

깃
Git

INDEX

- Git 이란 / 초기 설정
- 파일 관리 (개인)
- 파일 관리 (협업)
- 파일 관리 (관리자)

초기

Beginning

Git

»» 분산 버전 관리 시스템

- 소스 코드를 관리
- 파일의 변화를 시간에 따라 기록
- 과거의 특정 시점의 버전을 다시 불러올 수 있음
- VCS (Version Control System)

01

02

03

Setting

➤➤ 파일 선택 후 git 열기
(오른쪽 마우스 클릭)



Git Bash Here



MINGW64:/c/Users/xezou/Desktop/gitP

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/gitP  
$ |
```

01

02

03

Setting

≫≫ 사용자 설정

- `git config --global user.name (이름)`
사용자명 등록(필수)
- `git config --global user.email (이메일)`
이메일 주소 등록(필수)

01

02

03

Setting

»» 사용자 설정 성공시

- `git config --global --list`
사용자명 / 이메일 조회

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/sdf (master)
$ git config --global --list
user.email=xezout108@naver.com
user.name=xezout108
```

»» 사전 정보

- Git의 저장소는 3가지 단계로 나누어짐
 - 커밋한 소스가 보관되는 저장소
 - 현재 프로젝트 파일들이 있는 작업 트리
 - 저장소와 작업 트리사이의 버퍼영역으로 커밋될 대상이 저장되어 있는 스테이징 영역
- Git은 빈 디렉토리는 추적하지 않음
- 형상관리를 하지 않을 파일은 .gitignore 파일에 추가
- HEAD는 현재 브랜치의 가장 최신 커밋을 의미
- 기본 원격 저장소를 origin 이라고 부름

개인
Personal


프로젝트 생성

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

[Read the guide](#)[Start a project](#)

Owner

 xezout108 ▾

Repository name

Great repository names are short and memorable. Need inspiration? How about [jubilant-octo-umbrella](#).

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

[Create repository](#)

»» 프로젝트 생성 성공시

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** SSH <https://github.com/xezout108/ddd.git>

We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

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

```
echo "# ddd" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/xezout108/ddd.git
git push -u origin master
```

»» git에서 위의 코드 작성 (echo 옆에 들어가는 것은 Readme.md 내용)

