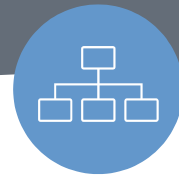
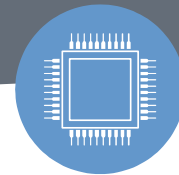




Git & GitHub 실습 (2)



2023년 3월 7일

Agenda

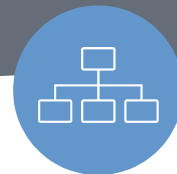
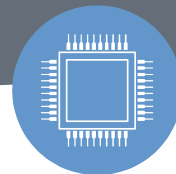
■ Git으로 버전 관리

- 커밋하기
 - 버전 관리에서 제외하기
 - 버전 비교하기
 - 버전 되돌리기
 - 태그 붙이기
- 작업 임시 저장하기
- 브랜치 사용하기
 - 만들기, 병합하기, 충돌해결하기

■ GitHub 사용하기

- GitHub와 SourceTree 연동하기
- 로컬 저장소와 원격 저장소 상호작용
 - Clone, Push, Fetch, Pull
- 여러 사람과 협력하기
 - Fork, Pull Request

Git으로 버전관리



커밋하기 (1) : 개념도



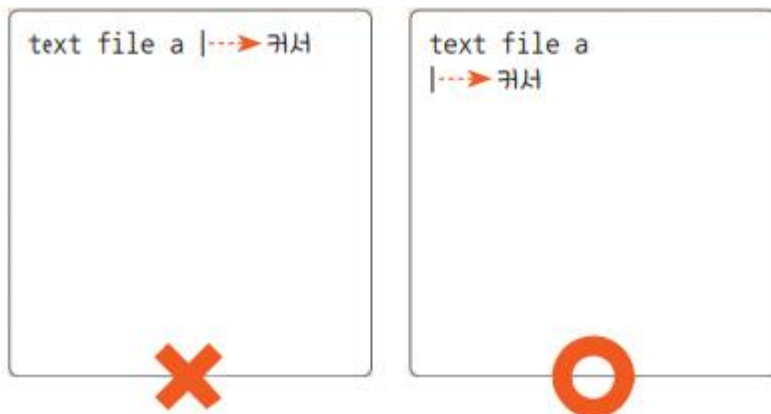
커밋하기 (2) : 첫 버전 만들기

로컬 저장소 만들기

- SourceTree로 GitTest 생성
- 파일탐색기에서 해당 폴더 내 .git 폴더 있는지 확인

버전 관리 대상 파일 만들기

- GitTest 폴더 내 a.txt, b.txt, c.txt 생성
 - 각 파일 내에 내용을 쓰고 마직은 <enter>를 치고 파일 저장



스테이지로 Add

- SourceTree의 <파일상태>에서 스테이지에 올라가지 않은 파일 확인
- <모드 스테이지에 올리기> 클릭
 - 파일별 Add : 파일명 옆의 <+> 클릭
 - 선택한 파일만 Add : <선택 내용 스테이지에 올리기>
- 스테이지에 올라간 파일명 옆의 <-> 클릭하면 스테이지에서 내려옴
- 각 파일을 클릭하면 각 파일의 변경 사항을 볼 수 있음

저장소로 Commit

- 커밋 메시지 작성 : What, How, Why
 - 제목, 한 줄 띄고 본문. 제목만 작성하여도 무관
- <커밋> 클릭 -> 새로운 버전 생성

커밋 내역 보기 : Source Tree의 <History>

- 커밋 메시지, 만들어진 시간, 작성자 등을 확인 가능

커밋하기 (3) : 변경 내역 커밋하기

■ 두번째 커밋 생성

- 작업 디렉토리(GitTest) 내 파일 수정
 - c.txt 삭제, a.txt 파일 내용 수정
- Add to Stage
 - SourceTree의 <History>에서 '커밋되지 않은 변경사항' 확인
 - SourceTree의 <파일상태>에서 변경내용 확인
 - <파일상태>에서 변경사항 스테이지에 올리기
- Commit
 - 커밋 메시지 작성 후 커밋하기
 - <History>에서 두번째 커밋 생성 확인

■ 세번째 커밋 생성

- 작업 디렉토리(GitTest) 내 d.txt 생성
- Add to State
- Commit

버전 관리 대상에서 제외하기 .gitignore

■ .gitignore 파일에 파일명 쓰기

- 해당 파일은 버전 관리에서 제외

- [실습]

