

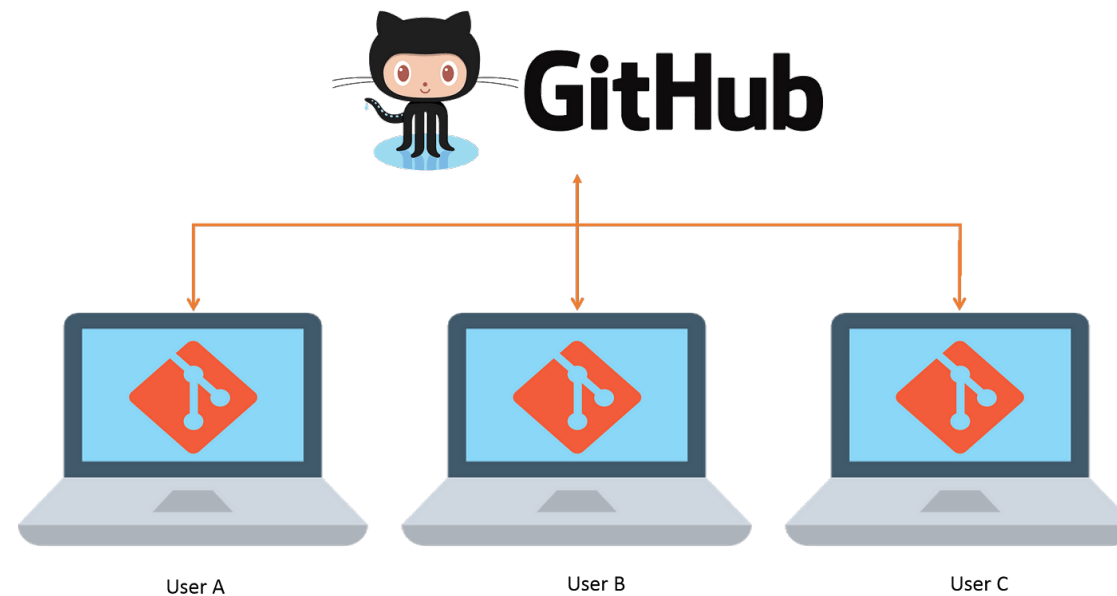
Lecture 2.

GitHub 기초 실습

김영빈 교수

실습 목표

- 개인별 GitHub Repository 생성 및 초기 설정
- Git 설치 및 GitHub 계정 연동
- Branch 생성 및 관리
- Pull Request(PR) 생성, 코드 리뷰, Merge 실행
- GitHub Issue & GitHub Action 기초 생성



Git과 GitHub



Local 컴퓨터에서 버전 관리를 담당하는 소프트웨어로, 파일 변경 사항을 추적하고 관리

```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
 1 file changed, 1 insertion(+)
 create mode 100644 README
$ git remote add origin git@github.com:cdown/thats.git
$ git push -u origin master
```



GitHub

Git을 기반으로 한 웹 호스팅 서비스로, 코드 저장소와 다양한 협업 기능을 제공



OpenAI

Verified

92.5k followers

<https://openai.com/>

Overview

Repositories 190

Projects

Packages

People 19

Pinned

openai-cookbook Public

Examples and guides for using the OpenAI API

MDX 62.1k 10k

whisper Public

Robust Speech Recognition via Large-Scale Weak Supervision

Python 77.7k 9.3k



git

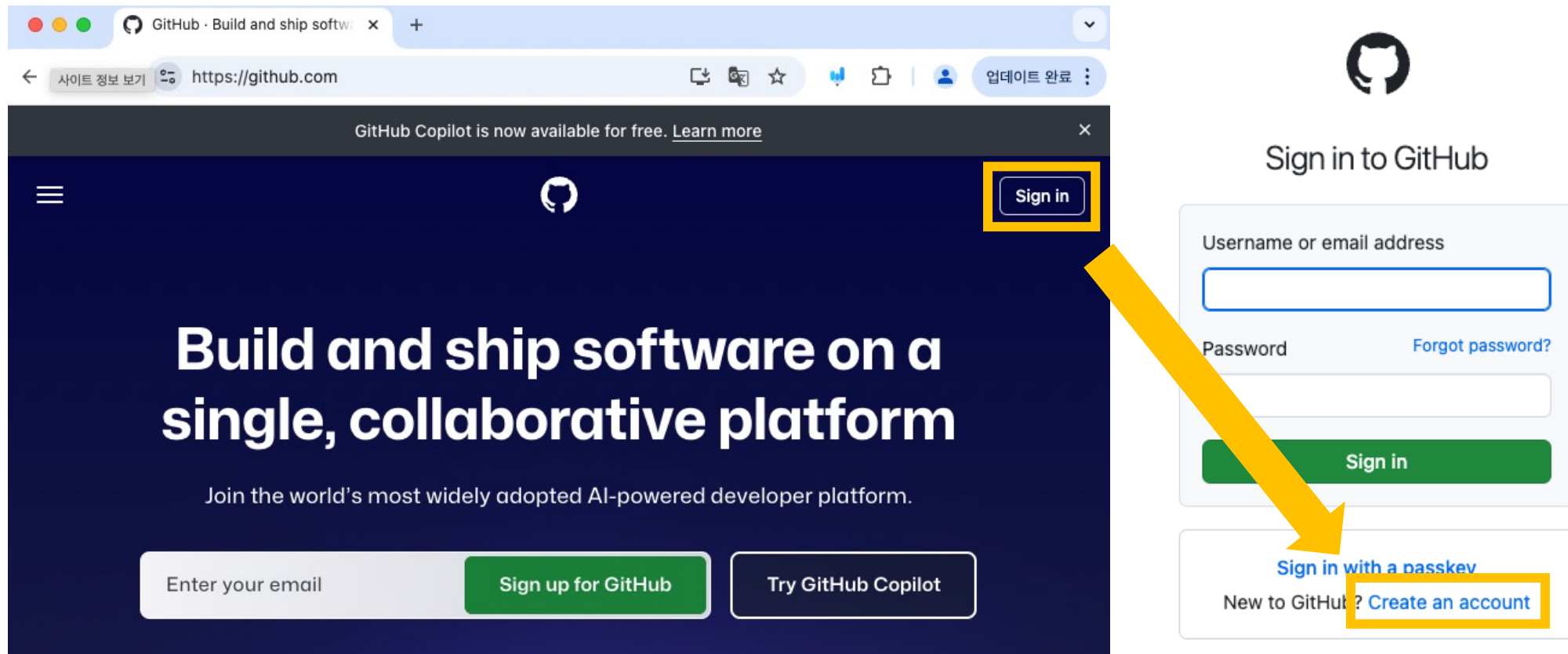


GitHub

	git	GitHub
기능	버전 관리 시스템(VCS)	클라우드 기반 협업 플랫폼
설치 위치	Local 컴퓨터 (개발자의 PC)	웹 서버 (github.com)
주요 역할	파일 변경 기록 저장, Branch 관리	코드 저장소 제공, 협업 기능 (PR, Issue)
네트워크 필요 여부	Local에서도 사용 가능	인터넷 필요
사용 방식	git commit, git merge	git push, git pull, PR

GitHub 계정 개설

- GitHub 접속 및 Sign up 및 Create an account 클릭, 회원가입 진행



GitHub 계정 개설

- 생성된 계정 확인

The screenshot shows a new GitHub profile page. At the top is a navigation bar with the GitHub logo, a search bar, and icons for repositories, projects, packages, and stars. Below the navigation bar, the profile section includes a large circular placeholder for a profile picture with a purple and white pixelated pattern, and an 'Edit profile' button. To the right of the profile picture, it says 'Popular repositories' and 'You don't have any public repositories yet.' Below the profile picture, there is a section for '0 contributions in the last year' with a calendar grid showing no activity. To the right of the calendar is a 'Contribution settings' dropdown and a list of years from 2020 to 2025, with 2025 selected. Below the calendar, there is a 'Contribution activity' section for 'March 2025' showing that 'bokju128 has no activity yet for this period.' and a 'Show more activity' button. At the bottom, there is a link to the 'GitHub profile guide'.