- git init
현재 폴더에 git 저장소를 생성

01

02

>> 코드 작성시

1 commit

1 branch

0 releases

1 contributor

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

hezout108 First Commit

Latest commit 2d2cd91 8 minutes ago

Readme.md

First Commit

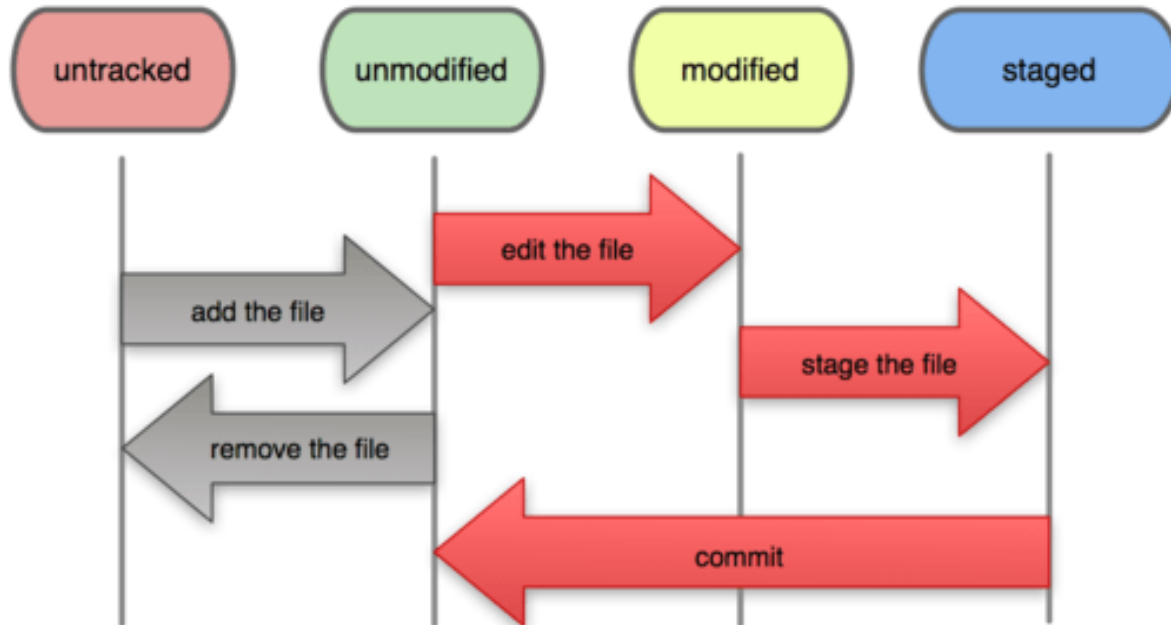
8 minutes ago

Readme.md

Prac01

Git의 파일 관리 시스템

File Status Lifecycle



파일 올리기

- git status
커밋되지 않은 변경사항 조회

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/sdf (master)
$ 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)

    ex01.txt

nothing added to commit but untracked files present (use "git add" to track)
```

- git add 파일명
untracked files의 파일들과 수정되었지만 아직 스테이징 영역에 올라가지 않은 파일들을 스테이징 영역에 올림
(파일명에 .을 쓰면 모든 파일에 대해 작용할 수 있음)

>> Add한 후

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ 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)

    new file:   ex01.txt
```

- git commit 파일명
스테이징 영역에 올라가 있는 파일들을 커밋함
(파일명에 .을 쓰면 모든 파일에 대해 작용할 수 있음)

>> Commit한 후

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (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 tree clean
```

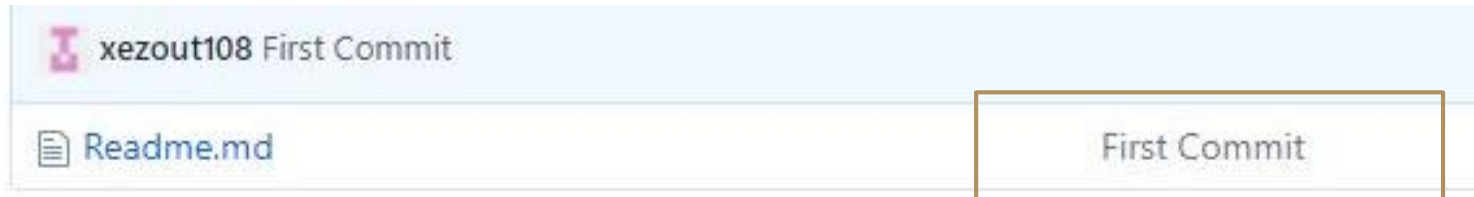
» Commit 추가 설명

[illegible]

- i 누르기 : 끼워 넣기 (콘솔창에 작성이 가능해짐)
(ESC 누를 시 종료)
- :wq : 위에 보이는 창 종료
- :q : 위에 보이는 창 종료(저장하지 않음)

≫≫ Commit 추가 설명

- 콘솔창에 작성한 메시지



- `git commit 파일명 -m “커밋 메시지”`
메시지 작성 창을 생략하고 바로 커밋할 수 있음

»» Commit 후 추가 수정시

- git diff
스테이징 영역과 현재 작업 트리의 차이점을 보여줌

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git diff
diff --git a/ex01.txt b/ex01.txt
index fea4135..11b1da4 100644
--- a/ex01.txt
+++ b/ex01.txt
@@ -1,2 @@
-First Commit
\ No newline at end of file
+First Commit
+Change
\ No newline at end of file
```

»» Commit 후 추가 수정시

- git diff
스테이징 영역과 현재 작업 트리의 차이점을 보여줌

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git diff
diff --git a/ex01.txt b/ex01.txt
index fea4135..11b1da4 100644
--- a/ex01.txt
+++ b/ex01.txt
@@ -1,2 @@
-First Commit
\ No newline at end of file
+First Commit
+Change
\ No newline at end of file
```

»» Commit 로그 조회

- git log
커밋 로그를 볼 수 있음 (-숫자 로 개수 지정 가능)
(--graph 추가시 브랜치 트리를 볼 수 있음)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git log
commit a57e518542acb059350d302a4ddd31ac8b330dc0 (HEAD -> master)
Author: xezout108 <xezout108@naver.com>
Date:   Mon Sep 18 01:06:13 2017 +0900

    asd

commit bdb331b7c702ba449c7c93da499058dce6749aa9
Author: xezout108 <xezout108@naver.com>
Date:   Mon Sep 18 01:03:26 2017 +0900

    sdf

commit 70e06b26e017477853522f5084d471eac60dcda8
Author: xezout108 <xezout108@naver.com>
Date:   Mon Sep 18 00:58:44 2017 +0900

    Update

commit 802dfb3815673252a976e1690a4428fc1c451713
Author: xezout108 <xezout108@naver.com>
Date:   Mon Sep 18 00:48:05 2017 +0900
```

