



# BI 코딩 실무 II

- 실무에서 사용하는 github

한주현

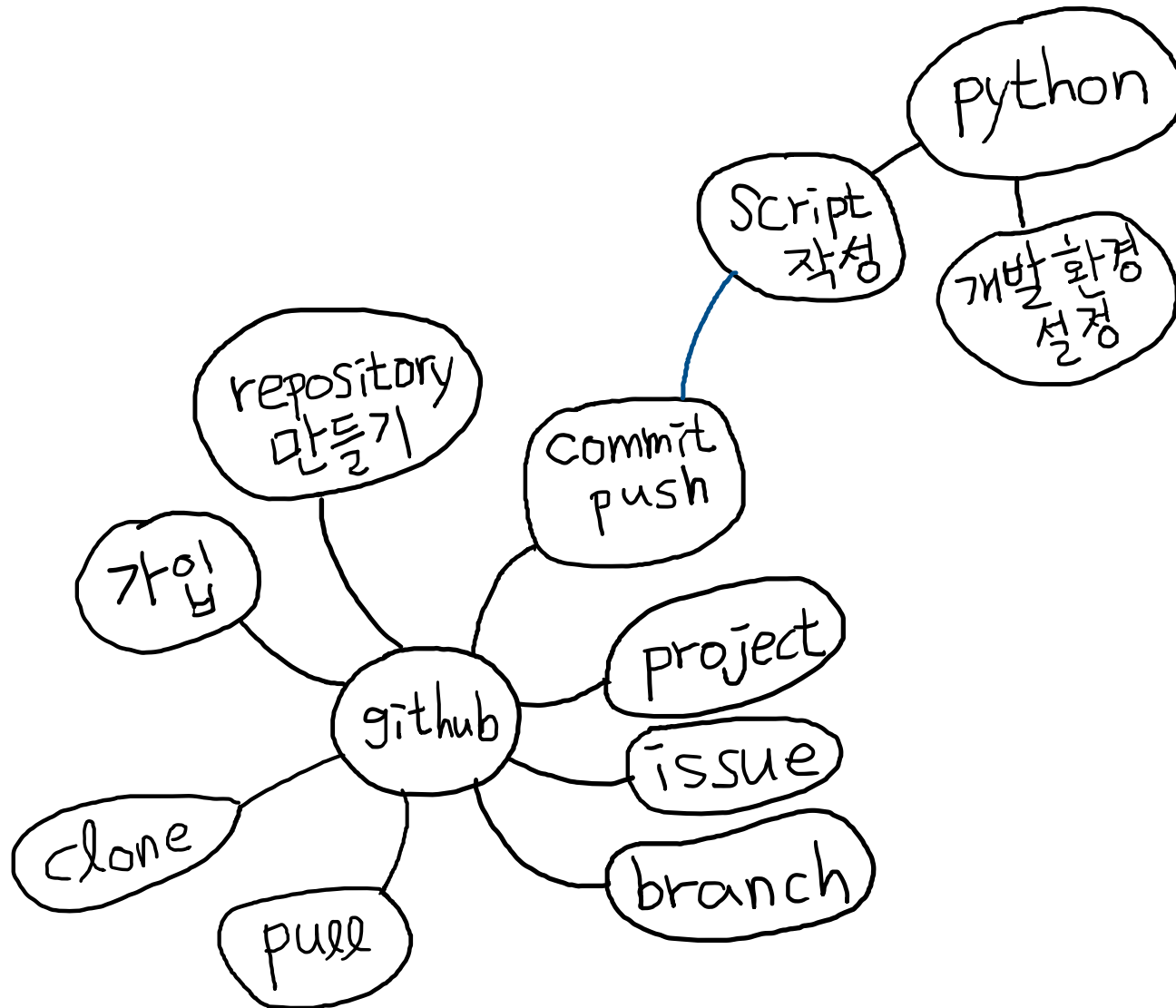
10/17/2020

[kenneth.jh.han@snu.ac.kr](mailto:kenneth.jh.han@snu.ac.kr)

# 강사소개

강사 소개	
이름	한주현
소속	3billion, Bioinformatics Engineer Seoul National University, Medical Informatics
메일	<a href="mailto:kenneth.jh.han@snu.ac.kr">kenneth.jh.han@snu.ac.kr</a>
주요 업무	Human Genome Analysis (WGS, WES) Rare Disease Analysis Bioinformatics Algorithms Analysis Pipeline / Platform Development Full Stack Development Cloud Computing
주 언어	Python, JAVA, JavaScript, Bash shell
저서	니콜라스 볼커 이야기 (2016.10, 금창원 외 공역) 바이오파이썬으로 시작하는 생물정보학 (2019.03, 한주현) 생명정보학 알고리즘 (2019.10, 한주현, 김태운 공역)
웹 페이지	<a href="https://korbillgates.tistory.com">https://korbillgates.tistory.com</a> (블로그)

# 금주 강의 내용



# 과제

- github 에 homework\_1 라고 하는 repository 를 생성한다.
- projec를 생성 후 과제를 issue로 등록한다.
- issue에 과제 수행한 내용을 정리한다.
- 다음 파일을 다운로드 받아서 옴기서열을 세는 파이썬 스크립트를 작성한다.
- [https://raw.githubusercontent.com/KennethJHan/Bioinformatics\\_Programming\\_101/master/059.fasta](https://raw.githubusercontent.com/KennethJHan/Bioinformatics_Programming_101/master/059.fasta)
- Issue에 정리한 내용을 바탕으로 간단하게 발표.

# Github

- Git



리눅스를 만든 리누스 토발즈가 만든  
버전 관리 시스템 (Version Control System)

