

# Git 기초교육

정체와 다루는 법에 대해

# 개요 - git 이란?

리누스 토르발스가 개발한 분산형 버전 관리 시스템

- 완성까지 2주 걸렸다고 함
- 참고로 이분이 리눅스도 만들었음
- 답내한(답답하니 내가 한다)의 표본

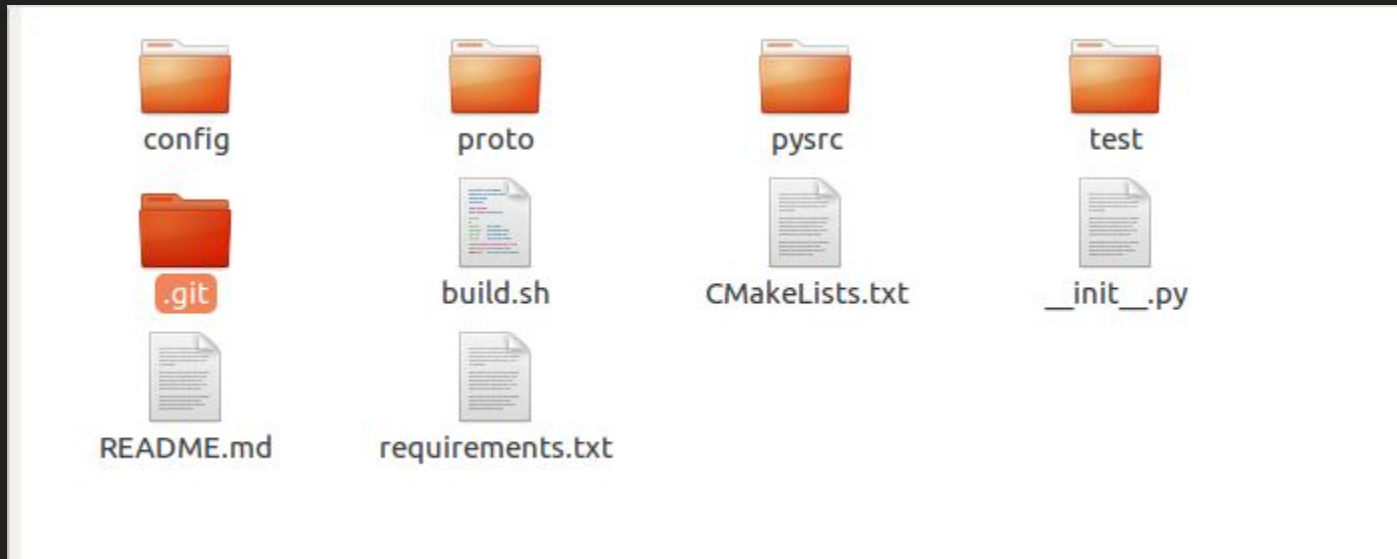


# 개요 - git 이란?

- SVN과 CVS(이전까지의 버전 관리 시스템들)을 혐오했던 리누스 토르발스
- 리누스 토르발스는 초기에 리눅스 개발시 버전 관리 시스템을 사용하지 않고 메일링 리스트로 기여 내용을 직접 관리(!!!)
- 분산 처리 기능과 비교적 빠른 성능을 위해 BitKeeper라는 소프트웨어를 사용하기 시작하나.....
- SourcePuller라는 BitKeeper와 유사한 소프트웨어가 나오자 BitKeeper측에서 개발자를 제한하기 시작함(Reverse Engineering을 의심함)
- 리누스가 필요로 하는 기능들을 포함한 버전 관리 시스템이 없었다!
  - CVS를 반면교사로 삼고
  - BitKeeper와 같은 분산된 작업 흐름을 유지하며
  - 데이터 무결성을 지키기 위한 체계를 잘 갖추고 있어야 한다
- 그래서 리누스는 (리눅스를 개발하던 와중에) git이라는 VCS를 만들었다!

발번역입니다.

# git의 정체



.git 폴더 안에 git이 기능하기 위한 것들이 다 들어있다.

.git이 git의 전부?

그럼 github은 뭔가요???

Bitbucket 은 또 무엇이고???

# git과 github의 관계

git의 기본 동작은 온라인에서 벌어지는게 아니다.

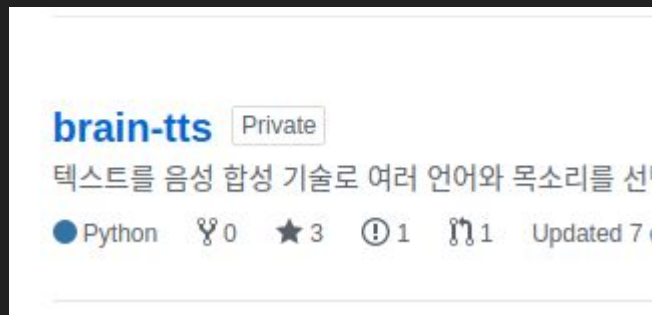
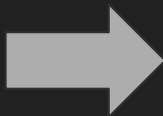
- git은 기본적으로 소프트웨어 이며, 자신의 폴더 안에 있는 파일들을 관리하는 것이 목적이다.
- 개발자가 코드를 관리하기 편히 하기 위해 쓰는것도 있지만 실제로는 어떤 파일이든 다 관리할 수 있다.
- 소프트웨어이기 때문에 당연히 오프라인에서도 작업이 가능하다.

github은 다양한 사용자들의 git을 사용한 작업들을 한곳에 저장 및 공유하기 위한 플랫폼이라 이해하는 것이 맞다.

# git 저장소 - git repository



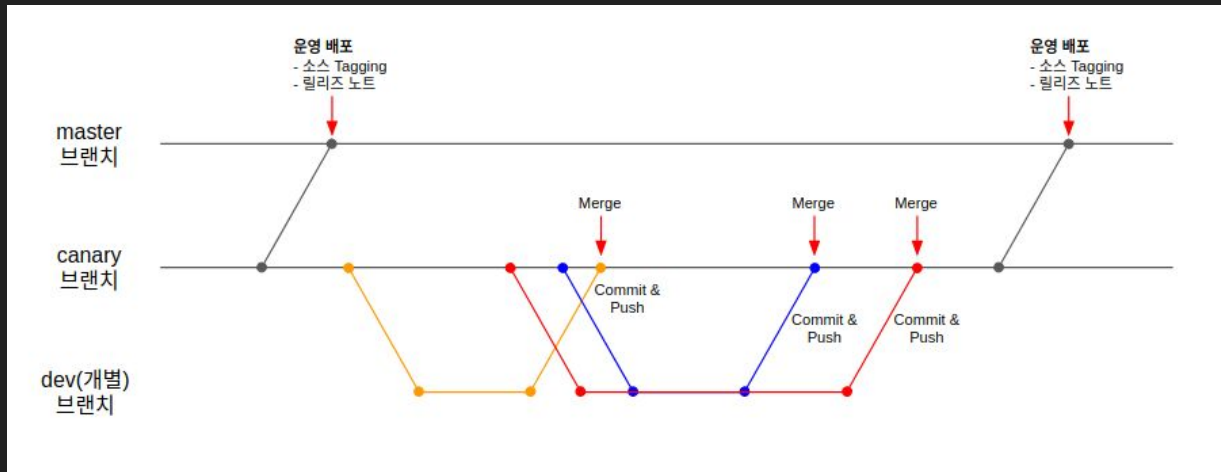
사용자의 local git repository



github 상의 git repository

사용자가 자신의 작업공간에 git을 사용하여 작업을 하고, 그것을 원격 (remote) 근원(origin)에 보내어 여러 작업자와 작업을 공유하여 작업을 진행하는 것이 핵심!