- .gitignore 파일 생성
- 파일을 열어서 e.txt 추가
- e.txt 생성
- SourceTree의 <파일상태>에서 Stage에 올라가지 않은 파일 확인

■ .gitignore 파일에 폴더명 쓰기

- 해당 폴더에 있는 모든 파일을 버전 관리에서 제외

- [실습]

- .gitignore 파일에 제외할 폴더명 'ignore/' 추가
- ignore 폴더 생성
- ignore 폴더 안에 아무 파일이나 여러 개 만들기
- SourceTree의 <파일상태>에서 Stage에 올라가지 않은 파일 확인

버전 비교하기

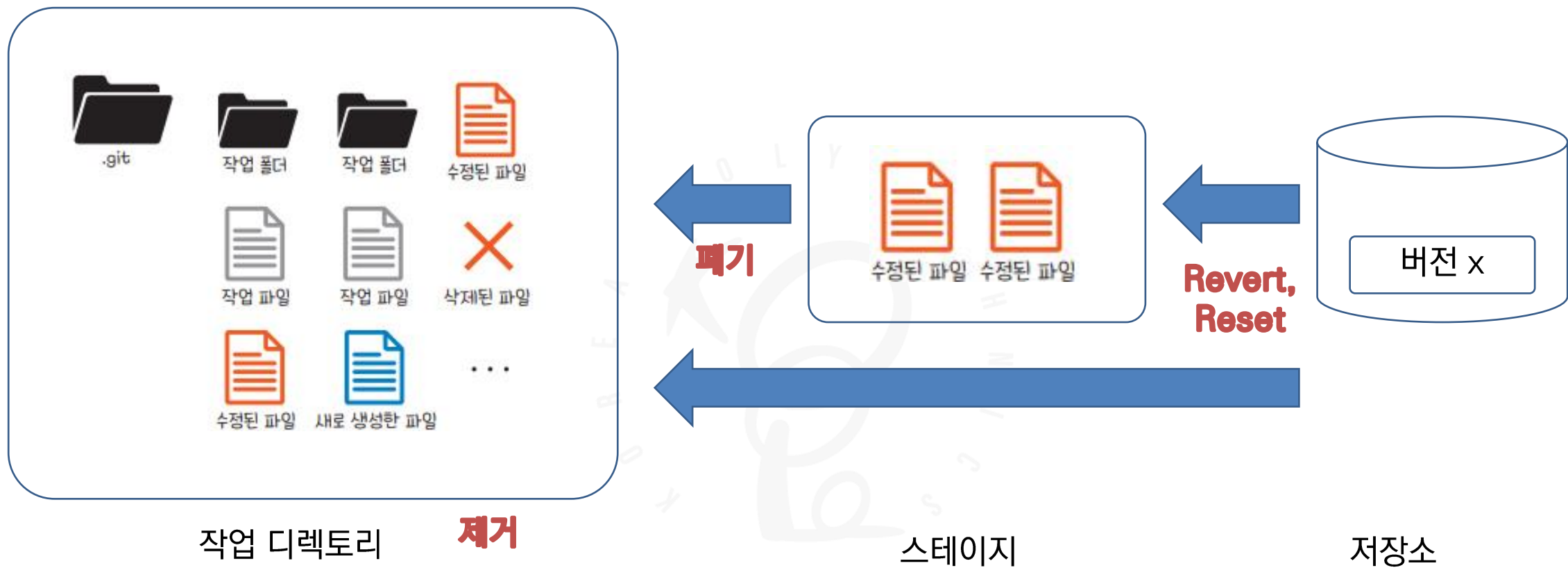
■ 특정 커밋의 파일 내용 확인

- SourceTree <History>에서 커밋 선택
- 보고자 하는 파일 선택 후 마우스 오른쪽 버튼 클릭
- <선택한 버전 열기>
- [실습]
 - 여러 커밋이 있는 파일 생성
 - Test.txt 파일 생성
 - 파일에 내용을 한 줄씩 추가하여 5개의 커밋 생성하기
 - 원하는 버전 내용 확인하기

■ 두개 버전 비교

- SourceTree <History>에서 커밋 선택
- 첫번째 파일 선택
- <Ctrl>을 누르고 두번째 파일 선택
- [실습]
 - 두번째 커밋과 네번째 커밋에서 Test.txt 파일 비교하기

