsil  
CONFLICT (content): Me
```

```
2  
++<<<<<< HEAD  
++=====
```

2

```
+ 3  
+ 4  
++>>>>>> 8ff029f... re
```

\$ git rebase - 병합 (squash)

// 메세지 edit

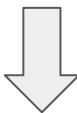
```
$ git log --pretty="%h %s"
8ff029f revert 2 :add 4
6f80d81 revert 2 :add 3
35d7d7e revert 2 :add 2
ab8b1bb revert 2 :add 1
b2d5950 revert 1
a48a813 14th
660f30a 13th
```

```
$ git rebase -i b2d5950
```

```
pick ab8b1bb revert 2 :add 1
pick 35d7d7e revert 2 :add 2
squash 6f80d81 revert 2 :add 3
pick 8ff029f revert 2 :add 4

# Rebase b2d5950..8ff029f onto 8ff029f
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, edit the commit message
# e, edit <commit> = use commit, edit the file(s)
# s, squash <commit> = use commit, combine the changes into a single commit
# f, fixup <commit> = like "squash" but only use the new changes
# x, exec <command> = run command(s), update commit message, and continue
```

```
# This is a combination of
# This is the 1st commit
revert 2 :add 2
# This is the commit message
revert 2 :add 3
# Please enter the commit message for
# with '#' will be ignored, or
#
```



```
# This is a combination of
# This is the 1st commit
revert 2 :add 2, 3
# Please enter the commit message for
# with '#' will be ignored, or
#
```

```
$ git log --pretty="%h %s"
999b2d8 revert 2 :add 4
3d98310 revert 2 :add 2, 3
ab8b1bb revert 2 :add 1
b2d5950 revert 1
a48a813 14th
660f30a 13th
```

\$ git rebase - 분리 (edit)

```
pick ab8b1bb revert 2 :add 1
edit 751ef88 revert 2 :add 2,3
pick 1faf6a/ revert 2 :add 4
```

```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ git log -1
commit 69e684dbed30b0ae05b7cb999df665269d3a6bc1 (HEAD)
Author: Eunsil <hosea0301@kaoni.com>
Date: Tue Sep 7 13:39:22 2021 +0900

    revert 2 :add 2,3
```

```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ vim eunsil
```

```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ git commit -a -m "revert takeOff 2"
[detached HEAD 845c0b6] revert takeOff 2
1 file changed, 1 insertion(+)
```

```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ vim eunsil
```

```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ git commit -a -m "revert takeOff 3"
[detached HEAD 8722cf3] revert takeOff 3
1 file changed, 1 insertion(+)
```

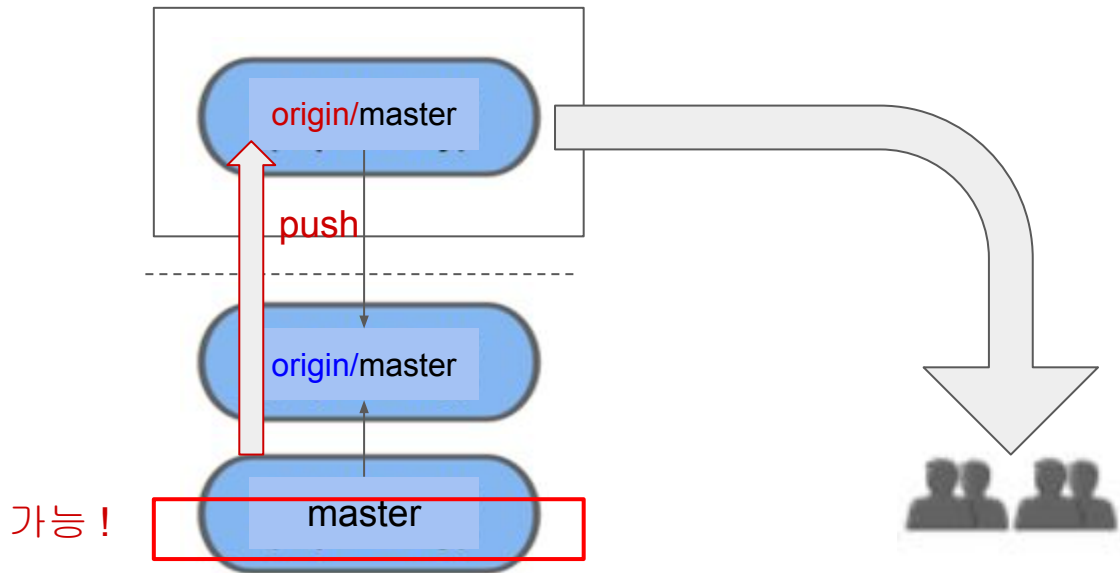
```
eunsil@DESKTOP-FVVK3TN MINGW64 /d/GitTemp/user2/userPrac
(kes2|REBASE 2/3)
$ git rebase --continue
Successfully rebased and updated refs/heads/kes2.
```

```
$ git log --pretty="%h %s" -7
90630a5 revert 2 :add 4
8722cf3 revert takeOff 3
845c0b6 revert takeOff 2
ab8b1bb revert 2 :add 1
b2d5950 revert 1
a48a813 14th
660f30a 13th
```

이력 고쳐쓰기의 위험성 : 공유

\$ git reset / revert / rebase (/ commit --amend)

: 이력을 고치려면 **pushing**을 하기 전에 !




