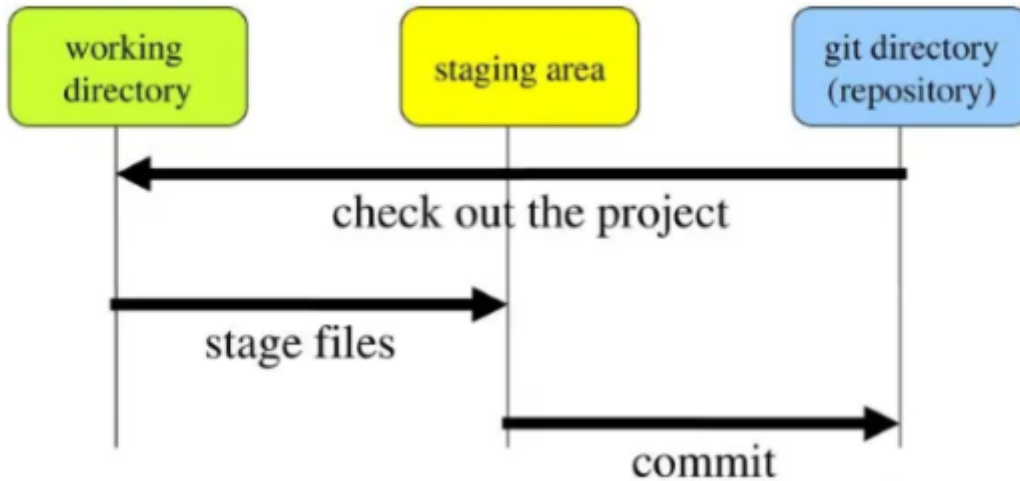


# GIT

버전 관리 툴

## GIT 관리 세가지 상태(status)

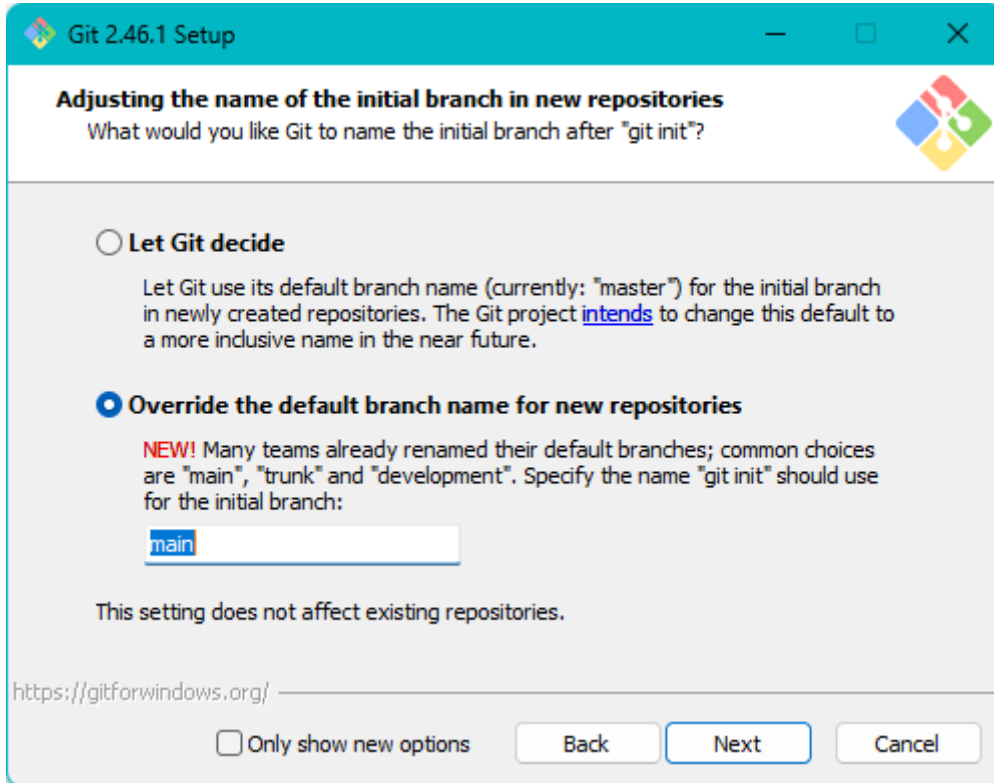
- Modified
- Staged
- Committed



- **modified:** working 디렉토리에 파일이 새로 생성되었거나 기존 관리 파일의 내용이 수정된 경우
- **staged:** commit 가능한 상태
  - working directory의 모든 파일이 형상관리 대상이 아닐 수 있다. 그래서 커밋 대상 파일을 먼저 선택한다.
  - staged 된 파일들은 staging area(index 라고도한다.)에 올려진다.
- **commit:** 변경된 내용을 Local repository에 저장한다.

# 설치

- <https://git-scm.com/download>
  - 운영체제에 맞춰 다운로드 및 설치
- 설치 시 **Adjusting the name of the initial branch in new repositories** 항목에서 **Override the default branch name for new repositories**를 선택한다.



- 나머지 항목은 **Next** 선택한다.

# 기본 명령어

- **git init**
  - 현재 폴더를 git을 통해 관리하는 working directory로 만든다.
  - .git (git local repository) 디렉토리가 생성된다.
- **git clone 원격저장소경로**
  - 원격 저장소(Remote Repository) 를 복사해서 Local 저장소를 만든다.
  - 이후 git push를 이용해 local 저장소에서 변경된 내용을 원격 저장소에 적용할 수 있다.
- **git add 파일**
  - 지정한 파일을 staged 상태로 만들어 준다.
  - staged 상태란 git에서 형상관리를 추적하는 대상파일을 말한다.
  - 파일 지정
    - 개별 파일 경로지정. **git add abc.txt**
