

# githube란?

GitHub(깃허브)는 개발자들이 코드를 저장하고, 공유하며, 함께 작업할 수 있도록 돕는 웹 기반 플랫폼이에요. 마치 개발자들을 위한 클라우드 저장소이자 소셜 네트워크 같은 곳이라고 생각할 수 있습니다.

핵심은 **\*\*Git(깃)\*\***이라는 **버전 관리 시스템**을 온라인으로 편리하게 사용할 수 있게 해주는 점이에요.

---

## 깃허브의 주요 기능

### 1. 코드 저장소 관리 (Repository)

- **저장소(Repository, 레포지토리):** 프로젝트의 모든 파일(코드, 문서, 이미지 등)과 변경 이력을 한데 모아 관리하는 공간이에요.
- **버전 관리:** 코드가 변경될 때마다 그 기록을 남겨요. 언제, 누가, 무엇을 바꿨는지 알 수 있고, 필요할 때는 **언제든지 이전 상태로 되돌아갈 수 있습니다.**

### 2. 협업 및 코드 공유

- **Fork (포크):** 다른 사람의 공개된 저장소를 내 계정으로 통째로 복사해오는 기능이에요. 원본에 영향을 주지 않고 나만의 방식으로 자유롭게 수정할 수 있습니다.
- **Pull Request (풀 리퀘스트, PR):** 내가 수정한 코드를 원본 저장소에 반영해달라고 요청하는 기능이에요. 이 과정에서 다른 팀원들이 코드를 **\*\*검토(코드 리뷰)\*\***하고 토론하여, 더 좋은 코드를 만들고 실수를 줄일 수 있습니다.
- **코드 리뷰:** PR 과정에서 동료 개발자가 변경된 코드를 확인하고 의견을 제시하며 품질을 개선하는 작업입니다.

### 3. 프로젝트 관리

- **이슈 트래킹 (Issues):** 버그 신고, 새로운 기능 제안, 개선 요청 등 프로젝트에서 해결해야 할 문제들을 기록하고 추적하는 기능입니다. 라벨을 붙이거나 담당자를 지정하여 체계적으로 관리할 수 있습니다.
  - **프로젝트 보드 (Project Boards):** 칸반 보드(Kanban Board)와 같은 형태로 현재 진행 중인 작업의 상태(할 일, 진행 중, 완료 등)를 시각적으로 관리할 수 있습니다.
-

# 1. Git 설치하기

Git을 설치하고 GitHub 같은 온라인 저장소(원격 저장소)와 내 컴퓨터의 폴더(로컬 저장소)를 연결하는 방법은 다음과 같습니다. 마치 내 컴퓨터의 파일을 클라우드에 백업하고 여러 명이 함께 작업할 수 있게 하는 과정이라고 생각하시면 돼요.

가장 먼저, **Git**이라는 프로그램이 컴퓨터에 설치되어 있어야 합니다. Git은 파일의 변화를 추적하고 관리해주는 도구입니다.

1. **Git 다운로드:** 웹 브라우저에서 ""를 검색하여 공식 Git 웹사이트로 이동합니다.
2. 사용하는 운영체제(Windows, macOS, Linux)에 맞는 설치 파일을 다운로드하고 실행합니다.
3. 설치 과정 중 특별히 설정을 바꾸지 않아도 되지만, 기본 설정을 따르면서 설치를 완료합니다.

설치가 끝났으면, 컴퓨터의 **터미널** 또는 **명령 프롬프트**를 열고 다음 명령어를 입력하여 Git이 제대로 설치되었는지 확인합니다.

```
git --version
```

버전 정보가 나타나면 성공입니다.

---

## ~~2. GitHub에 온라인 저장소 만들기~~ (추후 본인 거 만들때, 다른 저장소와 연결할때는 2단계 필요 X)

협업과 백업을 위해 코드를 올려둘 온라인 공간, 즉 **\*\*원격 저장소(Repository)\*\***를 GitHub에서 만듭니다.

1. **GitHub 웹사이트 접속:** GitHub ()에 접속하여 로그인합니다.
2. 화면 오른쪽 상단에서 '+' 아이콘을 클릭하고, **\*\*'New repository'\*\***를 선택합니다.
3. 다음 정보를 입력합니다.
  - **Repository name (저장소 이름):** 프로젝트 이름을 입력합니다. (예: `my-first-project`)
  - **Description (설명):** 프로젝트에 대한 간단한 설명을 작성합니다.
  - **Public/Private (공개/비공개):** 필요에 따라 선택합니다. (일반적으로 공개를 많이 합니다.)
4. **'Create repository'** 버튼을 클릭하여 저장소를 만듭니다.

저장소가 생성되면, GitHub는 이 저장소에 접속할 수 있는 고유한 주소(URL)를 보여줍니다. 이 주소가 로컬 저장소와 연결하는 데 필요합니다.

### 3. 오프라인(로컬) 저장소 준비 및 연결 🤝

이제 내 컴퓨터의 작업 폴더를 온라인 저장소와 연결할 차례입니다.

#### 3.1. 로컬 저장소 만들기

1. 내 컴퓨터에서 프로젝트를 진행할 폴더를 만듭니다. (예: `C:\Users\YourName\Desktop\my-first-project`)
2. 해당 폴더 안으로 이동하여 마우스 오른쪽 버튼을 클릭하고 '**Git Bash Here**' (Windows의 경우) 또는 터미널을 열어 폴더 위치로 이동합니다.
3. 이 폴더를 Git으로 관리하겠다고 선언합니다.

```
git init
```

이 명령어를 실행하면 폴더 안에 숨겨진 `.git` 폴더가 생겨나면서, 이 폴더가 이제 **로컬 저장소**가 됩니다.

#### 3.2. 온라인 저장소 주소 연결

Git에게 "내가 만든 온라인 저장소가 여기 있다"라고 알려줍니다.

1. GitHub에서 만든 저장소 페이지로 돌아가서, '**< > Code**' 버튼을 누르고 주소(URL)를 복사합니다. (보통 **HTTPS** 주소를 사용합니다.)
2. 로컬 저장소 폴더에서 다음 명령어를 입력합니다. 여기서 `[복사한 주소]` 부분에 GitHub 주소를 붙여 넣고, `origin` 은 이 원격 저장소의 별명입니다.

```
git remote add origin [복사한 주소]
```

#### 3.3. 파일 올리기 (Push)

이제 로컬 저장소의 파일을 온라인 저장소로 올릴(푸시, Push) 수 있습니다.

1. ~~로컬 폴더에 `README.md` 와 같은 파일을 만듭니다.~~(이미 온라인 저장소는 제가 만들었기 때문에 필요없습니다.)
2. Git에게 이 파일을 관리하라고 명령합니다. (**추적**)
3. 추적된 파일의 변경 내용을 하나의 버전으로 만듭니다. (**커밋**)

```
git add .
```

```
git commit -m "Initial project setup"
```

4. 이 버전을 온라인 저장소(`origin`)의 메인 가지(`main` 또는 `master`)로 올립니다. (**푸시**)

```
git push -u origin main
```

이 명령어를 실행하면, GitHub에 접속하여 이메일과 비밀번호(또는 개인 액세스 토큰)를 입력하라는 메시지가 나타날 수 있습니다. 인증을 완료하면, 로컬의 파일이 GitHub의 온라인

저장소로 전송되어 연결이 완료됩니다.