

# GitHub 실습

덕성여자대학교 컴퓨터학과  
최승훈

# GitHub 중요 용어

---

- ❑ commit

- 어느 순간 작업 공간의 상태를 저장소에 저장하는 행위

- ❑ branch

- 테스트를 해보거나 새로운 기능을 개발하기 위해 새로 만든 독립적인 공간

- ❑ master branch

- 새로운 프로젝트(저장소)마다 만들어지는 기본 브랜치
  - 배포할 최종 산출물이 마무리되는 작업 공간

- ❑ merge

- 한 브랜치에서 완성된 작업을 가져와 다른 브랜치에 포함하는 과정

# GitHub 중요 용어

---

## ❑ pull request

- 한 브랜치에서 완료된 작업을 다른 사람이 리뷰하고 마스터 브랜치로 병합하도록 요청하는 것

## ❑ issue:

- 수정해야 하는 버그나 새로 개발해야 하는 기능에 대한 토론을 위해 제기하는 문제(이슈)

## ❑ fork

- 다른 사용자의 계정에 프로젝트 복사본을 만드는 과정

## ❑ clone

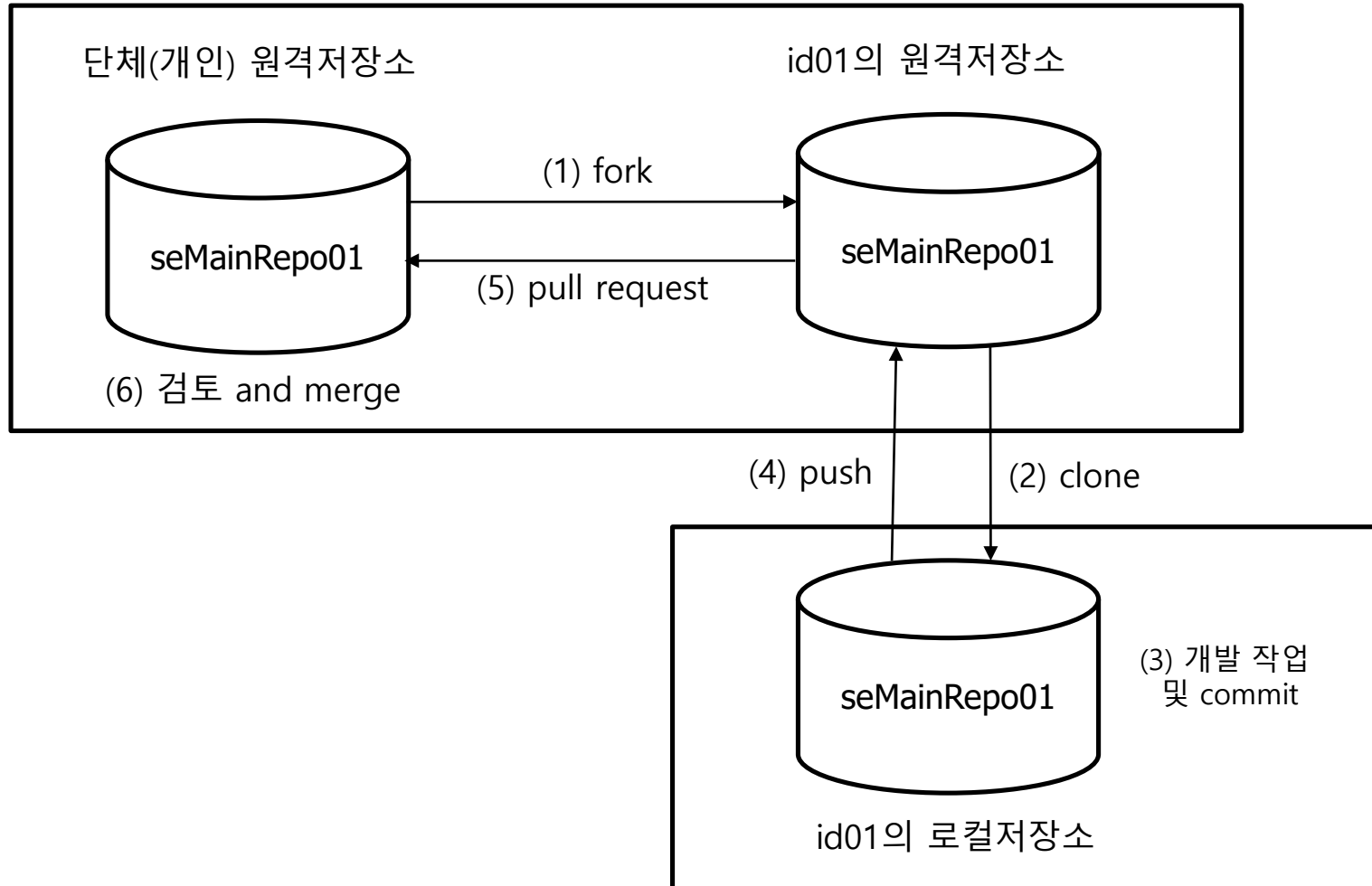
- 로컬에서 작업하기 위해 저장소의 내용을 복제하는 과정

## ❑ push

- 로컬 저장소와 원격 저장소 중 한 곳에서 새로 생성된 커밋을 다른 저장소에 반영시키는 것

# GitHub와 GitHub Desktop 동작 원리(fork)

GitHub

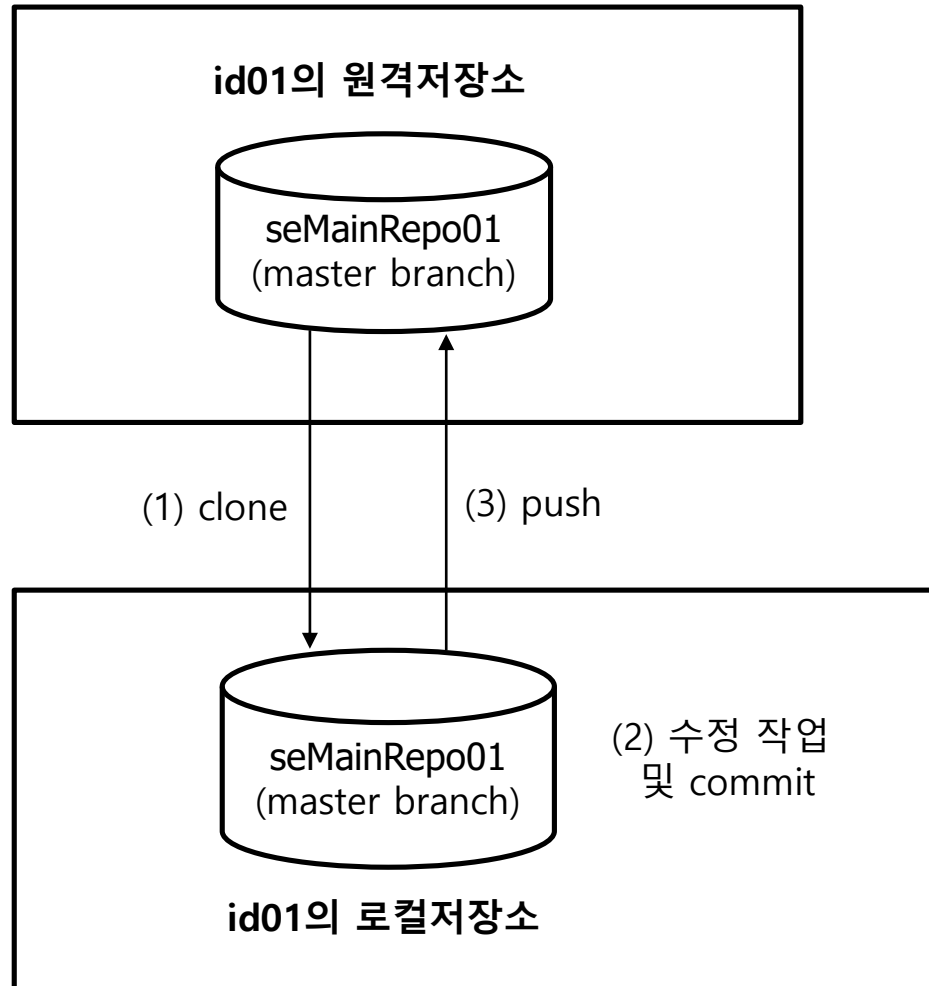


GitHub Desktop

