**Git 11강**

Commit 2개 5와 4를 삭제하고 3으로 돌아가고 싶을 때

1. Reset(초기화) : git reset (commit 3에 적혀 있는거) – hard + Enter

Vim + :q(나갈 때)

* git에서는 왠만한 정보를 삭제하지 않음. 다만 우리 눈에 보이지 않을 뿐. 필요할 때는 복구를 할 수 있는데, git의 원리를 알아야지 가능한 것
* 자신의 버전을 다른 곳에 공유하면 절대로 reset을 하면 안됨

Git log 했을 때 - - hard (나중에는 soft, option을 섬세하게 사용할 것)

1. Git revert: reset처럼 commit을 최소화 하는 것인데, 휙 날리는 것이 아닌 새로운 버전을 생성하는 것

**Git 12강**

1. 중국어 단어를 영어로 물어보는 것
2. 영어 단어가 중국어로 무엇인지의 표현을 물어 봄

내가 모르는 것을 알아낼 수 있는 방법을 알아내기

Git 역시 여러가지 부품(명령어)이 있음(commit add log diff init 순으로 많음 지금까지 공부한 비중은 40%를 차지함)

1. Git 복습

어떤 파일이 바뀔 때: Vim

Git add f1.txt

Git commit -> editor -> 버전 작성

Git log

clear

git commit - - help: commit 메시지에 대한 도움말을 볼 수 있다.

Vim f1.txt -> 4 저장

Git commit -a: add가 없이도 메시지를 작성할 수 있음 / 자동으로 파일을 git commit add 시킨다.

Git commit -am: editor를 띄우지 않고 바로 메시지를 작성한다. 후에 git log를 통해 결과 볼 수 있음

**13강: 중간단계**

**14강: git의 원리 입문**

Git의 명령어들이 어떤 매커니즘에 의해 동작하는지 소개

**15강: 분석도구 gistory**

1. Git의 첫번째 작업을 분석
2. 어떠한 명령을 내렸을 때 .git이라는 directory안에서 어떤 일이 일어나는지 살펴보기

조금 더 쉽게 git을 분석하는 방법: gistory (설치: <https://patiencelee.tistory.com/707>) 이후에 .git이라는 directory가 있어야 함)(local gist: 숫자 -> gistory에 들어갈 수 있음)

**16강: git add**

상위: cd ..

새 프로젝트 폴더: mkdir gitfth2

저장소 초기화: git init

Gistory + enter : 8805 -> 인터넷에서 gitstory 웹사이트 열어서 볼 수 있음

끌 때: ^C%

다시 킬 때: gistory (readonly일 경우: https://huskdoll.tistory.com/98)

이후 readonly를 하면 아무 일도 일어나지 않음

$ git add f1.txt를 하고 보면 index와 objects가 나옴

objects에는 우리가 적은 a라는 내용이 담겨 있으나, a가 담긴 파일의 이름은 index라는 파일에 적혀 있음(789819에 속하는 곳에 적혀 있다.)

즉, 파일의 이름: index, 파일의 내용: object에 담겨 있음

F2.txt를 눌러보고 변화 살펴보면 각각의 파일명과 파일 관련 정보가 index에 적혀 있는 것을 볼 수 있다.

Cp F1.txt f3.txt: index를 클릭했더니 f1.txt(7898) f2.txt(b680) f3.txt(7898) -> 파일을 저장할 때 이름이 달라도 내용이 같으면 같은 object 파일에 저장한다.

**17강: objects 파일명의 원리(내용이 같으면 파일명이 같다)**

Git은 SHA1 알고리즘을 통과 시켜서 두 글자를 떼서 object 밑에다가 aa라는 directory, f4라는 directory를 만들어서 hello를 만든다.

어떤 파일에 대해 git add를 하면 git은 파일의 내용을 보고, a라는 정보가 있으면 a+여러가지 파일을 추가해 압축하고, 이를 통과시켜서 값에 해당되는 objects라는 directory에 만들고 a라는 정보를 저장한다.

F1.txt를 입력해주면 작업이 끝난다.

**18강: commit의 원리**

Ls -al로 보고

Git commit을 해주고 commit message를 1로 해준다.

Reload를 하면 gitstory에 5개가 생기는데, commit 메시지에 해당하는 정보가 objects라는 곳 안에 저장이 되어 있다.

Commit 안에 tree(object: e42e: 우리가 작성한 버전의 파일 이름) + 내용

F1이라는 파일과 f2, f3에 무엇이 있는지 적혀 있는 것

F2.txt에 내용을 y z로 바꿔주기

Commit이 담긴 개체 수 + parent(이전 commit을 볼 수 있음)

Tree는 2개의 commit에 적혀 있는데, 이때 두 값이 다르다.

F2.txt는 z를 가리키는 object를 담고 있다.

F2.txt를 눌러보면 y z가 나온다.

1. Parent: 부모 값
2. Commit이 일어난 시점의 작업 directory에 있는 파일 이름과 파일 이름이 담고 있는 내용 사이의 tree가 담겨 있음

우리는 버전에 적혀 있는 tree를 통해서 project 폴더에 대한 상태를 볼 수 있다 = 사진을 찍었다(snap shoot)

\*\* directory 만들기

Git add d1/f1.txt 만들고 결과 살펴보면 각각의 parent, tree를 가지고 있음

Object 파일

1. 파일의 내용을 담는 것(blob)
2. 어떤 directory의 파일과 내용에 해당되는 blob에 대한 정보를 담고 있는 것(tree)
3. Commit

19강: status의 원리