- Github

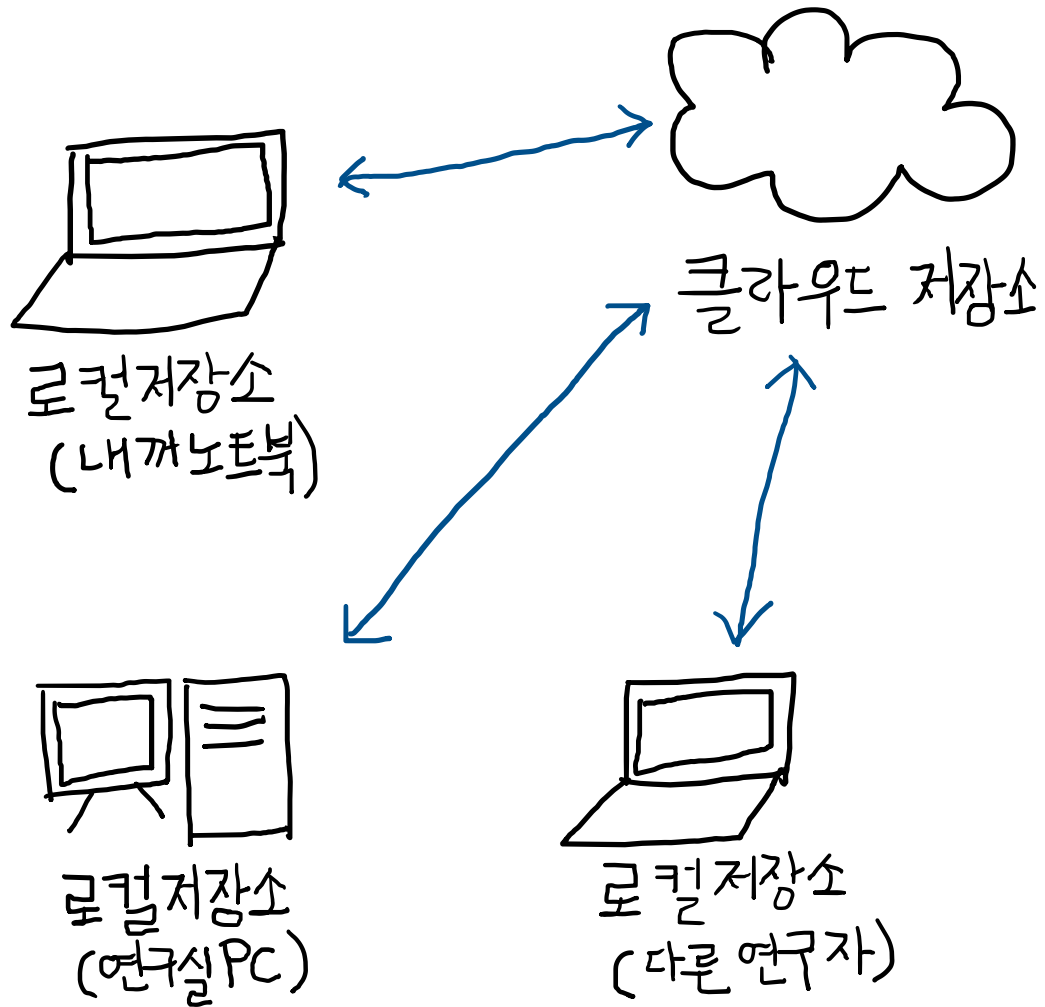
로컬에서 관리하는 Git을 클라우드 저장소에서  
관리해주는 버전 관리 시스템  
공개용 repository와 private repository를 사용할 수 있다  
2018년 MS 가 7,500,000,000 달러에 인수하였다



리누스 토발즈  
(1969.12.28 - )

 미국 USD	
	7,500,000,000 75억 달러
=	
 대한민국 KRW	
	8,756,250,000,000.00 8조 7,562억 5,000만 원

# git, github 개념



git  
로컬 저장소의 버전관리 시스템

github  
클라우드 저장소의 버전관리 시스템

git add  
파일을 등록

git commit  
파일을 저장소에 등록

git push  
파일을 클라우드 저장소에 등록

# 시작하기 전에

- git version 체크
- 리눅스

```
kenneth_jh_han@instance-1:~$ which git
/usr/bin/git
kenneth_jh_han@instance-1:~$ git --version
git version 2.17.1
kenneth_jh_han@instance-1:~$
```

- 윈도우

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\kenne>where git
C:\Program Files\Git\cmd\git.exe

C:\Users\kenne>git --version
git version 2.26.2.windows.1

C:\Users\kenne>
```

# github 가입하기

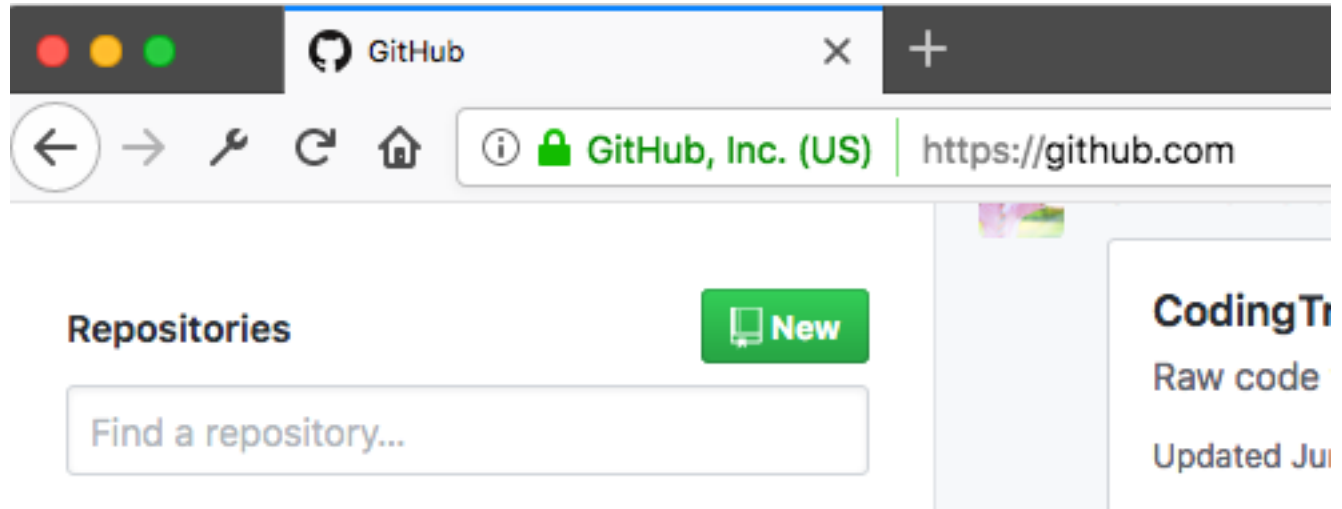
1) github 페이지에 접속합니다.

<https://github.com>

2) 회원가입을 합니다.

3) 로그인을 합니다.

4) New 버튼을 누릅니다.





# repository 만들기

5) 새로운 repository를 만듭니다.

Repository name은 test\_repository로 해봅시다.

Public, Private을 고를 수 있습니다. Public 으로 해봅시다.

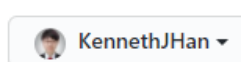
Initialize this repository with a README를 체크를 빼줍니다.

Create repository 버튼을 누릅니다.

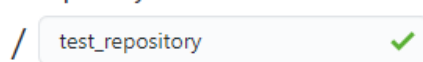
## Create a new repository

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

Owner \*



Repository name \*



Great repository names are short and memorable. Need inspiration? How about [sturdy-pancake?](#)

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

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

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

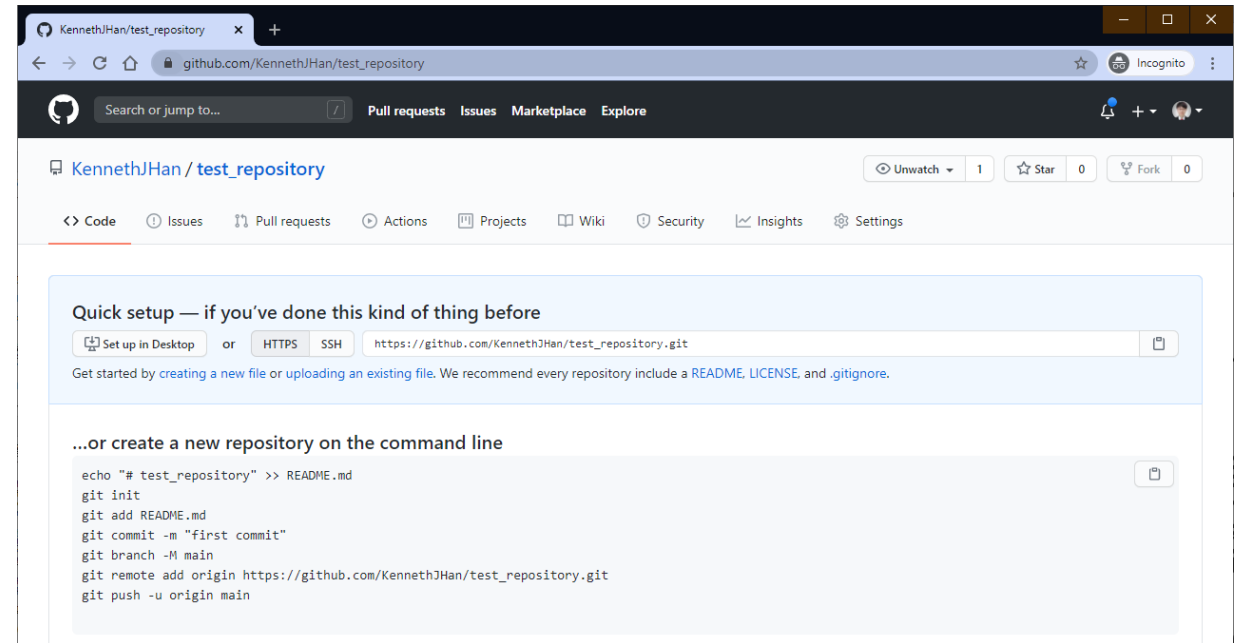
Create repository

# repository 만들기

6) repository 가 만들어졌습니다.

오른쪽 그림에서 지시하는 것 처럼  
실행해봅시다.

실행하다 보면 아래와 같이 나오는데,  
그러면 git config --global 로 환경설정을 해줍니다.