≫≫ 파일 Github에 올리기

- git push 리모트저장소(이름) 브랜치(이름)
현재 commit된 상태의 파일들을 github에 올림

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git push origin master
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (12/12), 1008 bytes | 0 bytes/s, done.
Total 12 (delta 0), reused 0 (delta 0)
To https://github.com/xezout108/prac1.git
 2d2cd91..a57e518 master -> master
```

01

02

>> Push 전

 xezout108 First Commit Readme.md

First Commit

>> Push 후

 xezout108 asd Readme.md

First Commit

 ex01.txt

asd

협업
Team

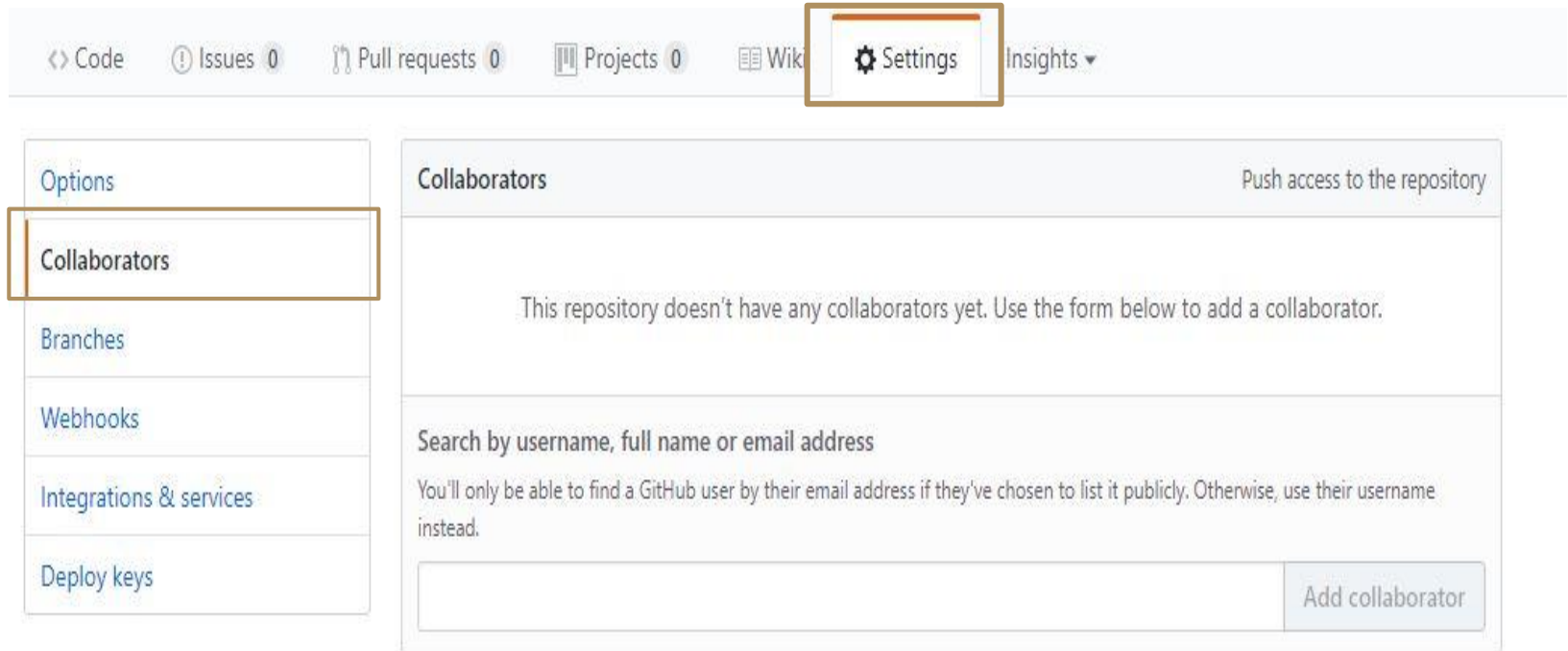
01

02

03

04

동료 추가하기



- Collaborator는 허가를 받을 필요 없이 commit을 할 수 있음

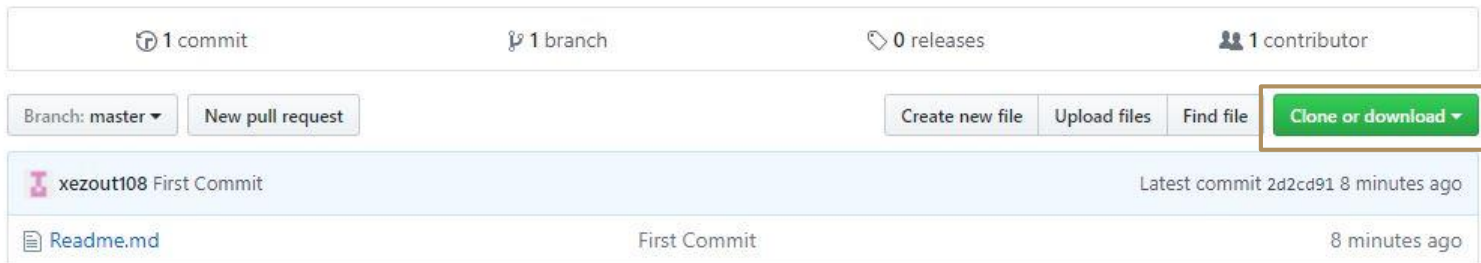
01

02

03

04

파일 가져오기



- git clone 저장소주소 폴더명
리모트 저장소를 복제하여 저장소를 생성함 (폴더 안에)
(폴더명 생략 가능 / 프로젝트 이름으로 폴더 생성됨)

» 리모트 저장소

- git remote
현재 존재하는 리모트 저장소 목록 조회
(-v 사용시 사용자명과 url을 볼 수 있음)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote
origin

xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote -v
origin https://github.com/xezout108/prac1.git (fetch)
origin https://github.com/xezout108/prac1.git (push)
```

