

수고하셨습니다~!

✓ main

default

- feat/bigwave100-main
- feat/hwlee-main
- feat/hyeonsang010716-main
- feat/jungsb0415-main
- feat/pinkgorae-main
- feat/wjm9765-main
- feat/xxxiv-34-main
- feat/yjs2673-main

[View all branches](#)

8 Open ✓ 0 Closed

- [FEAT] main 생성 #8 opened last week by hyeonwlee • Review required
- Feat/bigwave100 main #7 opened last week by bigwave100 • Review required
- [FEAT] create main.py #6 opened last week by jungsb0415 • Review required
- [FEAT] main 생성 #5 opened last week by pinkgorae • Review required
- [FEAT] main 생성 #4 opened last week by yjs2673 • Review required
- [FEAT] main 생성 #3 opened last week by xxxiv-34 • Review required
- [FEAT] main 생성 #2 opened last week by wjm9765 • Review required
- [FEAT] main 생성 #1 opened last week by hyeonsang010716 • Review required

모두 2주차 실습하시느라 고생하셨어요~!



지금까지 우리가 배운 것

- 리모트와 로컬 GitHub를 연결하고, 로컬에서 작업한 내용을 리모트 GitHub에 업로드하는 방법을 배웠음.
 - add
 - commit
 - push
- 협업 시 충돌을 방지하기 위해 브런치를 활용하여 병렬로 작업할 수 있도록 하고, Pull Request에 대해 배우며 협업 흐름을 익혔음.



이번 시간에 배울 것은

- 리모트와 로컬 GitHub를 연결하고, 로컬에서 작업한 내용을 GitHub에 업로드하는 방법을 배웠음.
 - add
 - commit
 - push
- 협업 시 충돌을 방지하기 위해 브런치를 활용하여 병렬로 작업할 수 있도록 하고, Pull Request에 대해 배우며 협업 흐름을 익혔음.
- 충돌이 발생했을 때...!



스프린트가 끝난 후, 해야 되는 작업

스프린트: 목표 했던 기능 구현이 끝난 상태

<현재 우리의 로컬 깃헙 상태>

- 각자의 브런치로 이동되어 있음.

현재 로컬이 어디 브런치에 있는지 확인하는 방법

git branch

```
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
  * feat/hyeonsang010716-main
    main
○ hyeonsang@johyeonsang-ui-MacBookPro git-study % █
```

스프린트가 끝난 후, 해야 되는 작업

<현재 우리의 리모트 깃헙 상태>

- 책임(팀장) 개발자가 풀리퀘를 승인해서, main 브런치로 머지한 상태

Git-Study Public

main 9 Branches 0 Tags

Go to file Add file Code

hyeonsang010716 Merge pull request #1 from SGCS-Release-Git-Project/feat/hyeonsang0... f78fa4d · now 45 Commits

| Author | Commit Message | Date |
|-----------------|-----------------------------|-----------|
| HwangInseok | [MOD] 디렉토리 위치 수정 | last week |
| bigwave100 | [FEAT] main 생성 | last week |
| docs | [ADD] Week 2 study PDF file | last week |
| hwlee | [FEAT] main 생성 | last week |
| hyeonsang010716 | [FEAT] main 생성 | last week |
| jungsb0415 | [FEAT] create main.py | last week |
| kyuhung | [FEAT] kyuhung README 생성 | last week |
| pinkgorae | [FEAT] main 생성 | last week |
| wjm9765 | [FEAT] main 생성 | last week |
| xxxiv-34 | [FEAT] main 생성 | last week |
| yjs2673 | [FEAT] main 생성 | last week |

스프린트가 끝난 후, 해야 되는 작업

<다음 스프린트를 위해 우리가 해야 할 일>

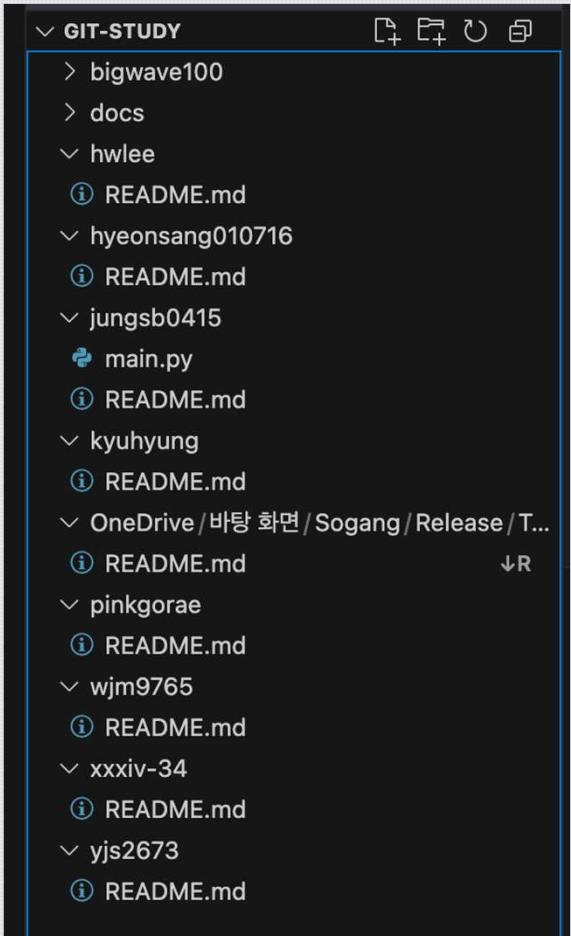
- 로컬에 있는 내 브런치를 main으로 이동하자.

git checkout main

```
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
 * feat/hyeonsang010716-main
   main
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git checkout main
Switched to branch 'main'
Your branch is behind 'origin/main' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
   feat/hyeonsang010716-main
 * main
○ hyeonsang@johyeonsang-ui-MacBookPro git-study % █
```

스프린트가 끝난 후, 해야 되는 작업

어라? main으로 이동했지만, 만들어둔 main.py가 없음.



```
└─ GIT-STUDY
    ├─ bigwave100
    ├─ docs
    └─ hwlee
        ├─ README.md
        └─ hyeonsang010716
            ├─ README.md
            └─ jungsbo415
                └─ main.py
        └─ README.md
    └─ kyuhyung
        └─ README.md
    └─ OneDrive / 바탕 화면 / Sogang / Release / T...
        └─ README.md
    └─ pinkgorae
        └─ README.md
    └─ wjm9765
        └─ README.md
    └─ xxxiv-34
        └─ README.md
    └─ yjs2673
        └─ README.md
```