Popular repositories

You don't have any public repositories yet.

0 contributions in the last year

Contribution settings

2025

2024

2023

2022

2021

2020

2019

Contribution activity

March 2025

bokju128 has no activity yet for this period.

Show more activity

Seeing something unexpected? Take a look at the [GitHub profile guide](#).

보유 Repository 목록

보유 Contribution 목록

본인을 소개하는 README 작성

- 자기 PR 및 전공 소개
- 사용 가능한 Skill 및 Tool 소개
- 현재 진행중인 프로젝트 소개 등
- (도전) 나를 어필하는 소개 README 만들기

The screenshot shows a GitHub profile page for user 'cowkite'. Annotations on the left side of the image point to specific parts of the profile:

- 본인 사진** (Profile picture): Points to the profile picture placeholder.
- 이름 소개 부분** (Name introduction part): Points to the bio section.
- 이메일 등 SNS 링크** (Email and SNS links): Points to the 'Edit profile' button.

The profile page itself includes the following sections:

- Overview**: Navigation tabs for Overview, Repositories (7), Projects, and Packages.
- Profile Header**: Includes social links (Blog, Email, LinkedIn, Twitter) and a bio: "Hi there! I'm mobile app developer using Android and iOS. Sometimes I develop cross-platforms like ReactNative or Flutter, but I prefer native. I enjoy hiking, swimming, dive and surf. I hope to develop every beautiful things."
- Skills**: A section titled "Platforms & Languages" showing skills like Android, iOS, ReactNative, Flutter, and Cordova, and "Tools" showing skills like Kotlin, Swift, Java, TypeScript, ReactiveX, Firebase, Realm, Bitrise, and Git.
- Pinned**: A section titled "I'm an early" showing a commit activity chart.
- 's GitHub Stats**: A section showing GitHub statistics: Total Stars (4), Total Commits (738), Total PRs (339), Total Issues (14), and Contributed to (2).
- 1,323 contributions in the last year**: A contribution graph showing activity from March to February.

Git과 GitHub



Local 컴퓨터에서 버전 관리를 담당하는 소프트웨어로, 파일 변경 사항을 추적하고 관리

```
$ git init
Initialized empty Git repository in /tmp/tmp.IMBYSY7R8Y/.git/
$ cat > README << 'EOF'
> Git is a distributed revision control system.
> EOF
$ git add README
$ git commit
[master (root-commit) e4dcc69] You can edit locally and push
to any remote.
 1 file changed, 1 insertion(+)
 create mode 100644 README
$ git remote add origin git@github.com:cdown/thats.git
$ git push -u origin master
```



GitHub

Git을 기반으로 한 웹 호스팅 서비스로, 코드 저장소와 다양한 협업 기능을 제공



OpenAI

Verified

92.5k followers

<https://openai.com/>

[Overview](#) [Repositories 190](#) [Projects](#) [Packages](#) [People 19](#)

Pinned

[openai-cookbook](#) Public

Examples and guides for using the OpenAI API

MDX 62.1k 10k

[whisper](#) Public

Robust Speech Recognition via Large-Scale Weak Supervision

Python 77.7k 9.3k



git

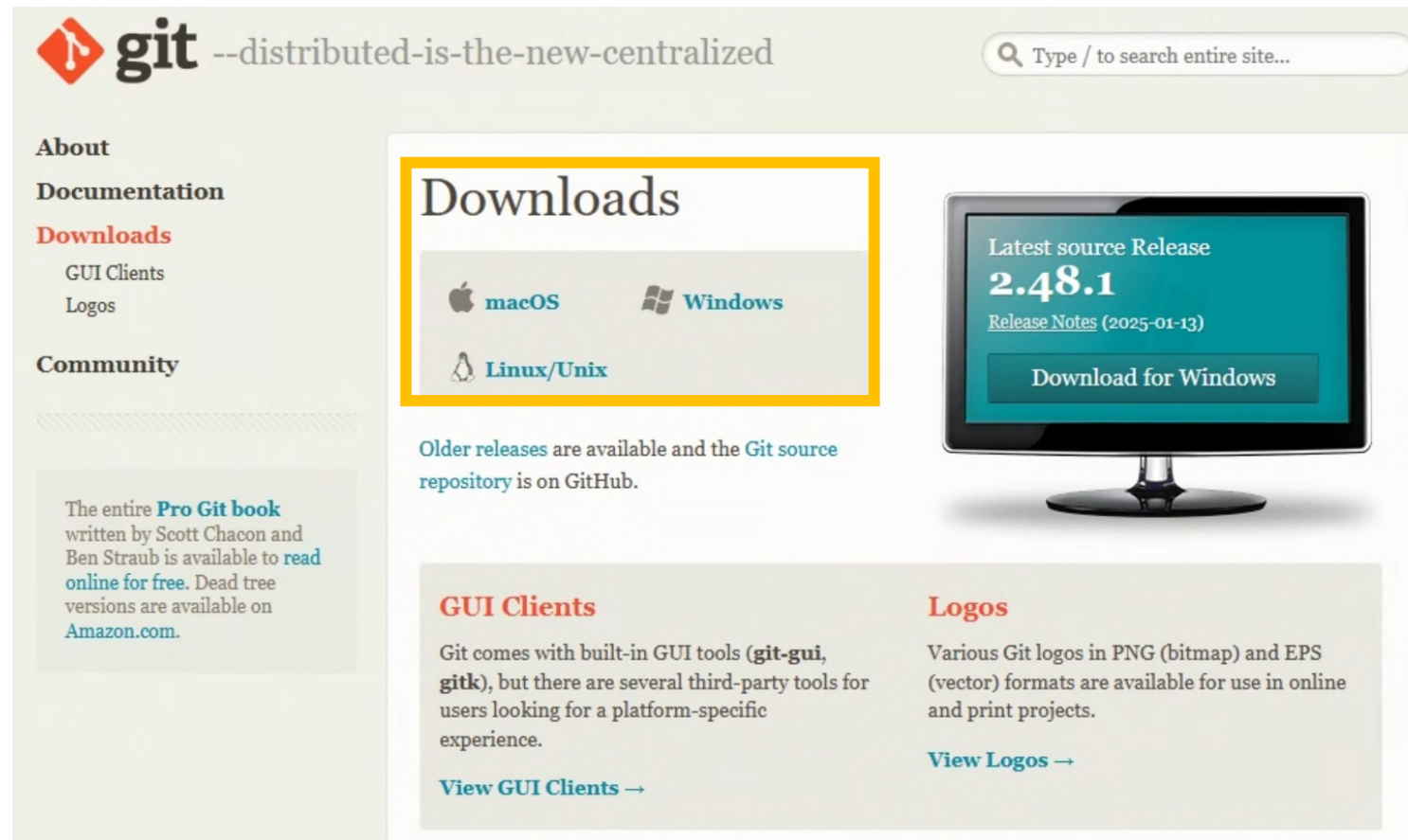


GitHub

기능	버전 관리 시스템(VCS)	클라우드 기반 협업 플랫폼
설치 위치	Local 컴퓨터 (개발자의 PC)	웹 서버 (github.com)
주요 역할	파일 변경 기록 저장, Branch 관리	코드 저장소 제공, 협업 기능 (PR, Issue)
네트워크 필요 여부	Local에서도 사용 가능	인터넷 필요
사용 방식	git commit, git merge	git push, git pull, PR

