

브랜치 개요

브랜치 : 커밋 사이를 가볍게 이동할 수 있는 포인터

HEAD : 작업 중인 브랜치의 최신 커밋이나 브랜치를 가리키도록 헤드 이동 가능

→ checkout, switch 명령 사용

결과 표시 (HEAD → main)

: main'은 마지막 커밋을 가리키고 HEAD는 현재 작업 브랜치인 main을 가리킴

새로운 브랜치 생성

\$ git branch bname : 생성하고 HEAD가 이동은 안함

\$ git switch -c bname, \$ git checkout -b bname : 생성 후 새로운 브랜치로 HEAD 이동 수행

설명	git checkout	git switch
이전 커밋으로 이동	\$ git checkout [이전커밋]	\$ git switch -d [이전커밋]
다른 브랜치로 이동	\$ git checkout [branch]	\$ git switch [branch]
새로운 브랜치를 생성 후 이동	\$ git checkout -b [newBranch]	\$ git switch -c [newBranch]

▶ 이전 브랜치로 이동 : \$ git switch - / \$ git checkout -

분리된 HEAD (detached) : HEAD가 현재 브랜치가 아닌 그 이전 커밋을 가리키는 상태

▶ \$ git checkout HEAD~ : 현재 브랜치에서 마지막 커밋 이전 커밋으로 이동

브랜치 삭제

- \$ git branch [-d | --delete] [branchName]
- \$ git branch -D [branchName] ▶ 강제 삭제

원격 저장소 복제 clone

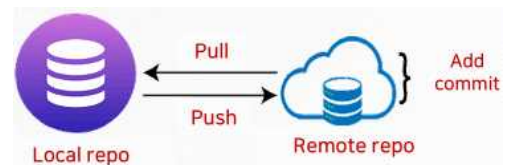
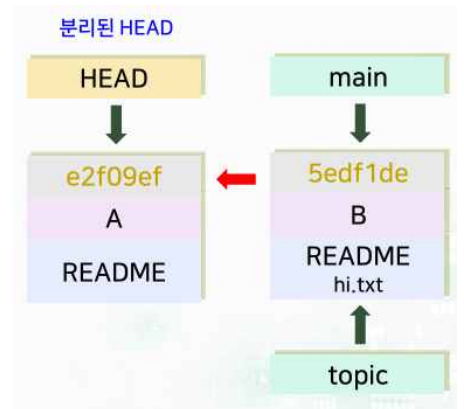
- \$ git clone [복사된 주소] : 원격저장소와 동일한 이름으로 복제
- \$ git clone [복사된 주소] [새로운 폴더명] : 하부 폴더에 복제
- \$ git clone [복사된 주소] . : 현재 폴더에 바로 복제

▶ \$ git remote : 원격 저장소 목록 (기본 이름 origin 점검)

\$ git remote -v : 저장소 주소 등 원격 저장소 정보 목록 표시

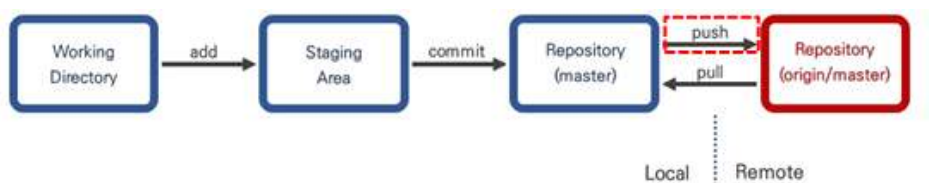
깃허브 원격 저장소 수정 후 pull

- 원격 저장소의 수정을 지역 저장소에 반영 (지역 저장소에서 pull)



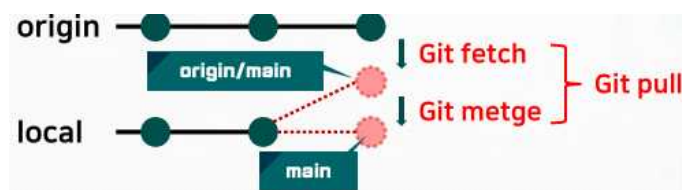
지역 저장소에서 push

- 쓰기 권한이 있어야 push 가능
- 로컬 저장소에서 남겨놓은 코드 변경 이력들이 원격 저장소로 전송
- \$ git push <저장소 별칭명> <브랜치명>
- \$ git push origin topic



명령 push에서 인자 생략하기

- 옵션 -u 사용 : 최소에 한 번만 저장소명과 브랜치명을 입력하고 그 이후는 모든 인자를 생략 가능
  - ▶ \$ git push origin topic
- 성공 이후 간단히 다음으로 가능
  - ▶ \$ git push



git pull = git fetch + git merge

- git pull 명령 : fetch 명령과 병합하는 merge 명령이 순차적으로 진행
- fetch 명령 : 원격 저장소의 정보를 로컬 저장소로 가져오는 명령 (remote/origin)
- merge 명령 : 변경된 정보를 로컬 저장소의 내용과 병합

→ fetch : 원격 저장소에서 로컬 저장소로 소스를 가져옴, 소스를 가져올 뿐 병합을 하지 않음

- fetch한 이후 origin/main, origin/HEAD가 더 최신 위치 (지역인 HEAD의 main이 이후에 있음)

▶ 깃허브의 내용을 반영하려면 fetch 후 merge / \$ git merge origin/main

정리

원격 저장소 수정 사항 pull로 지역 저장소로 가져오기	\$ git pull origin main / \$ git pull
원격 저장소 수정 사항 fetch로 지역 저장소로 가져와 병합하기	\$ git fetch origin main / \$ git fetch / \$ git merge origin/main
지역 저장소 수정 사항 push로 원격저장소 보내기	\$ git push origin main / \$git push