**Git의 스테이징 단계**

- git add으로 진행되는 형상 관리가 3가지 영역에서 진행된다.

① 워킹 디렉토리

- 소스 코드를 작업하는 영역으로 코드를 추가, 수정, 삭제하는 작업이 이루어지는 영역을 의미한다.

② 스테이징 영역

- 워킹 디렉토리에서 git add 명령을 실행하면 파일들은 Git의 스테이징 영역으로 이동하며 이를 통해

소스 코드의 상태 정보를 확인할 수 있다.

③ 저장소 영역

- 스테이징 영역에 있는 소스 코드에 git commit 명령을 실행하면 최종적으로 Git의 저장소에

반영된다.

**파일 관점에서 Git의 4가지 상태**

① Untracked : 워킹 디렉토리에 추가되었지만 Git에서 관리하지 않는 상태

② Unmodified : 신규로 파일이 추가되었을 때의 상태로 new file 상태와 동일

③ Modified : 파일이 추가된 이후 해당 파일이 수정되었을 때의 상태

④ Staged : Git의 스테이징 영역에 반영된 상태

**스테이징에 있는 파일 삭제 – git rm --cached 파일명**

**Git 이력 조회 및 변경 내용 비교**

➀ 이력 관리 기능

- 소스 코드가 언제 누구에 의해서 생성, 변경, 삭제를 했는 지 확인을 할 수 있다.

➁ 스냅샷

- Git에서는 커밋이 이루어질 때마다 모든 형상 관리 파일들을 바이너리 형태로 묶어서 관리하는 것

- Git에서 하나의 스냅샷은 하나의 커밋을 의미하며, 스냅샷 단위로 이력을 관리한다.

➂ git log

- 이력 정보 확인

- 커밋한 작업자, 일시, 메시지를 확인할 수 있다.

git log –p

- 버전과 버전사이의 코드 상태에서의 차이점을 비교

git log –-oneline

- commit id의 일부와 commit message만 보여준다.

➃ git diff

- 저장소의 파일과 워킹 디렉토리에 있는 파일을 비교해, 파일내의 콘텐츠 변경 부분을 확인할 수 있다.