- git remote show 이름
해당 리모트 저장소의 정보 조회

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/xezout108/prac1.git
Push URL: https://github.com/xezout108/prac1.git
HEAD branch: master
Remote branch:
  master tracked
Local branch configured for 'git pull':
  master merges with remote master
Local ref configured for 'git push':
  master pushes to master (up to date)
```

01

02

03

04

»» 리모트 저장소 생성 / 제거

- git remote add 이름 저장소주소
새로운 리모트 저장소 생성
- git remote rm 이름
해당 리모트 저장소를 제거

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote -v
origin  https://github.com/xezout108/prac1.git (fetch)
origin  https://github.com/xezout108/prac1.git (push)

xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote add pr1 https://github.com/pr1/prac1.git

xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git remote -v
origin  https://github.com/xezout108/prac1.git (fetch)
origin  https://github.com/xezout108/prac1.git (push)
pr1     https://github.com/pr1/prac1.git (fetch)
pr1     https://github.com/pr1/prac1.git (push)
```

02 >> 리모트 저장소 갱신

- git fetch 리모트저장소(이름) 리모트브랜치(이름)
리모트 저장소의 변경사항을 가져와서 리모트 브랜치를 갱신
(merge는 하지 않음)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git fetch origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/xezout108/prac1
 * branch          master      -> FETCH_HEAD
a57e518..ba8b33a  master      -> origin/master
```

- git pull 리모트저장소(이름) 리모트브랜치(이름)
리모트 저장소의 변경사항을 가져와서 로컬 브랜치에 합치는 작업을 한
꺼번에 함
(리모트저장소는 가져올 곳이고 리모트브랜치는 적용할 브랜치)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git pull origin master
From https://github.com/xezout108/prac1
 * branch          master      -> FETCH_HEAD
Updating a57e518..ba8b33a
Fast-forward
 ex01.txt | 3 ++-
1 file changed, 2 insertions(+), 1 deletion(-)
```