버전 되돌리기 : 개념도



버전 되돌리기 (1) : 스테이지 상황

■ 스테이지에 있는 파일을 내리기

- 스테이지에서 내리기
- 파일 선택 후 마우스 오른쪽 버튼 클릭하여 <폐기> 선택
- [실습]
 - 스테이지에 있는 상황 만들기
 - a.txt 파일에 내용 추가
 - SourceTree의 <파일상태>에서 스테이지에 올리기
 - 스테이지에 있는 파일 되돌리기

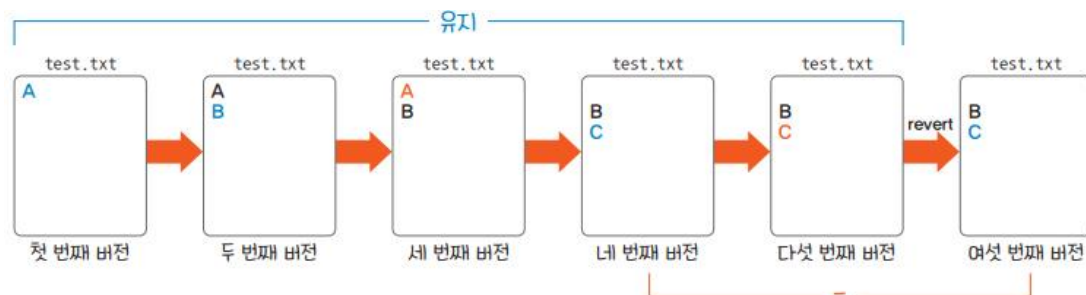
■ 스테이지에 올라가지 않은 파일 제거

- 파일 선택 후 마우스 오른쪽 버튼 클릭하여 <제거> 선택
- [실습]
 - 스테이지에 올라가지 않은 파일 생성
 - D.txt 생성
 - SourceTree에서 되돌리기 진행

버전 되돌리기 (2) : 커밋 되돌리기

■ Revert

- 기존 버전은 삭제하지 않고 되돌아간 상태를 새로운 버전으로 생성



• 절차

- <history>에서 되돌릴 버전 선택
- 마우스 오른쪽 버튼 클릭하여 <커밋 되돌리기> 선택
- 이전 커밋 상태로 현재 버전 새로 생성

• 실습

- 마지막 커밋 선택, <커밋 되돌리기>
- test.txt 파일 확인

■ Reset

- 되돌아갈 버전의 시점으로 완전히 돌아가는 방식
- 되돌아갈 버전 이후 버전은 삭제

• 유형

- Soft reset : 커밋만 되돌리기 (스테이지에 파일 있음)
- Mixed reset : 스테이지까지 되돌리기 (스테이지에 올라가지 않은 파일에 있음)
- Hard reset : 작업 디렉토리까지 되돌리기 (작업 디렉토리의 파일 내용 변경)

