

깃허브 가이드(Windows)

| | |
|-------|-------------------------|
| 🕒 생성일 | @2023년 11월 28일 오전 12:45 |
| ☰ 태그 | |

깃과 깃허브를 이용한 스터디 학습파일 제출 방법에 대한 가이드입니다.

깃 설치, 깃허브 가입, 토큰 발급, 레포지토리 포크 및 제출 방법 등을 설명합니다.

1. Git 설치
2. GitHub 가입
3. 토큰 발급
4. 스터디 학습파일 제출

깃이 설치되어 있고, 본인의 컴퓨터에서 깃 푸시를 진행해보신 분들은 4번 목차부터 보시면 됩니다.

깃(Git)은 코드 작성 시 파일을 관리하는 데 도움이 되는 프로그램으로, 파일의 수정 내용에 대해 버전을 부여하고, 이전 버전으로 파일을 이동시킬 수 있으며(branch를 통해 차원 분리)여러 명이 동시에 작업할 때 충돌을 방지할 수 있습니다.

깃은 혼자서 혹은 여러 사람과 함께 프로젝트를 진행할 때 안전하고 편리한 방법입니다.

깃허브(GitHub)는 깃이 사용되는 파일들을 웹사이트에서 관리해주는 저장소입니다.

1. Git 설치

Windows

<https://taewow.tistory.com/13> < 끝까지 읽어주세요!!

- 위 사이트와 같이 설치를 진행합니다. 내용을 전부 이해하지 않아도 되니 설치 방법만 참고합니다.
- 마지막에 입력하는 global name과 email은 저장소 GitHub와 별개로 깃 프로그램 자체에 입력되는 로컬 이름과 이메일입니다. 내 컴퓨터에서 깃 프로그램을 이용할 때는 해당 이름과 이메일로 깃 이력이 남습니다. 재입력이 가능하므로, 잘못 입력한 경우 git

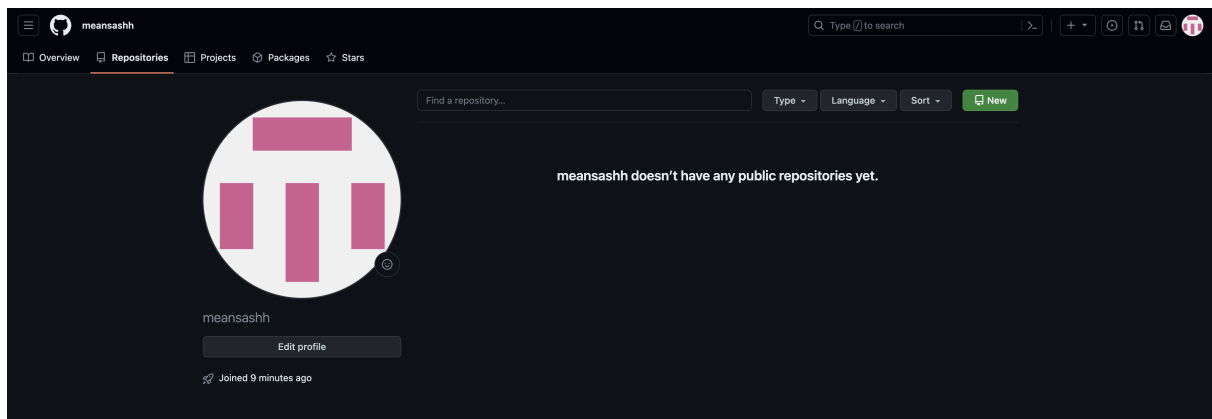
config user.name과 git config user.email로 내가 입력한 이름과 이메일을 확인할 수 있습니다. 확인 후, 이전과 같은 방법으로 재입력합니다.

