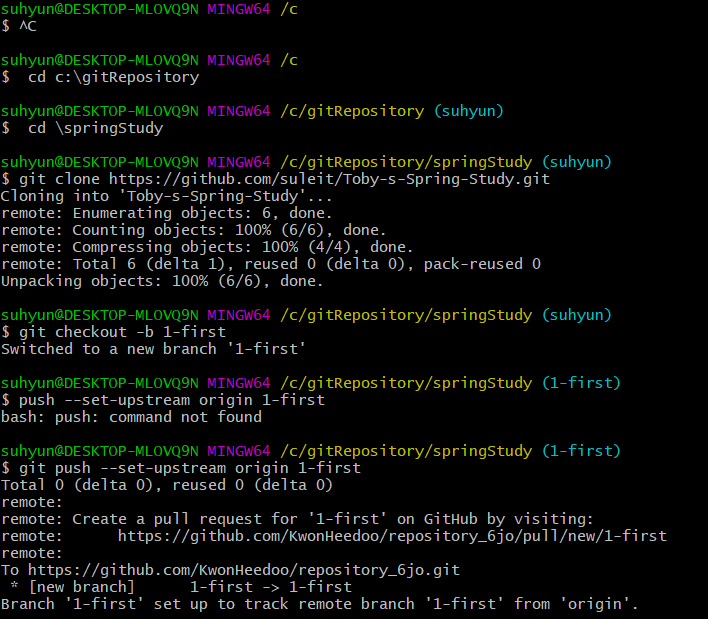
사용도구 툴: git Bash

필수세팅 git-hook 커밋시 이슈번호를 읽어들이도록 이슈기반 관리가 되도록 세팅할 것

---🡪 실패함 허스키 설치가 안됨



**브런치 생성(-b)하면서 생성한 브런치로 이동(checkout)하기**

Git checkout –b 브런치명

**원격 저장소로 새로 만든 브런치 push하기**

Git push –-set-upstream 원격저장소명(origin) 로컬브런치명

**Commit 및 push 하기**

**저장소 상태확인하**기 git status

**저장소 변경사항 버전에 전부 추가하기** git add . (.이 모두다 라는 의미)

**커밋하기** git commit –m “커밋메세지 괄호안에 넣어서 보내기”

-m은 뒤의 내용이 메시지라는 것을 가리킨다.

**Git push 로 마무리**

**브런치 병합하기**

**\*\*작업 브런치는 병합 후 제거하는 case**

**Git rebase master(🡨현재 작업중 브런치가 합쳐질 브런치명**

**Git checkout master( 메인 브런치로 이동)**

**Git pull –rebase=preserve (원격 브런치의 최신 사본 업데이트 받기)**

**Git merge 1-first (master에서 1-first 병합)**

**마지막으로 git push 로 로컬의 버전을 원격으로 반영해주면 끝~**

\*\*rebase 는 base를 다시 쌓는다는 뜻으로 메인 뿌리의 base를 master 하나로 재정립하겠다는 뜻임!

작업중인 브런치의 rebase 명령어를 주면 뿌리가 같아지기 때문에 병합하게 되면 작업이 없었던 master가 최신 상태로 버전을 전진하게 된다.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **요일** | **월** | **화** | **수** | **목** | **금** | **토** | **일** |
| **6:00~** | **스프링책** | **운동** |  | **운동** |  | **스터디모임** |  |
| **7:00~** |  |  |  |  |  |  |  |
| **8:00~** |  |  |  |  |  |  |  |
| **9:00~** | **정리** |  |  |  |  |  |  |
| **10:00~** | **귀가** |  |  |  |  |  |  |
| **11:00~** |  |  |  |  |  |  |  |
| **12:00~** |  |  |  |  |  |  |  |
| **01:00~** |  |  |  |  |  |  |  |