

16~18

클론(clone) : 원격 저장소를 지역 저장소에 복제

원격 저장소 생성과 복제

깃허브 원격저장소 생성 : 저장소 이름 git-clone → https 주소 복사

Pc 깃에서 원격저장소 복제(git clone)

\$ git clone [복사된-주소] : 원격저장소와 동일한 이름으로 복제

\$ git clone [복사된-주소] [새로운-폴더명] : 하부 폴더 [새로운-폴더명] 이름으로 복제

\$ git clone [복사된-주소] : 현재 폴더에 바로 복제

원격 저장소 별칭 이름 점검

\$ git remote : 원격 저장소 이름 목록

- 기본 이름 : origin

\$ git remote -v : 원격 저장소 주소와 이름 목록

원격 저장소 별칭 관리

\$ git remote add origin URL : 원격 저장소 별칭 저장

\$ git remote show origin : 자세한 정보

\$ git remote rename origin org : 이름 수정

\$ git remote rm org : 삭제

Vs code로 원격 저장소 관리 : clone git repository 클릭

push

쓰기 권한이 있는 자신의 저장소나 타인의 저장소의 협업자여야 push가 가능

push : 로컬 저장소에서 남겨놓은 코드 변경 이력들이 원격 저장소로 전송

\$ git push <저장소별칭명><브랜치명>

최초의 한 번만 저장소명과 브랜치명을 입력하고 그 이후에는 모든 인자를 생략 가능

대부분의 경우에는 로컬 저장소와 원격 저장소에서 동일한 브랜치 이름을 사용. 항상 현재 브랜치를 기준으로 git push 명령어가 작동한다면 매우 편리

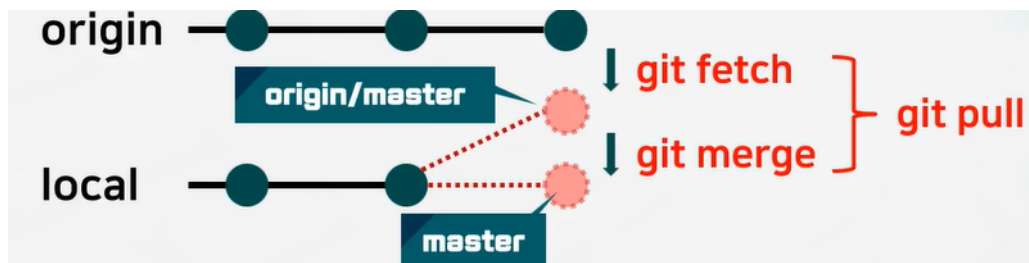
설정 변수 push.default를 current로 설정(\$ git config - -global push.default current)

어느 브랜치에서 작업을 하든 git push만 날리면 원격 저장소에 동일한 브랜치로 코드 변경이 업로드

git pull = git fetch+git merge

git pull 명령

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



fetch

원격 저장소에서 로컬 저장소로 소스를 가져와 병합을 미수행하는 명령

- \$ git fetch <remote>
- \$ git fetch origin

fetch 이후 origin/main,origin/HEAD가 더 최신의 위치에 있음

깃허브에 내용을 반영하려면 fetch 후 merge 해야함