```
kenneth_jh_han@instance-1:~$ mkdir test_repository
kenneth_jh_han@instance-1:~$ cd test_repository
kenneth_jh_han@instance-1:~/test_repository$ echo "# test_repository" >> README.md
kenneth_jh_han@instance-1:~/test_repository$ git init
Initialized empty Git repository in /home/kenneth_jh_han/test_repository/.git/
kenneth_jh_han@instance-1:~/test_repository$ git add README.md
kenneth_jh_han@instance-1:~/test_repository$ git commit -m "first commit"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <kenneth_jh_han@instance-1.us-west1-b.c.lecture-276814.internal>) not allowed
kenneth_jh_han@instance-1:~/test_repository$
```

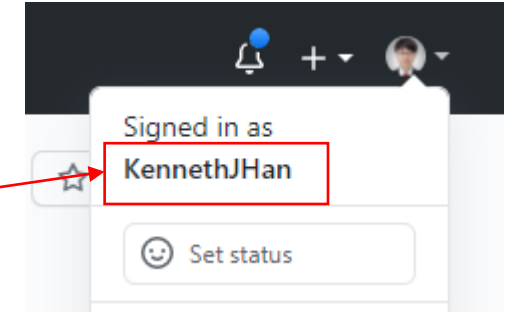
# repository 만들기

## 7) git config 설정

git config --global user.email "[여러분이 가입한 메일]"

git config --global user.name "[가입할 때 등록한 이름]"

이제 다시 commit 부분부터 진행합니다.



```
kenneth_jh_han@instance-1:~$ mkdir test_repository
kenneth_jh_han@instance-1:~$ cd test_repository
kenneth_jh_han@instance-1:~/test_repository$ echo "# test_repository" >> README.md
kenneth_jh_han@instance-1:~/test_repository$ git init
Initialized empty Git repository in /home/kenneth_jh_han/test_repository/.git/
kenneth_jh_han@instance-1:~/test_repository$ git add README.md
kenneth_jh_han@instance-1:~/test_repository$ git commit -m "first commit"

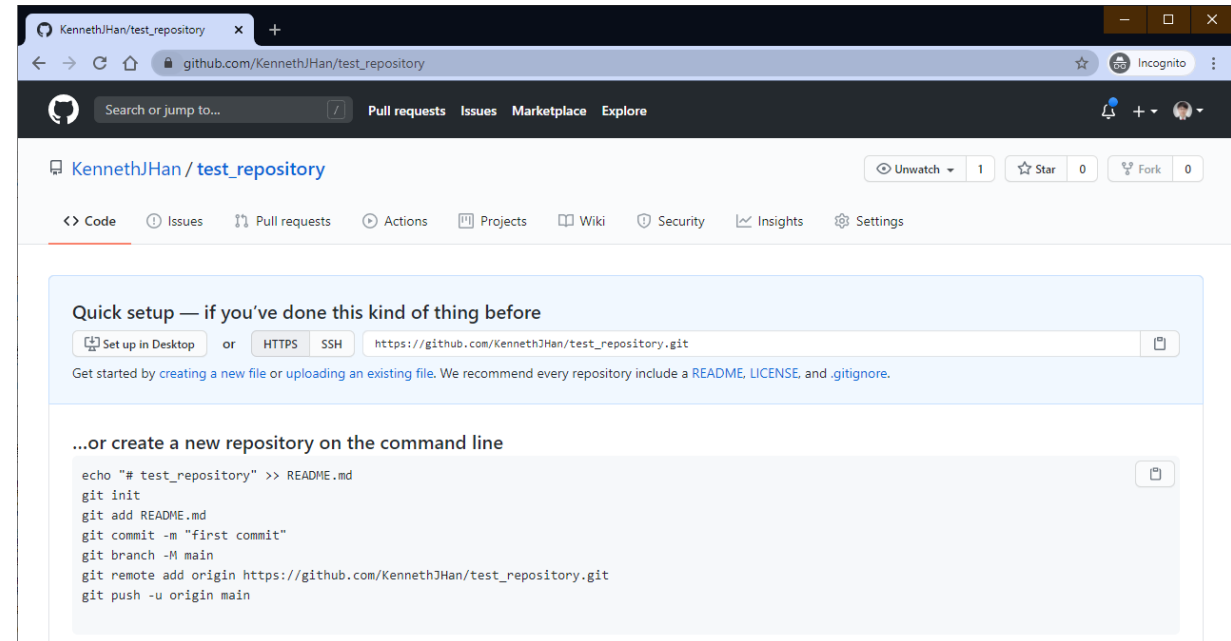
*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