# git repository의 관리



Local의 저장소의 경우 어떤 형식으로 사용하든 큰 문제가 발생하지 않는다.

다만 원격 저장소의 필요한 부분을 마구잡이로 건드리면 문제가 발생할 수 있어서 원격 저장소에 다양한 설정을 주기도 한다.



# git의 시작은 어디에서?

자신이 작업한 프로젝트 디렉토리에서 `git init`을 하면 새 저장소가 해당 디렉토리에 생성된다.

```
→ mindslab-git-test ls -la
total 8
drwxrwxr-x  2 aquashdw aquashdw 4096  5월  4 11:41 .
drwxrwxr-x 70 aquashdw aquashdw 4096  5월  4 11:41 ..
→ mindslab-git-test git init
Initialized empty Git repository in /home/aquashdw/git/mindslab-git-test/.git/
→ mindslab-git-test git:(master) ls -la
total 12
drwxrwxr-x  3 aquashdw aquashdw 4096  5월  4 11:41 .
drwxrwxr-x 70 aquashdw aquashdw 4096  5월  4 11:41 ..
drwxrwxr-x  7 aquashdw aquashdw 4096  5월  4 11:41 .git
→ mindslab-git-test git:(master) █
```

# git의 시작은 어디에서?

이후 원격에 생성된 저장소로 자신의 폴더에서의 작업사항을 넣을 수 있다.

## ...or create a new repository on the command line

```
echo "# mindslab-git-test" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:aquashdw0/mindslab-git-test.git
git push -u origin master
```



## ...or push an existing repository from the command line

```
git remote add origin git@github.com:aquashdw0/mindslab-git-test.git
git push -u origin master
```



# git의 시작은 어디에서?

이후 원격에 생성된 저장소로 자신의 폴더에서의 작업사항을 넣을 수 있다.

```
→ mindslab-git-test git:(master) git remote add origin git@github.com:aquashdw0/mindslab-git-test.git
→ mindslab-git-test git:(master) git push -u origin master
Counting objects: 3, done.
Delta compression using up to 16 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:aquashdw0/mindslab-git-test.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
→ mindslab-git-test git:(master) █
```

# git의 시작은 어디에서?


원격 저장소(관리 플랫폼)에서 새 저장소를 만들고, 해당하는 저장소를 복제(clone)해서 작업을 시작할 수도 있다.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?  
[Import a repository.](#)

---

**Owner**

 aquashdw0 ▾

/

**Repository name \***

mindslab-get-test ✓

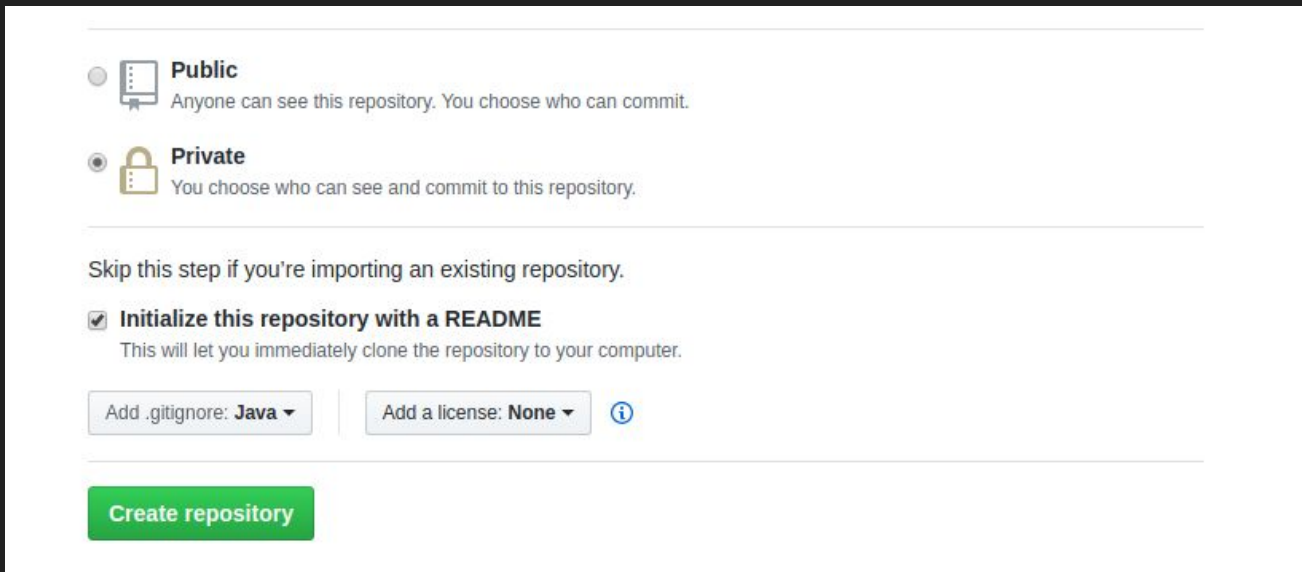
Great repository names are short and memorable. Need inspiration? How about **fuzzy-dollop**?

**Description (optional)**

git repository 원격 생성 예시

# git의 시작은 어디에서?

원격 저장소(관리 플랫폼)에서 새 저장소를 만들고, 해당하는 저장소를 복제(clone)해서 작업을 시작할 수도 있다.



The screenshot shows the GitHub 'Create repository' form. It has two radio buttons for visibility: 'Public' (selected) and 'Private'. Below these is a note to skip the step if importing an existing repository. There is a checked checkbox for 'Initialize this repository with a README'. At the bottom, there are dropdown menus for adding a .gitignore file (set to 'Java') and a license (set to 'None'), followed by a green 'Create repository' button.

☐ **Public**  
Anyone can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**  
This will let you immediately clone the repository to your computer.