스프린트가 끝난 후, 해야 되는 작업

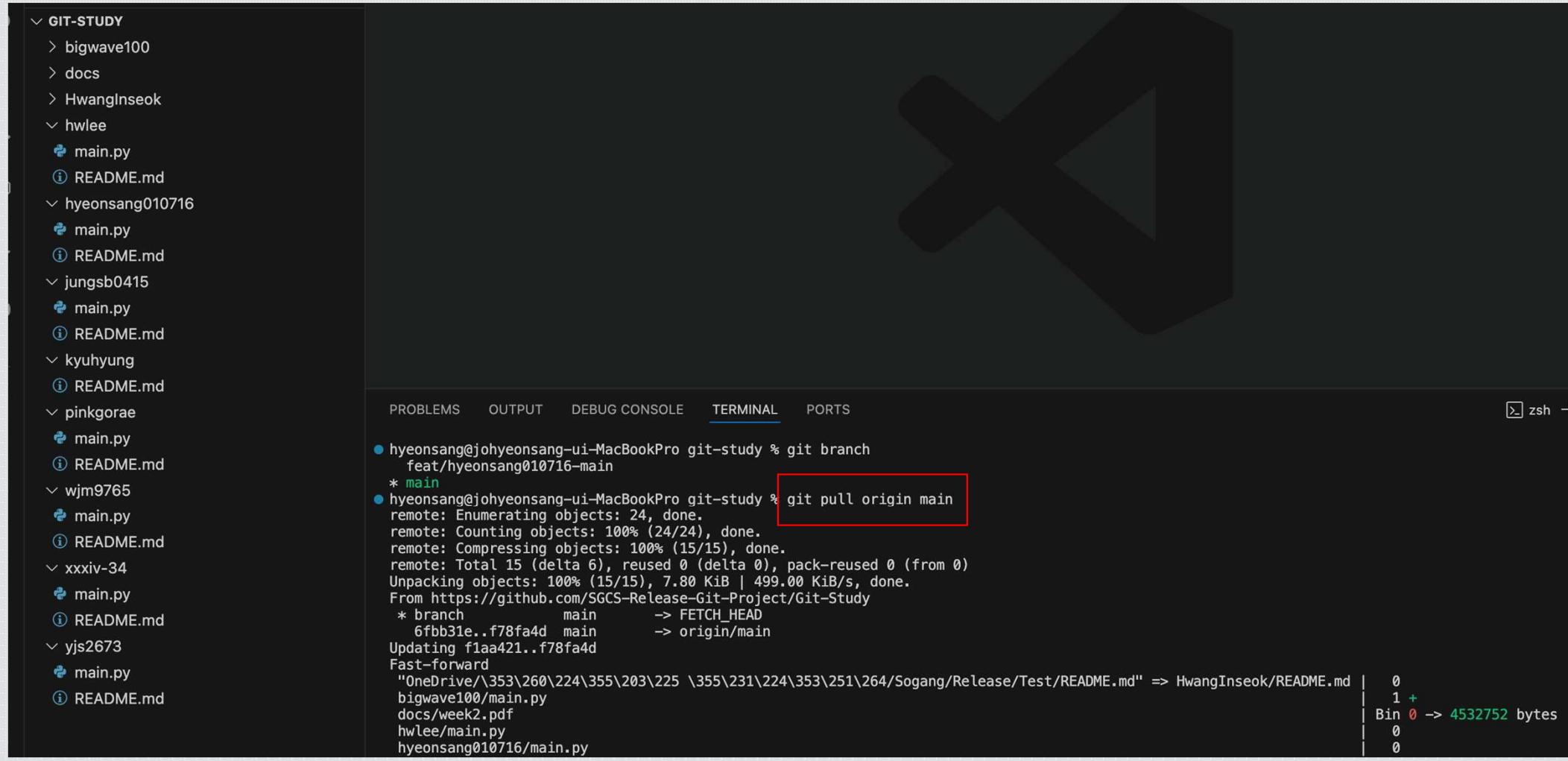
<주의할 점>

리모트에서 업데이트 된 사항은 로컬에 **자동** 반영이 안 됨!
반영하기 위해선 **pull**을 해야 함!

```
hyeonsang@johyeonsang-ui-MacBookPro git-study % git pull origin main
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 15 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (15/15), 7.80 KiB | 499.00 KiB/s, done.
From https://github.com/SGCS-Release-Git-Project/Git-Study
 * branch      main      -> FETCH_HEAD
   6ffb31e..f78fa4d  main      -> origin/main
Updating f1aa421..f78fa4d
Fast-forward
 "OneDrive\353\260\224\355\203\225 \355\231\224\353\251\264\Sogang\Release\Test\README.md" => HwangInseok/README.md |  0
 bigwave100/main.py          1 +
 docs/week2.pdf              0
 hwlee/main.py               0
 hyeonsang010716/main.py     4 +++-
 jungsb0415/main.py          0
 pinkgorae/main.py          0
 wjm9765/main.py             1 +
 xxxiv-34/main.py           1 +
 yjs2673/main.py             1 +
 10 files changed, 6 insertions(+), 1 deletion(-)
 rename "OneDrive\353\260\224\355\203\225 \355\231\224\353\251\264\Sogang\Release\Test\README.md" => HwangInseok/README.md (100%)
 create mode 100644 bigwave100/main.py
```

스프린트가 끝난 후, 해야 되는 작업

git pull origin main
업데이트 완료!



The screenshot shows a terminal window with a dark background and light-colored text. On the left, there is a file tree view of a directory named 'GIT-STUDY'. The terminal tab is active, and the command 'git pull origin main' is highlighted with a red rectangle. The output of the command is displayed below:

```
hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
  * main
hyeonsang@johyeonsang-ui-MacBookPro git-study % git pull origin main
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (15/15), done.
remote: Total 15 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (15/15), 7.80 KiB | 499.00 KiB/s, done.
From https://github.com/SGCS-Release-Git-Project/Git-Study
 * branch      main      -> FETCH_HEAD
   6fbb31e..f78fa4d main      -> origin/main
Updating f1aa421..f78fa4d
Fast-forward
  "OneDrive\353\260\224\355\203\225 \355\231\224\353\251\264\Sogang\Release\Test\README.md" => HwangInseok/README.md | 0
  bigwave100/main.py | 1 +
  docs/week2.pdf | 0
  hwlee/main.py | 0
  hyeonsang010716/main.py | 0
Bin 0 -> 4532752 bytes | 0
| 0
| 0
```

스프린트가 끝난 후, 해야 되는 작업

유지 보수적인 측면에서..

2주차 과제를 하면서, 만들었던 브랜치를 삭제 해야 함.

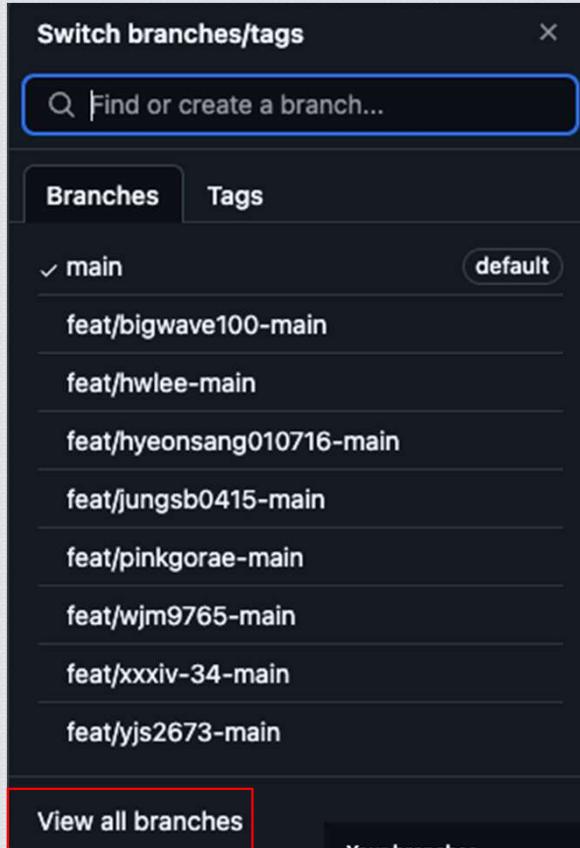
git branch -d 브랜치이름

Ex) git branch -d feat/hyeonsang010716-main

```
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
  feat/hyeonsang010716-main
* main
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch -d feat/hyeonsang010716-main
Deleted branch feat/hyeonsang010716-main (was 4a946ab).
● hyeonsang@johyeonsang-ui-MacBookPro git-study % git branch
* main
○ hyeonsang@johyeonsang-ui-MacBookPro git-study % █
```