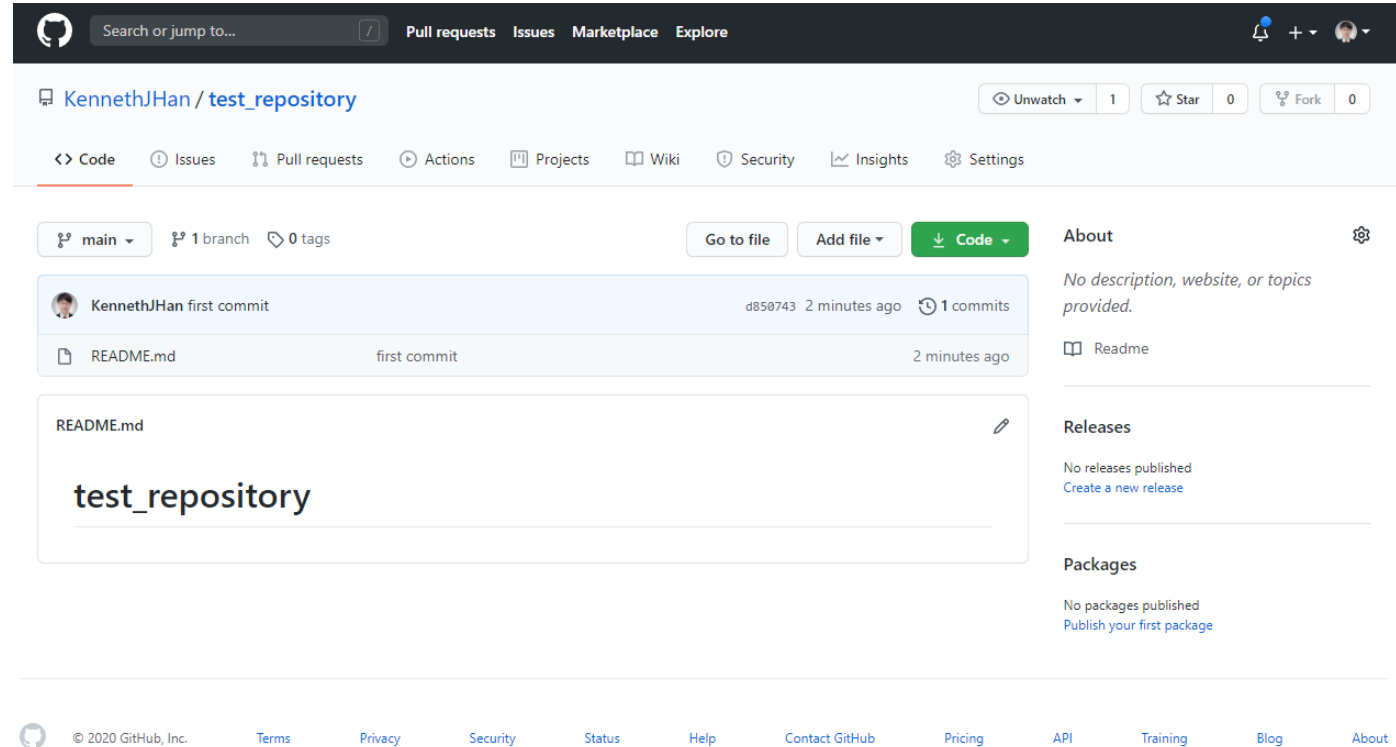
to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: empty ident name (for <kenneth_jh_han@instance-1.us-west1-b.c.lecture-276814.internal>) not allowed
kenneth_jh_han@instance-1:~/test_repository$ git config --global user.email "kenneth.jh.han@gmail.com"
kenneth_jh_han@instance-1:~/test_repository$ git config --global user.name "KennethJHan"
kenneth_jh_han@instance-1:~/test_repository$ git commit -m "first commit"
[master (root-commit) d850743] first commit
 1 file changed, 1 insertion(+)
 create mode 100644 README.md
kenneth_jh_han@instance-1:~/test_repository$ git branch -M main
kenneth_jh_han@instance-1:~/test_repository$ git remote add origin https://github.com/KennethJHan/test_repo
sitory.git
kenneth_jh_han@instance-1:~/test_repository$ git push -u origin main
Username for 'https://github.com': KennethJHan
Password for 'https://KennethJHan@github.com':
Counting objects: 3, done.
Writing objects: 100% (3/3), 231 bytes | 231.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/KennethJHan/test_repository.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
kenneth_jh_han@instance-1:~/test_repository$
```



# repository 만들기

8) 오른쪽 그림과 같이 정상적으로 생성되었는지 확인해봅니다.



# hello.py 작성

repository 디렉터리 안에 src 라는 디렉터를 만들어 그 안에서,

hello world를 출력하는 파이썬 스크립트 hello.py 를 작성해봅시다.

```
# hello.py
```

```
print("hello world")
```

```
kenneth_jh_han@instance-1:~/test_repository$ cd src
kenneth_jh_han@instance-1:~/test_repository/src$ ll
total 12
drwxrwxr-x 2 kenneth_jh_han kenneth_jh_han 4096 Oct 16 13:26 ./
drwxrwxr-x 4 kenneth_jh_han kenneth_jh_han 4096 Oct 16 13:21 ../
-rw-rw-r-- 1 kenneth_jh_han kenneth_jh_han  21 Oct 16 13:26 hello.py
kenneth_jh_han@instance-1:~/test_repository/src$ python3 hello.py
hello world
kenneth_jh_han@instance-1:~/test_repository/src$
```

# git status, add, commit, push

- git status 는 repository 내부에서 변동된 사항을 보여주는 명령어 입니다.

```
kenneth_jh_han@instance-1:~/test_repository$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

        src/

nothing added to commit but untracked files present (use "git add" to track)
kenneth_jh_han@instance-1:~/test_repository$
```

- git add 는 파일을 repository에 추가할 파일을 등록하는 명령어 입니다.  
git add [파일 이름] 을 하여 추가합니다.