Add .gitignore: **Java** ▼

Add a license: **None** ▼ ⓘ

**Create repository**

# git을 사용하기 전에..... - fetch, pull

설명을 하기 전에 먼저 보여줄 명령들

**fetch:** 원격 저장소의 데이터를 **local** 저장소로 가져온다.

**pull:** 원격 저장소의 데이터를 **local** 저장소로 가져오고, 그 데이터를 현재 프로젝트에 적용한다.

```
→ mindslab-git-test git:(master) git fetch
→ mindslab-git-test git:(master) git pull
Updating e54aa84..aa4dcc9
Fast-forward
 a | 1 +
 b | 1 +
 2 files changed, 2 insertions(+)
 create mode 100644 b
```

# git을 사용해 보자! - init, clone

새로운 프로젝트를 생성할 때

- 위의 슬라이드들을 참조하세요

원래 있던 저장소를 가져오거나, 또는 원격에 우선적으로 저장소를 생성할 때

```
→ git git clone git@github.com:aquashdw0/mindslab-git-test.git
Cloning into 'mindslab-git-test'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
Checking connectivity... done.
→ git cd mindslab-git-test
→ mindslab-git-test git:(master) █
```

init, clone

# git을 사용해보자! - init, clone

`git init`, `git init <directory\_name>` : 현재 또는 <directory\_name> 디렉토리에 새 git 저장소를 생성한다.

```
→ git mkdir mindslab-git-test
→ git git init mindslab-git-test
Initialized empty Git repository in /home/aquashdw/git/mindslab-git-test/.git/
→ git █
```

`git clone <remote\_repository\_url>` : 이미 원격에 존재하는 git 저장소를 자신의 현재 디렉토리의 <repository\_name> 디렉토리를 생성하고 동일한 git 저장소를 생성한다.

```
→ git git clone git@github.com:aquashdw0/mindslab-git-test.git
Cloning into 'mindslab-git-test'...
→ mindslab-git-test git:(master) ll
total 16
drwxrwxr-x  3 aquashdw aquashdw 4096 5월  4 12:06 ./
drwxrwxr-x 70 aquashdw aquashdw 4096 5월  4 12:06 ../
drwxrwxr-x  8 aquashdw aquashdw 4096 5월  4 12:07 .git/
```



# git을 사용해 보자! - status, add, commit, push

git이 아직 파악하지 않고있는 파일을 생성하고 원격으로 보내기

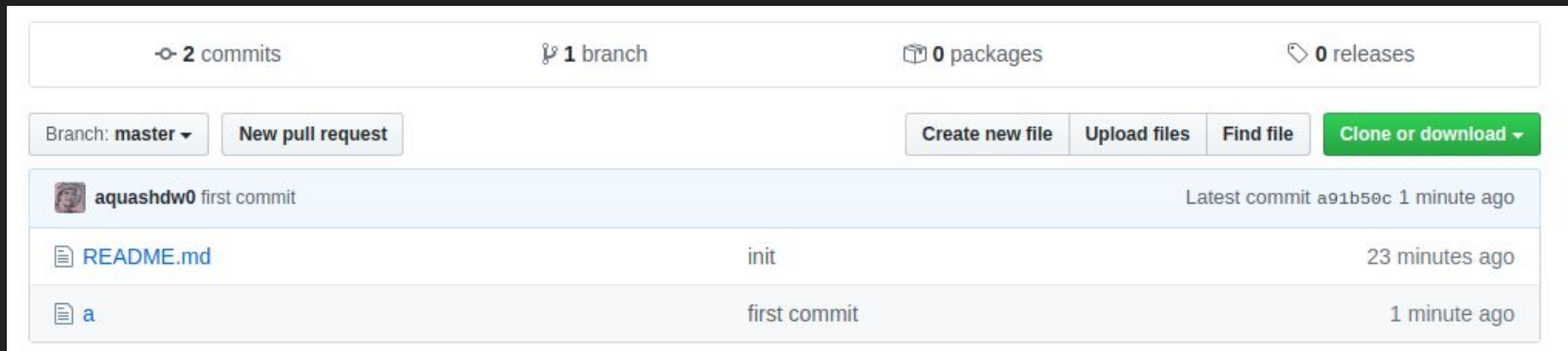
```
→ mindslab-git-test git:(master) vi a
→ mindslab-git-test git:(master) X git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        a

nothing added to commit but untracked files present (use "git add" to track)
→ mindslab-git-test git:(master) X git add a
→ mindslab-git-test git:(master) X git commit -m "first commit"
[master a91b50c] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 a
→ mindslab-git-test git:(master) git push
```

# git을 사용해보자! - status, add, commit, push

git이 아직 파악하지 않고있는 파일을 생성하고 원격으로 보내기



The screenshot shows a GitHub repository interface. At the top, it displays '2 commits', '1 branch', '0 packages', and '0 releases'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and a green 'Clone or download' button. The commit history table shows two commits: 'aquashdw0 first commit' (latest commit a91b50c, 1 minute ago) and 'init' (23 minutes ago). The file list shows 'README.md' and 'a'.

Commit	File	Commit Message	Time
aquashdw0 first commit			Latest commit a91b50c 1 minute ago
init	README.md		23 minutes ago
first commit	a		1 minute ago

status, add, commit, push

# git을 사용해 보자! - status, add, commit, push

`git status` : 현재 git 저장소의 상태를 확인한다.

`git add` : 현재 git 저장소의 상태에서 변화된 내용을 저장소에 저장하기 위해 지정한다.

`git commit` : add된 파일들에 대해서 git 저장소에 기록한다.

`git push` : local 저장소의 내용을 remote 저장소로 보낸다!

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
## mindslab-git-test
```

```
어머나 실수네여
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X 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 checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")
→ mindslab-git-test git:(master) X git checkout README.md
→ mindslab-git-test git:(master) git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
# mindslab-git-test  
git 생성 후 원격으로 보내는 예시입니다.
```

```
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X git add README.md
→ mindslab-git-test git:(master) X git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

→ mindslab-git-test git:(master) X git checkout README.md
→ mindslab-git-test git:(master) X git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X git reset HEAD
Unstaged changes after reset:
M      README.md
→ mindslab-git-test git:(master) X 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 checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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



# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working directory clean
→ mindslab-git-test git:(master) █
```

```
→ mindslab-git-test git:(master) git reset --soft HEAD~1
→ mindslab-git-test git:(master) X
→ mindslab-git-test git:(master) X
→ mindslab-git-test git:(master) X git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   README.md

→ mindslab-git-test git:(master) X git commit -m "commit that must NOT be in
note"
[master ad22ff7] commit that must NOT be in remote
 1 file changed, 1 insertion(+), 1 deletion(-)
→ mindslab-git-test git:(master) git reset --hard HEAD~1
HEAD is now at e54aa84 modify and push
→ mindslab-git-test git:(master) git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
# mindslab-git-test
git 생성 후 원격으로 보내는 예시입니다.
~
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
## mindslab-git-test
```

```
어머나 실수네여
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

```
~
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X git add README.md
→ mindslab-git-test git:(master) X git commit -m "commit that must NOT be in re
mote"
[master af228df] commit that must NOT be in remote
1 file changed, 1 insertion(+), 1 deletion(-)
→ mindslab-git-test git:(master) git reset --mixed HEAD~1
Unstaged changes after reset:
M      README.md
```