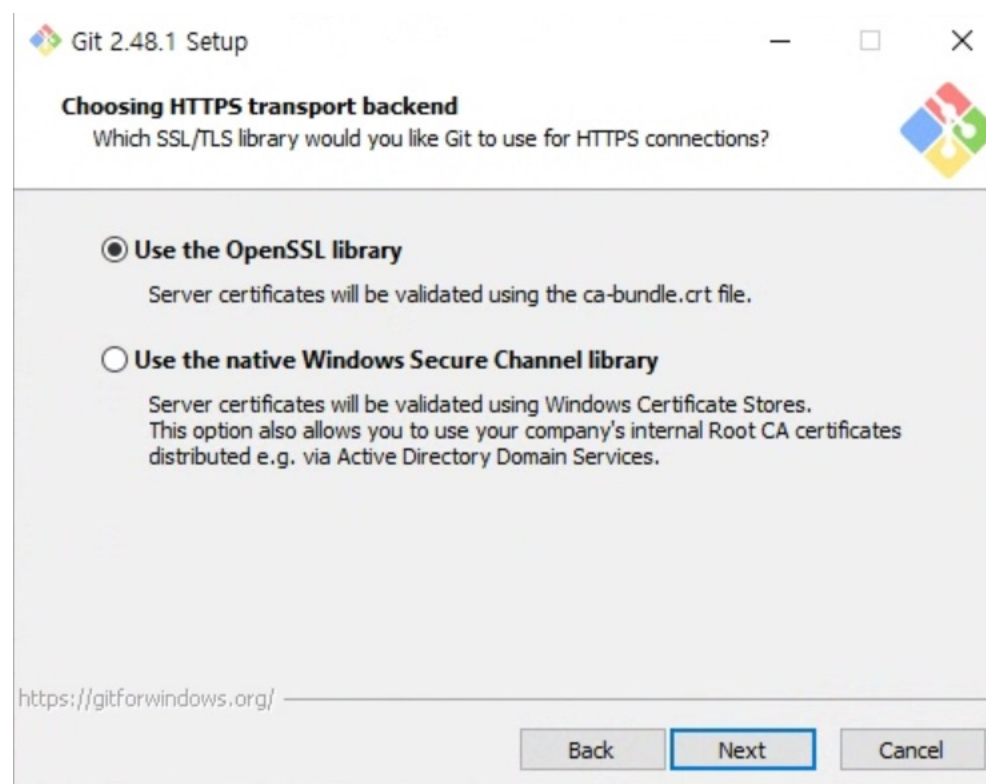
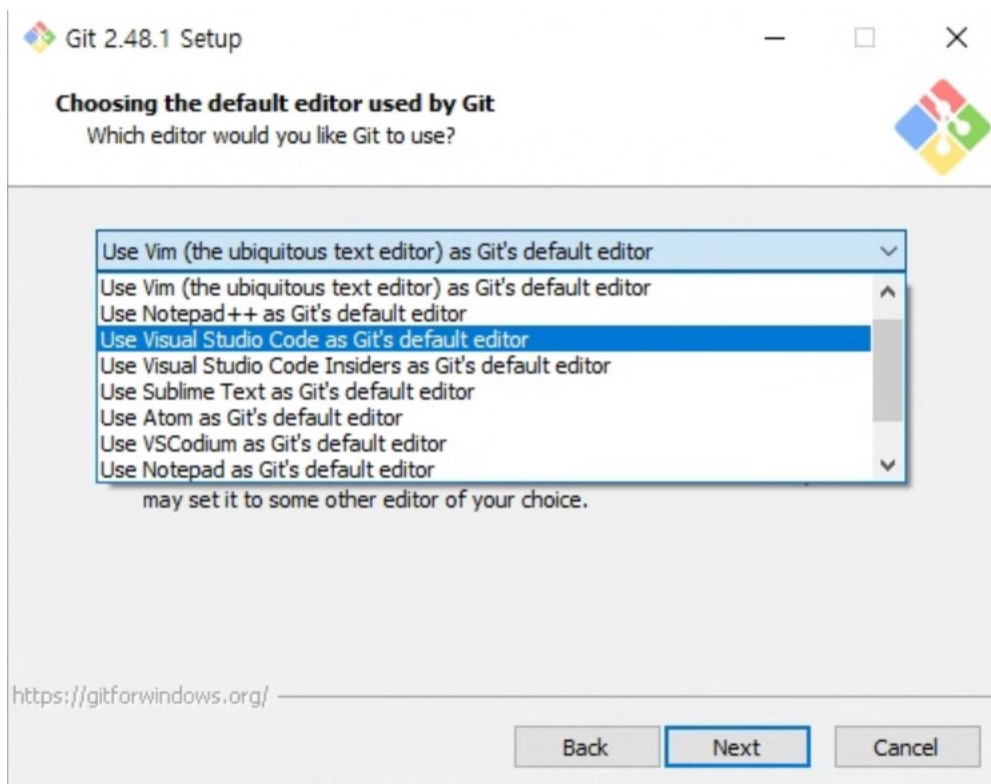
Git 설치 및 등록

- Git 다운로드: <https://git-scm.com/downloads/>



Git 설치 및 등록

- 설치 기본 옵션을 따르되, 아래 내용을 반영하여 진행



Git 설치 및 등록

- Git Bash를 통한 사용자 등록
 - git config --global user.name “
 - git config --global user.email “
 - git config --list

```
iip1@DESKTOP-NU4IEFG MINGW64 ~
$ git config --global user.name '사용자 이름'

iip1@DESKTOP-NU4IEFG MINGW64 ~
$ git config --global user.email '사용자 메일@naver.com'

iip1@DESKTOP-NU4IEFG MINGW64 ~
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\iip1\AppData\Local\Programs\Microsoft VS Code\bin\code" --wait
user.name=사용자 이름
user.email=사용자 메일@naver.com

iip1@DESKTOP-NU4IEFG MINGW64 ~
$
```

GitHub 개인 Repository 생성

- Repository name 및 Description, README 여부 설정

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *



Repository name *

이름 설정

Great repository names are short and memorable. Need inspiration? How about **silver-octo-guacamole** ?

Description (optional)

설명 작성



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.