```
kenneth_jh_han@instance-1:~/test_repository$ git add src
kenneth_jh_han@instance-1:~/test_repository$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   src/hello.py

kenneth_jh_han@instance-1:~/test_repository$
```

# git status, add, commit, push

- git commit -m “메시지” 를 사용하여 로컬 저장소(git)에 파일을 추가합니다.

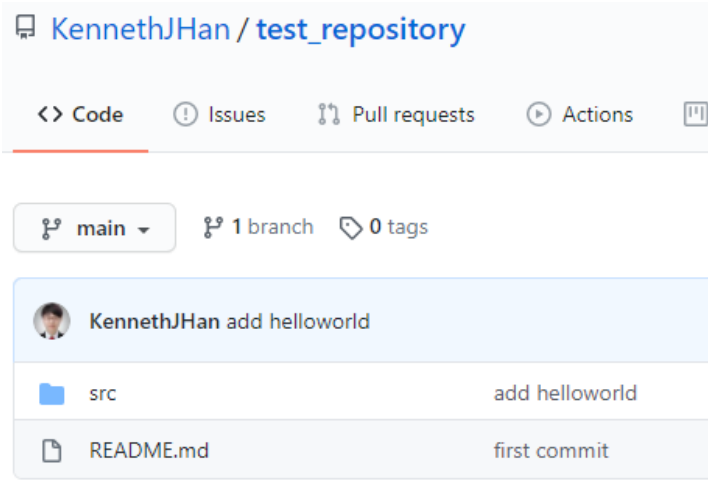
```
kenneth_jh_han@instance-1:~/test_repository$ git commit -m "add helloworld"
[main 0a41aa9] add helloworld
1 file changed, 1 insertion(+)
create mode 100644 src/hello.py
kenneth_jh_han@instance-1:~/test_repository$
```

- git push origin main 을 하여 원격 저장소(github)의 main branch에 파일을 추가합니다.

```
kenneth_jh_han@instance-1:~/test_repository$ git push origin main
Username for 'https://github.com': KennethJHan
Password for 'https://KennethJHan@github.com':
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 348 bytes | 348.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/KennethJHan/test_repository.git
   d850743..0a41aa9  main -> main
kenneth_jh_han@instance-1:~/test_repository$
```

# git status, add, commit, push

- 다음과 같이 원격 저장소에 잘 추가된 것을 확인할 수 있습니다.



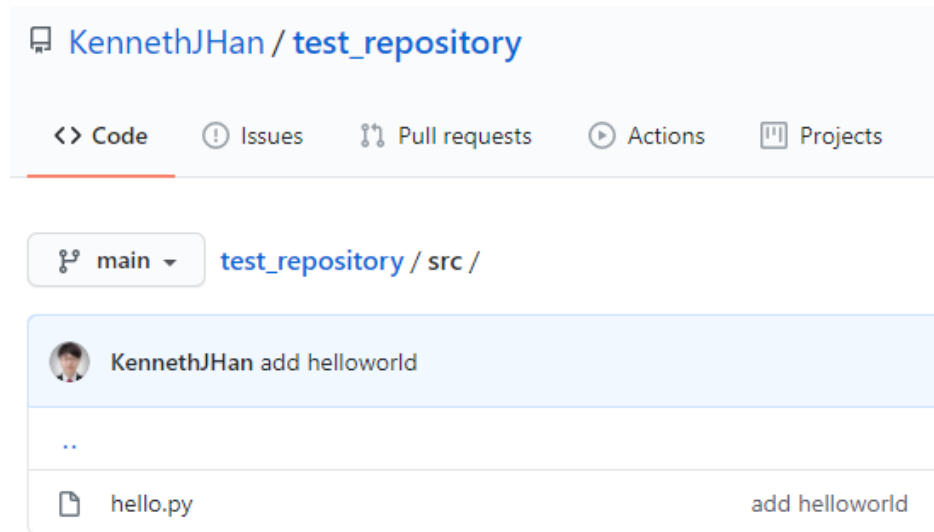
KennethJHan / test\_repository

<> Code Issues Pull requests Actions

main 1 branch 0 tags

KennethJHan add helloworld

src	add helloworld
README.md	first commit



KennethJHan / test\_repository

<> Code Issues Pull requests Actions Projects

main test\_repository / src /

KennethJHan add helloworld

..

hello.py	add helloworld
----------	----------------



KennethJHan / test\_repository

<> Code Issues Pull requests Actions Projects

main test\_repository / src / hello.py / <> Jump to

KennethJHan add helloworld

1 contributor

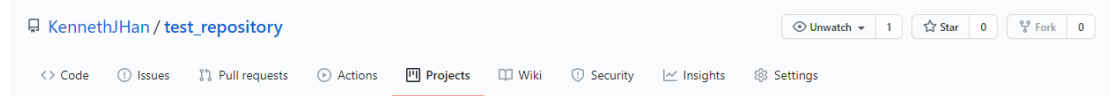
1 lines (1 sloc) | 21 Bytes

