

## GIT

파일의 이름을 바꾸지 않고, 파일의 변경사항은 컴퓨터가 관리하게 한다.  
버전을 관리하는 것 외에도 Backup, Recovery, Collaboration이 가능함.

CVS -> SVN -> GIT

이런 식으로 version control systems 의 발전을 이뤄감

나름대로 계보가 있고, 여러가지의 버전이 있음

본질적인 요소는 공통적인 요소 (변경사항을 관리하는 것)

혁신적인 요소는 새로운 버전 관리 시스템이 나왔을 때 그 전의 시스템을 뛰어넘을 수 있었던 차이점

## git init

- 현재 버전 관리를 하는 것
- 지우면 안됨! (버전 정보 다 날라감)

**cat : 입력하면 안의 내용들을 볼 수 있음**

**\*\*vim > 엔터 눌러서 안에 내용 작성가능**

작성 후 esc 누르고 : 누른 후 wq 누르면 저장 후 닫기

버전 : 의미있는 변화

```
$ git config --global user.name colawhite
```

```
user@DESKTOP-FU7U8CE MINGW64 ~/Documents/gitfth (master)
```

```
$ git config --global user.email cooker100@naver.com
```

이건 딱 한번만 하면 됨.

내 이름이랑 이메일이 명시되어 있어서 파악가능

이후에 이렇게 git commit 하면 이렇게 뜨는데

맨 위에 commit 메시지를 입력하면 됨



```
commit 3460e510316967ce5df0e157bf140df0b41f6641
```

```
Author: colawhite <cooker100@naver.com>
```

```
Date: Fri Sep 20 18:33:21 2024 +0900
```

이런식으로 버전 2를 기록해봄

[https://www.infllearn.com/course/lecture?  
courseSlug=%EC%A7%80%EC%98%A5%EC%97%90%EC%84%9C-%EC%98%A8-  
git&unitId=11599&tab=curriculum](https://www.infllearn.com/course/lecture?courseSlug=%EC%A7%80%EC%98%A5%EC%97%90%EC%84%9C-%EC%98%A8-git&unitId=11599&tab=curriculum)