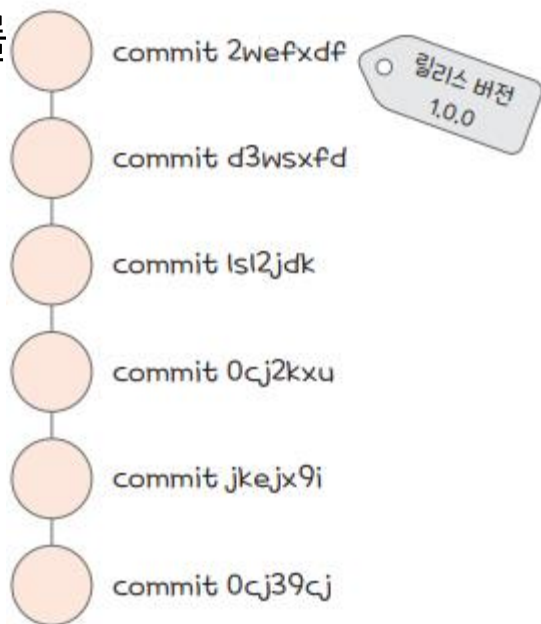
• 절차

- Reset할 버전을 선택
- 마우스 오른쪽 버튼 클릭하여 <이 커밋까지 현재 브랜치를 초기화> 선택
- Reset 유형 선택

태그 붙여서 릴리즈 하기

- 여러 커밋 중 유의미한 커밋을 사용자에게 Release

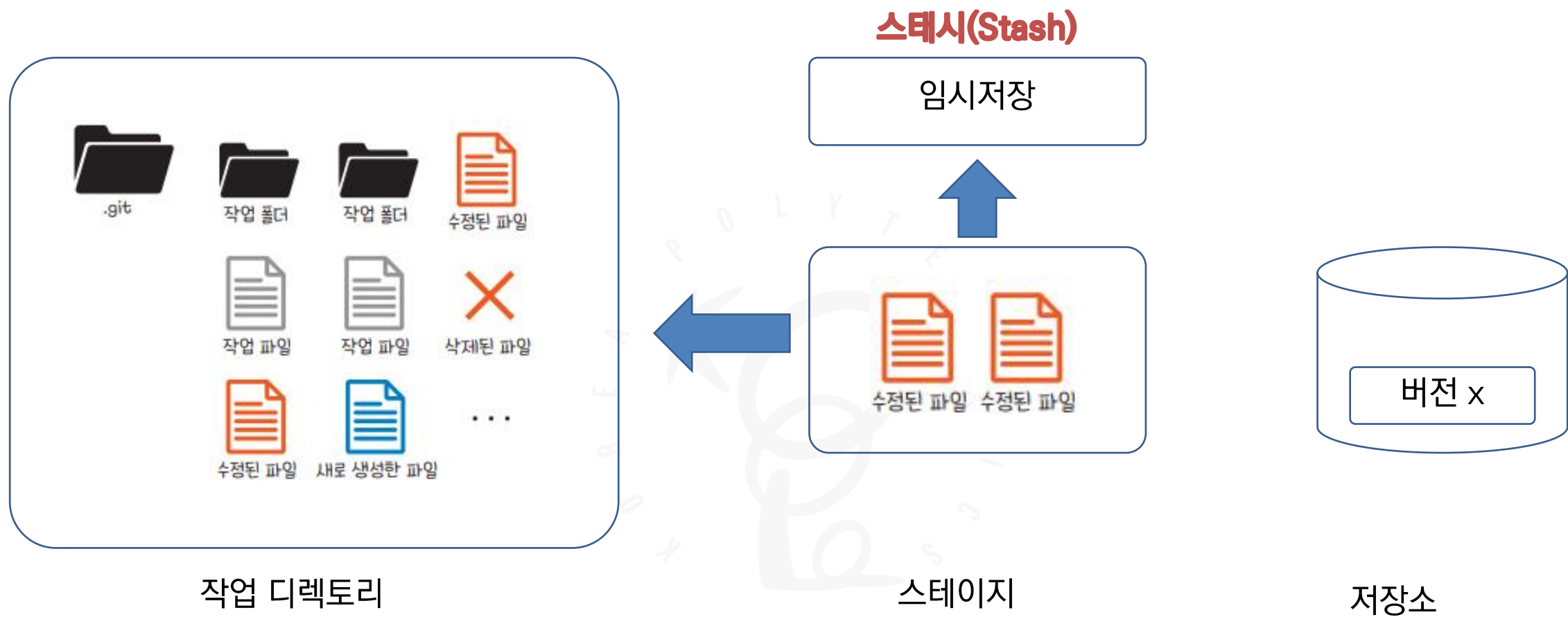
- 태그를



- [실습]

- SourceTree의 <History>에 release할 커밋 선택
- <Tag> 클릭
- Tag 이름에 v1.0.0 쓰고 enter
- <Tag추가> 선택

작업 임시 저장 : 개념도



작업 임시 저장 : 스테시(Stash)

■ 작업 임시 저장 생성

- 스테이지에 올리기 전 상태를 저장
- 절차
 - SourceTree의 <스테시> 클릭
 - 스테시 이름 입력 후 확인
 - 왼쪽의 <스테시>에서 임시 저장 목록 확인