# git을 사용해 보자! - checkout, reset

실수를 하였을때

```
→ mindslab-git-test git:(master) X 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 checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

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

```
# mindslab-git-test
어머나 실수네여
```

checkout, reset

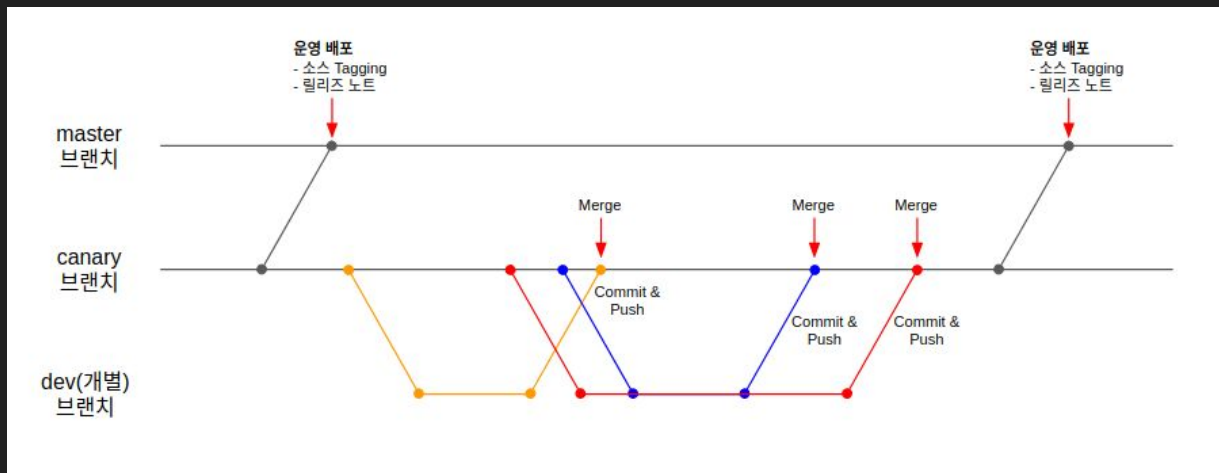
# git을 사용해 보자! - checkout, reset

`git checkout <filename>` : <filename> 파일의 상태를 현재 저장소의 상태로 복구시킨다.

`git reset --<option> HEAD~\${number}` : HEAD의 상태를 <option>에 따라 변경시킨다.

- **soft: number** 만큼의 commit을 뒤로 감고, git 변경사항들을 **staged** 상태로 유지한다.
- **hard: number** 만큼의 commit을 뒤로 감고, git 변경사항들을 전부 버리고 파일을 복구한다.
- **mixed: number** 만큼의 commit을 뒤로 감고, git 변경사항들을 전부 버린다.

# git을 활용하여 협업하자! - 개요



# git을 활용하여 협업하자! - branch, checkout

```
→ mindslab-git-test git:(master) git branch
* master
→ mindslab-git-test git:(master) git branch canary
→ mindslab-git-test git:(master) git branch
   canary
* master
→ mindslab-git-test git:(master) git checkout canary
Switched to branch 'canary'
→ mindslab-git-test git:(canary) git branch
* canary
   master
→ mindslab-git-test git:(canary) git push --set-upstream origin canary
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'canary' on GitHub by visiting:
remote:   https://github.com/aquashdw0/mindslab-git-test/pull/new/canary
remote:
To git@github.com:aquashdw0/mindslab-git-test.git
 * [new branch]      canary -> canary
Branch canary set up to track remote branch canary from origin.
```



# git을 활용하여 협업하자! - branch, checkout

**Default branch**

master Updated 42 minutes ago by aquashdw0


Default

Change default branch

**Your branches**

canary Updated 42 minutes ago by aquashdw0


0|0

New pull request 

**Active branches**

canary Updated 42 minutes ago by aquashdw0

0|0

New pull request 

checkout, branch

# git을 활용하여 협업하자! - branch, checkout

`git checkout <branch\_name>` : 작업중인 branch를 <branch\_name>으로  
정확한다.

`git branch` : 현재 저장소에 존재하는 branch들을 나열한다.

`git branch <branch\_name>` : 현재 저장소에 <branch\_name> branch를 생성한다.

`git branch -d <branch\_name>` : 현재 저장소에서 <branch\_name> branch를  
삭제한다.

# git을 활용하여 협업하자! - branch, checkout

```
→ mindslab-git-test git:(master) git branch
* master
→ mindslab-git-test git:(master) git branch canary
→ mindslab-git-test git:(master) git branch
   canary
* master
→ mindslab-git-test git:(master) git checkout canary
Switched to branch 'canary'
→ mindslab-git-test git:(canary) git branch
* canary
  master
→ mindslab-git-test git:(canary) git push --set-upstream origin canary
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'canary' on GitHub by visiting:
remote:   https://github.com/aquashdw0/mindslab-git-test/pull/new/canary
remote:
To git@github.com:aquashdw0/mindslab-git-test.git
 * [new branch]      canary -> canary
Branch canary set up to track remote branch canary from origin.
```

# git을 활용하여 협업하자! - branch, checkout

--set-upstream (or -u)

```
→ mindslab-git-test git:(canary) git branch dev
→ mindslab-git-test git:(canary) git checkout dev
Switched to branch 'dev'
→ mindslab-git-test git:(dev) git push

fatal: The current branch dev has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin dev

→ mindslab-git-test git:(dev) █
```

# git을 활용하여 협업하자! - branch, checkout

--set-upstream (or -u)

```
→ mindslab-git-test git:(dev) git push --set-upstream origin dev
Total 0 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'dev' on GitHub by visiting:
remote:   https://github.com/aquashdw0/mindslab-git-test/pull/new/dev
remote:
To git@github.com:aquashdw0/mindslab-git-test.git
 * [new branch]      dev -> dev
Branch dev set up to track remote branch dev from origin.
→ mindslab-git-test git:(dev) █
```

# git을 활용하여 협업하자! - branch, checkout

이건 덤

```
→ mindslab-git-test git:(canary) git checkout -b dev_branch  
Switched to a new branch 'dev_branch'  
→ mindslab-git-test git:(dev_branch) █
```

# git을 활용하여 협업하자! - 중간 점검

```
→ mindslab-git-test git:(canary) git branch
* canary
  dev_a
  dev_b
  master
→ mindslab-git-test git:(canary) █
```

# git을 활용하여 협업하자! - 중간 점검

```
Make a new file for git to track!  
modify the file!  
modify file for a!
```

```
→ mindslab-git-test git:(dev_a) X git add a  
→ mindslab-git-test git:(dev_a) X git commit -m "modify in branch a"  
[dev_a 9b823d5] modify in branch a  
1 file changed, 1 insertion(+)
```