```
$ git commit --amend
[kes2 458fb61] revert 2 :add 4 _ push 이후 amend
Date: Tue Sep 7 13:39:48 2021 +0900
1 file changed, 1 insertion(+)
```

// 바로 push 할 수 없음.

```
$ git push
To D:/GitTemp/originPrac/gitBare
! [rejected]        kes2 -> kes2 (r
error: failed to push some refs to '
hint: Updates were rejected because
hint: its remote counterpart. Integr
hint: 'git pull ...') before pushing
hint: See the 'Note about fast-forwa
```

// git pull 후 → push 가능.

```
$ git status
On branch kes2
Your branch and 'origin/kes2' have diverged,
and have 1 and 1 different commits each, respective
(use "git pull" to merge the remote branch into y

nothing to commit, working tree clean
```

```
$ git pull
```

// log: 다른 부분부터 다시 **commit & merge** 됨.

// commit --amend

```
$ git log --pretty="%h %s" -5
a684f7a Merge branch 'kes2' of D:/GitTemp/
458fb61 revert 2 :add 4 _ push 이후 amend
8ff029f revert 2 :add 4
6f80d81 revert 2 :add 3
35d7d7e revert 2 :add 2
```

// rebase-병합

```
$ git log --pretty="%h %s"
fa906bf Merge branch 'kes2' of
999b2d8 revert 2 :add 4
3d98310 revert 2 :add 2, 3
8ff029f revert 2 :add 4
6f80d81 revert 2 :add 3
35d7d7e revert 2 :add 2
ab8b1bb revert 2 :add 1
b2d5950 revert 1
a48a813 14th
660f20a 12th
```

```
'origin/kes2' by 3 commits.
(To publish your local commits)
```


태그 만들기 읽기전용/릴리즈

보통 '릴리스 브랜치(Release branch)'에서는 주석 태그를 사용하여 설명이나 서명을 넣은 보다 상세한 정보를 포함하는 태그를 사용하고, 로컬에서 일시적으로 사용하는 '토픽 브랜치(Topic branch)'에서는 이름만 만들어 붙이는 태그를 사용합니다.

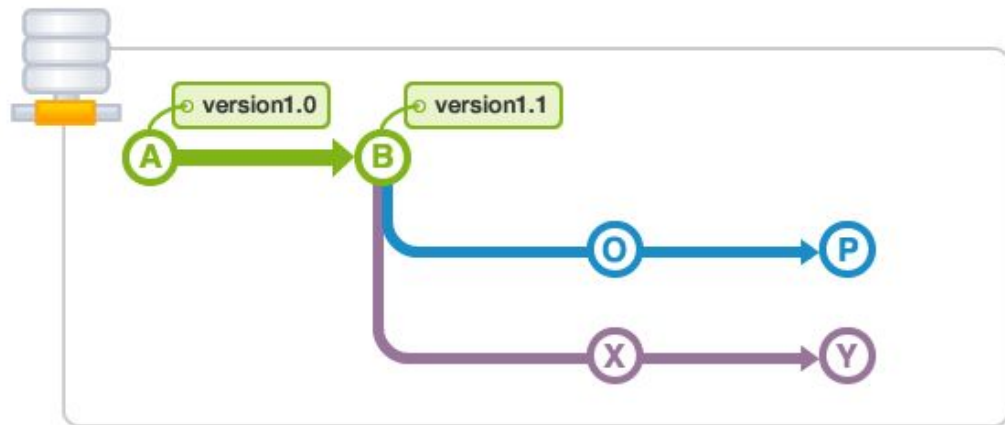
\$ git tag

- 일반 태그(Lightweight tag)

- 이름만 붙일 수 있어요

- 주석 태그(Annotated tag)

- 이름을 붙일 수 있어요
- 태그에 대한 설명도 포함할 수 있어요
- 서명도 넣을 수 있어요
- 이 태그를 만든 사람의 이름, 이메일과 태그를 만든 날짜 정보도 포함시킬 수 있어요



```
userPrac (res2)  
$ git checkout 0.1  
Note: switching to '0.1'.
```

```
eunsil@DESKTOP-FVVK3  
userPrac ((0.1))  
$
```

\$ git tag (일반 Lightweight)

