

# 컴퓨터프로젝트관리

## 2. commit & init

첫 번째 문제로 돌아가봅시다  
텍스트 파일이 하나 있고,  
수정한 내용을 기록해두고 싶다



“컵라면 1박스 구입” 이라는 수정이 발생

“수정”의 단계를 어느 정도까지 기록해야할까?

저장할 때 마다?

컵(Save)라(Save)면(Save)... 이런식으로  
저장했다해도 모두 기록해야할까?



# 규칙 1

“수정”의 단계는 “의미”를 기준으로

## 의미를 어떻게 알지?

SW가 의미를 인식할 수  
없으니 정할 수 있는 사람 즉,  
사용자가 정하는 것으로 하자

# 규칙 1

“수정”의 단계는 “의미”를 기준으로

# commit

이 파일의 수정이 끝났다  
“이 파일의 의미있는 수준의 수정 작업이 끝났음”  
이라고 git 에게 알려주는 일

# 세 번의 커밋

commit 3

2014년 7월 10일 17:00

컵라면 구매 할 일 추가

@홍길동

memo.txt

오늘의 할 일

- 5
- 6 L 마트에 가서 맥주 한박스
- 7 L 20리터 쓰레기 봉투 구입
- 8 L 컵라면 1박스 구입
- 9

commit 2

2014년 7월 9일 10:03

쓰레기 봉투 구매 할 일 추가

@홍길동

memo.txt

오늘의 할 일

- 5
- 6 L 마트에 가서 맥주 한박스
- 7 L 20리터 쓰레기 봉투 구입
- 8
- 9

commit 1

2014년 7월 6일 12:42