```
→ mindslab-git-test git:(dev_a) git push --set-upstream origin dev_a  
Counting objects: 3, done.  
Delta compression using up to 16 threads.  
Compressing objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 338 bytes | 0 bytes/s, done.  
Total 3 (delta 0), reused 0 (delta 0)  
remote:  
remote: Create a pull request for 'dev_a' on GitHub by visiting:  
remote:      https://github.com/aquashdw0/mindslab-git-test/pull/new/dev_a  
remote:  
To git@github.com:aquashdw0/mindslab-git-test.git  
* [new branch]      dev_a -> dev_a  
Branch dev_a set up to track remote branch dev_a from origin.
```



# git을 활용하여 협업하자! - 중간 점검

```
new file in branch b!  
~  
~  
~  
→ mindslab-git-test git:(dev_b) X git add b  
→ mindslab-git-test git:(dev_b) X git commit "add new in branch b"  
error: pathspec 'add new in branch b' did not match any file(s) known to git.  
→ mindslab-git-test git:(dev_b) X git commit -m "add new in branch b"  
[dev_b 7fe1e61] add new in branch b  
1 file changed, 1 insertion(+)  
create mode 100644 b  
→ mindslab-git-test git:(dev_b) git push --set-upstream origin dev_b
```

# git을 활용하여 협업하자! - 중간 점검

Active branches				
dev_b	Updated 1 minute ago by aquashdw0	0   1	<a href="#">New pull request</a>	
dev_a	Updated 3 minutes ago by aquashdw0	0   1	<a href="#">New pull request</a>	
canary	Updated 1 hour ago by aquashdw0	0   0	<a href="#">New pull request</a>	




이제 작업들을 합치는 일만 남았습니다!

# git을 활용하여 협업하자! - merge

```
→ mindslab-git-test git:(canary) git merge --ff dev_a
Updating e54aa84..9b823d5
Fast-forward
 a | 1 +
 1 file changed, 1 insertion(+)
→ mindslab-git-test git:(canary) █
```

Branch: **canary** ▾    **New pull request**    **Create new file**    **Upload files**    **Find file**    **Clone or download** ▾

This branch is 1 commit ahead of master.    [Pull request](#)    [Compare](#)

 <b>aquashdw0</b> modify in branch a	Latest commit 9b823d5 7 minutes ago
 <a href="#">README.md</a>	init 2 hours ago
 <a href="#">a</a>	modify in branch a 7 minutes ago

# git을 활용하여 협업하자! - merge

```
→ mindslab-git-test git:(canary) git merge --no-ff dev_b
```

```
GNU nano 2.5.3 File: ...uashdw/git/mindslab-git-test/.git/MERGE_MSG
```

```
Merge branch 'dev_b' into canary
```

```
# Please enter a commit message to explain why this merge is necessary,  
# especially if it merges an updated upstream into a topic branch.  
#  
# Lines starting with '#' will be ignored, and an empty message aborts  
# the commit.
```