2. GitHub 가입

<https://github.com/>

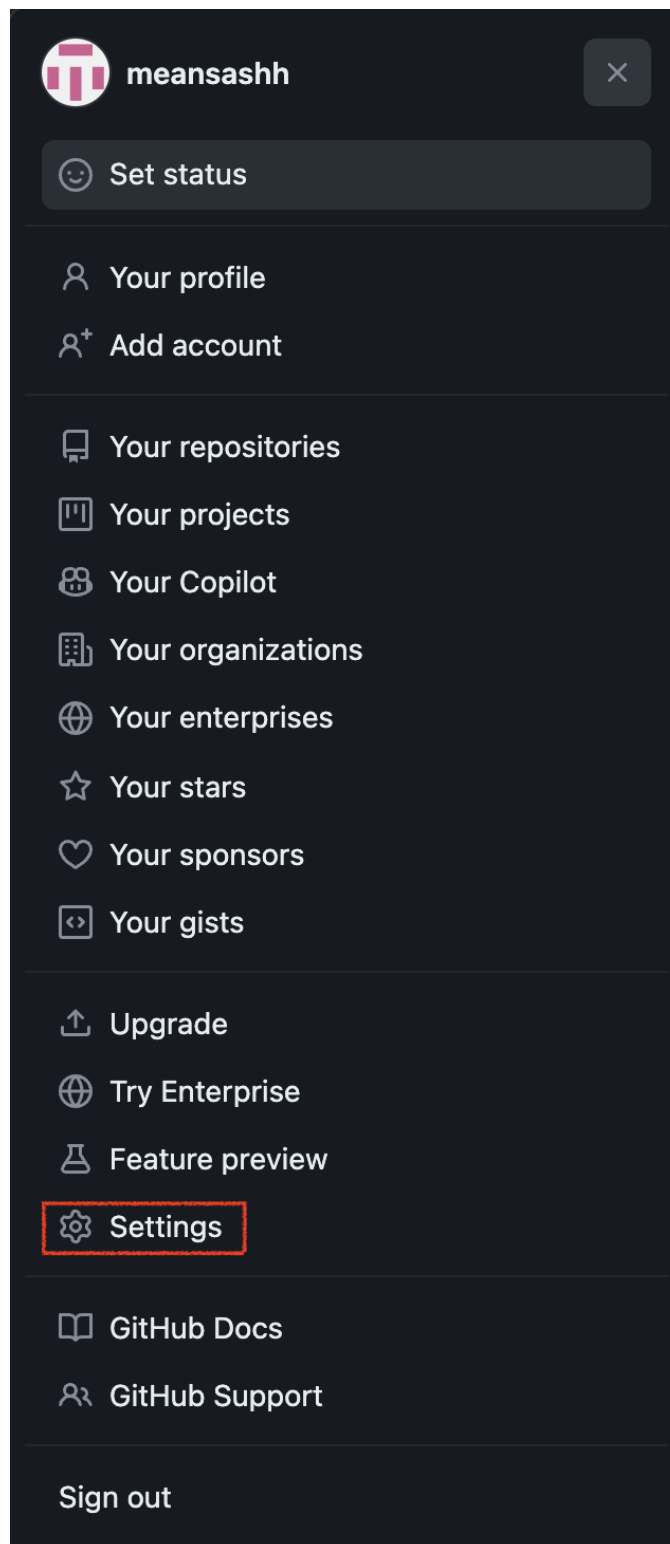
- 위 사이트에 들어가서 가입합니다.

가입 완료 시 화면(오른쪽 위 로고 버튼 클릭 후 your repositories 눌러도 나옵니다.)



3. 토큰 발급

- 컴퓨터에 토큰을 입력해야 합니다. 깃허브에서 발급받은 토큰을 등록된 컴퓨터에서만 저장소에 접근할 수 있습니다.
 - 먼저, 깃허브에서 토큰을 생성합니다. 다음 그림대로 진행합니다.



Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

Sessions

SSH and GPG keys

Organizations

Enterprises

Moderation

Code, planning, and automation

Repositories

Codespaces

Packages

Copilot

Pages

Saved replies

Security

Code security and analysis

Integrations

Applications

Scheduled reminders

Archives

Security log

Sponsorship log

Developer settings

Name

Your name may appear around GitHub where you contribute or are mentioned. You can remove it at any time.

