



git flow 기본 알려주기

■ 상태	시작 전
■ 마감일	@08/20/2025
👤 작업 유형	
■ 업데이트 시간	@2025년 8월 19일 오후 7:04

Git 가이드 📖

Git이란? 🤔

Git은 **버전 관리 시스템**입니다. 쉽게 말해서:

- 파일의 변경 이력을 추적하고 관리하는 도구
- 여러 사람이 함께 작업할 때 충돌을 방지하고 협업을 도와주는 시스템
- "최종.docx", "최종_진짜최종.docx" 같은 파일명 지옥에서 벗어날 수 있게 해주는 도구

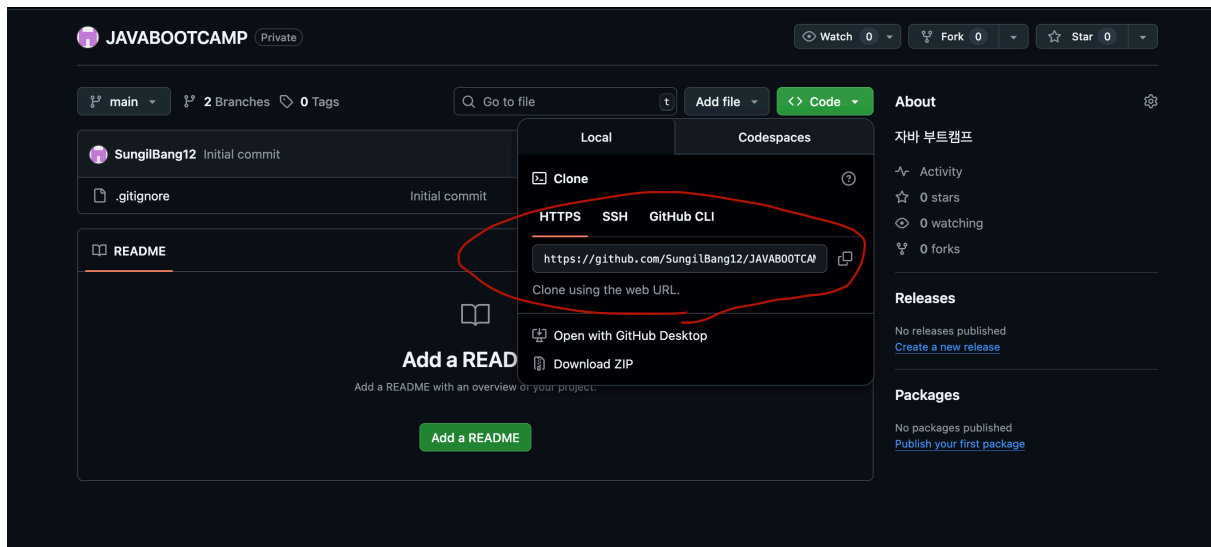
1단계: Git Bash 설치하기 🛠️

1. <https://git-scm.com/downloads> 접속
2. 운영체제에 맞는 버전 다운로드
3. 설치 프로그램 실행 (기본 설정으로 설치하면 됩니다)