```
^G Get Help  
^X Exit
```

```
^O Write Out  
^R Read File
```

```
^W Where Is  
^_ Replace
```

```
^K Cut Text  
^U Uncut Text
```

```
^J Justify  
^T To Spell
```

```
^C Cur Pos  
^_ Go To Line
```

```
Merge made by the 'recursive' strategy.
```

```
b | 1 +
```

```
1 file changed, 1 insertion(+)
```

```
create mode 100644 b
```

```
→ mindslab-git-test git:(canary)
```

# git을 활용하여 협업하자! - merge

Branch: canary ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 3 commits ahead of master.

Pull request Compare

aquashdw0 Merge branch 'dev\_b' into canary

Latest commit 345ac91 1 minute ago

README.md

init

2 hours ago

a

modify in branch a

10 minutes ago

b

add new in branch b

8 minutes ago

Merge branch 'dev\_b' into canary

aquashdw0 committed 2 minutes ago

345ac91

add new in branch b

aquashdw0 committed 8 minutes ago

7fe1e61

modify in branch a

aquashdw0 committed 10 minutes ago

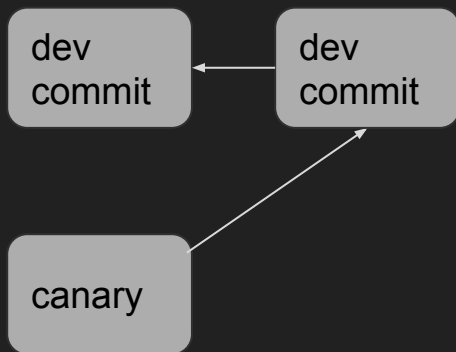
9b823d5

merge

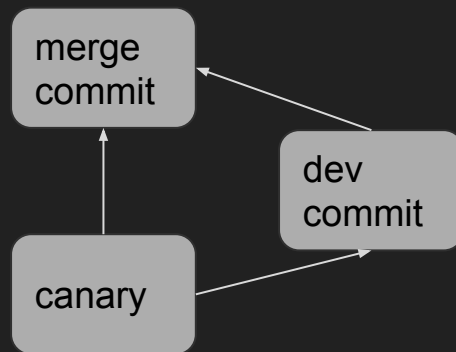
# git을 활용하여 협업하자! - merge

`git merge --<option> <branch\_name>` : 현재 branch에 <branch\_name>을 합친다.

- ff (fast-forward): 현재 branch 위에 합치려는 branch의 commit들을 얹는다.
- no-ff (no fast-forward): 두 branch를 합치는 과정을 commit으로 기록한다.



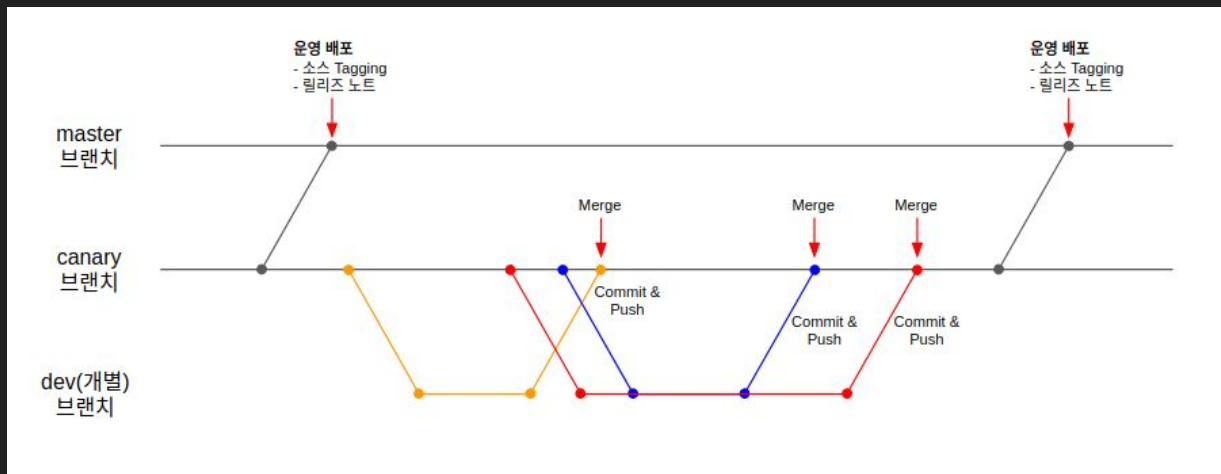
fast-forward



no-fast-forward

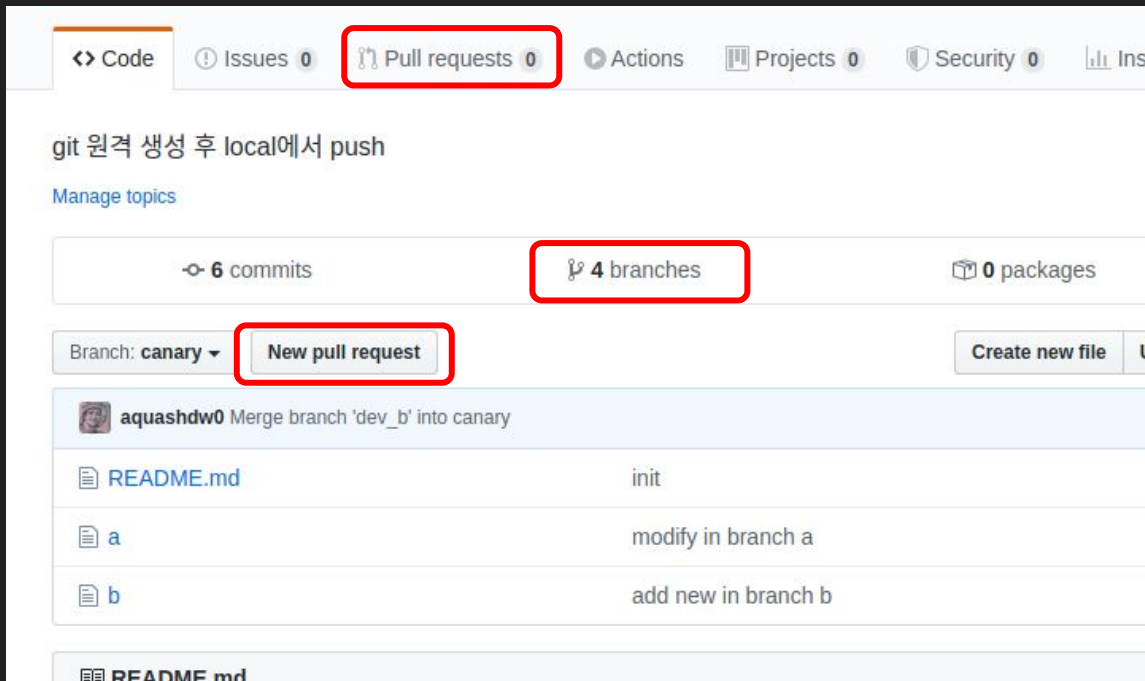
# git을 활용하여 협업하자! - Pull Request?

그럼 PR은 언제??



# git을 활용하여 협업하자! - Pull Request?

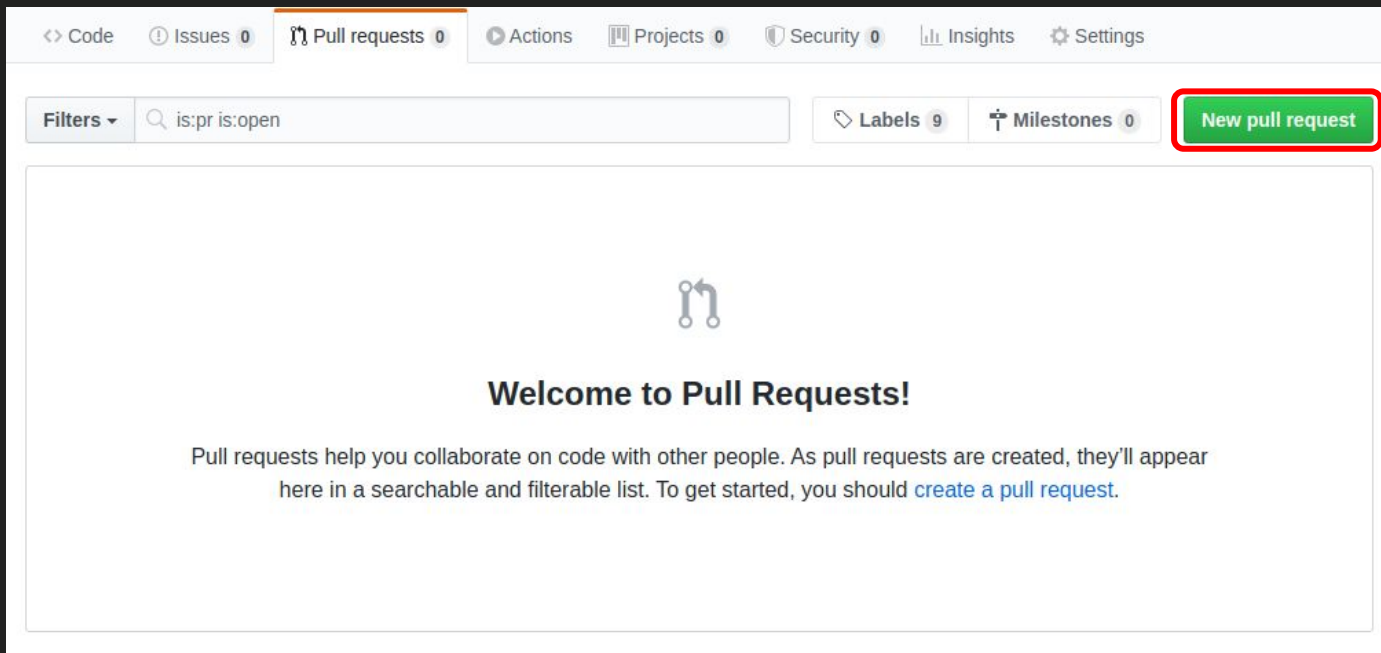
그럼 PR은 언제??





# git을 활용하여 협업하자! - Pull Request?

그럼 PR은 언제??



# git을 활용하여 협업하자! - Pull Request?

그럼 PR은 언제??

**Default branch**

canary Updated 15 minutes ago by aquashdw0

Default

Change default branch

**Active branches**

dev\_b Updated 21 minutes ago by aquashdw0

2|0

New pull request

dev\_a Updated 23 minutes ago by aquashdw0

2|0

New pull request

master Updated 2 hours ago by aquashdw0


3|0


New pull request

# git을 활용하여 협업하자! - Pull Request?

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

 base: **master** ◯ ◀ compare: **canary** ◯ ✓ **Able to merge.** These branches can be automatically merged.



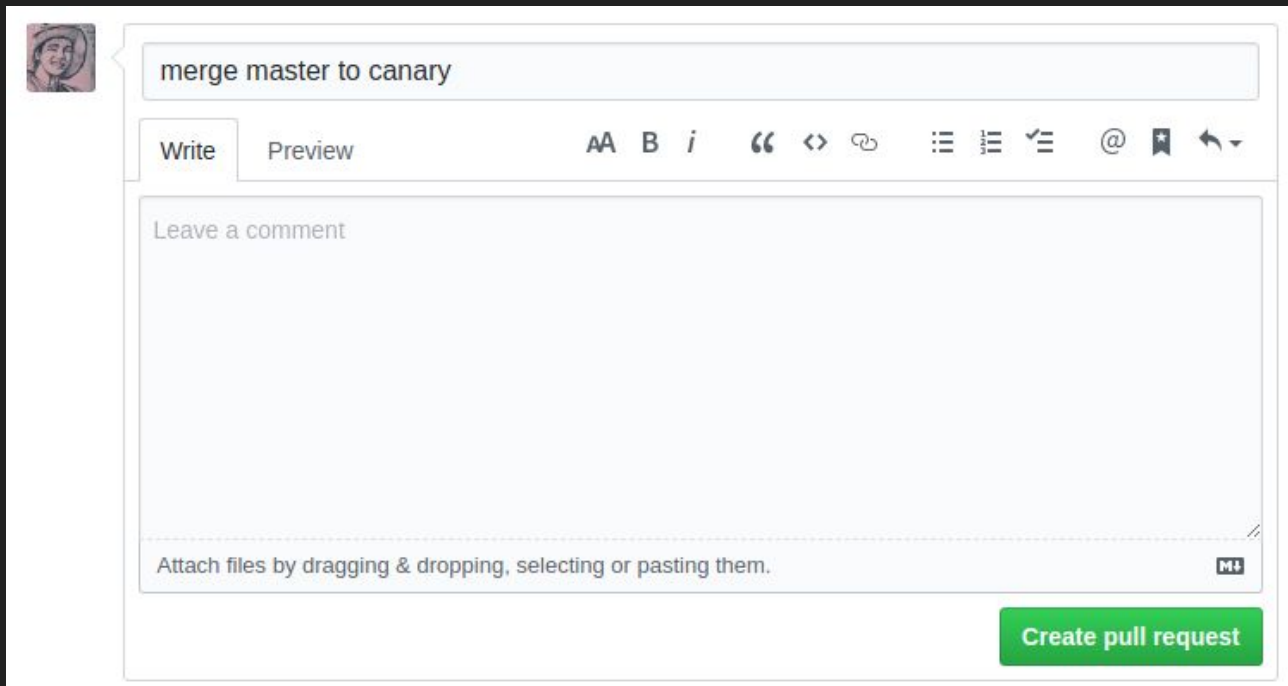
Write

Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

Leave a comment

# git을 활용하여 협업하자! - Pull Request?



A screenshot of a GitHub Pull Request interface. At the top left is a small profile picture of a person with a halo. To its right is a text box containing the title "merge master to canary". Below the title are two tabs: "Write" (selected) and "Preview". To the right of the tabs is a rich text editor toolbar with icons for bold (AA), italic (i), quote (”), code (<>), link (chain), list (three horizontal lines), ordered list (three horizontal lines with numbers), checkmark (checkmark), mention (@), bookmark (star), and undo (curved arrow). Below the tabs is a large text area with the placeholder text "Leave a comment". At the bottom of the text area is a dashed line and the text "Attach files by dragging & dropping, selecting or pasting them." followed by a small icon of a document with a plus sign. At the bottom right of the form is a green button with the text "Create pull request".