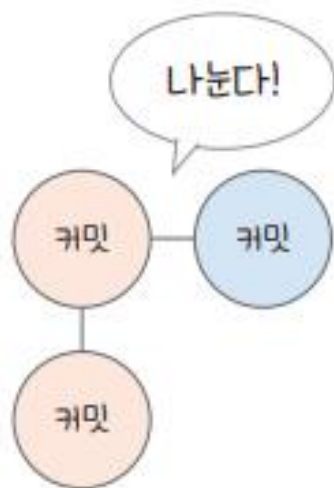
■ 임시 저장목록의 상태 적용

- 임시 저장상태를 작업 디렉토리에 반영원하는 임시저장 상태 선택
- 마우스 오른쪽 버튼 클릭하여 < ... 스테시 적용> 선택

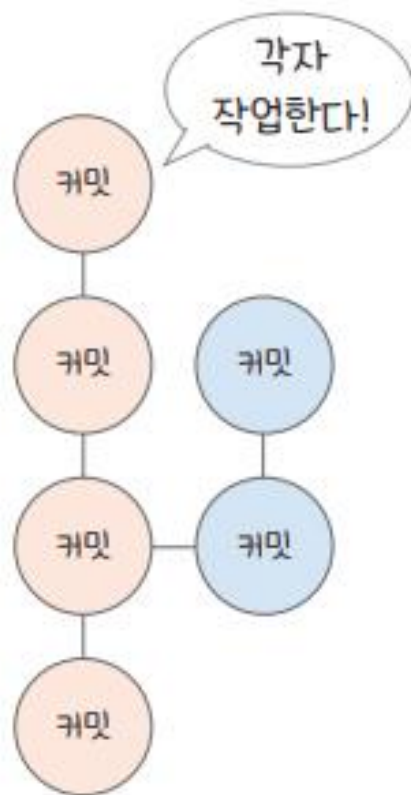
■ [실습]

- StashTest Repository 생성
- a.txt, b.txt, c.txt 생성하여 첫번째 커밋
- a.txt 수정, c.txt 삭제
- SourceTree에서 <스테시> 선택하여 '임시저장1' 생성
- 작업 디렉토리에서 파일 상태 확인
- d.txt 생성
- SourceTree에서 <스테시> 선택하여 '임시저장2' 생성
- 작업 디렉토리에서 파일 상태 확인
- <스테시>에서 '임시저장1' 적용
- 작업 디렉토리에서 파일 상태 확인