```
$ git tag  
0.1  
0.5  
1.0
```

```
$ git tag 1.0
```

```
commit a48a813d7451e25149b541128aa4da6ff9d1cde8  
(HEAD -> kes2, tag: 1.0, origin/kes2)  
Author: Eunsil <hosea0301@kaoni.com>  
Date: Tue Sep 7 12:12:38 2021 +0900
```

14th

```
$ git tag 0.5 kes3
```

```
commit 960c8af69148b7d30797150b7f3c7a860dff0643  
(tag: 0.5, kes3)  
Author: Eunsil <hosea0301@kaoni.com>  
Date: Tue Sep 7 11:54:42 2021 +0900
```

12th

```
$ git tag 0.1 138d5fab80e
```

```
$ git branch kes4 0.1
```

```
commit 138d5fab80e81  
(tag: 0.1)  
Author: Eunsil <hosea0301@kaoni.com>  
Date: Mon Sep 6 22:11:11 2021 +0900
```

ninth

```
commit 138d5fab80e81  
(tag: 0.1, kes4)  
Author: Eunsil <hosea0301@kaoni.com>  
Date: Mon Sep 6 22:11:11 2021 +0900
```

ninth

* / 를 이용해서 폴더처럼 조직화 할수도 있다.

\$ git tag (주석Annotated)

Annotated 태그

Annotated 태그를 만드는 방법은 간단하다. tag 명령을 실행할 때 `-a` 옵션을 추가한다.

```
$ git tag -a v1.4 -m "my version 1.4"
$ git tag
v0.1
v1.3
v1.4
```

`-m` 옵션으로 태그를 저장할 때 메시지를 함께 저장할 수 있다. 명령을 실행할 때 메시지를 입력하지 않으면 Git은 `편집기`를 실행시킨다.

`git show` 명령으로 태그 정보와 커밋 정보를 모두 확인할 수 있다.

```
$ git show v1.4
tag v1.4
Tagger: Ben Straub <ben@straub.cc>
Date: Sat May 3 20:19:12 2014 -0700

my version 1.4

commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date: Mon Mar 17 21:52:11 2008 -0700

    changed the version number
```

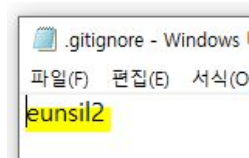
커밋 정보를 보여주기 전에 먼저 태그를 `만든 사람`이 누구인지, `언제` 태그를 만들었는지, 그리고 태그 `메시지`가 무엇인지 보여준다.

ezEKP_자바_개발자_가이드.docs

8.4. 사이트 오픈 후 처리

사이트에 업데이트를 적용할 때는 빌드한 소스(사이트 저장소의 사이트 브랜치)의 commit에 tagging을 하고 해당 tag를 팀 내부 레드마인의 '고객 사이트 정보' 게시판의 해당 사이트 게시물에 기록한다. tag의 형식은 '사이트아이디_업데이트날짜'와 같이 한다.(예) ryobi-20190212). 또한 버전 번호의 빌드날짜 부분을 변경하고 해당 사이트 게시물에서 별도의 버전관리를 수행한다.('서울시 마을공동체 종합지원센터 운영서버 정보' 게시물의 '버전번호 관리' 섹션 참조).

추적 무시하기 ignore



.gitignore가 제대로 작동되지 않아서 ignore처리된 파일이 자꾸 changes에 나올때가 있습니다.

git의 캐시가 문제가 되는거라 아래 명령어로 캐시 내용을 전부 삭제후 다시 add All해서 커밋하시면 됩니다.

```
git rm -r --cached .  
git add .  
git commit -m "fixed untracked files"
```


감사합니다.

참고 .git 제외하고 tar 나 zip 으로 묶기

\$ git archive

* 지원하는 포맷 list

```
$ git archive -l  
  
tar  
tgz  
tar.gz  
zip
```

아래 명령은 마스터 브랜치를 zip 형식으로 lesstif.zip 파일로 아카이빙 합니다.

CODE