Public email

Select a verified email to display

You have set your email address to private. To toggle email privacy, go to [email settings](#) and uncheck "Keep my email address private."

Bio

Tell us a little bit about yourself

You can @mention other users and organizations to link to them.

Pronouns

Don't specify

URL

Social accounts

Link to social profile

Link to social profile

Link to social profile

Link to social profile

Company

You can @mention your company's GitHub organization to link it.

Location

☐ Display current local time

Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our [privacy statement](#) to learn more about how we use this information.

Update profile

Profile picture

Edit

GitHub Apps

OAuth Apps

Personal access tokens

GitHub Apps

New GitHub App

Want to build something that integrates with and extends GitHub? [Register a new GitHub App](#) to get started developing on the GitHub API. You can also read more about building GitHub Apps in our [developer documentation](#).

GitHub Apps

OAuth Apps

Personal access tokens

1 Fine-grained tokens

2 Tokens (classic)

Personal access tokens (classic)

3

Generate new token (classic)

For general use

Need an API token for scripts or testing access to the [GitHub API](#).

Personal access tokens (classic) function like a password for Git over HTTPS, or for [Authentication](#).

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

2024_personal_token

Expiration *

Custom...



2025. 03. 01.



(무기한으로 만료기간을 설정하는 것은 권장되지 않으며, 보안상 위험합니다. 따라서 일단 1년으로 설정합니다. 1년 후에는 내 컴퓨터에 토큰을 새로 등록해야만 깃허브에 접근할 수 있습니다.)

Select scopes

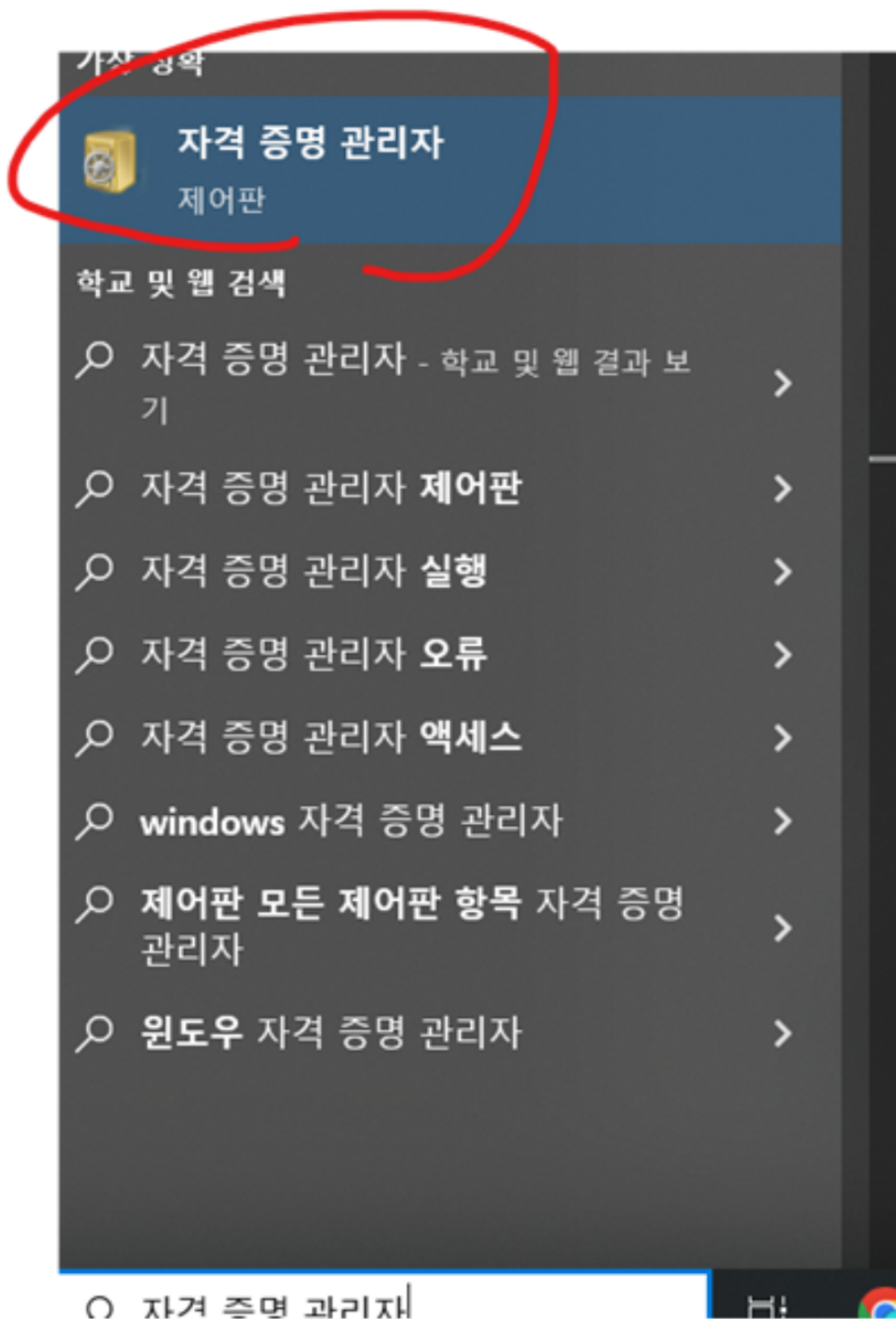
Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