01

02

03

04

»» 다른 사람 수정 내용 보기

Fetch를 한 경우 (Pull을 하면 바로 merge하기 때문에 안됨)

- `git diff 브랜치명 리모트저장소(이름)/브랜치명`
자신의 저장소에 있는 브랜치와 외부에 저장된 브랜치를 비교하여 다른 사람의 수정 내용을 봄

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git diff master origin/master
diff --git a/ex01.txt b/ex01.txt
index 11b1da4..05fe6f2 100644
--- a/ex01.txt
+++ b/ex01.txt
@@ -1,2 +1,3 @@
 First Commit
-Change
\ No newline at end of file
+Change1
+Change2
\ No newline at end of file
```

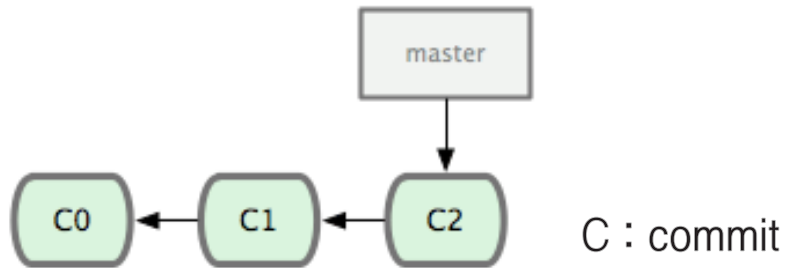
01

02

03

04

브랜치



- git branch
현재 존재하는 브랜치를 조회 (-r 추가시 원격저장소의 브랜치 확인 가능)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git branch
* master
  prac1
```

- git checkout
현재 작업 중인 브랜치 확인

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git checkout
Your branch is up-to-date with 'origin/master'.
```

»» 새로운 브랜치 생성

- `git branch 브랜치명B 브랜치명A`
브랜치명A에서 새로운 브랜치 브랜치명B를 만듦
(checkout은 하지 않음)
(기본 브랜치 이름은 master임)
(현재 브랜치에 생성시 브랜치명A는 생략 가능)
- `git checkout -b 브랜치명B 브랜치A`
브랜치명A에서 브랜치명B라는 새로운 브랜치를 만들어서 checkout함
(현재 브랜치에 생성시 브랜치명A는 생략 가능)

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git branch prac2

xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git checkout -b prac3 master
Switched to a new branch 'prac3'

xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (prac3)
$ git branch
  master
  prac1
  prac2
* prac3
```

01

02

03

04

➤➤ 브랜치 지우기 (거의 사용 안함)

- `git branch -d 브랜치명`
해당 브랜치를 지움
(github에서는 지워지지 않음)
- `git push origin --delete 브랜치명`
해당 브랜치를 github에서 지움

01

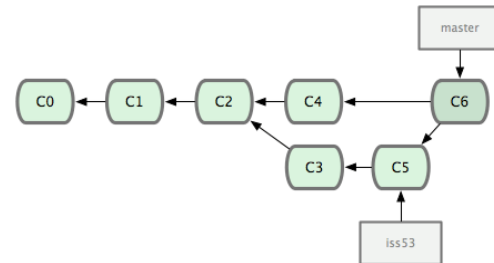
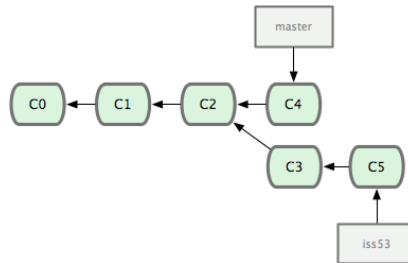
02

03

04

≫≫ 브랜치 병합 (merge / rebase)

- git merge 브랜치명
해당 브랜치를 현재 브랜치로 합침



```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/dididi (master)
$ git merge prac3
Updating a57e518..ba8b33a
Fast-forward
 ex01.txt | 3 ++-
 1 file changed, 2 insertions(+), 1 deletion(-)
```


01

02

03

04

태그

➤➤ Annotated 태그

- `git tag -a 태그명 -m “태그 메시지 “`
태그를 만든 사람, 이메일, 날짜, 메시지, GPG서명 까지 가능한 태그

➤➤ Lightweight 태그

- `git tag 태그명`
특정 커밋을 가리키는 역할만 하는 태그

관리자

Master

Pull Request

➤➤ 1. Pull Request 넣기 (Contributor)

