git

일련번호는 주로 7자리로 구분
바로이전으로 되돌리기
git reset --hard HEAD~
HEAD=커서
헤드를 내려서 이전 저장본을 헤드로 만든다
branch pointer 같이 내려와야 함
HEAD, branch pointer 같이 끌어내리는 reset --hard HEAD~

```
       p3 added v <-master     <--HEAD
       p2 added o <-master     <--
       p1 added o <-master     <--
index.html add o <-master      <--
        branch o pointer       Cursor
                                       reset --hard HEAD~
       p3 added o <-master     <--
       p2 added v <-master     <--HEAD
       p1 added o <-master     <--
index.html add o <-master      <--
        branch o pointer       Cursor
                                        reset --hard HEAD~
       p3 added o <-master     <--
       p2 added o <-master     <--
       p1 added v <-master     <--HEAD
index.html add o <-master      <--
        branch o pointer       Cursor
```

안전한 공간에서 수정을 해보고 싶다
lab1이라는 branch를 만들어서 수정을 해려함
default branch point - master

```
             p3 added v <-master     <--HEAD
             p2 added o <-master     <--
             p1 added o <-master     <--
       index.html add o <-master     <--
               branch o pointer      Cursor
                                              git branch lab1
                                              git checkout lab1
```

HEAD--> lab1->p3 added v <-master <--
p2 added o <-master <--
p1 added o <-master <--

index.html add o <-master <--
Cursor branch o pointer

```
    working tree
```

HEAD--> lab1->p3 added v o <-master <--
p2 added o <-master <--
p1 added o <-master <--
index.html add o <-master <--
Cursor branch o pointer

```
                                                    git checkout master
                                    working tree
     lab1-> p3 added o v <-master     <--  HEAD
              p2 added o <-master     <--
              p1 added o <-master     <--
        index.html add o <-master     <--
                branch o pointer           Cursor
```

병합하기
git merge

```
                                    working tree
     lab1-> p3 added o v <-master     <--  HEAD
              p2 added o <-master     <--
              p1 added o <-master     <--
        index.html add o <-master     <--
                branch o pointer           Cursor

                                                    git merge lab1
                                    working tree
           p3 added v   <-master,lab1 <--  HEAD
              p2 added o <-master
              p1 added o <-master
        index.html add o <-master
                branch o pointer           Cursor

같은 부분을 수정한 여러 branch가 있다면 병합할 때 충돌이 일어날 수도 있다.

Git
```

git init을 한 폴더는 저장소가 됨
git local repository
git remote repository(git hun, bitbucket)

```
local에서 remote로 저장을 하게 되면 어느곳에서든 업로드 다운로드 할 수 있음
social coding도 가능
```

Untracked file - init은 했지만 add하지 않은 file
commit 한 직후 working tree clean

staging - git add .
picturing - git commit -m""

- `add . // commit -m""`

working tree --> stage --> local repositary
0000 --> (staged) --> (commited)
0001 --> (staged) --> (commited)
0011 --> (staged) --> (commited)

git gui
git gui & 독립적인 프로세스

fast-forward merging
같은 줄의 branch를 올리는 일

git branch -d 삭제

같은 자리를 수정하게 되면 충돌이 나게 되는데 충돌을 해결하고 add, commit 하면 해결됨 (<<<<<<<<<<<<<<<<<

> 지워줘야함)

두 commit들이 갈라졌었으니 merge commit이 생긴다

깃허브
git remote add origin 00000000
git remote -v 저장소 목록 보기

account -settings -developer settings -personal access tokens

local에는 원격저장소를 tracking하는 저장소가 있다 (origin/master)
Your branch is ahead of 'origin/master' by 2 commits.
ㄴafter push -up to date

CLONE
저장소의 상태 그대로 가져오기
자동으로 remote repositary를 tracking하는 local repositary가 생긴다
commit, push 가능
저장할 폴더에 git clone *address*(한번만함)
github에 있던 폴더가 생성된다.
fetch - 커밋 내려받기 merge - 커밋 합치기(fetch+merge=pull)
내려받은 working tree 에서는 origin/master가 master보다 앞선다 merge를 해줘야함

남의 저장소 사용하는 방법

1. 그냥 다운로드
2. 나의 컴퓨터의 특정 폴더로 clone
3. 남의 github 저장소를나의 github으로 fork한다
   나의 github에 있는 fork 된 저장소를
   나의 computer의 특정 폴더로 저장소를 clone 해서 사용

git commit 만 단독으로 사용하게 된다면 editor가 열림
vi editor
edit mode(INTERT)
commit message를 적는다. 여러줄 적을 수 있음
명령모드로 바꿈 (esc)
:wq

```
        command mode(default)명령모드
              edit mode로 가려면 enter i (INSERT)
```