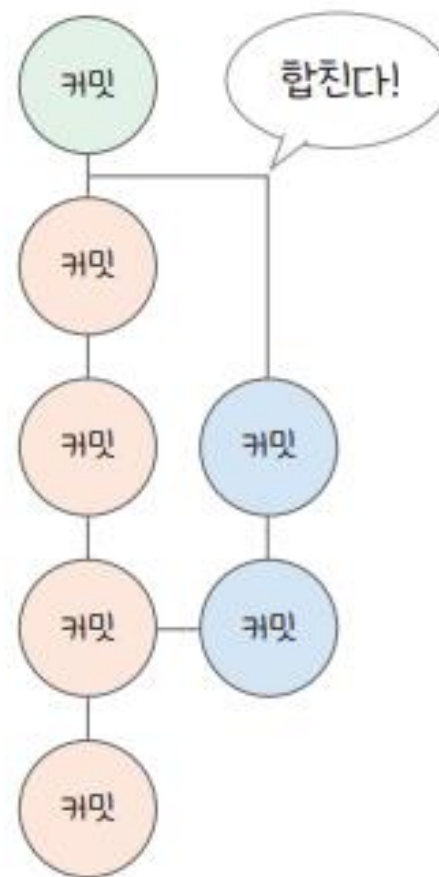
브랜치 (Branch) : 개념도



(1) 브랜치를 나눈다



(2) 각자의 브랜치에서 작업한다

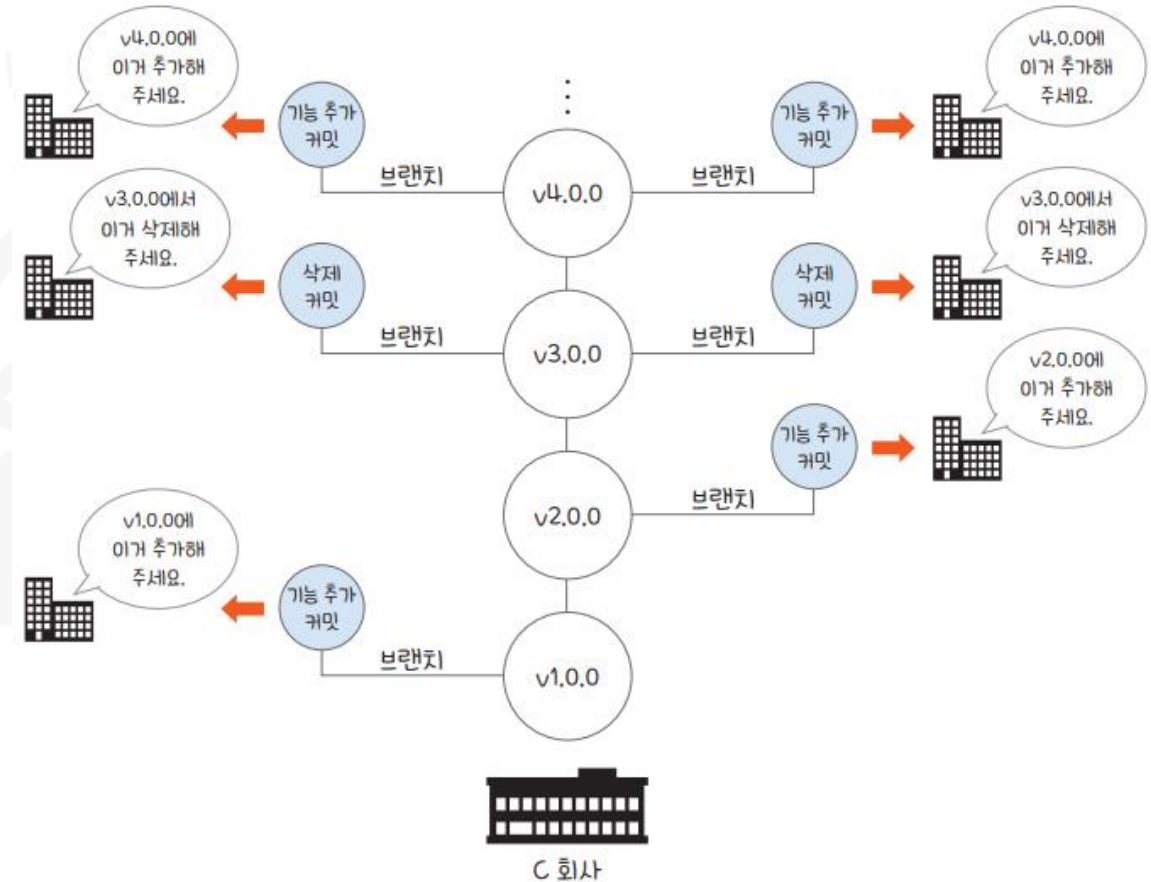
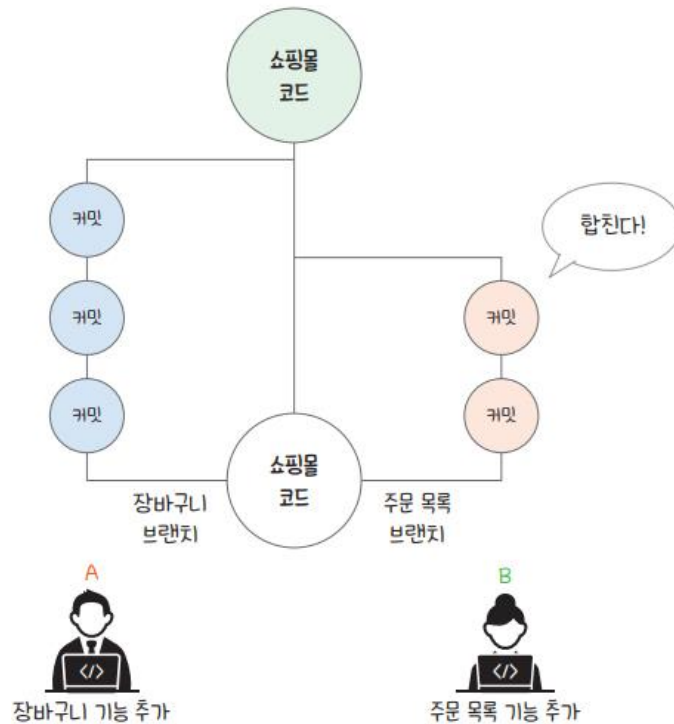


(3) 나눈 브랜치를 합친다

브랜치 : 필요한 경우

■ 브랜치가 필요한 상황

- 이미 방대한 코드가 있는 상황에서 여러 사람이 각자의 기능을 추가해야하는 경우
- 새로운 요구 사항이 Client별로 다른 경우
- 새로운 요구 사항이 버전별로 다른 경우



브랜치 나누고, 합쳐보기

- 최초의 브랜치 : master 브랜치

- 브랜치 나누기

- 원하는 버전에서 <브랜치> 선택 -> 브랜치 이름, <새로운 브랜치로 체크아웃> 선택하고 생성

- 브랜치 합치기 (merge)