The image shows two GitHub repository pages. The top page is for 'xezout108 / prac1'. It has a 'Watch' button with a dropdown, a 'Star' button with '0', and a 'Fork' button with '0'. The 'Fork' button is highlighted with a brown border. Below the repository name, there are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. The 'Code' tab is selected. Below the tabs, there is a message 'No description, website, or topics provided.' and an 'Edit' button. Below that, there is a bar showing '6 commits', '1 branch', '0 releases', and '1 contributor'. The bottom page is for 'khu-lee / prac1', which is a fork of 'xezout108/prac1'. It has a 'Watch' button with a dropdown, a 'Star' button with '0', and a 'Fork' button with '1'. Below the repository name, there are tabs for 'Code', 'Pull requests 0', 'Projects 0', 'Wiki', 'Settings', and 'Insights'. The 'Pull requests 0' tab is selected. Below the tabs, there is a message 'No description, website, or topics provided.' and an 'Edit' button. Below that, there is a bar showing '6 commits', '1 branch', '0 releases', and '1 contributor'.

xezout108 / prac1

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Edit

Add topics

6 commits 1 branch 0 releases 1 contributor

khu-lee / prac1
forked from xezout108/prac1

Watch 0 Star 0 Fork 1

Code Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. Edit

Add topics

6 commits 1 branch 0 releases 1 contributor

➤➤ Fork한 저장소 clone

- `git clone` (Fork했을 때 생긴 URL)

➤➤ 로컬 저장소에 원본 프로젝트 리모트 저장소 추가

- `git remote add` 이름 URL

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (master)
$ git remote -v
sub      https://github.com/freckie/pr.git (fetch)
sub      https://github.com/freckie/pr.git (push)
upstream https://github.com/xezout108/pr.git (fetch)
upstream https://github.com/xezout108/pr.git (push)
```

≫≫ 작업할 때는 새로운 브랜치를 만들어서 작업함

- `git checkout -b 브랜치명B 브랜치A`
브랜치명A에서 브랜치명B라는 새로운 브랜치를 만들어서 checkout함
(현재 브랜치에 생성시 브랜치명A는 생략 가능)

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (master)
$ git checkout -b prac1
Switched to a new branch 'prac1'
```

»» 작업 완료 후 본인의 저장소(github)에 올리기

- git push 리모트저장소(이름)(개인) 브랜치명

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (master)
$ git remote -v
sub      https://github.com/freckie/pr.git (fetch)
sub      https://github.com/freckie/pr.git (push)
upstream https://github.com/xezout108/pr.git (fetch)
upstream https://github.com/xezout108/pr.git (push)
```

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (master)
$ git push sub prac1
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/freckie/pr.git
 * [new branch]      prac1 -> prac1
```


01

02

>> Github에 올린 후

Your recently pushed branches:

🔗 **prac1** (less than a minute ago)

🔗 Compare & pull request

Branch: master ▼

New pull request

Create new file

Upload files

Find file

Clone or download ▼

This branch is even with xezout108:master.

🔗 Pull request 📄 Compare

🔗 **xezout108** File Update

Latest commit 78dcb18 11 minutes ago

📄 [Readme.md](#)

First Commit

18 minutes ago

📄 [ex01.txt](#)


File Update

11 minutes ago


01

02

>> Github에 올린 후

 base fork: xezout108/pr base: master ... head fork: freckie/pr compare: prac1

✓ **Able to merge.** These branches can be automatically merged.



Change ex01 [Pull Request 제목]

Write

Preview

AA ▾ B i “ < > 🔗 ☰ ☷ ☰ ☷ ↶ @ 📌

Leave a comment

[Pull Request 내용]

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

02

➡➡ Pull Request 보내기 성공시

🔍 1 Open ✓ 0 Closed

Author ▾ Labels ▾ Projects ▾ Milestones ▾ Reviews ▾ Assignee ▾ Sort ▾

🔍 ex01 내용 수정함

#1 opened 2 minutes ago by freckie

≫ 2. Pull Request 받기 (Master)

xezout108 / prac1

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Settings Insights

No description, website, or topics provided. [Edit](#)

[Add topics](#)


6 commits 1 branch 0 releases 1 contributor

1 Open 0 Closed




Author Labels Projects Milestones Reviews Assignee Sort

ex01 내용 수정함
#1 opened 2 minutes ago by freckie


» 2. Pull Request 받기 (Master)


 **Open** freckie wants to merge 1 commit into xezout108:master from freckie:prac1

Conversation 3 Commits 1 **Files changed 1**

 freckie commented 8 minutes ago First-time GitHub contributor  


ex01 내용을 추가했어요...