```
1 | $ git archive --format=zip master -o ../lesstif.zip
```

아래는 v1.0 태그를 tar.gz 으로 아카이빙하며 접두사로 v1.0-를 붙이는 예제입니다.

CODE

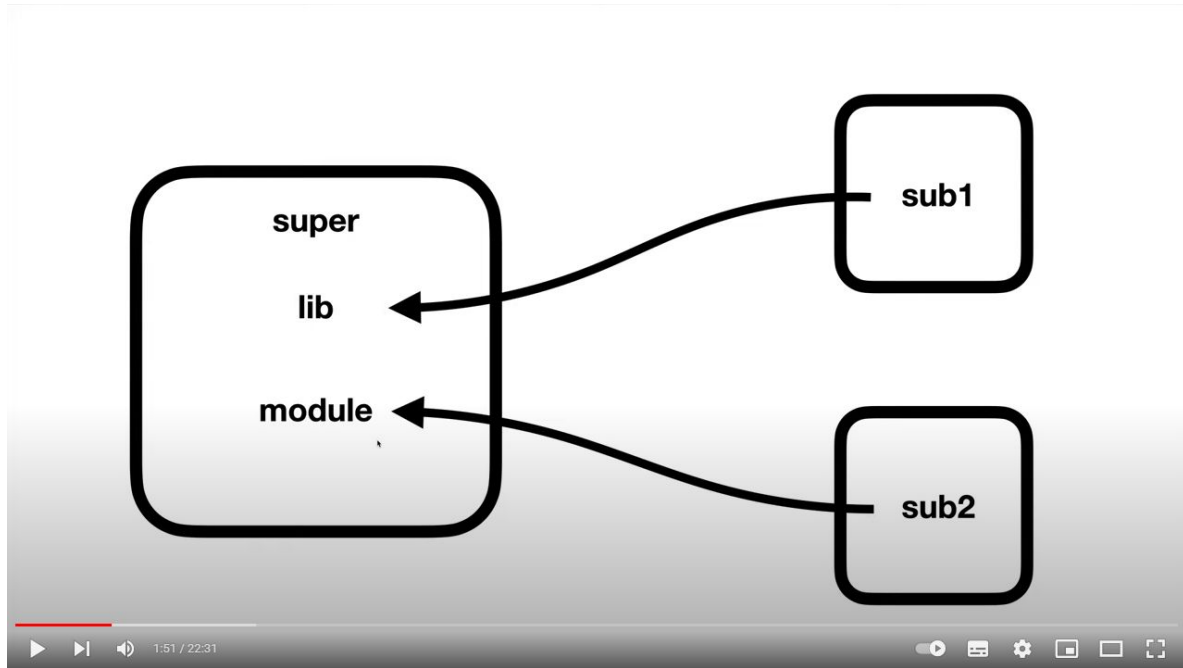
```
1 | $ git archive --format=tar.gz --prefix=v1.0- v1.0 -o lesstif-v1.0.tar.gz
```

<https://www.lesstif.com/gitbook/git-archive-git-tar-zip-54952880.html>

참고 서브모듈

<https://youtu.be/TAe4uZqYt6c>

\$ git submodule



참고 patch 파일

<https://codechacha.com/ko/git-patch/>

Git - Patch 만들기 & 적용하기

By JS | Last updated: October 07, 2019

GIT TIPS

오픈소스 프로젝트에 참여하면 git을 사용하게 되고, patch 파일을 접하게 됩니다. 만약 다른 사람으로부터 patch를 전달받았다면 나의 로컬 소스에 적용할 수 있습니다.

이 글에서는 리눅스 개발환경에서 Git 패치를 만드는 방법과 적용하는 방법을 알아보겠습니다.

git commit 패치 만들기 & 적용

Git을 사용할 때, 다음 명령어를 사용하면 현재 commit부터 입력한 commit 이전까지 패치를 만들어줍니다.

```
$ git format-patch [commit-id]
```

실제로 패치를 만들어보면 commit 당 1개의 패치 파일을 만들어줍니다.

```
$ git log
commit 2eadda0d1931632a4dcdcd2bce170f55dc9928a2 (HEAD -> master)
...
commit 53192f622e94f814f56352248fb33fc7bd017184
```

참고 git 추천 사이트

<https://backlog.com/git-tutorial/kr/contents/>



누구나 쉽게 이해할 수 있는 Git 입문

버전 관리를 완벽하게 이용해보자

트윗

좋아요 421개

목차

입문 편

발전 편

찾아보기

목차

입문편

Git의 기본

시작하기

이력을 관리하는 저장소

변경을 기록하는 커밋

작업 트리(Work tree)와 인덱스(Index)

튜토리얼1 Git의 기본

Git 설치

초기 설정

새 저장소 만들기

파일 커밋(Commit)하기

발전편

브랜치 (Branch)

브랜치란?

브랜치 만들기

브랜치 전환하기

브랜치 통합하기

토픽 브랜치와 통합 브랜치에서의 작업 흐름 파악하기

튜토리얼1 브랜치를 사용해 보자

0. 사전 준비

1. 브랜치 만들기

찾아보기

기본 조작

저장소를 만들고 싶다

파일과 폴더를 인덱스에 등록하고 싶다

인덱스에 추가된 파일을 커밋하고 싶다

변경된 파일의 목록을 확인하고 싶다

변경된 파일의 내용을 확인하고 싶다

커밋 로그를 보고 싶다

커밋의 상세내용을 확인하고 싶다

파일과 폴더명을 변경하거나 이동시키고 싶다

파일을 삭제하고 싶다