# GitHub와 GitHub Desktop 동작 원리

---

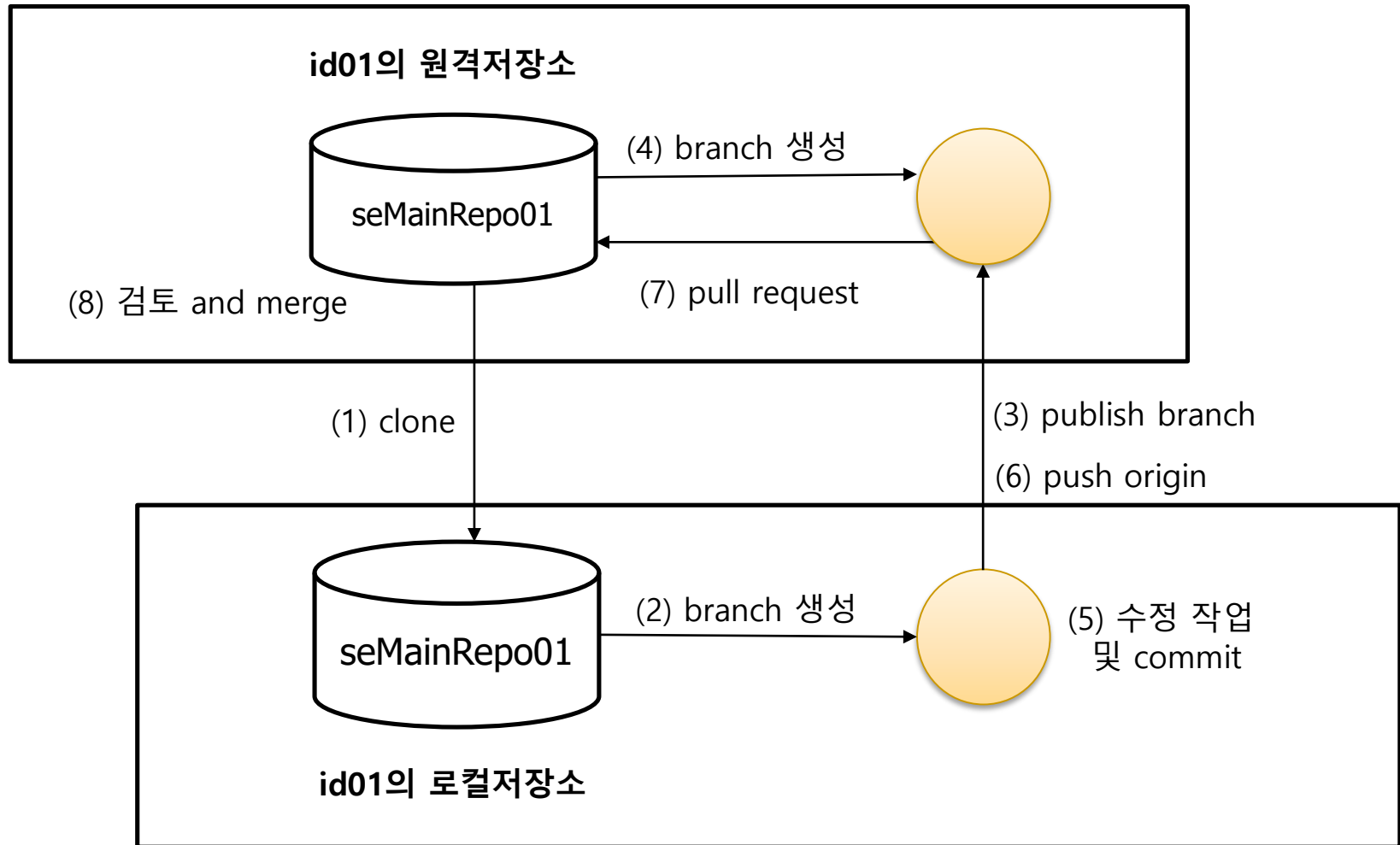
GitHub



GitHub Desktop

# GitHub와 GitHub Desktop 동작 원리(branch)

GitHub



GitHub Desktop

# 01. GitHub 실습

---

## □ repository 생성 위치

- 단체 또는 개인 사용자 계정 아래에 생성 가능함
- 단체에 생성 시
  - 새로운 team을 만들고, team에 collaborator 멤버를 추가해서 프로젝트에 접근하도록 함
- 개인 사용자 계정에 생성 시
  - collaborator를 프로젝트에 직접 추가