스프린트가 끝난 후, 해야 되는 작업

리모트에서 브런치 삭제하는 방법



리모트 브런치는 풀리퀘 머지 후, 보통 책임(팀장) 개발자가 삭제 함.

여기까지 완료하면 스프린트 후, 작업 끝!

Your branches

| Branch | Updated | Check status | Behind | Ahead | Pull request |
|---------------------------|-----------|--------------|--------|-------|--------------|
| feat/hyeonsang010716-main | last week | 22 0 | #1 | ... | |

충돌이 났을 경우

항상 충돌이 안 나면 좋겠지만, 언제 어느 상황에서 충돌이 발생할지 모른다..
한 개의 브런치 안에서 한 명의 개발자가 안 할 때도 있음.

```
④ hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push  
To https://github.com/hyeonsang010716/test-repo.git  
! [rejected]          main -> main (fetch first)  
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'  
hint: Updates were rejected because the remote contains work that you do  
hint: not have locally. This is usually caused by another repository pushing  
hint: to the same ref. You may want to first integrate the remote changes  
hint: (e.g., 'git pull ...') before pushing again.  
hint: See the 'Note about fast-forwards' in 'git push --help' for details.  
○ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

언제나 보기 싫은 에러..

충돌이 났을 경우 – 1(히스토리 충돌)

<상황>

A개발자와 B개발자가 같은 브런치에서 작업을 하고 있음.

A개발자가 푸쉬를 해서, 리모트 깃헙이 최신화 되었지만 B개발자는 변경 사실을 모름.

B개발자가 pull 하지 않고, push를 진행 함.

| Name | Last commit message | Last commit date |
|-----------|---|------------------|
| README.md | Update README.md | 7 minutes ago |
| func.py | [FEAT] 1+1=2 기능 구현 | last week |
| hello.py | Create hello.py | now |
| main.py | A개발자가 푸쉬한 소스 코드 [FEAT] hello world 기능 구현 | last week |

충돌이 났을 경우 – 1(히스토리 충돌)

<상황>

A개발자와 B개발자가 같은 브런치에서 작업을 하고 있음.

A개발자가 푸쉬를 해서, 리모트 깃헙이 최신화 되었지만 B개발자는 변경 사실을 모름.

B개발자가 pull 하지 않고, push를 진행 함.

The screenshot shows a terminal window with the following output:

```
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add world.py
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[FEAT] world 기능 구현"
[main 4d986e8] [FEAT] world 기능 구현
 1 file changed, 1 insertion(+)
   create mode 100644 world.py
④ hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
  ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
○ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

The last command, `git push`, is highlighted with a red box. The error message indicates that the push was rejected because the remote repository contains work that the local repository does not have, specifically regarding the `main` branch.

깃헙 히스토리가 최신이 아니기 때문에, 충돌 발생!

충돌이 났을 경우 – 1(히스토리 충돌)

아래와 같은 상황임!