| | |
|---|---|
| <input checked="" type="checkbox"/> repo | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status | Access commit status |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status |
| <input checked="" type="checkbox"/> public_repo | Access public repositories |
| <input checked="" type="checkbox"/> repo:invite | Access repository invitations |
| <input checked="" type="checkbox"/> security_events | Read and write security events |
| <input type="checkbox"/> workflow | Update GitHub Action workflows |
| <input type="checkbox"/> write:packages | Upload packages to GitHub Package Registry |
| <input type="checkbox"/> read:packages | Download packages from GitHub Package Registry |
| <input type="checkbox"/> delete:packages | Delete packages from GitHub Package Registry |
| <input type="checkbox"/> admin:org | Full control of orgs and teams, read and write org projects |
| <input type="checkbox"/> write:org | Read and write org and team membership, read and write org projects |
| <input type="checkbox"/> read:org | Read org and team membership, read org projects |
| <input type="checkbox"/> manage_runners:org | Manage org runners and runner groups |
| <input type="checkbox"/> admin:public_key | Full control of user public keys |
| <input type="checkbox"/> write:public_key | Write user public keys |
| <input type="checkbox"/> read:public_key | Read user public keys |
| <input type="checkbox"/> admin:repo_hook | Full control of repository hooks |
| <input type="checkbox"/> write:repo_hook | Write repository hooks |
| <input type="checkbox"/> read:repo_hook | Read repository hooks |
| <input type="checkbox"/> admin:org_hook | Full control of organization hooks |
| <input type="checkbox"/> gist | Create gists |
| <input type="checkbox"/> notifications | Access notifications |
| <input checked="" type="checkbox"/> user | Update ALL user data |
| <input checked="" type="checkbox"/> read:user | Read ALL user profile data |
| <input checked="" type="checkbox"/> user:email | Access user email addresses (read-only) |
| <input checked="" type="checkbox"/> user:follow | Follow and unfollow users |

Select scopes 항목은 repo와 user 만 체크해주시면 됩니다 😊

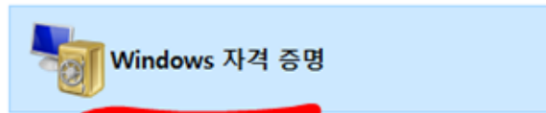
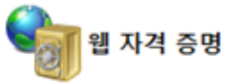
토큰을 생성하고 나면 토큰페이지가 나오는데, 그냥 넘어가면 다시 볼 수 없으므로, 해당 페이지에서 바로 토큰을 메모장이나 다른 곳에 복사해둡니다.

