

단위문제명	단계1: 반달곰 커피 IT 혁신팀의 첫걸음		
분야	컴퓨터공학	구분	DevOps일

무엇을 위함인가? Git 숙달!
 뭘 해야 하는가? GitHub!

수행과제

문제1 커피를 좋아하는 동료들을 위해서

Github 공식 문서 사이트에서 시작하기(빠른시작) 내용을 확인한다.

Github 사이트에 가입하고 로그인 한다.

Github의 유료 요금제와 무료 요금제의 차이점을 확인한다.

GitHub에서 토큰을 생성한다.

이후에 토큰을 사용 확인 할 수 있게 텍스트 파일로 저장해둔다. 인증 후에 텍스트 파일은 삭제한다. -> 오프라인 / 폰 등에 메모해 두기 □ (비번 대신 사용 가능함!)

토큰 정보를 PC에 저장하여 GitHub 자동 로그인 되도록 설정한다.

- > 확인할 것! □
- > 선발과정 아이디 기억할 것! □
- > GPT에게 물어보거나 검색! □
- > 설정 가서 생성하기! GPT! □

-> 필요할 때 하면 됩니다! □

문제2 함께 일하는 즐거움을 위해서..

GitHub에서 원격 저장소를 david라는 이름으로 새로 만든다.

app.py가 저장되어 있는 PC의 작업 디렉토리로 이동하여 원격 저장소를 origin이라는 이름으로

추가한다. app.py -> 선발과정에서 만들었던 거 갖다 쓰기!

PC의 저장소 main 브랜치의 커밋 내역을 원격저장소의 main 브랜치로 올린다.

PC의 저장소 add-image 브랜치의 커밋 내역을 원격저장소의 add-image 브랜치로 올린다.

추가한 두 개의 브랜치가 원격저장소에 잘 전달되었는지 확인한다.

-> github.com (웹)에서 진행하면 됩니다! □

-> 요건 terminal에서 git 명령어 써서 진행 □

Git 원격 저장소 주소 등록(terminal):

↳ git remote set-url origin https://github.com/user/[변경할repo.git]

-> (터미널) git commit -m "commit 메시지" □

-> (터미널) git push origin main □

-> github에서 확인 □

단위문제명	단계1: 반달곰 커피 IT 혁신팀의 첫걸음		
분야	컴퓨터공학	구분	DevOps일

문제3 소스코드 관리는 꼭 필요한 것만

GitHub david 원격 저장소에서 .gitignore 파일을 추가한다.

.gitignore 파일을 아래 내용으로 편집한다.

__pycache__

.venv

원격 저장소로부터 코드를 내려받기 한다. (터미널)-> git pull origin main

GitHub david 원격 저장소에 위 두개의 .gitignore에서 지정한 **pycache**, .venv 디렉토리가 있다면

삭제한다.

변경사항을 Git 커밋 후 원격저장소의 main 브랜치로 다시 올린다.

1. .gitignore 생성 및 편집

```
bash
cd ~/your/project/path # 프로젝트 경로로 이동
echo "__pycache__/" >> .gitignore
echo ".venv/" >> .gitignore
```

3. .gitignore 에서 지정한 폴더 삭제

GitHub 원격 저장소에 이미 올라가 있다면 **tracking**을 끄고 삭제해야 합니다.

```
bash
# 캐시에서 제거
git rm -r --cached __pycache__
git rm -r --cached .venv

# 로컬에서도 삭제하려면 (선택 사항)
rm -rf __pycache__ .venv
```

4. 변경 사항 커밋

```
bash
git add .gitignore
git commit -m ".gitignore 추가 및 __pycache__, .venv 제거"
```

5. 원격 저장소(main 브랜치)에 다시 업로드

```
bash
git push origin main
```

보너스 과제

문제 1: 인기 있는 오픈소스 Fork 및 토큰 관리 보안

Star가 100개 이상 달린 타인의 GitHub 저장소를 본인의 GitHub 계정으로 Fork 한다.

Personal Access Token을 텍스트 파일로 저장한 뒤 삭제해야 하는 이유를 조사한다. ->GPT 및 구글링해서 공부하기 □

문제 2: 저장소 복제

다른 PC나 디렉토리에서 원격 저장소를 clone 한다.

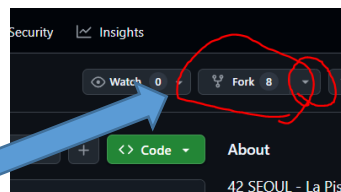
문제 3: 파이썬 프로젝트에서 .gitignore 사용 이유

파이썬 개발 시 __pycache__와 .venv 디렉토리가 생성되는 이유를 조사한다.