맥주 구매 할 일 추가

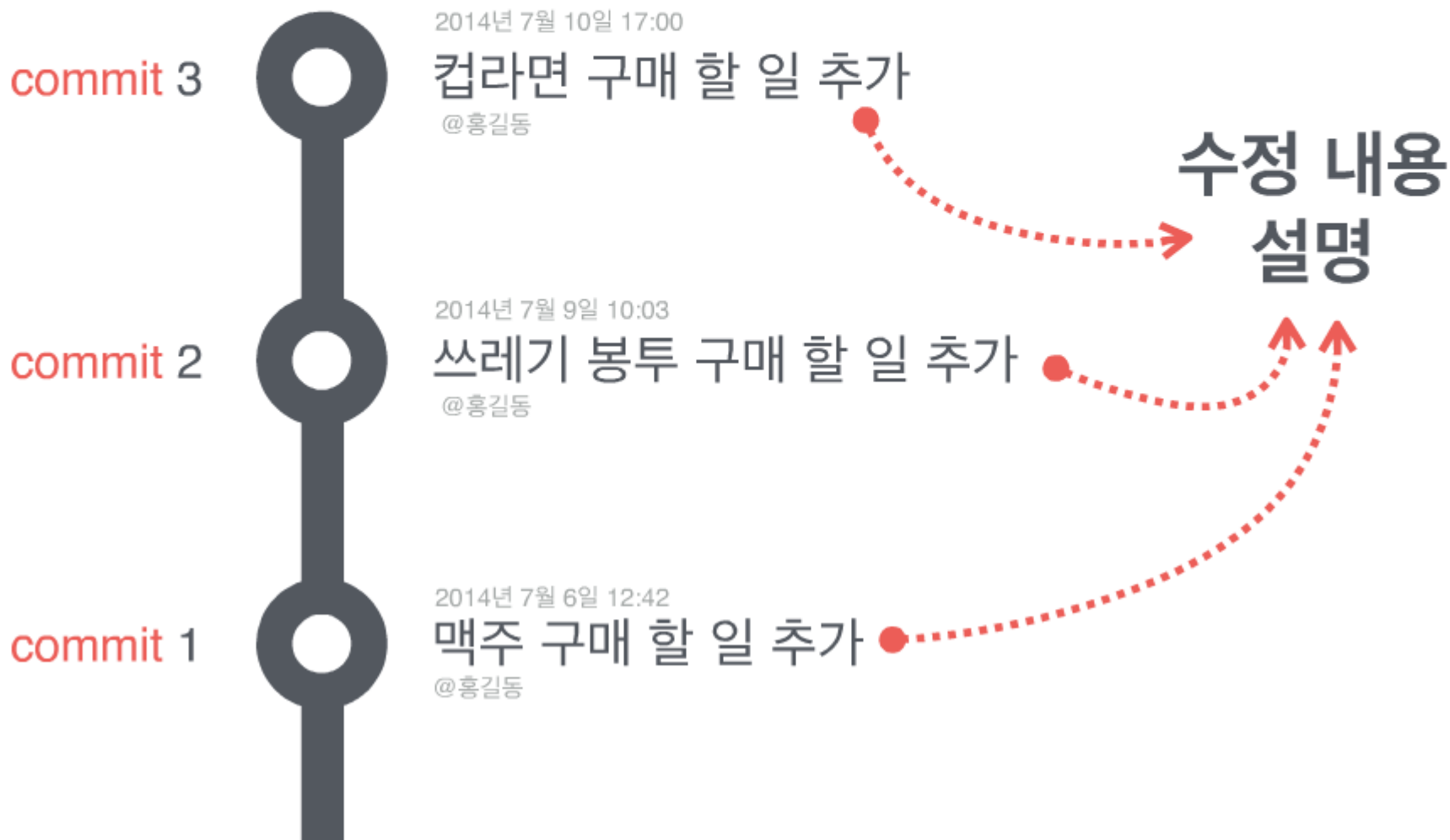
@홍길동

memo.txt

오늘의 할 일

- 5
- 6 L 마트에 가서 맥주 한박스
- 7
- 8

# 수정한 내용을 간략하게 표현하기 위한 “커밋 메시지” 기록



# 두번만 할 수도 있겠죠?

커밋의 기준을 정하는건 사용자.

커밋의 단위를 작게 할 수록 변경 내용의 기록

수준은 정밀해지며 파악할 수 있는 정보도 많아진다

commit 2

2014년 7월 10일 17:00

쓰레기 봉투, 컵라면 구매 할 일 추가

@홍길동



memo.txt

오늘의 할 일

5  
6 L 마트에 가서 맥주 한박  
7 L 20리터 쓰레기 봉투 구  
8 L 컵라면 1박스 구입

commit 1

2014년 7월 6일 12:42

맥주 구매 할 일 추가

@홍길동



memo.txt

오늘의 할 일

5  
6 L 마트에 가서 맥주 한박  
7  
8



# git 의 시작

지정한 폴더의 변경 내용을 추적하기위한 준비

# init

선택된 폴더의 변경 내용을 추적하기 위해 git 저장소를 만든다. 저장소가 만들어지면 git은 지정한 폴더를 포함하여 하위 폴더의 모든 변경 내용을 커밋 단위로 추적한다.

# 정리

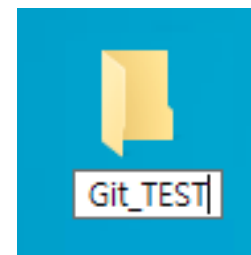
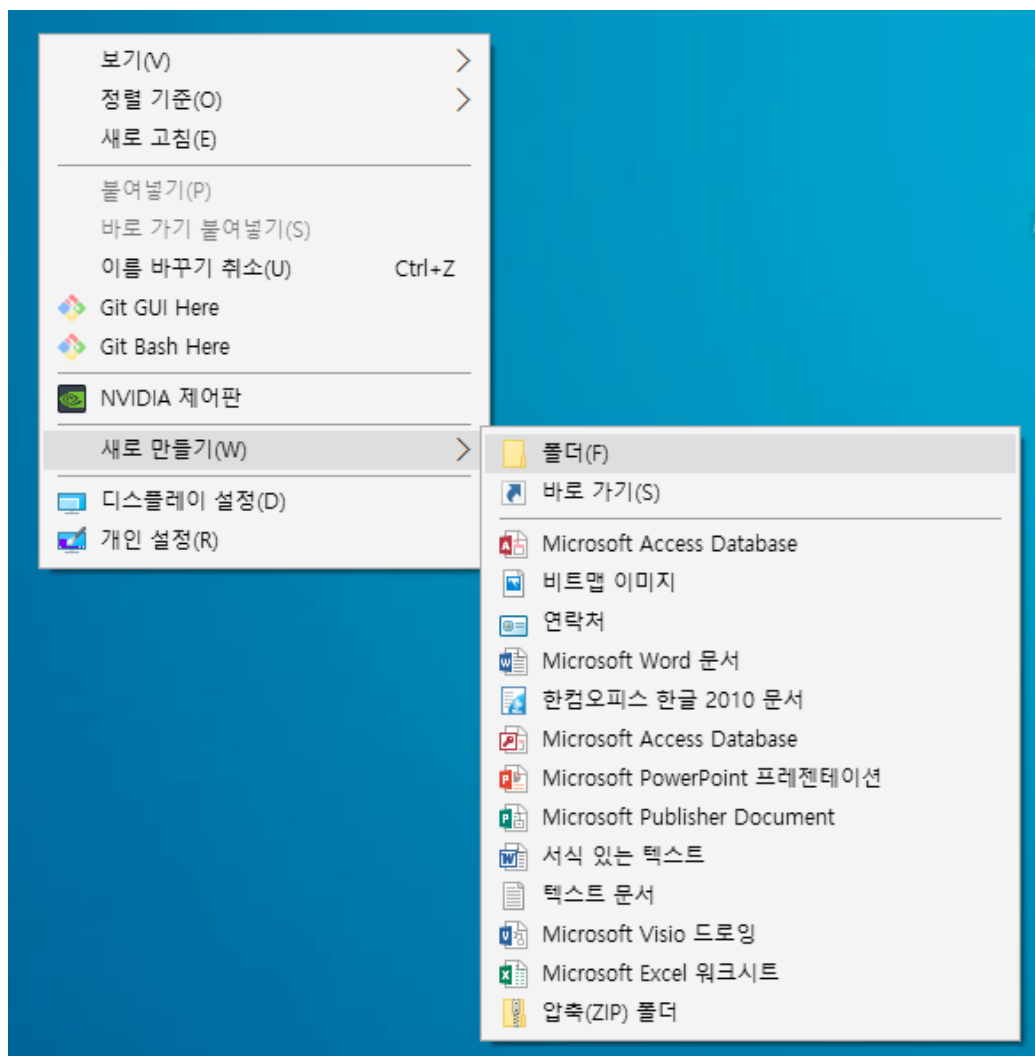
**commit**