    - wild card 문자 이용. **git add \***, **git add \*.txt**
- **git reset 파일명**
  - git add 취소 하여 unstaged 상태로 만든다.
  - 파일명을 생략하면 add된 모든 파일을 unstaged 상태로 변경한다.
- **git commit**
  - 커밋(최종 확정) 처리
  - staged 상태의 파일들을 커밋한다.
  - 커밋은 하나의 작업단위가 종료되면 처리한다.
  - 커밋 메시지를 작성하기 위해 editor 가 실행되며 메시지 작성후 저장하면 커밋처리됨.
  - **git commit -m "커밋 메시지"** 로 editor 없이 메시지를 작성할 수 있다.
- **git pull**
  - 원격저장소의 상태로 local 저장소로 다운로드
  - 작업을 시작 하기 전에 **가장 먼저 실행한다.**
- **git push 원격저장소 branch**
  - 원격저장소에 local 저장소의 커밋한 내용 올리기
  - ex) **git push origin main**

## gitignore파일

- Git을 이용한 관리 대상에서 제외할 파일들과 디렉토리들을 설정하는 파일
- 파일명: **.gitignore**
- 한 줄에 한개 파일 또는 한개 디렉토리를 지정한다.
  - 명시적으로 파일명 또는 디렉토리 이름을 작성한다.
  - 디렉토리를 지정할 경우 뒤에 **/** 를 붙인다.
    - ex) **env/**: env 디렉토리를 관리대상에서 제외
    - 디렉토리를 지정하면 그 안의 파일들 모두 관리 대상에서 제외된다.
- 파일을 지정할 때 **wild card 문자**를 이용해 확장자 기반으로 설정할 수 있다.
  - **\*.dat** : 확장자가 dat인 모든 파일들을 관리 대상에서 제외.
- 프로그래밍 언어 별로 .gitignore 파일을 생성해 주는 사이트
  - [gitignore.io](https://gitignore.io)
  - Github에서 Repository를 생성할 때 생성할 수 있다.