- Merge하려는 브랜치로 checkout
- 방법 1. 왼쪽 메뉴에서 원하는 브랜치 마우스 오른쪽 버튼 클릭 -> <현재 브랜치로 OOO 병합> 선택
- 방법 2. 상단의 <브랜치> 선택, 병합할 커밋 선택

- 브랜치 삭제

- 더 이상 사용하지 않은 브랜치 삭제하기

- 실습

- 최초 브랜치 : Master 브랜치에 3개의 커밋
 - Master-1.txt, master-2.txt, master-3.txt
- 브랜치 나누기
 - 첫번째 브랜치 branch1에 2개의 커밋
 - Branch1-1.txt, Branch1-2.txt
 - 두번째 브랜치 branch2에 3개의 커밋
 - Branch2-1.txt, Branch2-2.txt, Branch2-3.txt
- 브랜치 합치기
 - Branch1을 master에 병합(merge)
 - Branch1 삭제
 - Branch2의 첫번째 커밋을 master에 병합
 - Branch2의 세번째 커밋을 병합

충돌 해결 하기

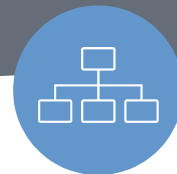
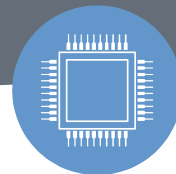
■ 충돌 발생 상황

- 브랜치별로 같은 파일을 다른 내용으로 수정한 경우
 - 충돌 해결 : 어떤 브랜치의 내용을 반영할지 직접 선택
 - 스테이지에 올라가지 않은 파일을 오른쪽 마우스 버튼 클릭
 - <충돌해결> 선택
 - <내것을 이용해 해결>과 <저장소것을 사용하여 해결> 중 선택하여 해결
 - 다시 커밋

■ 실습

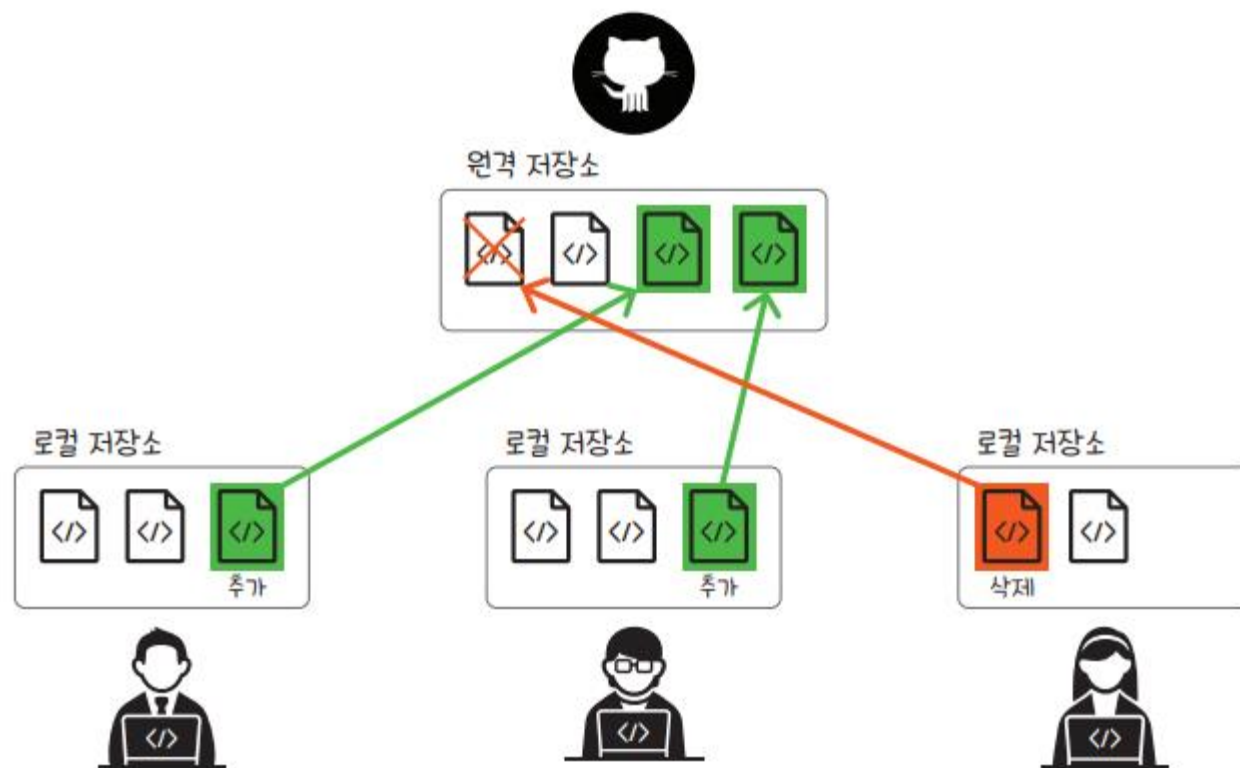
- Master 브랜치 내 a.txt : a 쓰기 → 커밋
- 브랜치 branch3 생성, a.txt 내용으로 b로 변경
- Master에 branch3 병합
- 충돌 발생 확인
- <내 것을 이용해 해결> 선택