2단계: GitHub에서 프로젝트 다운로드하기 📁

GitHub Repository URL 찾기

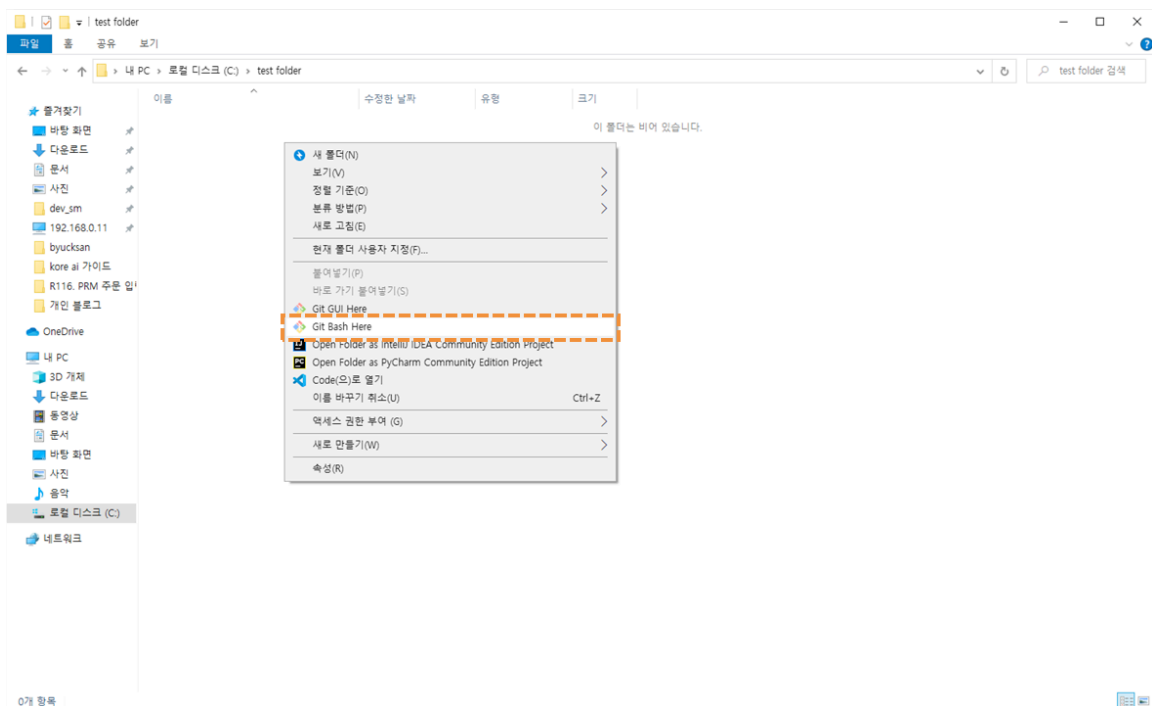
GitHub 저장소 페이지에서 초록색 "Code" 버튼 클릭
→ HTTPS 탭에서 URL 복사
예: <https://github.com/username/project-name.git>



Git Bash로 다운로드하기

1. 원하는 폴더에서 Git Bash 열기

- 폴더에서 마우스 우클릭 → "Git Bash Here" 선택



2. Git 저장소 초기화

git init

💡 이 명령어는 "이 폴더를 Git으로 관리하겠다"는 선언입니다. 처음 한 번만 실행하면 됩니다.

3. 원격 저장소 연결

```
git remote add origin [repository URL]
```

💡 "origin"은 원격 저장소의 별명입니다. GitHub 저장소와 연결한다는 의미입니다.

4. 파일 다운로드

```
git pull origin main
```

💡 GitHub의 main 브랜치에서 모든 파일을 내 컴퓨터로 가져옵니다.

3단계: 변경사항을 GitHub에 올리기

Git의 파일 업로드 과정을 이해해봅시다:

작업 폴더	→	Staging Area	→	Local Repository	→	Remote Repository
↓		↓		↓		↓
수정		git add		git commit		git push

단계별 실행

1. 수정된 파일을 Staging Area에 추가

```
# 특정 파일만 추가
git add 파일이름.java

# 모든 변경된 파일 추가
git add .
```

💡 Staging Area는 "업로드 준비실"이라고 생각하면 됩니다.

2. 변경사항을 Local Repository에 저장

```
git commit -m "변경 내용에 대한 설명"
```

⚠ 커밋 메시지는 구체적으로 작성하세요!

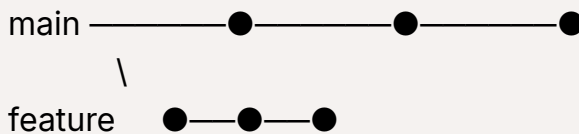
- 좋은 예: "로그인 기능 버그 수정"
- 나쁜 예: "수정", "1", "ㅇㄴㅇㄹ"

3. GitHub에 업로드

```
git push origin main
```

4단계: 브랜치 변경하기 🌿

브랜치는 작업의 분기점입니다. 마치 평행우주처럼 다른 버전의 프로젝트를 동시에 관리할 수 있습니다.



브랜치 변경 방법

```
git checkout 브랜치이름
```

⚠ 중요한 주의사항

브랜치를 변경하기 전에 **현재 작업을 정리**해야 합니다:

1. 모든 변경사항을 커밋: `git commit -m "메시지"`
2. 또는 GitHub에 푸시: `git push origin 현재브랜치`
3. 또는 임시 저장: `git stash` (나중에 `git stash pop` 으로 복원)

branch 관련 추가 명령어 정리 📋

명령어	설명
<code>git branch</code>	브랜치 목록 보기
<code>git branch 새브랜치명</code>	새 브랜치 만들기

트러블슈팅

"fatal: not a git repository" 에러

→ `git init` 을 먼저 실행하세요.

브랜치 변경이 안 될 때

→ `git status` 로 확인 후, 변경사항을 커밋하거나 stash하세요.

푸시가 안 될 때

→ `git pull origin main` 으로 최신 버전을 먼저 받아보세요.