공개 여부 설정

Initialize this repository with:



Add a README file

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

README 설정

Git으로 GitHub 업로드하기

- 'Calculator/calculate.py' 경로에 사칙연산 계산기 코드 작성

```
### 사칙연산 함수 정의
def plus(a, b):
    return a + b

def minus(a, b):
    return a - b

def mul(a, b):
    return a * b

def divide(a, b):
    return a / b
```

```
if __name__ == '__main__':

    ### 사용자 입력
    print('\n첫번째 숫자를 입력하세요.')
    input1 = input('입력: ')

    print('\n원하는 사칙연산 기호 중 하나를 선택하세요. (+, -, *, /)')
    act = input('기호: ')

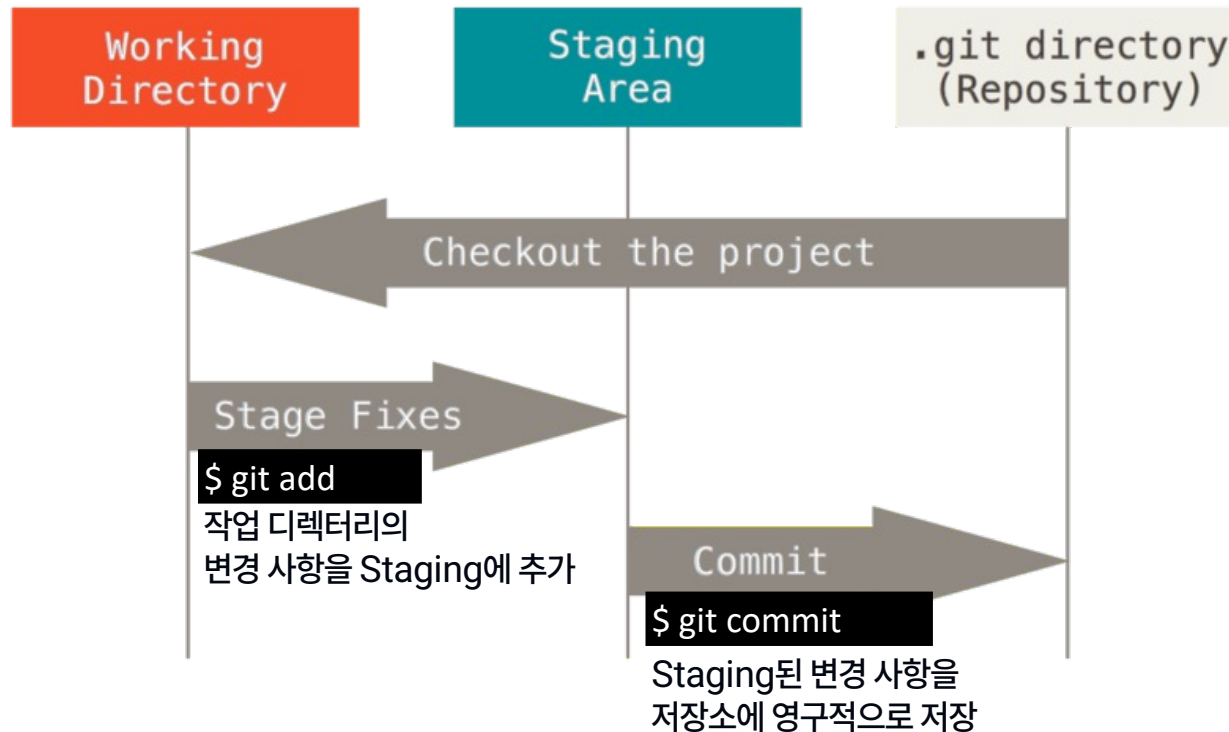
    print('\n두번째 숫자를 입력하세요.')
    input2 = input('입력: ')

    ### 연산 수행
    if act == '+':
        result = plus(input1, input2)
    elif act == '-':
        result = minus(input1, input2)
    elif act == '*':
        result = mul(input1, input2)
    elif act == '/':
        result = divide(input1, input2)
    print(f'사칙연산 결과는 {result}입니다.')
```

Git의 3가지 주요 영역

- Working Directory / Staging Area / Repository

▼ Git의 3가지 주요 영역



Working Directory

- 현재 작업 중인 파일이 위치하는 공간
- Git이 관리하지 않는 파일도 포함

Staging Area

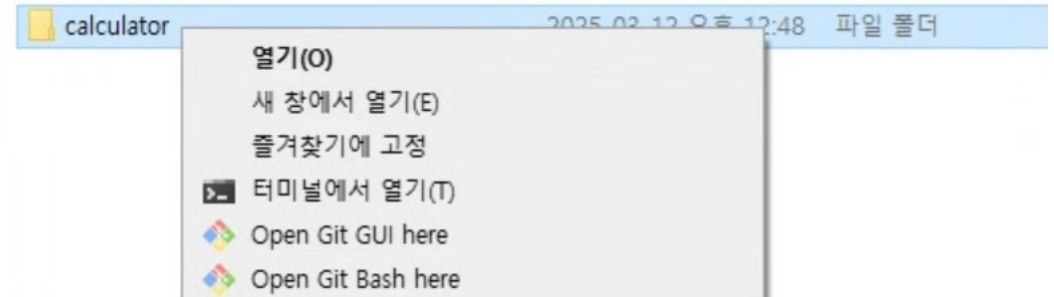
- `$ git add` 명령어로 변경 사항을 임시 저장
- Commit할 파일을 선택하는 단계

Repository

- `$ git commit` 로 변경 사항을 확정 저장
- SHA-1 해시 값과 함께 저장소에 영구 저장

Git으로 GitHub 업로드하기

- 해당 경로에서 Open Git Bash here 선택



- 파일 준비
 - git init
 - git add .
 - git status
 - git commit -m “
(commit message 작성)

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator
$ git init
Initialized empty Git repository in C:/Users/iip1/Desktop/OSSW/calculator/.git/

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git add .

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   calculate.py

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git commit -m 'upload'
[master (root-commit) f079623] upload
1 file changed, 37 insertions(+)
create mode 100644 calculate.py
```


Git으로 GitHub 업로드하기

- 파일 업로드
 - `git remote add origin https://github.com/사용자 이름/레포지토리 이름`
 - `git push -u origin master`

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git remote add origin https://github.com/사용자 이름/레포지토리 이름

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 563 bytes | 563.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/사용자 이름/레포지토리 이름/pull/new/master
remote:
To https://github.com/사용자 이름/레포지토리 이름
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Pull & Request (PR)

Pull Request(PR) : 개발자가 코드 변경 사항을 병합하기 전에 검토 받을 수 있도록 하는 GitHub의 협업 기능

- 새로운 기능을 개발한 후, **Main Branch에 Merge하기 전 코드 리뷰를 요청하는 과정**

새로운 기능 개발

- 작업 별 Branch를 생성하여 독립적으로 개발 진행

PR 생성

- 개발이 완료되면 기존 Main Branch로 병합을 요청하는 PR을 생성

코드 검토 및 피드백

- 팀원들이 코드 변경 사항을 검토하고 피드백을 제공
- 코드 스타일, 버그, 최적화 가능성 등을 논의

수정 및 반영

- 리뷰 내용을 반영하여 코드 수정 후 다시 푸시

Merge 및 Branch 정리

- 코드 리뷰가 완료되면 PR을 승인한 후 Main Branch로 병합
- 불필요한 Feature Branch는 삭제하여 관리

※ PR 흐름도

1. Fork 생성

→

2. 변경 작업

→

3. PR 생성

→

4. 코드 리뷰

→

5. Merge

Pull & Request (PR)

- push 후 웹에서 GitHub Repository 확인, 필요에 따른 코드 리뷰 진행

master had recent pushes 2 seconds ago [Compare & pull request](#)

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](#) or [learn more about diff comparisons](#).

base: main ← compare: master

There isn't anything to compare.
main and master are entirely different commit histories.

Showing 1 changed file with 37 additions and 0 deletions. Split Unified