토큰 발급을 완료한 후에, 내 컴퓨터 윈도우에 토큰을 등록해줍니다. 다음 그림대로 진행합니다.



자격 증명 관리

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.



자격 증명 백업(B) 자격 증명 복원(R)

Windows 자격 증명

Windows 자격 증명 추가

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명

인증서 기반 자격 증명 추가

인증서가 없습니다.

일반 자격 증명

일반 자격 증명 추가

GitHub - <https://api.github.com/mount-ain>

수정한 날짜: 2022-05-17 (v)

GitHub - <https://api.github.com/sungsaaan>

수정한 날짜: 2021-09-12 (v)

git:<https://github.com>

수정한 날짜: 오늘 (v)

MicrosoftAccount:user=ejaoseja@naver.com

수정한 날짜: 오늘 (v)

일반 자격 증명 추가를 클릭한 뒤, 네트워크 주소에 `git:https://github.com`을 입력하고, 이름과 암호를 입력합니다. 이름은 자유롭게 설정 가능하고, 암호는 방금 발급받은 토큰을 입력해줍니다. 일반 자격 증명에 이미 `git:https://github.com` 가 있을 경우, 해당 자격 증명에서 이름과 암호만 변경해줍니다.

이제 토큰 등록이 완료됐습니다. `git bash`를 켜다가 새로 열어줍니다.

4. 스터디 학습파일 제출

✅ 깃허브에 깃푸시를 진행해본 분은 여기부터 진행해주세요 😊

- 각 스터디 레포지토리에 해당하는 주소로 이동합니다.

C - https://github.com/meansash/Amicom_C_Study

Python - https://github.com/meansash/Amicom_Python_Study

Algorithm - https://github.com/meansash/Amicom_Algorithm_Study

- 아래 화면의 **Fork** 를 클릭합니다.

The screenshot shows the GitHub interface for the repository 'Amicom_Algorithm_Study' by user 'meansash'. The repository is public. At the top, there are navigation tabs: Code, Issues, Pull requests, Actions, Projects, Wiki, Security, and Insights. Below the repository name, there are buttons for Pin, Unwatch (1), Fork (0), and Star (0). The 'Fork' button is highlighted with a red box. Below the repository name, there is a section for the main branch 'main' with a 'Go to file' button and a 'Code' button. The commit history shows an update to 'README.md' by 'meansash' 1 hour ago. The README content is displayed, showing the title 'Amicom_Algorithm_Study' and the text 'Amicom 알고리즘 스터디입니다😊'. On the right side, there is an 'About' section with a description 'No description, website, or topics provided.' and a list of links: Readme, Activity, 0 stars, 1 watching, and 0 forks. Below the 'About' section, there is a 'Releases' section with the text 'No releases published' and a link 'Create a new release'. At the bottom, there is a 'Packages' section with the text 'No packages published' and a link 'Publish your first package'.

- 아래와 같이 설정 후 **Create fork** 를 진행합니다.

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.

Required fields are marked with an asterisk (*).

Owner *

Choose an owner ▼

Repository name *

Amicom_Algorithm_Stu

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

Description (optional)

☒ Copy the `main` branch only

Contribute back to meansash/Amicom_Algorithm_Study by adding your own branch. [Learn more.](#)

Create fork

- Owner 부분에는 본인의 아이디를 선택해주세요.
- `Fork` 를 진행하면 나의 아이디에 해당 레포지토리 내용이 복사된 새로운 레포지토리가 생성됩니다.

Git bash를 열고 아래 명령어를 차례대로 작성합니다.

- `git clone https://github.com/본인깃허브아이디/포크한레포지토리이름.git`
- `cd 포크한레포지토리이름`
- `git checkout -b 본인깃허브아이디`

`pwd` 명령어를 사용하여 저장 경로를 확인합니다.