GitHub에서 .gitignore 템플릿 중 Python을 선택했을 때 포함되는 항목을 확인한다.

Flask 기반 프로젝트를 기준으로 .gitignore에 추가되어야 할 항목들을 나열한다.

아무 깃헙 리포지토리나 가서->



Bash: cd 원하는/디렉/토리 로 가서: □

\$ git clone https://github.com/YOUR-USERNAME/YOUR-REPOSITORY

GPT 및 구글링해서 공부하기 □

단위문제 소개

단위문제명	단계2 : 늘어나는 이슈들		
분야	컴퓨터공학	구분	DevOps일

수행과제

문제4. 협업 환경 구성 및 이슈 관리

GitHub 원격 저장소의 Collaborators에 팀원(또는 동료학습자)을 추가한다.

Milestone을 “메뉴판 추가”로 생성하고, 마감일은 1주일 후로 설정한다.

이슈 “메뉴 화면추가”를 등록하고, 본인을 Assignee, “메뉴판 추가”를 milestone으로 지정한다.

로컬에서 add-menu 브랜치를 생성하고 이동한다. (터미널) git branch add-menu

문제5. 메뉴 화면 추가 기능 구현 및 커밋

add-menu 브랜치에서:

app.py에 /menu 라우트를 추가한다. → 선발과정 app.py 적극 수정할 것!

templates/menu.html 파일을 생성하여 버튼 3개가 있는 메뉴 페이지를 작성한다. 동료교육생 / GPT에게 도움!

menu.html은 파일 이름으로, app.py는 hunk 단위로 스테이지에 추가한다. (터미널) git add 파일.이름 -p

다음 규칙에 따라 커밋한다:

제목: Feat: 메뉴 화면추가

본문: 추가한 기능 설명

꼬리말: Resolves: #1

(터미널) git commit -m "Feat: 메뉴 화면
추가
추가한 기능 설명
Resolves: #1 "

-> 따옴표를 닫지 않은 상태에서 커밋 메
시지를 입력해준 뒤에, 끝내기를 원하는
위치에서 따옴표를" 닫아준 뒤 엔터를 치
면 커밋 메시지가 종료됨!

I-nkamanda (작성자 ID)

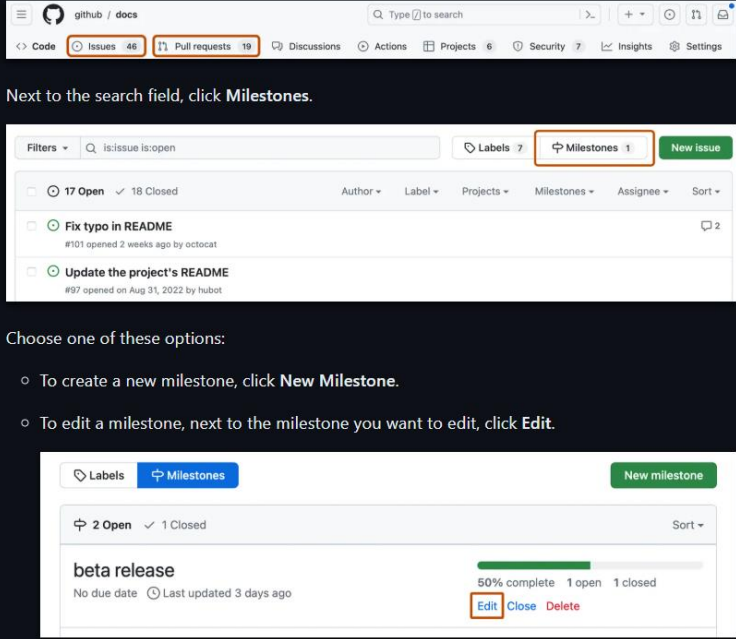
1 On GitHub, navigate to the main page of the repository.

2 Under your repository name, click Issues or Pull requests.

3 Next to the search field, click Milestones.

4 Choose one of these options:

- To create a new milestone, click New Milestone.
- To edit a milestone, next to the milestone you want to edit, click Edit.



단위문제 소개

단위문제명	단계2 : 늘어나는 이슈들		
분야	컴퓨터공학	구분	DevOps일

문제6. Pull Request 생성 및 반려 처리

add-menu 브랜치를 원격 저장소에 push하고 Pull Request(PR)를 생성한다.

