

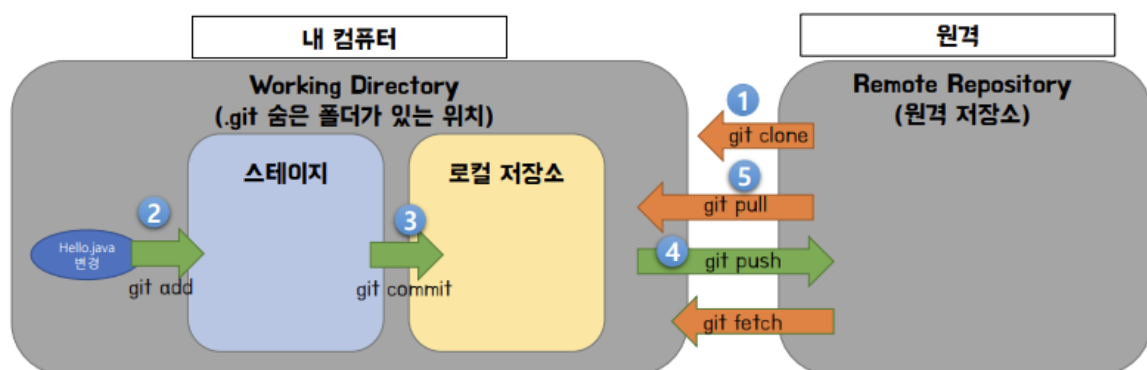
# GIT 기본 설명 및 사용법

☰ Subject	GIT
☰ index	4
📅 날짜	@2024/01/03
☰ 배운 내용 요약	GIT과 GITHUB의 개념, GITHUB 사용해보기, commit 같은 용어 설명
☼ 상태	완료

## Git이란?

- 버전 관리 시스템
- 개발자들이 서로 협업을 용이하게 할 수 있게 하는 수단
- 히스토리를 볼 수 있어서 필요 시 되돌릴 수 있음
- 작성한 코드를 리뷰 할 수 있음
- 대부분의 IDE에서 git을 연동할 수 있으며 자신의 GitHub 계정과 연동이 가능

## Git의 작동 구조



- 내가 내 컴퓨터(로컬 저장소)에서 작업한 것을
- 원격 저장소로 올리거나(push)
- 원격 저장소에서 최신 버전을 내 작업 공간(로컬저장소)으로 당겨옴(pull)

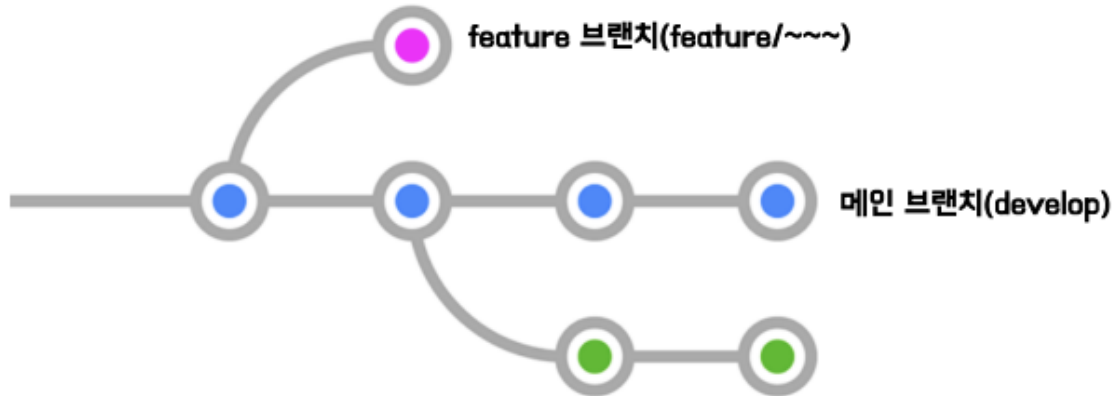
## CLI 와 GUI

- CLI : Command Line Interface(명령줄 인터페이스)의 줄임말로 글자를 입력해 컴퓨터에게 명령을 내리는 방식
- GUI : Graphical User Interface(그래픽 유저 인터페이스)의 줄임말로 CLI와 다르게 사용자가 보기 쉽게 그래픽으로 이루어짐