저장 경로 안에다 아무 파일(README.md 등...) 내용을 변경 혹은, 메모장 파일 하나를 추가한 뒤

- `git add .`

(꼭 add뒤에 스페이스바 한번 해주신 뒤 . 찍어주세요)

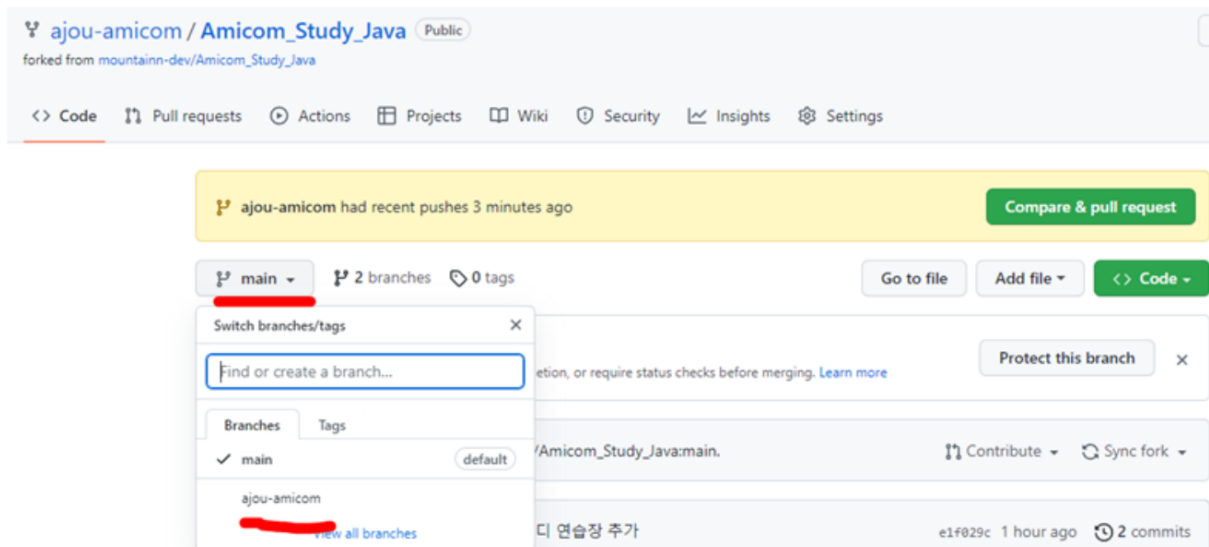
- `git commit -m "[docs] Study_언어_n주차"`

ex) `git commit -m "[docs] Study_Python_1주차"`

- `git push origin 본인깃허브아이디`

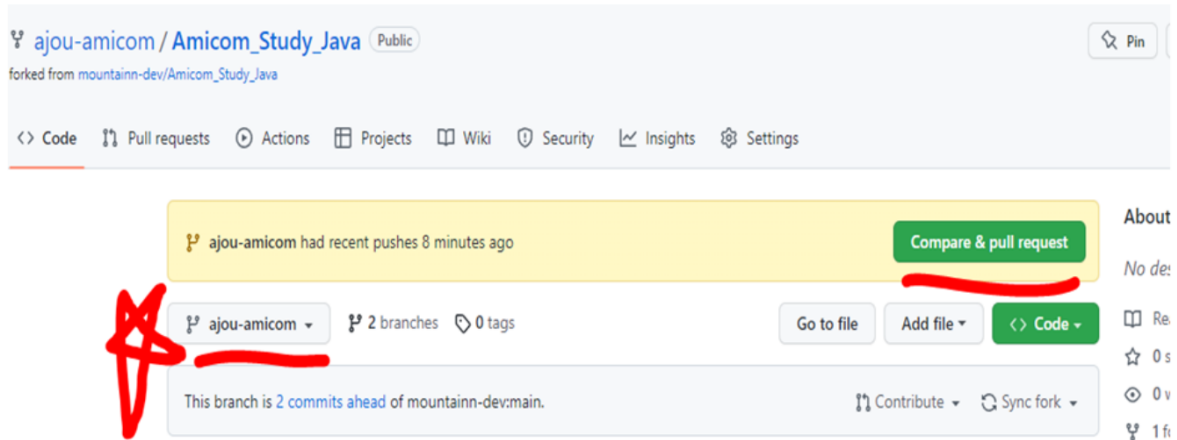
위 명령어를 차례로 실행해주세요.