merge master to canary

Write Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ ★ ↶


Leave a comment


Attach files by dragging & dropping, selecting or pasting them. 📎


Create pull request

# git을 활용하여 협업하자! - Pull Request?


Add more commits by pushing to the **canary** branch on **aquashdw0/mindslab-git-test**.



 **Continuous integration has not been set up**  
GitHub Actions and [several other apps](#) can be used to automatically catch bugs and enforce style.

 **This branch has no conflicts with the base branch**  
Merging can be performed automatically.


[Merge pull request](#) or [view command line instructions](#).



Write Preview

AA B i “ <> 🔗 ☰ ☷ ✓ @ 📌 ↶

Leave a comment

Attach files by dragging & dropping, selecting or pasting them. 

[Close pull request](#) [Comment](#)

# git을 활용하여 협업하자! - Pull Request?


Branch: master ▾




New pull request

Create new file

Upload

This branch is 1 commit ahead of canary.

 aquashdw0 Merge pull request #1 from aquashdw0/canary ...

 README.md	init
 a	modify in branch a
 b	add new in branch b

# git을 활용하여 협업하자! - conflict!

¶ Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Adipiscing commodo elit at imperdiet dui accumsan sit. Enim praesent elementum facilisis leo vel fringilla. Volutpat ac tincidunt vitae semper quis lectus nulla at volutpat. Quis commodo odio aenean sed adipiscing. Mattis vulputate enim nulla aliquet porttitor. In hac habitasse platea dictumst vestibulum rhoncus est. Volutpat lacus laoreet non curabitur gravida. Ultrices mi tempus imperdiet nulla. Sollicitudin nibh sit amet commodo nulla facilisi. In massa tempor nec feugiat. Viverra orci sagittis eu volutpat odio facilisis mauris sit. Pulvinar elementum integer enim neque volutpat ac. Pellentesque elit eget gravida cum sociis natoque penatibus et. Eu feugiat pretium nibh ipsum consequat nisl vel pretium. Vel turpis nunc eget lorem dolor sed. Dignissim cras tincidunt lobortis feugiat vivamus at. Urna dui convallis convallis tellus. Id ornare arcu odio ut sem nulla pharetra diam sit. Quam adipiscing vitae proin sagittis.

At augue eget arcu dictum varius duis at consectetur. Ut aliquam purus sit amet luctus venenatis. Eget lorem dolor sed viverra. Mauris cursus mattis molestie a iaculis at erat pellentesque adipiscing. Vitae turpis massa sed elementum tempus. Sapien pellentesque habitant morbi tristique. Vestibulum lorem sed risus ultricies tristique. Cursus sit amet dictum sit amet justo donec enim diam. Eget felis eget nunc lobortis mattis aliquam. Massa sed elementum tempus egestas sed sed risus pretium. Auctor urna nunc id cursus metus aliquam. Mus mauris vitae ultricies leo. Ante in nibh mauris cursus. Habitant morbi tristique senectus et netus. Morbi tincidunt augue interdum velit euismod.



# git을 활용하여 협업하자! - conflict