GitHub 사용하기



GitHub란

- 개발자의 SNS
- 원격 저장소 호스팅 서비스
 - 백업
 - 협업



원격 저장소와 SourceTree와 연동하기

■ SourceTree 와 GitHub 연동하기

- 〈Remote〉
- 〈계정추가〉
 - Authentication (인증)

■ GitHub의 저장소 Clone하기

- 원격 저장소를 로컬 저장소로 복제하기
- [실습]
 - 원격 저장소
 - <https://github.com/namhyung/uftrace>
 - SourceTree에서 〈Clone〉
 - 원격 저장소 URL, 로컬 저장소 위치 입력
 - 〈클론〉

내 원격 저장소 생성 및 복제

■ GitHub에 내 저장소 생성

- New Repository, Create Repository
 - Name
 - Add README file check
 - 〈Create〉
- 생성된 Repository 정보 확인
- 삭제는 설정의 Danger Zone에서 ...
- 프로필에서 확인

■ 내 원격 저장소 Clone하기

- SourceTree의 〈Remote〉에서 내 원격 저장소 목록 확인
- 〈clone〉 선택
- 폴더 생성 확인
- Main 브랜치 원격 저장소 주소 확인

상호 작용 : Push, Fetch

■ Push

- 로컬 저장소의 변경 내용을 원격 저장소에 반영
- 실습
 - 로컬 저장소에 파일 추가 a.txt 후 커밋
 - 원격저장소와 상태 비교
 - <push> 선택
 - 원격저장소와 상태 비교
 - A.txt 삭제, b.txt 생성 후 커밋
 - 원격저장소와 상태 비교
 - <push> 선택
 - 원격저장소와 상태 비교

■ Fetch (가져오기만 함)

- 원격 저장소의 변경 내용을 로컬 저장소에 반영
 - 로컬 저장소에 반영으로 되나 별도의 branch로 반영됨
- 실습
 - 원격 저장소의 a.txt 수정 후 커밋
 - 커밋 수 확인
 - SourceTree에서 <Fetch> 선택
→ origin/main 브랜치 생성
 - Origin/main을 병합하기

상호 작용 : Pull

■ Pull (가져와서 합치기)

- Fetch + Merge
- 실습
 - 원격저장소에 c.txt 생성한 후 커밋
 - SourceTree <Pull> 선택
 - Branch 통합된 상태 확인
 - 로컬저장소의 c.txt 확인



GitHub로 협력하기

■ Fork

- 기여할 Project를 내 저장소로 가져오기
- 실습
 - 기여할 Project를 내 원격 저장소로 fork
 - 원하는 프로젝트의 원격저장소로 이동
 - `github.com/hyejeong1019/CollaboTest`
 - <Fork> 실
 - 내 원격저장소에서 확인 (forked ...)
 - 내 원격저장소로 fork된 프로젝트를 내 로컬 저장소로 clone
 - SourceTree의 Remote에서 해당 프로젝트 선택하여 clone
 - 작업 디렉토리 확인

■ Pull Request

- Push 권한이 없는 Project를 수정하여 Push를 한 후, 해당 Project 권한 보유자에서 Pull을 요청
- Project 담당자와 요청받은 Pull을 반영
- 실습
 - 새로운 브랜치 생성 : add_이름
 - 해당 브랜치로 checkout
 - 작업 디렉토리에 본인이름.txt 생성 후 커밋
 - 작업한 브랜치 Push
 - Request 받은 사람이 Merge pull request 실행

수고하셨습니다~~~ ^^