- 푸시를 완료하면 다음 그림과 같이, main 밑에 본인 아이디의 새로운 branch가 생성됩니다.
- 해당 브랜치를 클릭하여 브랜치를 변경해줍니다.



- 이제 pull-request 를 진행합니다. Pr은 내가 Fork해온 레포지토리 원본 저장소에 코드 리뷰를 요청하는 것이라고 보면 됩니다. 아래 그림대로 진행합니다. 먼저 Pull request 를 클릭합니다.

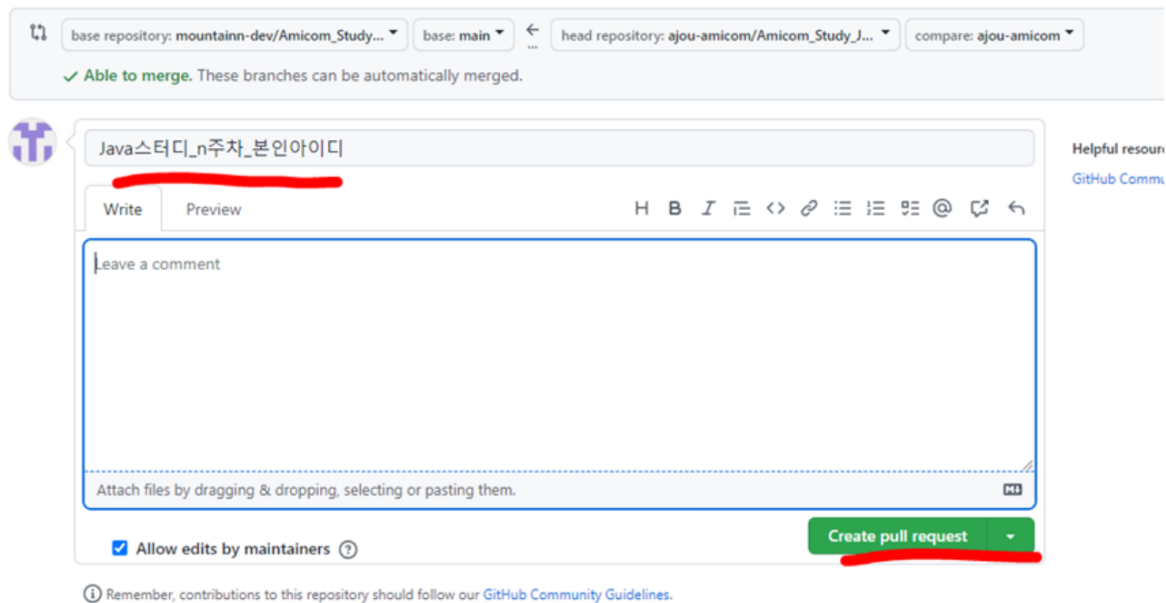
! 반드시 현재 브랜치가 `main` 이 아닌 새로 생성한 브랜치로 변경되어 있어야 합니다.



- 제목을 아래와 같이 스터디명_본인아이디로 작성하고 제출을 완료합니다.

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](#).



최초 제출때만 4번 항목 내용 진행, 그 이후엔 제출시 아래 내용만 진행

2회차(최초 제출 이후)부터는, 매 주차마다 학습 후 터미널에서

- `git add .`
- `git commit -m "[docs] Study_언어_n주차"`
- `git push origin 본인깃허브아이디`

위 명령어만 실행해주시면 됩니다.

고생하셨습니다😊