수정 내역을 사용자 기준의 의미로 기록한다

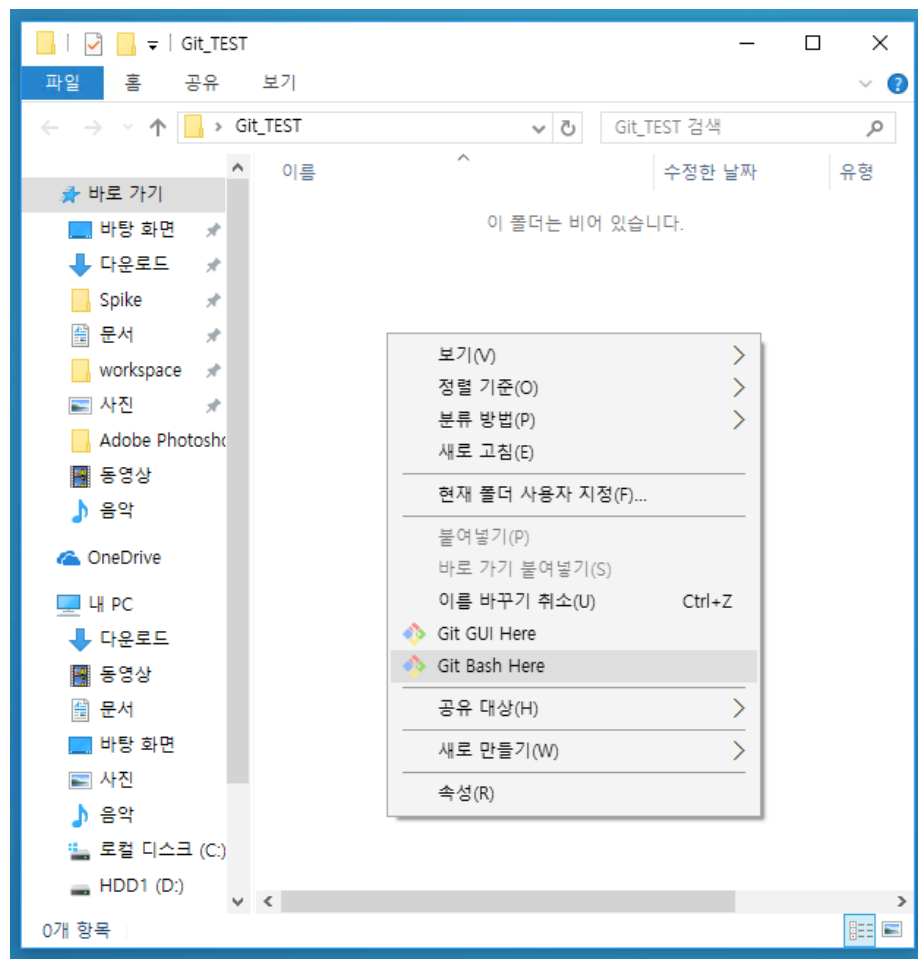
**init**

지정된 폴더의 