깃헙 히스토리 예제

Code Issues Pull requests Actions Projects Security Insights Settings

test-repo · Private

main · 2 Branches · 0 Tags

hyeonsang090716 [MOD] README 수정 · Edit · now · 7 Commits

[MOD] README 수정 · now

README

test-repo

A 개발자

히스토리

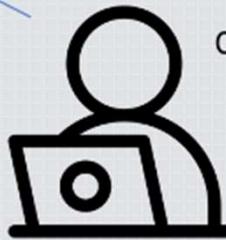
[MOD] 수정1
[MOD] README
수정

B

A



A개발자



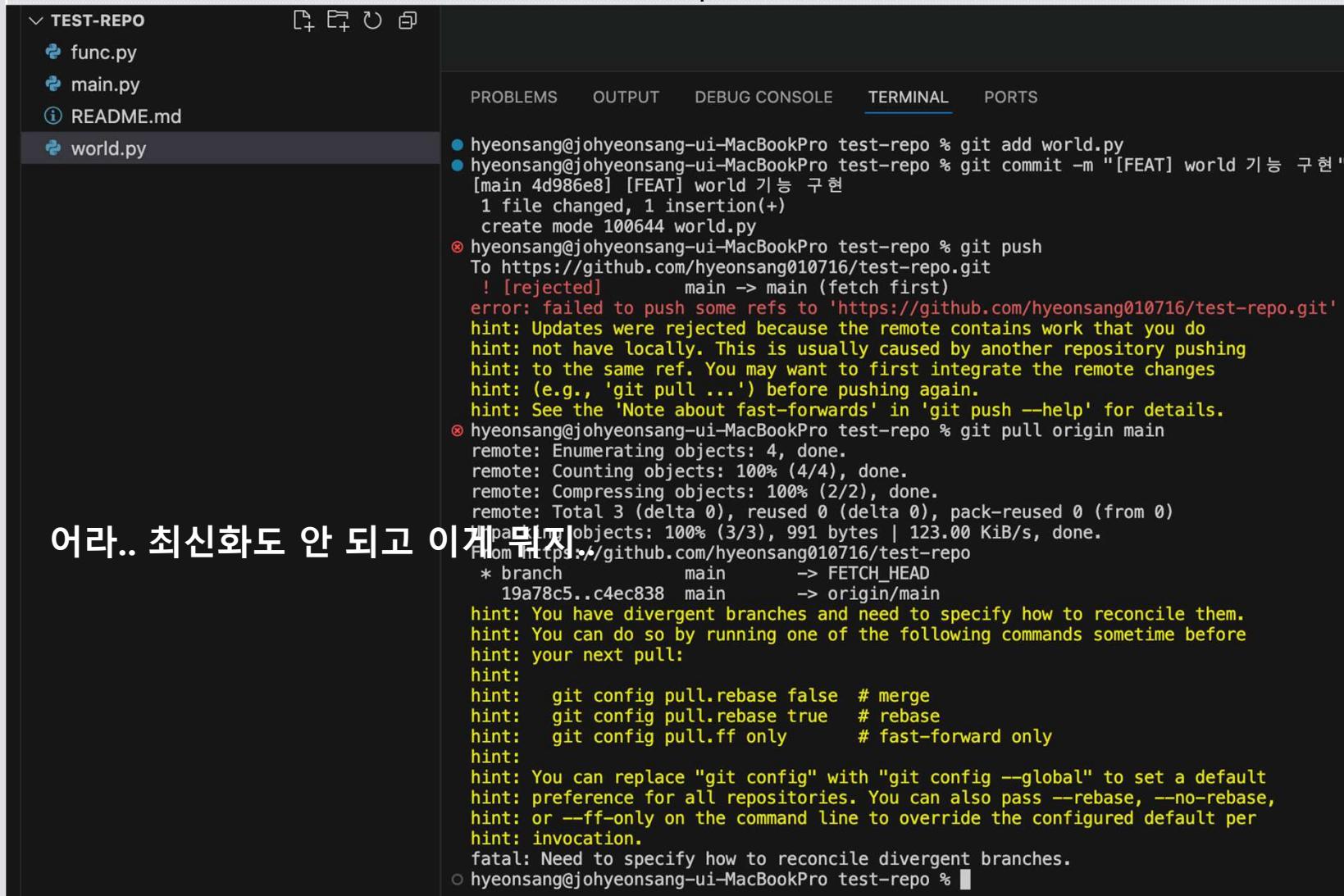
B개발자

<시나리오>
B 개발자가 리모트를 수정해서 푸쉬 함.
하지만 충돌!

이유) 최신 히스토리가 아니라서!

충돌이 났을 경우 – 1(히스토리 충돌)

먼저 히스토리를 최신화 시키기 위해서, pull 작업을 진행하려고 함.



The screenshot shows a terminal window within a code editor interface. The left sidebar displays a file tree for a repository named 'TEST-REPO' containing files func.py, main.py, README.md, and world.py. The 'world.py' file is currently selected. The terminal tab is active at the bottom of the window. The output of the terminal shows the following sequence of commands and their results:

```
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add world.py
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[FEAT] world 기능 구현"
[main 4d986e8] [FEAT] world 기능 구현
  1 file changed, 1 insertion(+)
   create mode 100644 world.py
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Delta compression objects: 100% (3/3), 991 bytes | 123.00 KiB/s, done.
From https://github.com/hyeonsang010716/test-repo
 * branch            main      -> FETCH_HEAD
   19a78c5..c4ec838  main      -> origin/main
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:  git config pull.rebase false # merge
hint:  git config pull.rebase true  # rebase
hint:  git config pull.ff only    # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

어라.. 최신화도 안 되고 이게 뭐지?

충돌이 났을 경우 – 1(히스토리 충돌)

아래와 같은 에러가 발생한 이유: git pull을 했을 때, 히스토리 정리 조건을 선택하지 않아서 임.
(만약 아래 사진처럼 안 나온다면, git config 등록을 이전에 미리한 상태)

히스토리 충돌이 났을 경우, 2가지 히스토리 정리 조건이 있음.

1. git config pull.rebase false # merge
2. git config pull.rebase true # rebase

한 번만 설정 해주면, 다음부터 히스토리 충돌이 났을 경우 아래 hint가 발생하지 않음

```
hint: You have divergent branches and need to specify how to reconcile them.
hint: You can do so by running one of the following commands sometime before
hint: your next pull:
hint:
hint:   git config pull.rebase false # merge
hint:   git config pull.rebase true  # rebase
hint:   git config pull.ff only     # fast-forward only
hint:
hint: You can replace "git config" with "git config --global" to set a default
hint: preference for all repositories. You can also pass --rebase, --no-rebase,
hint: or --ff-only on the command line to override the configured default per
hint: invocation.
fatal: Need to specify how to reconcile divergent branches.
```

충돌이 났을 경우 – 1(히스토리 충돌)

먼저 git config pull.rebase false # merge

로컬 깃헙 히스토리 상황: A

리모트 깃헙 히스토리 상황: B

A + B 를 합쳐서 새로운 C 히스토리를 만들어서 정리하겠다.

```
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git config pull.rebase false
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
From https://github.com/hyeonsang010716/test-repo
 * branch            main      -> FETCH_HEAD
Merge made by the 'ort' strategy.
\ hello.py | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 hello.py
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 10 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5), 558 bytes | 558.00 KiB/s, done.
Total 5 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/hyeonsang010716/test-repo.git
  c4ec838..a2bdcd1 main -> main
hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

충돌이 났을 경우 – 1(히스토리 충돌)

먼저 git config pull.rebase false # merge

로컬 깃헙 히스토리 상황: A

리모트 깃헙 히스토리 상황: B

A + B 를 합쳐서 새로운 C 히스토리를 만들어서 정리하겠다.

```
Merge branch 'main' of https://github.com/hyeonsang010716/test-repo  
hyeonsang010716 committed 4 minutes ago  
[FEAT] world 기능 구현  
hyeonsang010716 committed 2 hours ago  
Create hello.py  
hyeonsang010716 authored 2 hours ago
```

The screenshot shows a GitHub commit history for a repository. There are three commits listed:

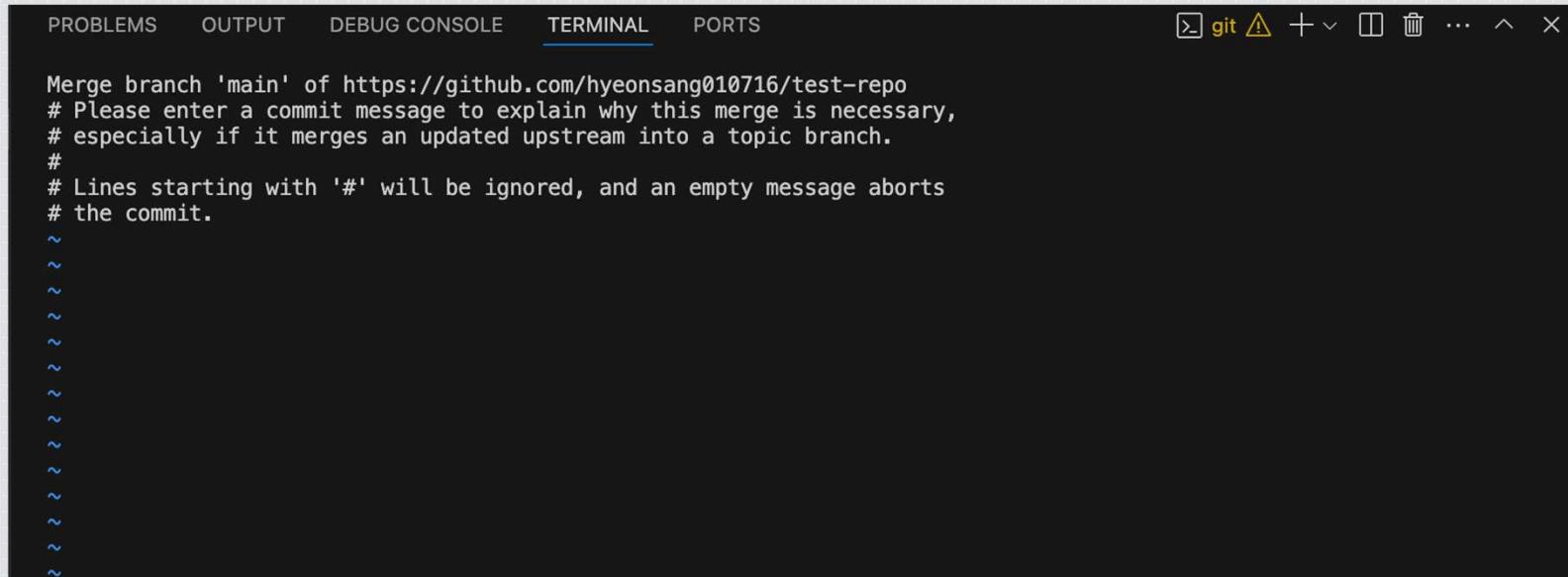
- Commit C: Merge branch 'main' of https://github.com/hyeonsang010716/test-repo by hyeonsang010716, committed 4 minutes ago. This commit is at the top of the list.
- Commit B: [FEAT] world 기능 구현 by hyeonsang010716, committed 2 hours ago. This commit is the middle one.
- Commit A: Create hello.py by hyeonsang010716, authored 2 hours ago. This commit is at the bottom of the list and is highlighted with a red rectangular box around its author, date, and hash.

