branch

* HEAD : 현재 속한곳
* master : 기본 브랜치
* git branch : 브랜치 목록
* git branch [name] : 현재 머물고 있는 브랜치를 복사한 name 브랜치 생성
* git branch [name] [hash] : 해당 해쉬복사복인 name 브랜치 생성
* git checkout [branch\_name] : 해당 브랜치로 HEAD 이동
* git checkout -b [branch\_name] : 브랜치 생성과 HEAD 이동 한번에

브랜치 병합 marge

* git merge [b\_name] : 현재 브랜치에 b\_name 브랜치를 병합함
  + 바로 커밋창 나오는데 알아서 채우면 됨
* merge 후 충돌 과정
  + merge후 자동적으로 merge 안되고 충돌이 됐다는 글이 나오면
  + 해당 파일로 가면 ===== >>>>>masger 같은 표시로 충돌 나는 부분을 명시함
  + 충돌 과정 수정후 add하고 cim이 아닌 ci를 해주면 자동으로 merge branch 라는 내용이 들어감 커밋내용 알아서 하고 wq하고 나오면 충돌 해결 끝
* 3 way merge : 2개의 브랜치의 공통 조상을 가져와 merge을 더더욱 자동화 하는 방법
* p4merge : 충돌tool 다운로드
  + p4merge git config 설정해주면 사용 가능
  + 사용법은 구글링 해야함
  + tool가서 수정후 bash에서 바꾼파일은 자동으로 add 되고 .orig파일을 삭제(원래의 충돌 파일 == >> 포함된)하고 커밋하면된다.

checkout vs reset

* checkout은 HEAD을 제어하는것이다. 꼭 브랜치로 갈필요 없고 특정 버전으로 갈수 있다. 이제 이런 상태를 detached 상태하고 부른다.
* reset은 HEAD가 속한 브랜치(현재 브랜치)를 제어하는 것이다. HEAD는 해당 브랜치를 가리키고 있으니 당연히 HEAD도 같이 움직이긴한다.