```
1 print("hello world")
```



# Project 추가하기

- Project 탭을 클릭
- Create a project 를 클릭
- Project board name을 my\_project 라고 쓰고 Create project 클릭
- Add a column 을 클릭
- To Do 라고 쓰고 Create column 을 클릭
- Add column 을 클릭
- In Progress 라고 쓰고 Create column 을 클릭
- 같은 방식으로 Done 을 만듦



## Organize your issues with project boards

Did you know you can manage projects in the same place you keep your code? Set up a project board on GitHub to streamline and automate your workflow.

### Create a new project

Coordinate, track, and update your work in one place, so projects stay transparent and on schedule.

Project board name

my\_project

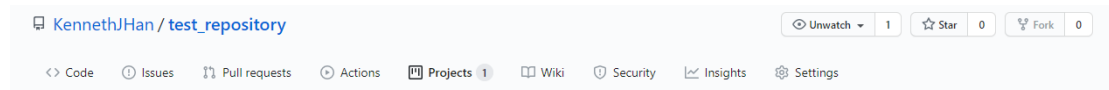
Description (optional)

Project template

Save yourself time with a pre-configured project board template.

Template: None

Create project



my\_project  
Updated now

Filter cards

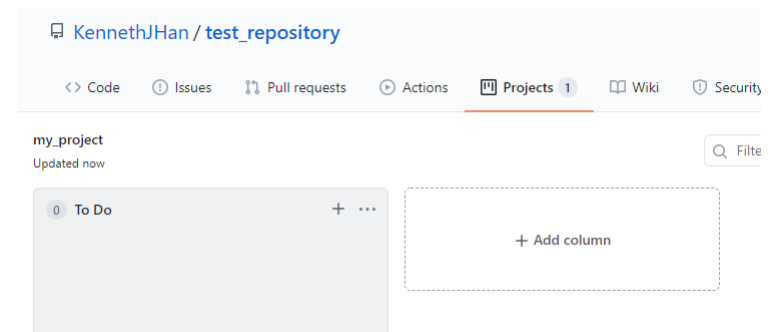
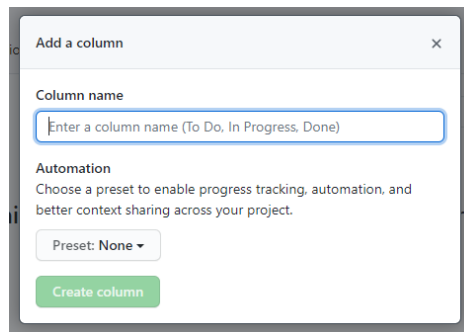
+ Add cards

Fullscreen

Menu

This project doesn't have any columns or cards.

Add a column



# Project 추가하기

KennethJHan / test\_repository

Unwatch

1

Star

0

Fork

0

Code

Issues

Pull requests

Actions

Projects 1

Wiki

Security

Insights

Settings

my\_project

Updated now

Filter cards

+ Add cards

Fullscreen

Menu

0 To Do

+ ...

0 In Progress

+ ...

0 Done

+ ...

+ Add column

- 완성된 모습
- 여기에 Issue를 등록하여 card를 넣을 수 있습니다.

# Issue 등록하기

- Issue 탭 클릭
- New Issue 클릭
- 오른쪽과 같이 쓴다
- Assignees 는 자신을 등록한다.
- Projects 는 my\_project를 지정한다.
- Submit new issue를 클릭.

The screenshot displays the GitHub 'Issues' tab for a repository. At the top, a navigation bar includes links for Code, Issues (selected), Pull requests, Actions, Projects (1), Wiki, Security, Insights, and Settings. Below this, a notification banner states: 'Label issues and pull requests for new contributors. Now, GitHub will help potential first-time contributors discover issues labeled with good first issue.' A search bar shows the filter 'is:issue is:open'. To the right of the search bar are buttons for 'Labels 9' and 'Milestones 0', and a green 'New issue' button.

The main content area shows the 'New Issue' form. The title field contains 'hello world 수정'. Below the title are 'Write' and 'Preview' tabs. The 'Write' tab is active, showing a text area with the following content:

```
# hello world 수정
## hello.py 에 사용자로 부터 값을 입력 받아 hello [사용자 값] 을 출력하도록 수정

sys 모듈을 사용한다.

```python
import sys
```
```


Below the text area is a note: 'Attach files by dragging & dropping, selecting or pasting them.' and a small 'MD' icon. At the bottom of the form is a green 'Submit new issue' button.

On the right side of the form, there are sections for 'Assignees' (showing 'KennethJHan'), 'Labels' (showing 'None yet'), and 'Projects' (showing a list of projects with 'my\_project' selected). At the bottom of the page, a footer note states: 'Remember, contributions to this repository should follow our GitHub Community Guidelines.'

# Issue 등록하기



- 다음과 같이 이슈가 완성됨.
- 다시 Project에 들어가서,  
등록된 issue를 끌어서 To Do에  
넣는다.

hello world 수정 #1

 Open KennethJHan opened this issue now · 0 comments



KennethJHan commented now

Owner  

## hello world 수정

hello.py 에 사용자로 부터 값을 입력 받아 hello [사용자 값] 을 출력하도록 수정

sys 모듈을 사용한다.









```
import sys
```



KennethJHan self-assigned this now

 KennethJHan / test\_repository

 Unwatch 1  Star 0  Fork 0

 Code  Issues 1  Pull requests  Actions  Projects 1  Wiki  Security  Insights  Settings

my\_project

Updated 6 minutes ago

 Filter cards

< + Add cards



is:open

You can use the filters available in [issue search](#).

Triage (1) Hide










 hello world 수정

#1 opened by KennethJHan



Search results

 KennethJHan / test\_repository

 Code  Issues 1  Pull requests  Actions  Projects 1  Wiki  Security  Insights  Settings

my\_project

Updated 6 minutes ago

 Filter cards

1 To Do

+ ...

 hello world 수정

#1 opened by KennethJHan



0 In Progress

+ ...

0 Done

+ ...

# git branch 생성

- git checkout -b “[branch 이름]” 로 branch 를 생성한다.  
branch 로 잘 이동되었는지 확인한다.

```
kenneth_jh_han@instance-1:~/test_repository$ git checkout -b "issue-#1-modify-helloworld"  
Switched to a new branch 'issue-#1-modify-helloworld'  
kenneth_jh_han@instance-1:~/test_repository$
```

- 다음과 같이 hello.py 를 수정하고 git status 로 확인해본다.

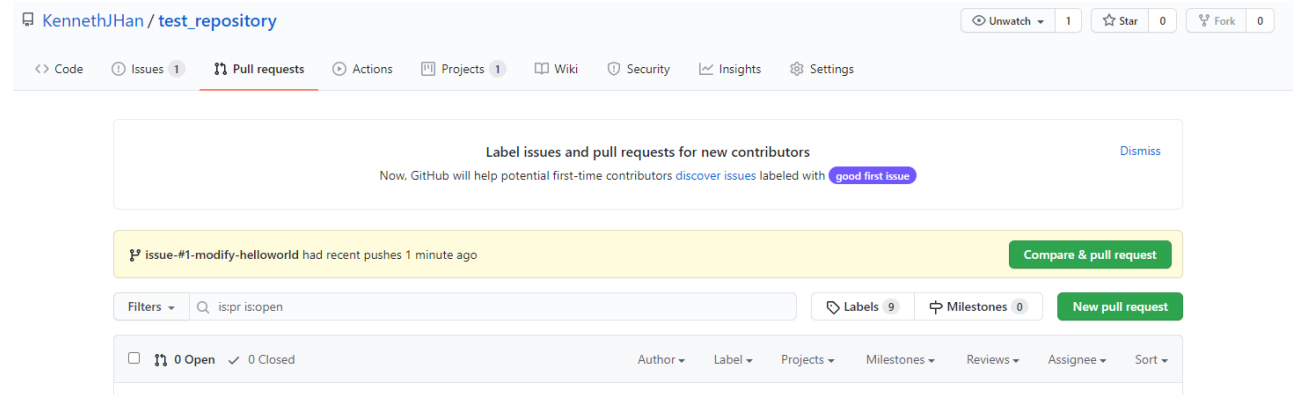
```
kenneth_jh_han@instance-1:~/test_repository$ cd src  
kenneth_jh_han@instance-1:~/test_repository/src$ ll  
total 12  
drwxrwxr-x 2 kenneth_jh_han kenneth_jh_han 4096 Oct 16 13:26 ./  
drwxrwxr-x 4 kenneth_jh_han kenneth_jh_han 4096 Oct 16 13:21 ../  
-rw-rw-r-- 1 kenneth_jh_han kenneth_jh_han  21 Oct 16 13:26 hello.py  
kenneth_jh_han@instance-1:~/test_repository/src$ vi hello.py  
kenneth_jh_han@instance-1:~/test_repository/src$ cat hello.py  
import sys  
  
print("hello", sys.argv[1])  
  
kenneth_jh_han@instance-1:~/test_repository/src$ git status  
On branch issue-#1-modify-helloworld  
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")  
kenneth_jh_han@instance-1:~/test_repository/src$
```

# git branch 에서 작업하기