 Change ex01 cb94fc7


 xezout108 reviewed a minute ago View changes

ex01.txt

...	...	@@ -1 +1,2 @@
1	1	Original File


 xezout108 a minute ago Owner

여기도 수정좀


 Reply...

ex01.txt

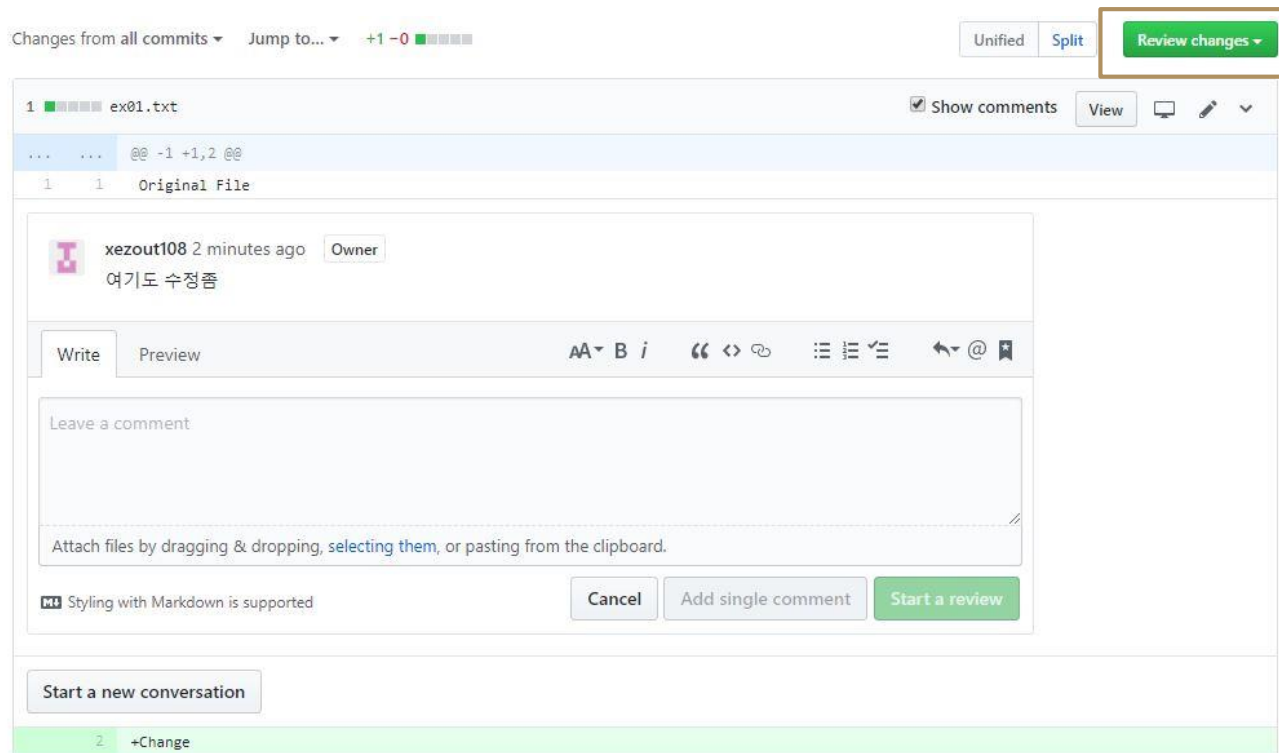
...	...	@@ -1 +1,2 @@
1	1	Original File
2		+Change

 xezout108 a minute ago Owner

이거 내용좀 바꿔바

 Reply...