```
37 calculate.py
... @@ -0,0 +1,37 @@
1 + ### 사칙연산 함수 정의
2 + def plus(a, b):
3 +     return a + b
4 +
5 + def minus(a, b):
6 +     return a - b
```

PR 생성 → 리뷰어 피드백 제공 → 수정 후 다시 Push → 최종 리뷰 후 PR Merge

Pull & Request (PR, 실습)

- 코드 에러 수정 후 새롭게 commit & push하기
- 현재 작성한 코드에 따르면, 사칙연산이 올바르게 작동하지 않음 ($5 + 1! = 51$)
- 숫자 입력 부분에 텍스트를 넣어도 코드가 계속 동작함

```
(base) PS C:\Users\wiip\desktop\OSSW\calculator> python calculate.py
첫번째 숫자를 입력하세요.
입력: 5
원하는 사칙연산 기호 중 하나를 선택하세요. (+, -, *, /)
기호: +
두번째 숫자를 입력하세요.
입력: 1
사칙연산 결과는 51입니다.
```

```
(base) PS C:\Users\wiip\desktop\OSSW\calculator> python calculate.py
첫번째 숫자를 입력하세요.
입력: 안녕
원하는 사칙연산 기호 중 하나를 선택하세요. (+, -, *, /)
기호: +
```

Pull & Request (PR, 실습)

- 사칙연산이 올바르게 수행되도록 네가지 연산 기호와 관련한 코드를 모두 점검 후, 네가지 사칙연산을 올바르게 수정 했다는 commit message를 작성하여 push하기
- 새롭게 push한 후 GitHub에서 코드 수정 내역 및 commit message 확인하기

...	...	@@ -1,6 +1,6 @@
1	1	### 사칙연산 함수 정의
2	2	def plus(a, b):
3	-	return a + b
	3	+ return float(a) + float(b)

GitHub 코드 수정 내역 확인

Commits

History for OSSW-week2 / calculate.py on master

Commits on Mar 12, 2025

fix plus operation

bokju128 committed 1 minute ago

upload

bokju128 committed 1 hour ago

End of commit history for this file

Commit History에서
추가된 commit message 확인

Pull & Request (PR, 실습)

- (도전) 숫자를 입력할 때까지 입력을 계속 반복하도록 코드를 수정하고, 해당 내용을 올바르게 수정 했다는 commit message를 작성하여 push하기
- 새롭게 push한 후 GitHub에서 코드 수정 내역 및 commit message 확인하기

잘못된 입력, 에러 메세지 출력

잘못된 입력, 에러 메세지 출력

올바른 입력, 이후 과정 수행

```
(base) PS C:\Users\wiip\desktop\OSSW\calculator> python calculate.py
첫번째 숫자를 입력하세요.
입력: 안녕
잘못된 입력입니다, 다시 첫번째 숫자를 입력하세요.
첫번째 숫자를 입력하세요.
입력: hello world
잘못된 입력입니다, 다시 첫번째 숫자를 입력하세요.
첫번째 숫자를 입력하세요.
입력: 104
원하는 사칙연산 기호 중 하나를 선택하세요. (+, -, *, /)
기호:
```

Git으로 GitHub 업로드하기 (실습)

- 해당 Repository 내 README에 실행 방법 및 에러 수정 로그 남기기
- Markdown 작성법을 검색 후 다음과 똑같은 양식으로 작성하기
- (도전) 구글링으로 계산기 이미지 불러와서 README 안에 이미지 넣기

README

OSSW-week2

나만의 사칙연산 계산기 만들기

실행 방법 🚧

터미널(혹은 아나콘다)을 열어서 `python calculator.py`를 실행한다.
실행 이후 숫자로 된 입력 및 사칙연산 기호를 입력하여 계산 결과를 확인한다.

에러 수정 로그 🔍

날짜	수정
250309	사칙연산 계산기 첫 구현
250310	입력값 숫자로 제한하도록 수정
250312	숫자 아닌 입력값 반복적으로 다시 입력하도록 수정

#을 활용한 제목 크기 설정

Markdown을 통한 테이블 작성

새 branch 생성 및 등록

- 현재 branch 확인 및 새 branch 생성
 - git checkout
 - git checkout -b imsy

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git checkout
M       calculate.py
Your branch is up to date with 'origin/master'.

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git checkout -b imsy
Switched to a new branch 'imsy'
```

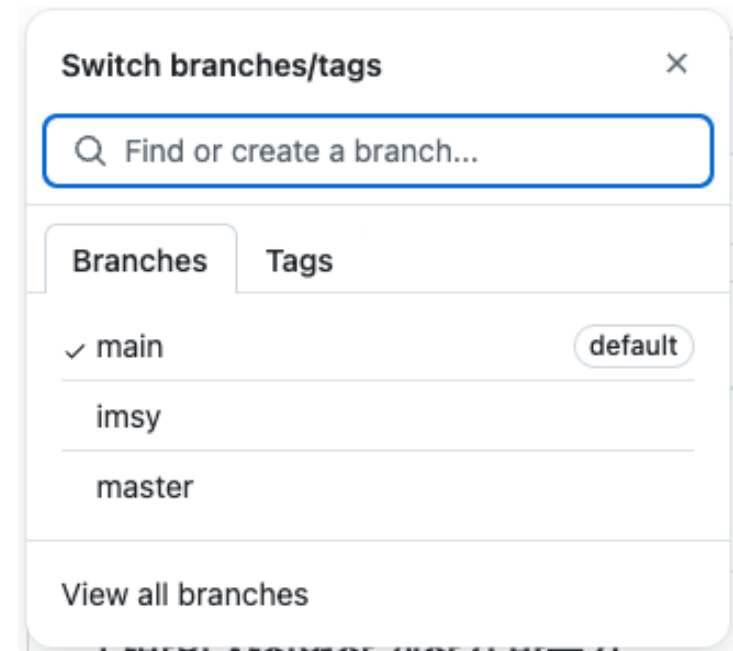
- 로컬 저장소와 원격 저장소의 브랜치 연결
 - git branch --set-upstream-to origin/master

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (imsy)
$ git branch --set-upstream-to remotes/origin/master
branch 'imsy' set up to track 'origin/master'.
```

새 branch 생성 및 등록

- 새 branch로 업로드 및 GitHub에 업데이트 된 branch 현황 확인
 - git push origin imsy
- GitHub에 표시되는 새 branch 확인

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (imsy)
$ git push origin imsy
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'imsy' on GitHub by visiting:
remote:   https://github.com/사용자 이름/레포지토리 이름/pull/new/imsy
remote:
To https://github.com/사용자 이름/레포지토리 이름
 * [new branch]      imsy -> imsy
```



Merge 방식 비교

사진출처: <https://kotlinworld.com/277>

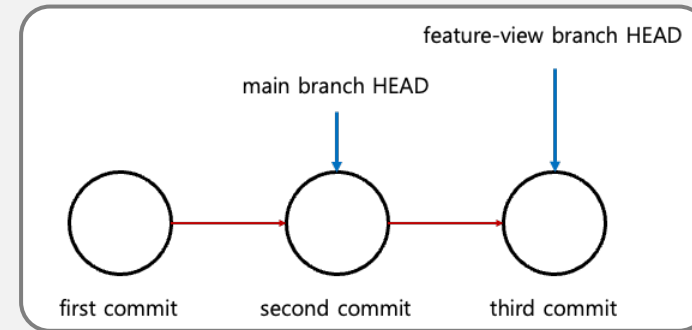
Fast-forward Merge

: Branch가 분기되지 않고,
한쪽 Branch가 다른 Branch의 최신 Commit을 그대로 포함하는 직선적 흐름
(단순한 병합에 사용)