# 01. GitHub 실습

---

- [github.com](https://github.com)에 id 및 repository 생성
  - 로그인 id 생성 및 로그인 (예: "id01")
  - organization(단체) 생성
    - 오른쪽 상단 '+' 메뉴 > New Organization 클릭
    - Choose Team for Open Source 선택 > 데이터 입력(팀명: "seTeam1") > 멤버 추가
  - 단체 아래에 Repository 생성 및 파일 업로드  
(개인 계정에 repository를 생성해도 됨)
    - 'Create a new Repository' 선택
    - Repository name 입력: "seMainRepo01"
    - JUnit 실습 시 작성했던 파일 업로드 및 커밋
      - se.maven.examl의 SimpleCalculator.java
      - 폴더 명 추가 se/maven/example



# 01. GitHub 실습

---

## □ fork 실습

- 오른쪽 상단 fork 메뉴 선택
  - id01로 fork 지정
- "id01/seMainRepo01" repository가 생성됨
  - 이 repository에서 파일 수정 작업 후 커밋
  - seTeam01/seMainRepo01에 pull request를 신청

# 01. GitHub 실습

---

## □ pull request 실습

- SimpleCalculator.java 수정 및 커밋
  - 3번째 라인에 다음 문자열 삽입
    - `“// 1: remote by id0”`
- 상단의 “Pull requests” 선택 > “New Pull Request” 버튼 클릭
- Comparing changes 화면
  - base repository, pull 방향 등을 확인
  - 아래 수정 내용 확인
  - “Create pull request” 버튼 클릭
  - 내용 작성
  - “Create pull request” 버튼 클릭
- seTeam01/id01 에서
  - comment 추가 또는 “Merge pull request” 클릭 > confirm merge
  - 코드 수정이 반영되었는지 확인

# 01. GitHub 실습

---

## □ branch 실습

- id01로 로그인
- "branch:master" 버튼 클릭 > 새로운 branch 명 입력
  - branch 명: update\_readme
- README.md 파일 수정 및 커밋
  - md: mark down
    - HTML보다 간단한 구조의 문법으로 콘텐츠 작성 가능한 마크업 언어
    - 예: #This is a H1, **\*\*\***, *\*single asterisks\**, **\*\*double asterisks\*\*** 등
- Pull request 생성
  - 상단 "Pull requests" 클릭 > "New pull request" 버튼 클릭
  - base 및 pull 방향 확인
    - 현재 repository의 master branch <- update\_readme branch
  - Create pull Request 버튼 클릭 > 한번 더 > Merge pull request 버튼 클릭 > Confirm Merge 버튼 클릭

# 01. GitHub 실습

---

- ❑ branch 실습(계속)
  - README.md 수정 작업을
    - id01/seMainRepo01에서 seTeam01/seMainRepo01로의 pull request 작성
      - 상단의 Pull request 선택 및 pull request 생성

## 02. GitHub Desktop 실습

---

- ❑ GitHub Desktop 설치
  - <https://desktop.github.com/> 접속
  - 다운로드 및 설치
- ❑ SublimeText 설치하기
  - <https://www.sublimetext.com/> 접속
  - 다운로드 및 설치



## 02. GitHub Desktop 실습

---

### □ 원격 저장소 clone 하기

- id01로 로그인
- 메뉴: File > Clone Repository 선택
  - id01/seMainRepo01 선택
- 복제된 repository의 위치
  - C:\Users\wssh\OneDrive\Documents\GitHub\seMainRepo01

### □ 로컬 저장소 파일 수정하기

- "Open in Sublime Text" 선택
  - se/maven/example/SimpleCalculator.java 선택
  - 파일 내용 수정 및 저장
    - "// 1: local by id01" 삽입