≫ 2. Pull Request 받기 (Master)



➤➤ 2. Pull Request 받기 (Master)



- 변경 사항이 원래 저장소에 merge됨
(Pull Request 받기 완료)

➤➤ Pull Request 받기 완료 후

- `git pull` 리모트저장소(이름) 브랜치명

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (prac1)
$ git pull upstream master
remote: Counting objects: 1, done.
remote: Total 1 (delta 0), reused 1 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From https://github.com/xezout108/pr
 * branch            master       -> FETCH_HEAD
 * [new branch]      master       -> upstream/master
Updating cb94fc7..503fe4b
Fast-forward
```


» Pull Request 받기 완료 후

- `git log --graph`
(커밋 로그를 그래프 형태로 볼 수 있음)

```
callab@ENI-B7-13 MINGW32 ~/Desktop/aa/pr (prac1)
$ git log --graph
* commit 503fe4ba48b3c4e2ac4e613a85ffa26f4165c720 (HEAD -> prac1, upstream/master)
|
| Merge: 78dcb18 cb94fc7
| Author: LeeSH <30400648+xezout108@users.noreply.github.com>
| Date: Mon Sep 18 15:37:00 2017 +0900
|
| Merge pull request #1 from freckie/prac1
|
| ex01 내용 수정함
|
* commit cb94fc7b4750aa6f1f1410307c80097c5d444bdd (sub/prac1)
| Author: ENI-B7-13\callab <freckie@frec.kr>
| Date: Mon Sep 18 15:20:44 2017 +0900
|
| Change ex01
|
* commit 78dcb18a2bf9cf85d1fdb77d91bcc9e24bf88d7d (sub/master, sub/HEAD, master)
| Author: xezout108 <xezout108@naver.com>
| Date: Mon Sep 18 15:13:21 2017 +0900
|
| File Update
|
* commit 1a6ff34b84ebbee333d1e526a1452adb122efd28
```

추가 내용

More

➤➤ 1. 파일 목록 조회

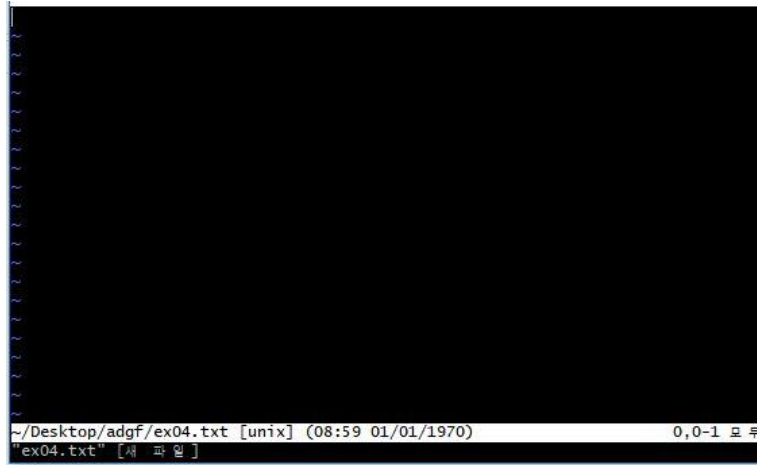
- `ls -al`
현재 폴더에 있는 파일의 이름과 생성 시간의 목록 조회

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/adgf (master)
$ ls -al
total 18
drwxr-xr-x 1 xezou 197609 0 9월 18 16:30 ./
drwxr-xr-x 1 xezou 197609 0 9월 18 16:28 ../
drwxr-xr-x 1 xezou 197609 0 9월 18 16:30 .git/
-rw-r--r-- 1 xezou 197609 6 9월 18 16:29 ex01.txt
-rw-r--r-- 1 xezou 197609 0 9월 18 16:29 ex02.txt
-rw-r--r-- 1 xezou 197609 0 9월 18 16:30 ex03.txt
-rw-r--r-- 1 xezou 197609 7 9월 18 16:29 Readme.md
```

➤➤ 2. 파일 생성 및 편집기

```
xezou@LAPTOP-BM7IDRTA MINGW64 ~/Desktop/adgf (master)  
$ vim ex04.txt
```

- vim 파일명 (확장명까지 작성해야 함)
해당 파일이 없을 시에는 해당 파일을 생성하고 편집함
해당 파일이 존재할 시 그 파일의 내용을 표시하고 편집함



- i 누르기 : 끼워 넣기 (콘솔창에 작성이 가능해짐)
(ESC 누를 시 종료)
- :q : 위에 보이는 창 종료(저장하지 않음)
- :wq : 위에 보이는 창 종료(저장함)

Thank you