```
kenneth_jh_han@instance-1:~/test_repository/src$ git status
On branch issue-#1-modify-helloworld
nothing to commit, working tree clean
kenneth_jh_han@instance-1:~/test_repository/src$ git push origin issue-#1-modify-helloworld
Username for 'https://github.com': KennethJHan
Password for 'https://KennethJHan@github.com':
Counting objects: 4, done.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 379 bytes | 379.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
remote:
remote: Create a pull request for 'issue-#1-modify-helloworld' on GitHub by visiting:
remote:   https://github.com/KennethJHan/test_repository/pull/new/issue-%231-modify-hello
remote:   world
remote:
To https://github.com/KennethJHan/test_repository.git
 * [new branch]      issue-#1-modify-helloworld -> issue-#1-modify-helloworld
kenneth_jh_han@instance-1:~/test_repository/src$
```

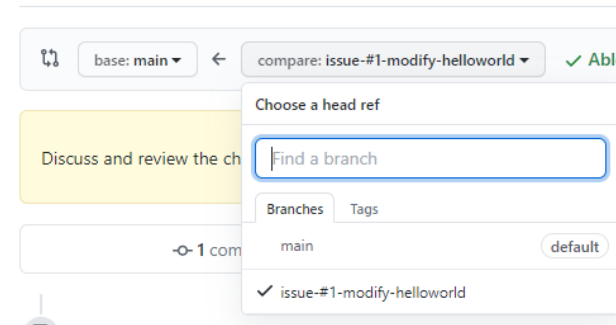
# github pull request

- Pull requests 탭에 들어감
- New pull request 를 클릭
- 생성했던 branch 를 클릭
- 바뀐 내용을 확인
- 삭제된 내용은 빨간색으로 표기됨
- 추가된 내용은 초록색으로 표기됨
- Create pull request 를 클릭



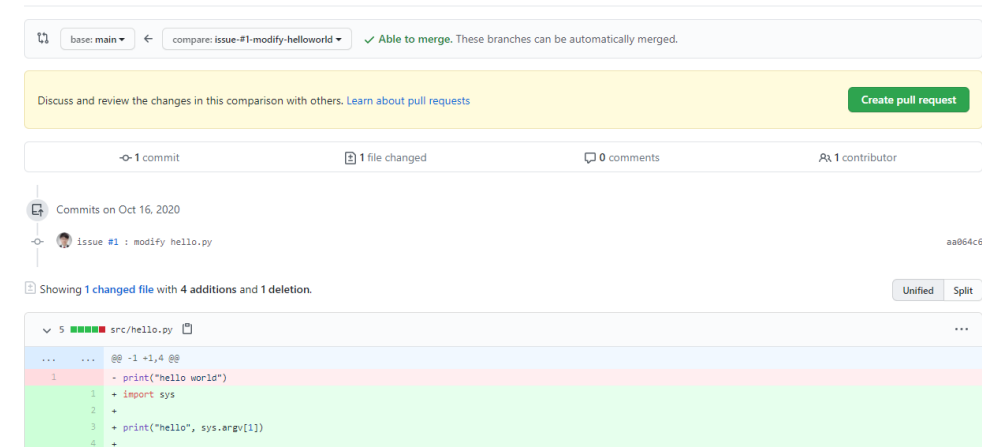
## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If y



## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



# github pull request

pull request 에 대한 내용을 씀

Assignees 를 자신으로 지정

Projects 를 지정

Create pull requests 를 클릭

Merge pull request 를 클릭

Confirm merge 를 클릭

Delete branch 를 클릭

## 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: main ← compare: issue-#1-modify-helloworld ✓ Able to merge. These branches can be automatically merged.

issue #1 : modify hello.py

Write Preview

H B I ≡ <> 🔗 ≡ ≡ ☑ @ 📎 ↶

hello.py 를 수정

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

Create pull request

Reviewers

No reviews

Assignees

KennethJHan

Labels

None yet

Projects

my\_project

Milestone

No milestone

Linked issues

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

## issue #1 : modify hello.py #2

Open KennethJHan wants to merge 1 commit into main from issue-#1-modify-helloworld

Conversation 0

Commits 1

Checks 0

Files changed 1

KennethJHan commented now

hello.py 를 수정

issue #1 : modify hello.py

aa064c6

KennethJHan self-assigned this now

Add more commits by pushing to the issue-#1-modify-helloworld branch on KennethJHan/test\_repository.

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 You can also open this in GitHub Desktop or view [command line instructions](#).

## issue #1 : modify hello.py #2

Open KennethJHan wants to merge 1 commit into main from issue-#1-modify-helloworld

Conversation 0

Commits 1

Checks 0

Files changed 1

KennethJHan commented now

hello.py 를 수정

issue #1 : modify hello.py

KennethJHan self-assigned this now

Add more commits by pushing to the issue-#1-modify-helloworld branch on KennethJHan/test\_repository.

Merge pull request #2 from KennethJHan/issue-#1-modify-helloworld

issue #1 : modify hello.py

Confirm merge Cancel

## issue #1 : modify hello.py #2

Merged KennethJHan merged 1 commit into main from issue-#1-modify-helloworld now

Conversation 0

Commits 1

Checks 0

Files changed 1

KennethJHan commented 1 minute ago

hello.py 를 수정

issue #1 : modify hello.py

aa064c6

KennethJHan self-assigned this 1 minute ago

KennethJHan merged commit f547a7c into main now

Revert

Pull request successfully merged and closed

You're all set—the issue-#1-modify-hell branch can be safely deleted.

Delete branch



# git pull

- git checkout main 을 타이핑 하여 main branch 로 돌아감  
로컬 저장소의 main branch 에 있는 hello.py는  
원격 저장소의 hello.py 보다 버전이 느림