Each commit has a copy icon and a diff icon to its right. Commit A also has a green "Verified" badge next to its hash.

충돌이 났을 경우 – 1(히스토리 충돌)

git pull origin main 했을 때, 아래와 같은 화면이 보인다면
Vim 환경과 동일함.

1. i를 눌러서 commit 메시지를 작성
2. 작성 후, esc
3. :wq로 저장



The screenshot shows a dark-themed terminal window with a light gray header bar. The header bar contains several icons: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. To the right of the header are more icons: a square with 'git' and a triangle, a plus sign, a downward arrow, a square with a diagonal line, a three-dot ellipsis, a double upward arrow, and a close button. The main area of the terminal is black and displays the following text:

```
Merge branch 'main' of https://github.com/hyeonsang010716/test-repo
# 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.
~
```

충돌이 났을 경우 – 1(히스토리 충돌)

먼저 git config pull.rebase true # rebase

로컬 깃헙 히스토리 상황: A

리모트 깃헙 히스토리 상황: B

A -> B 순서로 히스토리 정리

● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git config pull.rebase true

```
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add world.py
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[FEAT] world 구현"
[main 1d6c909] [FEAT] world 구현
 1 file changed, 0 insertions(+), 0 deletions(-)
  create mode 100644 world.py
❸ hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
  ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 921 bytes | 460.00 KiB/s, done.
From https://github.com/hyeonsang010716/test-repo
 * branch            main      -> FETCH_HEAD
   304b6f9..ac82e76  main      -> origin/main
Successfully rebased and updated refs/heads/main.
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 10 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 299 bytes | 299.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hyeonsang010716/test-repo.git
  ac82e76..04970de  main -> main
❹ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

[FEAT] world 구현
hyeonsang010716 committed 1 minute ago

B

Create hello.py
hyeonsang010716 authored 2 minutes ago

A

04970de ↗ ↘

Verified ac82e76 ↗ ↘

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

A개발자가 hello.py 를 수정해서 리모트에 푸쉬 했음.

The screenshot shows a GitHub commit history for the file 'hello.py' in a repository named 'test-repo'. The commit was made by 'hyeonsang010716' with the message 'Update hello.py'. Below the commit, there are two tabs: 'Code' (selected) and 'Blame'. The code content is as follows:

```
1 print("hello")
2 # A개발자가 고침.
```

만약 동시에 B개발자도 hello.py를 건드리게 된다면..?

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

먼저 git config pull.rebase true # rebase

The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: OPEN EDITORS (world.py, hello.py), TEST-REPO (func.py, hello.py, main.py, README.md, test.py, world.py).
- OPEN EDITORS** panel: world.py (content: print("hello")), hello.py (content: print("hello")
B 개발자가 고침.).
- TERMINAL** tab: Shows a git push command that failed due to a fast-forward conflict.
- PROBLEMS** tab: Lists errors from the git push command.
- OUTPUT**, **DEBUG CONSOLE**, and **PORTS** tabs are also visible.

```
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add hello.py
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[MOD] B개발자가 고침"
[main 1681454] [MOD] B개발자가 고침
  1 file changed, 1 insertion(+)
② hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
  ! [rejected]          main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
○ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

이번에는 git pull origin main 만 해서는 해결 안 됨...

The screenshot shows a Visual Studio Code (VS Code) interface with a dark theme. On the left is the Explorer sidebar showing a file tree with files like world.py, hello.py, func.py, main.py, README.md, test.py, and another world.py. The hello.py file is open in the editor, showing a merge conflict between two developers (A and B). The conflict markers are shown as '<<<<<' and '=====' around the code. A tooltip 'Resolve in Merge Editor' is visible over the conflict area. Below the editor is the Terminal tab, which displays the following git command history:

```
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add hello.py
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[MOD] B개발자가 고침"
[main 1681454] [MOD] B개발자가 고침
 1 file changed, 1 insertion(+)
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
 ! [rejected]      main -> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 941 bytes | 470.00 KiB/s, done.
From https://github.com/hyeonsang010716/test-repo
 * branch            main      -> FETCH_HEAD
   04970de..8211878  main      -> origin/main
Auto-merging hello.py
CONFLICT (content): Merge conflict in hello.py
error: could not apply 1681454... [MOD] B개발자가 고침
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 1681454... [MOD] B개발자가 고침
hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

The status bar at the bottom indicates '8211878! (Rebasing)'.

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

The screenshot shows a GitHub repository named 'TEST-REPO' in the Explorer sidebar. The 'hello.py' file is open in the editor, displaying a merge conflict between two commits. The conflict is shown in the following code snippet:

```
1 print("hello")
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
2 <<<<< HEAD (Current Change)
3 # A 개발자가 고침.
4 =====
5 # B 개발자가 고침.
```

A blue button labeled 'Resolve in Merge Editor' is visible in the gutter between the two code blocks. The status bar at the bottom indicates 'Rebase' is in progress, specifically '8211878! (Rebasing)'.

충돌난 소스 코드를 보여 줍니다.

```
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add hello.py
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "[MOD] B개발자가 고침"
[main 1681454] [MOD] B개발자가 고침
1 file changed, 1 insertion(+)
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
! [rejected]      main --> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 941 bytes | 470.00 KiB/s, done.
From https://github.com/hyeonsang010716/test-repo
 * branch            main      --> FETCH_HEAD
   04970de..8211878  main      --> origin/main
Auto-merging hello.py
CONFLICT (content): Merge conflict in hello.py
error: could not apply 1681454... [MOD] B개발자가 고침
hint: Resolve all conflicts manually, mark them as resolved with
hint: "git add/rm <conflicted_files>", then run "git rebase --continue".
hint: You can instead skip this commit: run "git rebase --skip".
hint: To abort and get back to the state before "git rebase", run "git rebase --abort".
Could not apply 1681454... [MOD] B개발자가 고침
```

> OUTLINE Rebase로 이동해서, 에러를 해결하려고 함.

Not Committed Yet Ln 7, Col 1 Spaces: 4 UTF-8 LF {} Python 3.13.0 ('3.13.0') Go Live Prettier

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

먼저 충돌된 코드를 고쳐 줌.
<< == >> 를 없애줘야 함.

고쳐진 코드로 반영할 것이라는 뜻

The screenshot shows the VS Code interface. On the left is the Explorer sidebar with icons for files, folders, and a repository. A blue circle with the number '1' is on the repository icon. The repository section is titled 'TEST-REPO' and contains files: func.py, hello.py (marked with a red exclamation point), main.py, and README.md. In the center is the 'OPEN EDITORS' pane, which is currently expanded to show two files: 'world.py' and 'hello.py'. The 'world.py' tab is active. To the right is the code editor pane, showing the content of 'hello.py':

```
1 print("hello")
2 # B 개발자가 고침.
```

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

충돌 난 부분을 고친 후

git add hello.py : hello.py가 충돌 났는데, 내 코드를 반영하겠다.

git rebase --continue : 충돌 해결했으니, rebase를 떠나겠다.