- History가 깨끗하게 유지됨
- 새로운 병합 Commit이 생성되지 않음

한계

- 중간에 다른 Commit이 있을 경우 동작하지 않음
- 동일한 파일을 수정한 Commit이 여러 Branch에 존재하면 충돌발생



- HEAD 포인터를 앞으로 이동
- HEAD 포인터를 앞으로 이동하여 병합 (새로운 Commit 생성 없음)

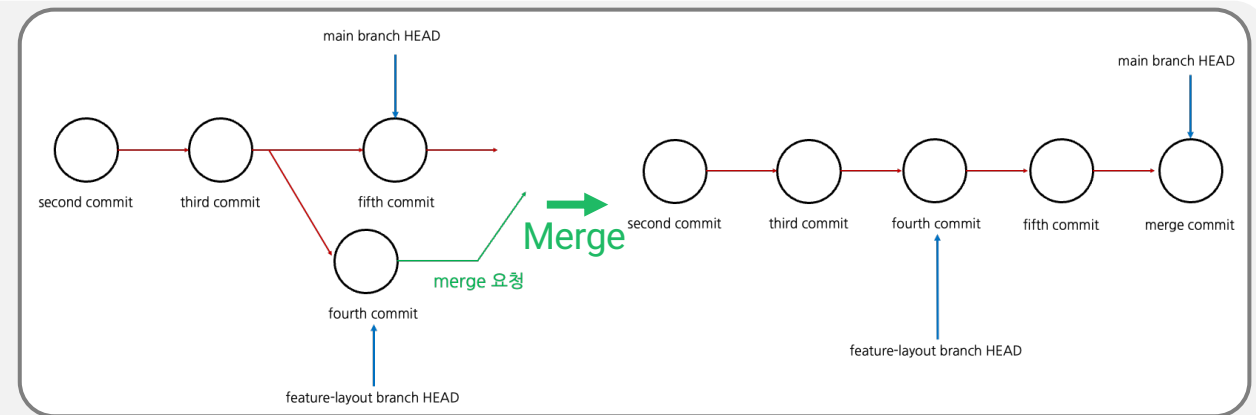
```
$ git checkout -b feature # feature 브랜치 생성 및 이동
$ git commit -m "Add feature" # 새로운 커밋 생성 (feature 브랜치에서)
$ git checkout main # main 브랜치로 이동
$ git merge feature # Fast-forward Merge 수행
```

Three-way Merge

: 병합하려는 두 Branch가 서로 다른 Commit을 가지고 있어 공통 조상이 존재
(분기된 Branch 병합에 사용)

- Branch가 독립적으로 변경된 후 병합될 때 사용됨
- 새로운 병합 Commit이 생성되어 History에 병합 흔적이 남음

```
$ git checkout main # main 브랜치로 이동
$ git merge feature # 병합 커밋 생성
```



- 3개의 Commit 비교로 새 병합 Commit 생성
- 공통 조상과 두 Branch의 최신 Commit을 비교하여 새로운 Merge Commit 생성

Branch 병합 (Fast-forward Merge)

- 병합하려는 branch로 이동, 이후 병합
 - git checkout master
 - git merge imsy
- Commit message 작성 및 push

```
iipl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (imsy)
$ git checkout master
M       calculate.py
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

iipl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git merge imsy
Already up to date.
```

```
iipl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git add .

iipl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git commit -m 'merge imsy'
[master 31a1e43] merge imsy
1 file changed, 9 insertions(+), 2 deletions(-)

iipl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/calculator (master)
$ git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 392 bytes | 392.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/사용자 이름/레포지토리 이름
3096ec6..31a1e43 master -> master
```

Branch 병합 (Three-way Merge)

- 새 폴더 'three-way-merge' 생성 후 Git 저장소 초기화

- mkdir three-way-merge
- cd three-way-merge
- git init

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ mkdir three-way-merge

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ cd three-way-merge

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git init
Initialized empty Git repository in C:/Users/iip1/Desktop/OSSW/three-way-merge/.git/
```

- Main branch에서 파일 생성 및 첫 커밋

- echo " " > file.txt
- git add file.txt
- git commit -m " "

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ echo 'first file in mian branch' > file.txt

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next time Git touches it
```

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git commit -m 'initial commit'
[master (root-commit) a5fd2bd] initial commit
1 file changed, 1 insertion(+)
create mode 100644 file.txt
```

Branch 병합 (Three-way Merge)

- 새 branch 생성 후 해당 파일 수정
 - git checkout -b feature
 - echo " " > file.txt
 - git add file.txt
 - git commit -m " "

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git checkout -b feature
Switched to a new branch 'feature'

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (feature)
$ echo 'feature branch modification' > file.txt

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (feature)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (feature)
$ git commit -m 'modify in feature branch'
[feature 6aa6131] modify in feature branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

- Main branch에서 해당 파일을 다른 내용으로 수정
 - git checkout master
 - echo " " > file.txt
 - git add file.txt
 - git commit -m " "

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (feature)
$ git checkout master
Switched to branch 'master'

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ echo 'main branch modification' > file.txt

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git commit -m 'modify in main branch'
[master 9cf2b16] modify in main branch
1 file changed, 1 insertion(+), 1 deletion(-)
```

Merge Conflict

충돌 발생 원인

같은 파일의 같은 위치를 여러 사람이 수정했을 때 충돌 발생

예시) 각 Branch에서 README.md의 동일한 부분을 다르게 수정

```
Main branch >>
Welcome to my project!

Feature branch >>
Welcome to our awesome project!
```

```
$ git merge feature
CONFLICT (content): Merge conflict in README.md
```

→ Main Branch와 Feature Branch가 같은 줄을 다르게 수정했기 때문에 **충돌발생**

충돌 해결 방법

충돌이 발생하면 Git은 해당 파일 내에서 충돌 부분을 자동으로 표시

예시) 충돌 발생 후 README.md 파일 내용

```
<<<<<< HEAD
Welcome to my project!
=====
Welcome to our awesome project!
>>>>>> feature
```

현재(HEAD, 즉 main branch)의 내용

변경된 내용을 구분하는 기준선

feature branch의 내용

해결 단계)

1. 충돌 파일을 편집하여 최종 내용으로 수정
2. 충돌 마커 제거 (<<<, ==, >>>)
3. 수정된 파일을 Git에 추가 → `git add README.md`
4. 충돌 해결 완료 후 Commit → `git commit`

Branch 병합 (Three-way Merge)

- 새 branch를 main branch로 병합하는 과정에서,
앞서 main/feature branch에서 동일한 파일을 서로 다르게 수정한 파일이 충돌함
 - git merge feature

충돌 감지 및 병합 중단

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master)
$ git merge feature
Auto-merging file.txt
CONFLICT (content): Merge conflict in file.txt
Automatic merge failed; fix conflicts and then commit the result.
```

- 해당 파일을 열어 충돌 내용 확인
 - vi file.txt
 - :q! (enter)

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master|MERGING)
$ vi file.txt
```

```
<<<<<< HEAD
main branch modification
```

main branch 수정 내용

```
=====
feature branch modification
>>>>>> feature
```

새 branch 수정 내용

Branch 병합 (Three-way Merge)

- 이 경우 해당 파일을 직접 수정 후 저장
 - vi file.txt
 - i, (수정), esc, :wq!

```
modification to make no conflicts!|
~
~
~
file.txt[+] [dos] (14:06 14/03/2025) 1,35 All
— INSERT —
```

- 이후 수정한 파일 git 추가 및 commit
 - git add file.txt
 - git commit -m “

```
h1pl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master|MERGING)
$ git add file.txt

h1pl@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/three-way-merge (master|MERGING)
$ git commit -m 'resolve merge conflict between main and feature'
[master a95d299] resolve merge conflict between main and feature
```

Issue 생성 및 관리

- 특정 Repository 내 Issues 탭을 활용하여 개발자 간 소통 가능

사칙연산 계산기 입력 문제 #1

✓ Closed



i opened 3 minutes ago

...