## 02. GitHub Desktop 실습

---

- ❑ 로컬 저장소에서 원격 저장소로 수정사항 보내기(push)
  - GitHub Desktop 선택
  - SimpleCalculator.java 파일의 수정 사항 보여줌
    - 빨간색 부분: 삭제된 부분
    - 초록색 부분: 추가한 부분
  - 왼쪽 하단의 “commit to master” 버튼 클릭
  - “Push origin” 버튼 클릭
  - GitHub에서 원격 저장소 확인
    - Code 메뉴 선택 > SimpleCalculator.java 내용 확인



## 02. GitHub Desktop 실습

---

- ❑ 원격 저장소 수정사항을 로컬 저장소로 fetch 하기
  - 원격 저장소에서 SimpleCalculator.java 수정
    - 다음 문장 삽입 및 커밋
      - "2: remote by shchoi8347"
  - GitHub Desktop에서 상단의 "fetch" 선택
  - Changes 탭 선택 > "Pull origin" 버튼 클릭
  - 로컬 저장소의 SimpleCalculator.java 확인





## 02. GitHub Desktop 실습

---

- ❑ GitHub Desktop: branch 실습
  - 새로운 branch 생성
    - 상단 Branch 메뉴 > New Branch 선택 > branch 이름 입력
      - branch 명: localBranch01
    - Create branch 버튼 클릭
  - 생성된 branch를 원격지로 publish > 원격지에서 확인
    - "Publish branch" 버튼 클릭
  - 로컬저장소에서 current branch 확인
  - sublimeText에서 파일 수정 작업 및 저장
    - 다음 문장 삽입
      - 1: localBranch01 by shchoi8347
  - GitHub Desktop에서 커밋

## 02. GitHub Desktop 실습

---

- ❑ GitHub Desktop: branch 실습(계속)
  - “Push origin” 버튼 클릭
    - 원격지에서 확인
    - 원격지의 localBranch01에도 수정 사항이 적용됨
  - Create Pull Request 클릭
    - GitHub의 pull request 창이 열림
    - 원격지의 어떤 branch로 적용 요청할 지 선택 후 pull request 생성

## 03. 기타 GitHub 실습

---

### □ 프로젝트 wiki 페이지 작성

- 상단의 Settings 메뉴 선택
  - Features 부분에서 Wikis 체크 되어 있는 지 확인
- 상단의 Wiki 메뉴 선택 > Create the first page 버튼 클릭
  - 내용 편집 및 저장
- 새로운 페이지 추가
  - New page 버튼 클릭 및 내용 작성
- 두 페이지 연결
  - 오른쪽 Pages 리스트 > 연결대상 페이지 선택 > URL 복제
  - 오른쪽 Pages 리스트 > 연결주체 페이지 선택 > Edit 버튼 > 링크 아이콘 버튼 선택 > 복제했던 URL 붙여넣기

## 03. 기타 GitHub 실습

---

### ❑ 협력자 추가

- 상단의 Settings 메뉴 선택 > Collaborators 선택
  - 협력자의 user name 입력 > Add collaborator 버튼 클릭

### ❑ 프로젝트 홈 페이지 자동 생성

- 상단의 Settings 메뉴 선택 > GitHub Pages 부분 설정

### ❑ 단체에 팀 추가

- 왼쪽 상단의 고양이 아이콘 클릭 > Manage Organization 클릭
- 수정할 단체 선택 > 상단의 Teams 선택 > New Team 클릭 > team 데이터 입력
- 팀 멤버 추가: 왼쪽의 '+' 기호 이용

## 03. 기타 GitHub 실습

---

- ❑ 기존 이클립스 프로젝트를 로컬저장소로 만든 후 publish 하기
  - GitHub Desktop 실행 > File 메뉴 > New Repository 선택
    - Repository 명: 이클립스 프로젝트 폴더명과 동일한 이름
    - Local path: 이클립스 프로젝트 폴더의 상위 폴더를 선택
  - 이클립스 실행
    - 파일 수정
    - GitHub Desktop 에서 commit > push origin