```
→ mindslab-git-test git:(canary) X git branch
* canary
  dev_a
  dev_b
  master
```

>Lorem dolor sit amet, consectetur adipiscing elit. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

dev\_a

>Lorem ipsum sit amet, consectetur adipiscing elit. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

dev\_b



# git을 활용하여 협업하자! - conflict!

```
→ mindslab-git-test git:(canary) git merge dev_a  
→ mindslab-git-test git:(canary) git merge dev_b
```

```
→ mindslab-git-test git:(canary) git merge dev_a  
Updating ed049ad..8a506b5  
Fast-forward  
 c | 2 +-  
 1 file changed, 1 insertion(+), 1 deletion(-)  
→ mindslab-git-test git:(canary) git merge dev_b  
Auto-merging c  
CONFLICT (content): Merge conflict in c  
Automatic merge failed; fix conflicts and then commit the result.  
→ mindslab-git-test git:(canary) X
```

```
<<<<<< HEAD  
Lorem dolor sit amet, conse  
unt ut labore et dolore mag  
ccumsan sit. Enim praesent  
ncidunt vitae semper quis l  
adipiscing. Mattis vulputa  
tea dictumst vestibulum rho  
a. Ultrices mi tempus imper  
facilisi. In massa tempor n  
cilisis mauris sit. Pulvina  
que elit eget gravida cum s  
psum consequat nisl vel pre  
cras tincidunt lobortis fe  
s. Id ornare arcu odio ut s  
in sagittis.  
=====  
Lorem ipsum sit amet, conse  
unt ut labore et dolore mag  
ccumsan sit. Enim praesent  
ncidunt vitae semper quis l  
adipiscing. Mattis vulputa  
tea dictumst vestibulum rho  
a. Ultrices mi tempus imper  
facilisi. In massa tempor n  
cilisis mauris sit. Pulvina  
que elit eget gravida cum s  
psum consequat nisl vel pre  
cras tincidunt lobortis fe  
s. Id ornare arcu odio ut s  
in sagittis.  
>>>>>> dev_b
```

# git을 활용하여 협업하자! - conflict!

merge conflict

두개 이상의 브랜치를 merge하려고 할 때

A 브랜치와 B 브랜치에서 동일한 파일의  
동일한 줄을 수정할 경우 conflict가  
발생한다

# git을 활용하여 협업하자! - conflict!

```
→ mindslab-git-test git:(canary) ✗ vi c
→ mindslab-git-test git:(canary) ✗ git status
On branch canary
Your branch is ahead of 'origin/canary' by 1 commit.
  (use "git push" to publish your local commits)
You have unmerged paths.
  (fix conflicts and run "git commit")

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

        both modified:   c

no changes added to commit (use "git add" and/or "git commit -a")
→ mindslab-git-test git:(canary) ✗ git add c
→ mindslab-git-test git:(canary) ✗ git commit -m "merge conflict"
[canary a98e26e] merge conflict
```

# git을 활용하여 협업하자! - rebase

```
→ mindslab-git-test git:(canary) ll
total 36
drwxrwxr-x  3 aquashdw aquashdw 4096 5월 4 15:16 ./
drwxrwxr-x 70 aquashdw aquashdw 4096 5월 4 12:06 ../
-rw-rw-r--  1 aquashdw aquashdw   70 5월 4 14:02 a
-rw-rw-r--  1 aquashdw aquashdw   22 5월 4 14:04 b
-rw-rw-r--  1 aquashdw aquashdw 4302 5월 4 15:16 c
-rw-rw-r--  1 aquashdw aquashdw   42 5월 4 15:16 d
drwxrwxr-x  8 aquashdw aquashdw 4096 5월 4 15:16 .git/
-rw-rw-r--  1 aquashdw aquashdw   75 5월 4 13:22 README.md
→ mindslab-git-test git:(canary) git status
On branch canary
Your branch is up-to-date with 'origin/canary'.
nothing to commit, working directory clean
```

# git을 활용하여 협업하자! - rebase

`git log` : 현재 branch의 commit log를 확인한다

```
commit 70af2badf4059d36df96968fd3b632fddc7ced7b
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:16:47 2020 +0900
```

add d

```
commit a98e26e9e6bec79e1f41d5c360261de74e692bab
Merge: 8a506b5 7b8dba2
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:12:32 2020 +0900
```

merge conflict

```
commit 7b8dba2f42ba67384958010de46b10e37fc5df78
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:06:30 2020 +0900
```

modify c in branch dev\_b

canary

```
commit af4eae90f27b07ae807e509b7f6d075282650758
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:16:01 2020 +0900
```

add another paragraph in dev\_c

```
commit a98e26e9e6bec79e1f41d5c360261de74e692bab
Merge: 8a506b5 7b8dba2
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:12:32 2020 +0900
```

merge conflict

```
commit 7b8dba2f42ba67384958010de46b10e37fc5df78
Author: 지호 박 <aquashdw@mindslab.ai>
Date:   Mon May 4 15:06:30 2020 +0900
```

add feature to branch dev\_c  
dev\_c



# git을 활용하여 협업하자! - rebase

```
→ mindslab-git-test git:(canary)
→ mindslab-git-test git:(canary) git checkout dev_c
Switched to branch 'dev_c'
Your branch is up-to-date with 'origin/dev_c'.
→ mindslab-git-test git:(dev_c) git rebase canary
First, rewinding head to replay your work on top of it...
Applying: add another paragraph in dev_c
```

```
total 36
drwxrwxr-x 3 aquashdw aquashdw 4096 5월 4 15:19 ./
drwxrwxr-x 70 aquashdw aquashdw 4096 5월 4 12:06 ../
-rw-rw-r-- 1 aquashdw aquashdw 70 5월 4 14:02 a
-rw-rw-r-- 1 aquashdw aquashdw 22 5월 4 14:04 b
-rw-rw-r-- 1 aquashdw aquashdw 5243 5월 4 15:19 c
-rw-rw-r-- 1 aquashdw aquashdw 42 5월 4 15:19 d
drwxrwxr-x 8 aquashdw aquashdw 4096 5월 4 15:20 .git/
-rw-rw-r-- 1 aquashdw aquashdw 75 5월 4 13:22 README.md
→ mindslab-git-test git:(dev_c) █
```

```
commit c50f80e8b219143287bd17caa74d92f03a5c6fce
Author: 지호 박 <aquashdw@mindslab.ai>
Date: Mon May 4 15:16:01 2020 +0900
```

add another paragraph in dev\_c

```
commit 70af2badf4059d36df96968fd3b632fddc7ced7b
Author: 지호 박 <aquashdw@mindslab.ai>
Date: Mon May 4 15:16:47 2020 +0900
```

add d

```
commit a98e26e9e6bec79e1f41d5c360261de74e692bab
Merge: 8a506b5 7b8dba2
Author: 지호 박 <aquashdw@mindslab.ai>
Date: Mon May 4 15:12:32 2020 +0900
```

merge conflict

```
your branch is up-to-date with 'origin/canary'
→ mindslab-git-test git:(canary) git merge dev_c
Updating 70af2ba..c50f80e
Fast-forward
 c | 2 ++
 1 file changed, 2 insertions(+)
→ mindslab-git-test git:(canary) █
```

# QA

reference: <https://dev.to/lydiahallie/cs-visualized-useful-git-commands-37p1>