## 기본적인 용어

- Working Directory
  - 작업자의 현재 시점
  - 파일 수정, 저장 등의 작업을 하는 디렉토리
  - 작업 디렉터리라고도 함
- Repository(저장소)
  - 크게 2가지로 나뉨
  - Remote Repository(원격 저장소)
    - 소스 코드는 원격 저장소에서 관리되며 여러 사람이 함께 공유가 가능함
  - Local Repository(로컬 저장소)
    - 내 PC에 파일이 저장되는 개인 전용 저장소
    - 저장소를 만드는 방법은 다음과 같음
      1. 저장소를 로컬에 새로 만듦
      2. 이미 만들어져있거나 방금 만든 원격 저장소를 로컬 저장소로 Clone(복사)함

- Branch(브랜치)
  - 가지 혹은 분기점을 의미
  - 작업할 때 메인 브랜치에서 뿔어나와 브랜치를 생성하고, 해당 브랜치에서 작업한 후에 작업이 완료되면 메인 브랜치로 Merge(병합)하여 개발을 완료함



- Merge(병합)
  - 다른 브랜치의 내용을 현재 브랜치로 가져와 합치는 작업을 의미함
- Commit(커밋)
  - 현재 변경된 작업을 저장소에 저장하는 작업
- Checkout (= Switch)
  - 다른 Branch 작업을 불러오는 것
- pull VS fetch
  - pull : 변경 사항을 내 컴퓨터에 가져옴
  - fetch : 변경 사항을 내 컴퓨터에 가져오지 않음
- Pull Request(PR)
  - 바로 merge하면 굉장히 위험하기에 생김
  - PR을 하면 다른 이들이 코드를 보고 merge해도 문제가 없다고 생각되면 request를 승인함
  - 일정한 수의 승인을 받으면 merge가 됨

# Git의 역할

소스 코드를 관리할 수 있음. 기능으로는 아래 명령어들이 존재함

- 소스 병합 (merge, rebase)
- 소스 리비전 관리 (reset, commit, branch)
- 소스 릴리즈 (push)
- 소스 태깅 (tag)
- 소스 변경사항 검토 (diff, log)

## GitHub란?

- Git을 기반으로 온라인으로 서비스하는 형태
- Git과 GitHub는 다른 것
- 개발자들은 매우 자주 사용함

GitHub는 GitHub 사이트에서 관리가 가능하기도 하고 SourceTree와 같은 프로그램을 통해서 관리가 가능하기도 함

우리는 SourceTree를 사용했음

## Git의 기능, 명령

Git은 터미널 혹은 GUI 두 가지 방법으로 관리를 할 수 있다

터미널 명령어의 경우 다음과 같다

- 초기화 관련 명령어
  - git config —global user.name “사용자 이름” : 사용자 이름을 변경하는 명령어
  - git config —global user.email 메일주소 : 사용자 메일 주소를 변경하는 명령어
  - git init : git을 초기화 하는 명령어(단 GitHub에서 클론을 받은 경우에는 이 명령어를 쓰지 않아도 됨)

- git remote add origin 깃경로 : git을 원격 저장소에 저장하는 엔드포인트인 git 리모트 설정
- git clone 깃경로 : 원격 저장소에 저장된 파일을 컴퓨터로 복사하는 명령어
- 프로젝트 변경 사항 확인 및 추가 관련 명령어
  - git status : git의 변경 사항을 확인하는 명령어
  - git add 코드 : 파일을 추가하는 명령어
  - git add : 모든 파일을 추가하는 명령어
  - git commit : 변경 사항을 commit하는 명령어
    - 만약 commit시 바로 commit 메시지를 작성하고 싶다면 다음 명령어를 사용
      - git commit -m “쓰고 싶은 메시지”
  - git log : 현재 commit한 로그를 확인하는 명령어
- (참고) Vi 입력 모드

작업	Vi 명령어	상세
입력 시작	i	명령어 입력 모드에서 텍스트 입력 모드로 전환
입력 종료	ESC	텍스트 입력 모드에서 명령어 입력 모드로 전환
저장 없이 종료	:q	
저장 없이 강제 종료	:q!	입력한 것이 있을 때 사용
저장하고 종료	:wq	입력한 것이 있을 때 사용
위로 스크롤	k	git log 등에서 내역이 길 때 사용
아래로 스크롤	j	git log 등에서 내역이 길 때 사용

GUI의 경우 눈으로 바로 보고 수행 가능함