계산기에서 숫자를 입력해야 되는데 문자도 입력되는 것 같아요
이거 어떤 문제죠?

Create sub-issue



i now

Owner

Author

...

제가 이후 commit을 추가적으로 진행함으로써 문제를 해결하였습니다.
관련 commit을 참고하시기 바랍니다.



i closed this as completed now

Issue 생성 및 관리

- 실제 유명한 Open Library에서는 최근까지도 활발한 이슈가 논의 중
- (도전) 평상시 자주 사용하는 Library GitHub 접속 후 최근 이슈 확인하기

The screenshot displays the GitHub Issues page for the Python pandas library. At the top, there are filters for 'Open' (3,621) and 'Closed' (23,637) issues, along with dropdown menus for 'Author', 'Labels', 'Projects', 'Milestones', 'Assignees', and 'Types'. A 'Newest' sort button is also present. The list of issues includes:

- BUG: Can only compare identically-labeled Series objects (string vs. object)** (Bug, Needs Discussion, Strings) - #61099 · wahsmail opened yesterday
- ENH: DatetimeIndex.set_freq()** (Enhancement, Needs Triage) - #61094 · rwijtvliet opened 2 days ago
- ENH: Support dict to pd.set_option for cleaner code** (Enhancement, Needs Triage) - #61093 · yasirroni opened 2 days ago
- BUG: Collision between equivalent frequencies 'QS-FEB' and 'QS-NOV'** (Bug, Datetime) - #61086 · rwijtvliet opened 3 days ago (2 comments)
- BUG: pd.api.types.infer_dtype on scalar input** (Bug, Docs) - #61081 · gnotisauton opened 4 days ago (1 fork, 8 comments)
- BUILD: installation using wheels fails when using nix + uv2nix** (Build) - #61079 · cpcloud opened 5 days ago (2 comments)

Python pandas Library의 Issues

GitHub Actions

- GitHub Actions을 수행하기 위한 디렉토리 추가 및 Git 저장소 초기화
- GitHub Actions 워크플로우 파일 추가
 - `mkdir -p .github/workflows`
 - `touch .github/workflows/ci.yml`
 - `vi .github/workflows/ci.yml`

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ mkdir github-actions

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ cd github-actions

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ git init
Initialized empty Git repository in C:/Users/iip1/Desktop/OSSW/github-actions/.git/

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ echo 'github actions demo' > README.md

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ git add README.md
warning: in the working copy of 'README.md', LF will be replaced by CRLF the next time Git touches it

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ git commit -m 'initial commit'
[master (root-commit) e0b2a54] initial commit
1 file changed, 1 insertion(+)
create mode 100644 README.md
```

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ mkdir -p .github/workflows

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ touch .github/workflows/ci.yml
touch: cannot touch '.github/workflows/ci.yml': No such file or directory

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ touch .github/workflows/ci.yml

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/github-actions (master)
$ vi .github/workflows/ci.yml
```

GitHub Actions

- GitHub Actions 워크플로우 파일 내용 작성

워크플로우 이름 설정

실행 조건 명시
(main branch로 코드를 push하거나 PR이 생성될 때)

'test'라는 작업을 정의하고
이를 GitHub에서 제공하는 ubuntu 환경에서 실행

코드를 가져오고,

Node.js 버전 16을 통해 환경을 구성하고,

관련 의존성을 설치하고,

테스트 실행

```
name: CI Pipeline

on:
  push:
    branches:
      - main
  pull_request:
    branches:
      - main

jobs:
  test:
    runs-on: ubuntu-latest

    steps:
      - name: 코드 체크 아웃
        uses: actions/checkout@v3

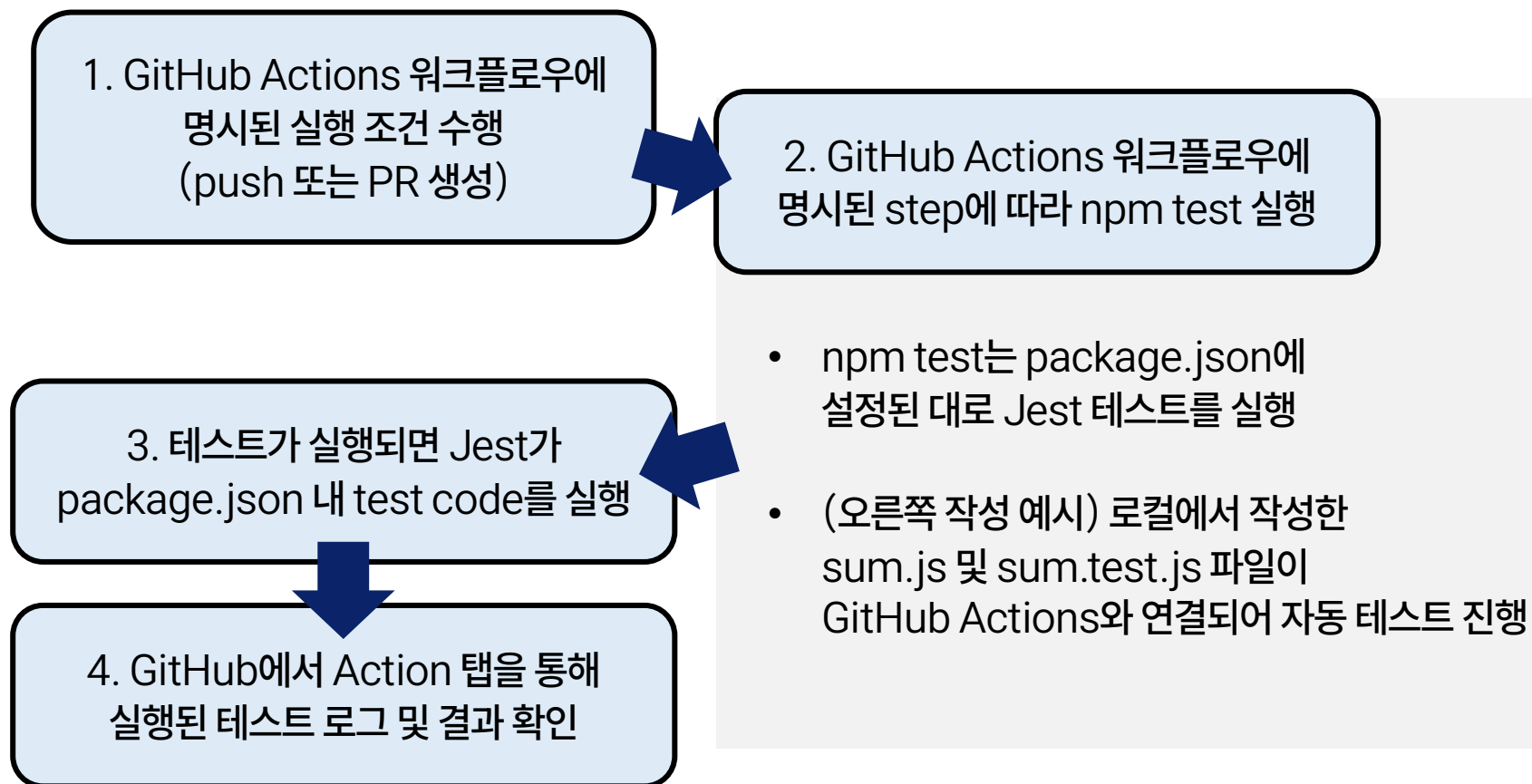
      - name: Node.js 설정
        uses: actions/setup-node@v3
        with:
          node-version: '16'

      - name: 의존성 설치
        run: npm install

      - name: 테스트 실행
        run: npm test
```