제목: 메뉴 화면추가 완료

내용: 이슈 #1 검토 요청 드립니다.

milestone: 메뉴판 추가

PR에서 의견 "녹차 메뉴만 버튼 사이즈가 이상하네요."를 남기고 **Close with comment**로 반려한다.

문제7. 수정 후 PR 재등록 및 병합

menu.html에서 녹차 버튼의 사이즈를 70pt로 수정하고 커밋 후 push한다.

새로운 PR을 등록하고 "수정사항 확인했습니다."라는 댓글을 남긴다.

PR을 **Merge**하고, 로컬 main 브랜치에서 변경사항을 pull 한다.

웹브라우저에서 메뉴를 확인한 뒤, 해당 이슈를 커밋 번호와 함께 **Close with comment**로 종료한다.

보너스 과제

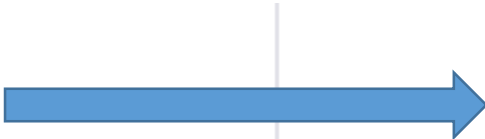
이슈별로 댓글을 남기고 완료 시 직접 Close 해본다.

프로젝트 내용을 설명하는 README.md를 작성하여 저장소 루트에 추가한다.

저장소를 **Private** → **Public**으로 전환해 접근성을 테스트한다.

전체 작업을 **Visual Studio Code**에서 다시 수행하고 CLI 방식과 비교한다.

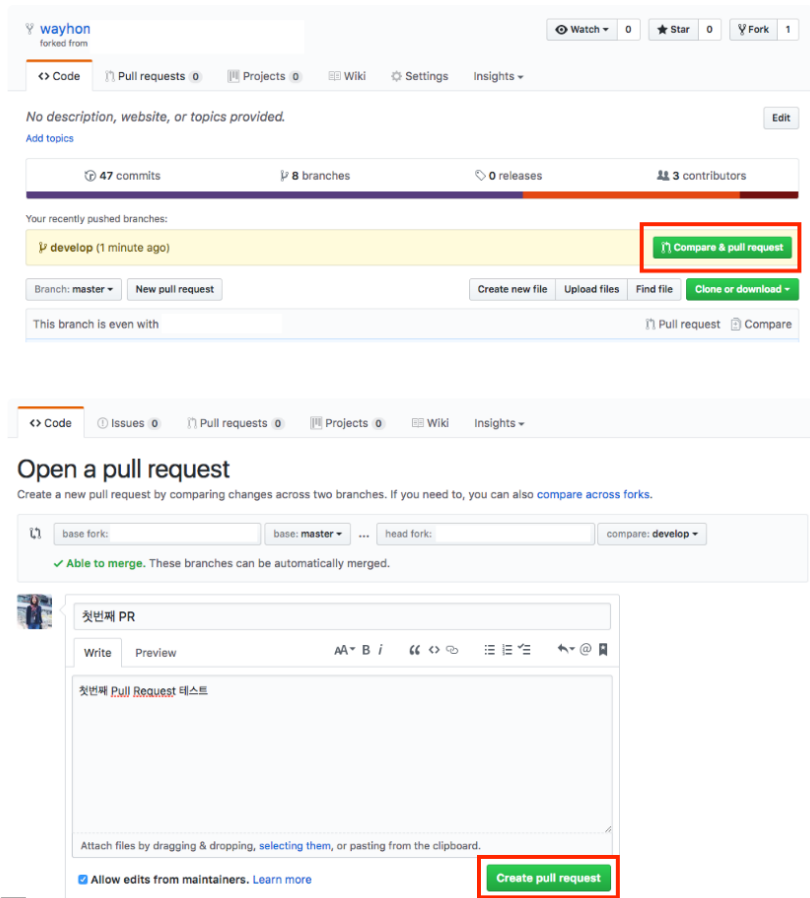
GitHub **Wiki**에 프로젝트 설명을 5줄 이상 기술한다.



막히는 게 있다면:
동료 <-> GPT!

5. Pull Request 생성

- push 완료 후 본인 계정의 github 저장소에 들어오면 **Compare & pull request** 버튼이 활성화 되어 있다.
- 해당 버튼을 선택하여 메시지를 작성하고 PR을 생성한다.



- > issue 들어가서 직접
- > github.com 들어가면 보임
- > 평가자에게 잘 설명할 수 있게 공부!
- > Github CLI 깔아서 진행해보기
- > github.com 들어가서 할 수 있습니다