```
kenneth_jh_han@instance-1:~/test_repository/src$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
kenneth_jh_han@instance-1:~/test_repository/src$ ll
total 12
drwxrwxr-x 2 kenneth_jh_han kenneth_jh_han 4096 Oct 16 14:02 ./
drwxrwxr-x 4 kenneth_jh_han kenneth_jh_han 4096 Oct 16 13:21 ../
-rw-rw-r-- 1 kenneth_jh_han kenneth_jh_han 21 Oct 16 14:02 hello.py
kenneth_jh_han@instance-1:~/test_repository/src$ cat hello.py
print("hello world")
kenneth_jh_han@instance-1:~/test_repository/src$
```

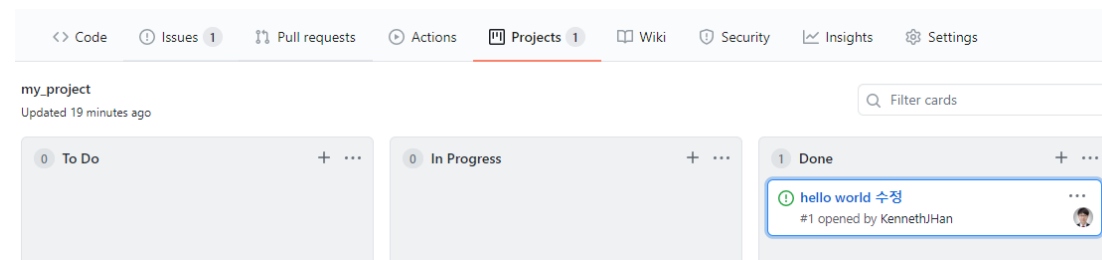
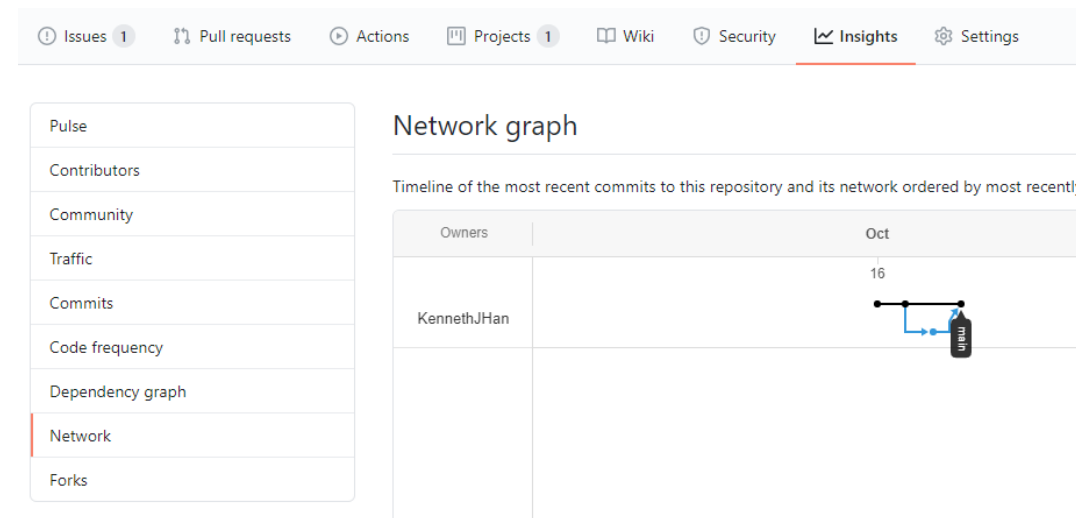
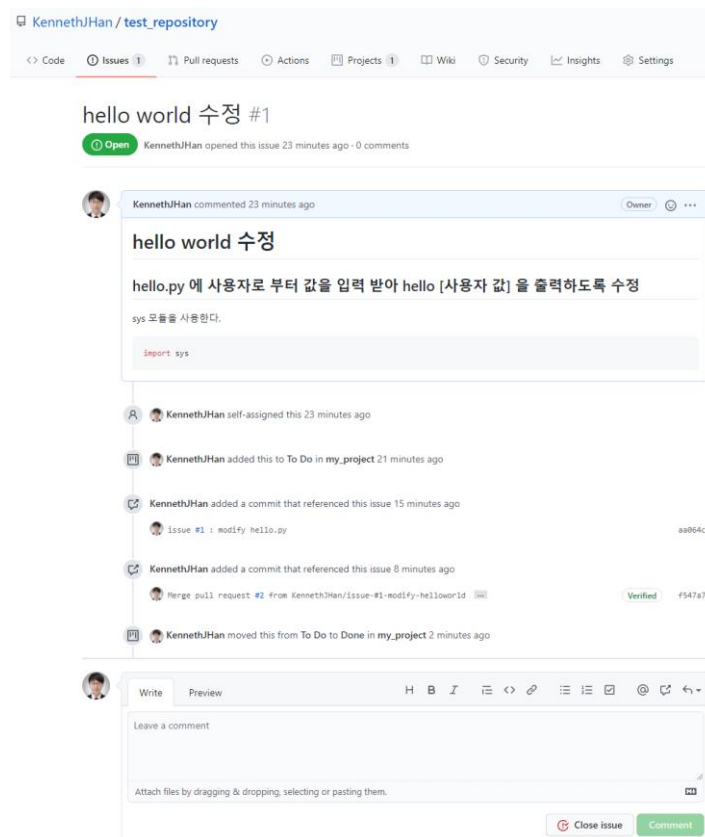
- 원격 저장소의 파일을 받아와야 함
- git pull origin main 으로 원격 저장소의 파일을 받아 옴

```
kenneth_jh_han@instance-1:~/test_repository/src$ git pull origin main
remote: Enumerating objects: 1, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 1 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (1/1), done.
From https://github.com/KennethJHan/test_repository
* branch          main          -> FETCH_HEAD
   0a41aa9..f547a7c main          -> origin/main
Updating 0a41aa9..f547a7c
Fast-forward
   &rfzMel&happye&,54+insertions(+), 1 deletion(-)
kenneth_jh_han@instance-1:~/test_repository/src$ cat hello.py
import sys

print("hello", sys.argv[1])
kenneth_jh_han@instance-1:~/test_repository/src$
```

# 작업 정리

- Insights 탭에 Network 를 클릭하면 branch 의 관계를 확인할 수 있음
- Projects 에 들어가서 issue를 Done으로 넘김
- Issues 에 들어가서 진행했던 issue를 close 해줌



# 과제

- github 에 homework\_1 라고 하는 repository 를 생성한다.
- projec를 생성 후 과제를 issue로 등록한다.
- issue에 과제 수행한 내용을 정리한다.
- 다음 파일을 다운로드 받아서 옴기서열을 세는 파이썬 스크립트를 작성한다.
- [https://raw.githubusercontent.com/KennethJHan/Bioinformatics\\_Programming\\_101/master/059.fasta](https://raw.githubusercontent.com/KennethJHan/Bioinformatics_Programming_101/master/059.fasta)
- Issue에 정리한 내용을 바탕으로 간단하게 발표.