GitHub Actions

- node.js가 설치되어 있다는 가정하에 GitHub Actions는 다음 동작으로 수행됨



```
{
  "name": "github-actions-demo",
  "version": "1.0.0",
  "description": "",
  "main": "sum.js",
  "scripts": {
    "test": "jest"
  },
  "devDependencies": {
    "jest": "^29.6.1"
  }
}
```

<package.json 작성 예시>

GitHub Fork & Clone

- Sorting Algorithms을 다루는 다음 GitHub 저장소로 진행
 - https://github.com/B3ns44d/Python_Sorting_Algorithms
- 이후 https://github.com/사용자 이름/Python_Sorting_Algorithms 확인

The screenshot shows the GitHub repository page for 'Python_Sorting_Algorithms' by user 'B3ns44d'. The repository is public and has 27 forks and 22 stars. The 'Fork' button is highlighted with a yellow box. A large yellow arrow points from this box to the 'Create fork' button in the 'Create a new fork' sidebar on the right, which is also highlighted with a yellow box.

Create a new fork

A fork is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (*).

Owner * / Repository name *
Python_Sorting_Algorithm

Python_Sorting_Algorithms is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)

Python - Sorting Algorithms

☒ Copy the master branch to your default branch. Contribute back to B3ns44d/Python_Sorting_Algorithms by adding your own branch. [Learn more.](#)

① You are creating a fork in your personal account.

Create fork

GitHub Fork & Clone

- Clone 수행
 - git clone '~.git'
- 폴더 내 변경 진행
 - cd Python_Sorting_Algorithms
 - echo " " > file.txt
- Commit & Push
 - git add file.txt
 - git commit -m " "
 - git push origin master

```
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ git clone https://github.com/사용자 이름 Python_Sorting_Algorithms.git
Cloning into 'Python_Sorting_Algorithms' ...
remote: Enumerating objects: 43, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (29/29), done.
remote: Total 43 (delta 14), reused 39 (delta 11), pack-reused 0 (from 0)
Receiving objects: 100% (43/43), 10.47 KiB | 2.62 MiB/s, done.
Resolving deltas: 100% (14/14), done.

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW (master)
$ cd Python_Sorting_Algorithms

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/Python_Sorting_Algorithms (master)
$ echo 'clone succeeded' > file.txt

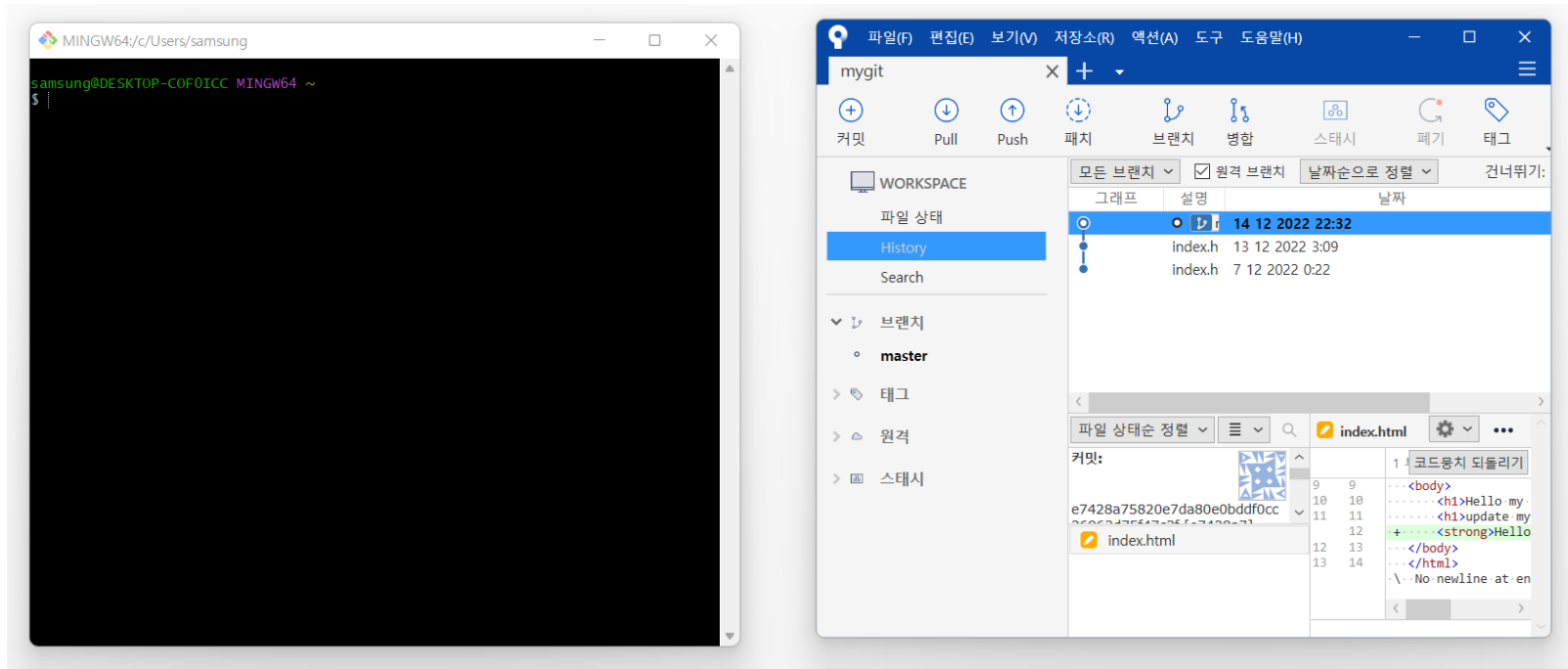
iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/Python_Sorting_Algorithms (master)
$ git add file.txt
warning: in the working copy of 'file.txt', LF will be replaced by CRLF the next
time Git touches it

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/Python_Sorting_Algorithms (master)
$ git commit -m 'add file.txt'
[master 3717c11] add file.txt
1 file changed, 1 insertion(+)
create mode 100644 file.txt

iip1@DESKTOP-NU4IEFG MINGW64 ~/Desktop/OSSW/Python_Sorting_Algorithms (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 281 bytes | 281.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/사용자 이름 Python_Sorting_Algorithms.git
cf6c6d2..3717c11 master -> master
```

SourceTree

- Git을 GUI(Graphical User Interface)로 사용하는 방식
- Git bash를 통한 명령어 입력 없이 직관적인 버튼 클릭만으로 Git 활용이 가능



좌측 CUI(Command User Interface)와 우측 GUI

GitKraken

- Git을 GUI(Graphical User Interface)로 사용하는 방식
- SourceTree와는 달리 PR 및 코드 리뷰까지 함께 제공하여 협업에 강점을 가짐
- SourceTree는 windows/macOS에서만 지원하며, GitKraken은 Linux도 지원