```
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add hello.py  
hyeonsang@johyeonsang-ui-MacBookPro test-repo % git rebase --continue  
[detached HEAD 14f6c21] [MOD] 충돌 해결한 코드
```

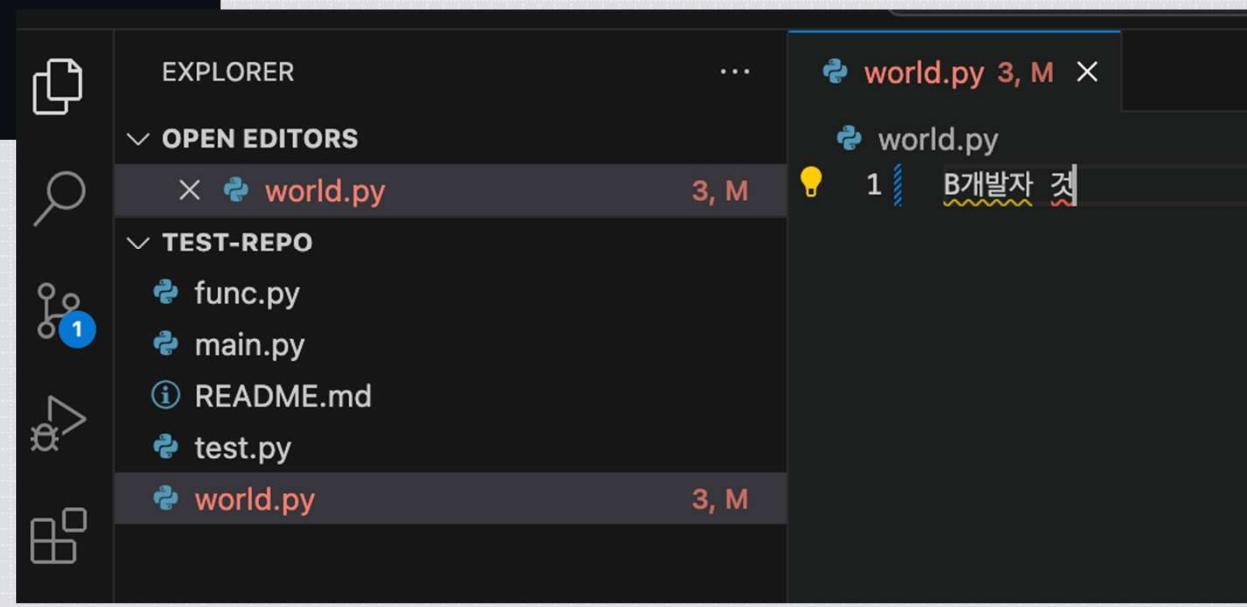
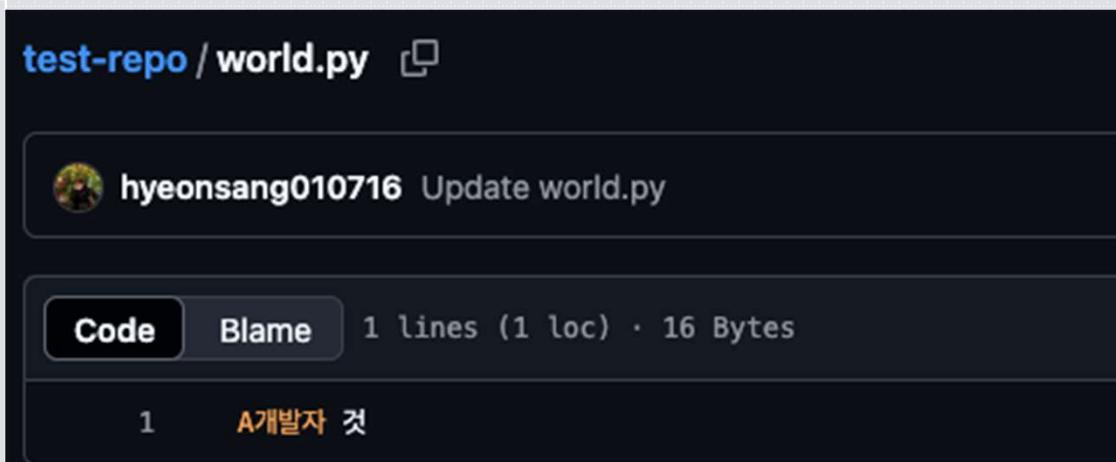
The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar: Shows files in the repository: func.py, hello.py, main.py, README.md, test.py, and world.py. The file `hello.py` is currently selected.
- OPEN EDITORS** panel: Shows two open editors: `world.py` and `hello.py`. The `hello.py` editor is active.
- hello.py Editor Content:**

```
1 print("hello")
2 # B 개발자가 고침.
```

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

두 번째, git config pull.rebase false # merge



충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

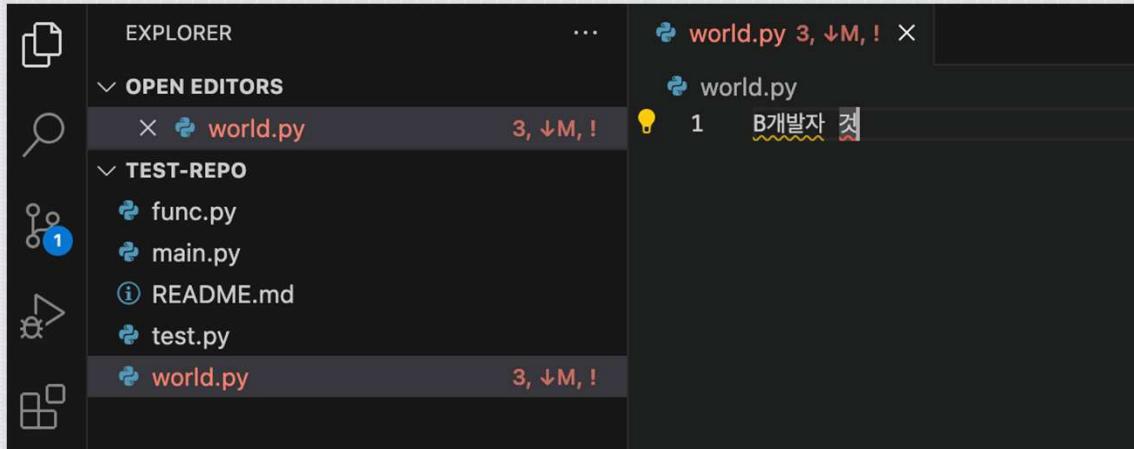
The screenshot shows a Visual Studio Code (VS Code) interface with a dark theme. On the left is the Explorer sidebar, which lists files: func.py, main.py, README.md, and test.py under 'TEST-REPO'. Two instances of 'world.py' are shown in the 'OPEN EDITORS' section, both with status '9, ↓M, !'. The main editor area displays a diff between two versions of 'world.py'. The top part of the diff is labeled 'HEAD (Current Change)' and contains the text 'B개발자 것'. The bottom part is labeled '(Incoming Change)' and contains the text 'A개발자 것'. A blue button labeled 'Resolve in Merge Editor' is located at the bottom right of the editor area. Below the editor is a tab bar with 'PROBLEMS' (9), 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (underlined), and 'PORTS'. The terminal tab shows the following command-line session:

```
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git config pull.rebase false
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add .
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "B개발자 것"
[main 909f399] B개발자 것
 1 file changed, 1 insertion(+)
④ hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
To https://github.com/hyeonsang010716/test-repo.git
  ! [rejected]      main --> main (fetch first)
error: failed to push some refs to 'https://github.com/hyeonsang010716/test-repo.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
④ hyeonsang@johyeonsang-ui-MacBookPro test-repo % git pull origin main
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 912 bytes | 130.00 KiB/s, done.
From https://github.com/hyeonsang010716/test-repo
 * branch            main       -> FETCH_HEAD
   040fbcc..c8221e5  main       -> origin/main
Auto-merging world.py
```

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

코드 수정!

<< == >> 를 없애줘야 함.



수정 완료 후에는 git add 파일 이름으로 수정이 끝났다는 것을 알려주기!

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS

● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git add world.py
○ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

충돌이 났을 경우 – 2(히스토리 & 같은 소스 코드 충돌)

그 다음, 병합 메시지 작성!

```
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git commit -m "B개발자로 바꿈"
[main 3aa9309] B개발자로 바꿈
● hyeonsang@johyeonsang-ui-MacBookPro test-repo % git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 10 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 421 bytes | 421.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/hyeonsang010716/test-repo.git
  c8221e5..3aa9309  main -> main
○ hyeonsang@johyeonsang-ui-MacBookPro test-repo %
```

| | | | | |
|-----------------|---|---------------------|--|--|
| B개발자로 바꿈 | C | 3aa9309 | | |
| B개발자 것 | B | 909f399 | | |
| Update world.py | A | Verified c8221e5 | | |

3주차 실습

- 간단한 알고리즘을 가지고 깃 협업 해보기!
- 주의사항) 커밋 컨밴션, 브런치 네이밍 컨밴션, 풀리퀘 컨밴션 맞출 것 (**실습 목적**)

- FEAT: 새로운 기능 추가
- FIX: 버그 수정
- DOCS: 문서 수정
- STYLE: 스타일 관련 기능(코드 포맷팅, 세미콜론 누락, 코드 자체의 변경이 없는 경우)
- REFACTOR: 코드 리팩토링
- TEST: 테스트 코드 추가
- CHORE: 빌드 업무 수정, 패키지 매니저 수정(ex .gitignore 수정 같은 경우)

커밋 컨밴션: [FEAT] , [MOD] , [FIX] ~

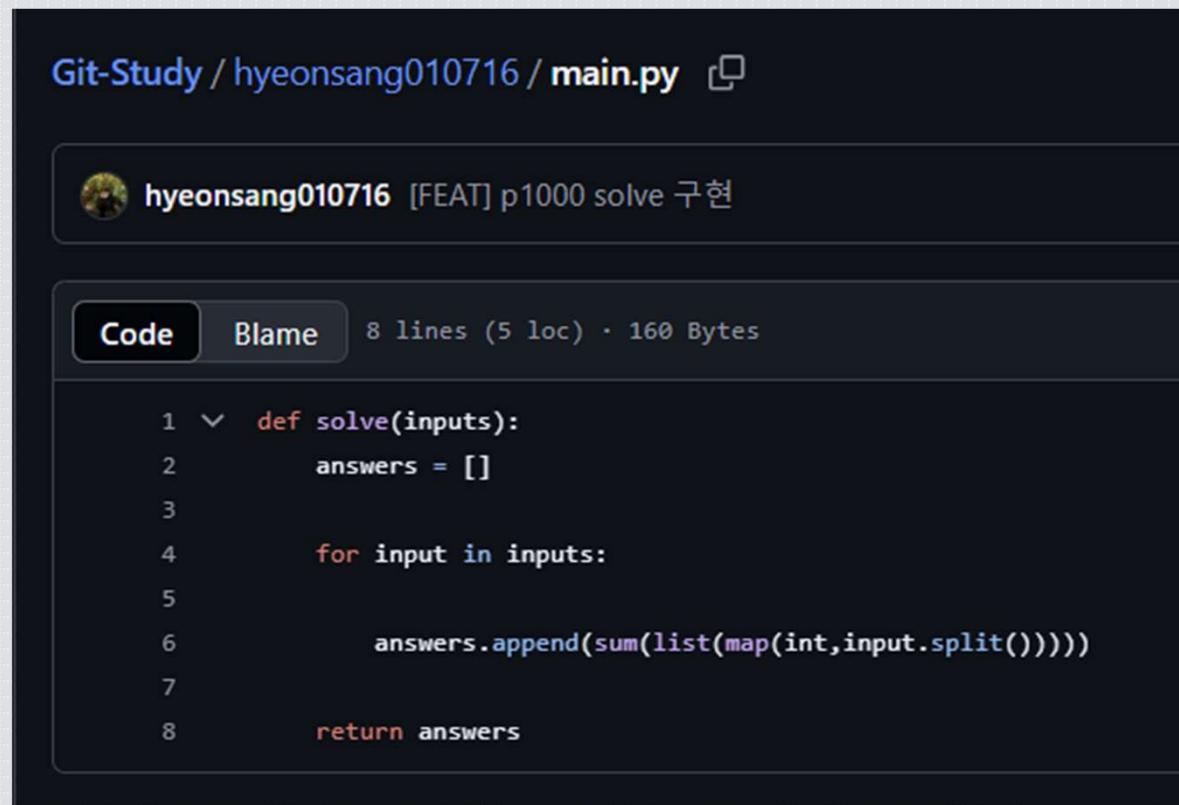
브런치 컨밴션: 깃협 계정 아이디/백준 문제 유형-백준 문제 번호
(만약 hyeonsang010716이 math 유형 백준 1000번을 풀었을 경우 → hyeonsang010716/math-1000)

풀리퀘 컨밴션: [백준 문제 유형]문제 번호-최초 풀리퀘를 보낸 날짜
(만약 2025.3.16일날 math 1000번을 풀어서 풀리퀘를 보낼 경우 → [math]1000-2025.3.16)

그리고 풀리퀘 상세 설명에 성공한 화면 사진 붙여넣기

실습

2주차 실습에 만들었던, main.py에 solve 함수를 만들고
서버에서 보내주는 inputs를 가지고, 정답을 return 합니다.



The screenshot shows a GitHub repository page for 'Git-Study / hyeonsang010716'. The 'main.py' file is displayed. The code is as follows:

```
1  def solve(inputs):
2      answers = []
3
4      for input in inputs:
5
6          answers.append(sum(list(map(int,input.split()))))
7
8      return answers
```

실습

추가 설명

inputs는 리스트고 띄어쓰기로 구분되어 있습니다.

예제 입력 1 복사

[1 2]

['1 19', '4 4', '23 14', '0 0']

예제 입력 1 복사

1 19
4 4
23 14
0 0

정답은 리스트로 return 해주세요.

Git-Study / hyeonsang010716 / main.py



hyeonsang010716 [FEAT] p1000 solve 구현

Code

Blame

8 lines (5 loc) · 160 Bytes

```
1 def solve(inputs):
2     answers = []
3
4     for input in inputs:
5
6         answers.append(sum(list(map(int,input.split()))))
7
8     return answers
```

print(inputs)로 디버깅하면서 하면 더 쉽습니다!

먼저 루트 디렉토리에서 pip install -r requirements.txt 를 사용해서 필요한 라이브러리를 다운 받아줍니다.

- hyeonsang@johyeonsang-ui-MacBookPro git-study % pip install -r requirements.txt
Requirement already satisfied: beautifulsoup4==4.12.3 in /Users/hyeonsang/.pyenv/versions/3.13.0/lib/python3.13/site-packages (from -r requirements.txt (line 1)) (4.12.3)
Requirement already satisfied: requests==2.32.3 in /Users/hyeonsang/.pyenv/versions/3.13.0/lib/python3.13/site-packages (from -r requirements.txt (line 2)) (2.32.3)
Requirement already satisfied: streamlit==1.37.0 in /Users/hyeonsang/.pyenv/versions/3.13.0/lib/python3.13/site-packages (from -r requirements.txt (line 3)) (1.37.0)

실습

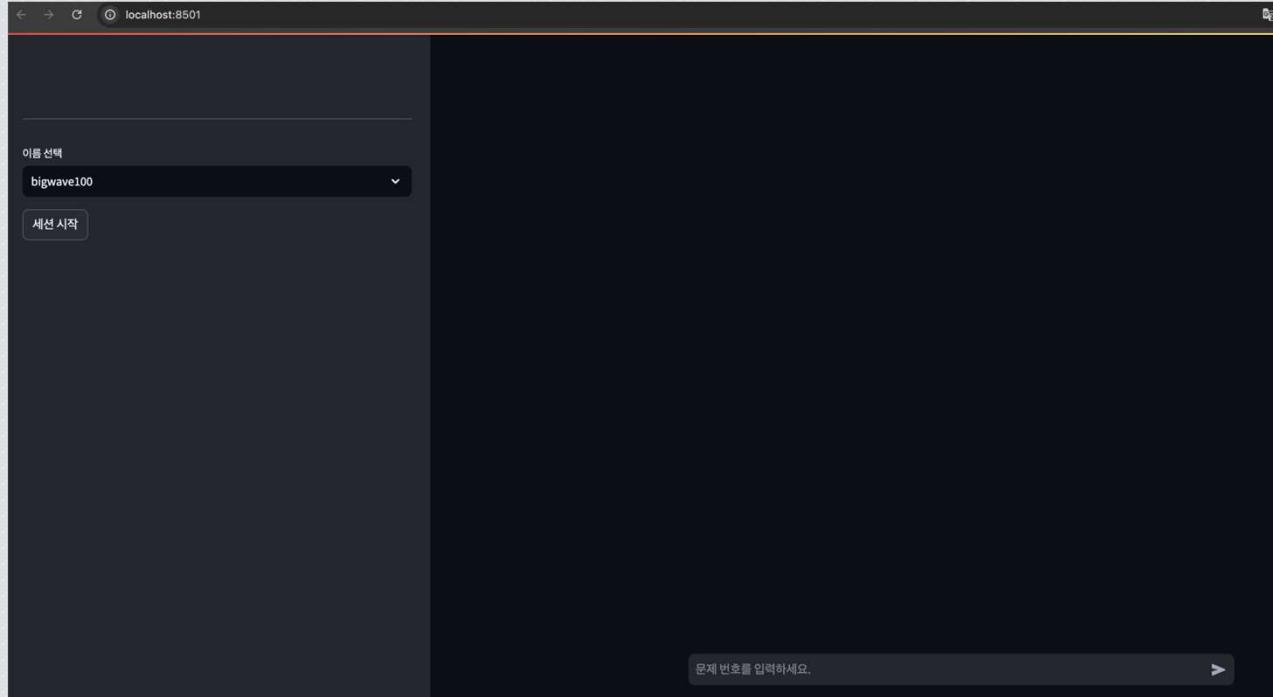
다음, streamlit run Home.py 를 입력해서, 프론트 서버를 켜줍니다.
아래 <http://localhost:8501> 을 들어가줍니다.

- hyeonsang@johyeonsang-ui-MacBookPro git-study % streamlit run Home.py

You can now view your Streamlit app in your browser.

Local URL: <http://localhost:8501>

Network URL: <http://192.168.0.10:8501>



실습

자신의 아이디 이름을 선택해줍니다.

이름 선택

hyeonsang010716



세션 시작

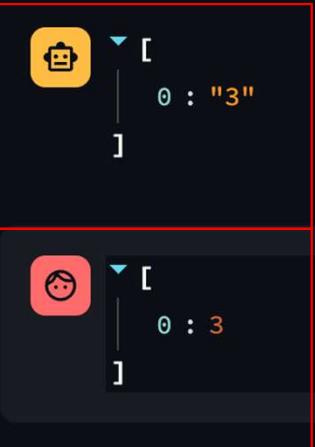
자신이 푼 문제 번호를 입력해줍니다.

1000



위, 아래가 같으면 성공!
이 사진을 캡쳐해서 풀리퀘 설명에 넣어주세요.

문제: 1000



백준 정답

내 정답

실습

끝

Add a title

[math]1000-2025.4.7

Add a description

Write Preview

문제: 1000

 [0 : "3"]

 [0 : 3]

Create pull request

시험 기간

| 달력 | | 양음력변환 | | | 전역일계산 | | |
|--------------------|----------------------|----------------------------------|----------------|-----------------|--------------------|---------------------|--|
| | | 2025.04 ▾ | | | | | |
| 일 | 월 | 화 | 수 | 목 | 금 | 토 | |
| 30 | 31 | 1 수산인의 날 | 2 세계 자폐... | 3 4.3 희생자... | 4 청명 향토예비... | 5 식목일 한식 | |
| 6 | 7 보건의 날 | 8 | 9 | 10 | 11 임시정부 ... | 12 음 3.15 도서관의 날 | |
| 13 | 14 | 15 | 16 국민안전... | 17 | 18 | 19 4.19혁명 | |
| 20 곡우 장애인의 날 | 21 과학의 날 | 22 지구의 날 새마을의 날 정보통신... | 23 세계 책과... | 24 | 25 법의 날 | 26 세계지적... | |
| 27 | 28 음 4.1 충무공 탄... | 29 | 30 | 1 | 2 | 3 | |