변경 내용을 추적하기 위한 준비

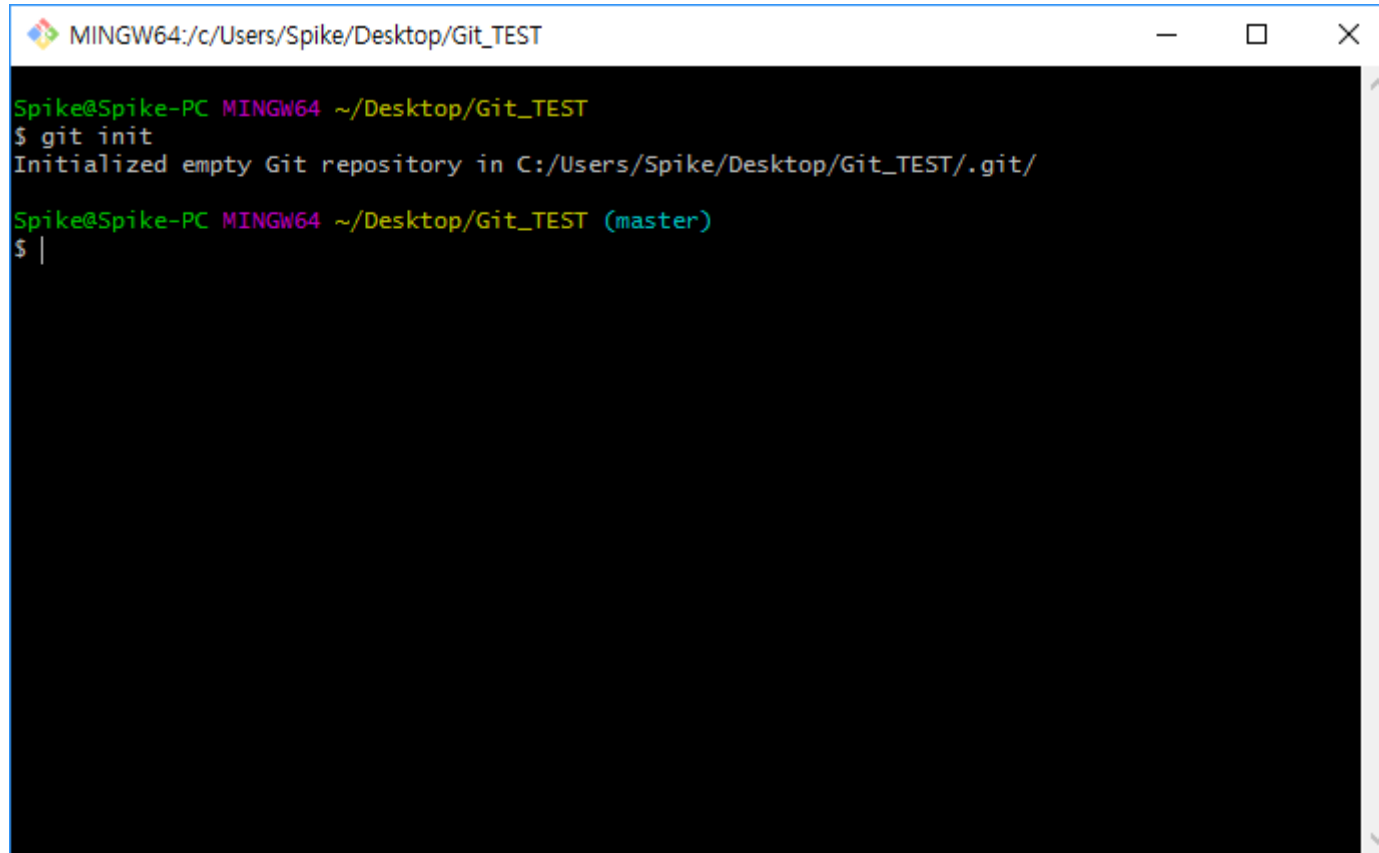
# 폴더 만들기



# Git bash 실행

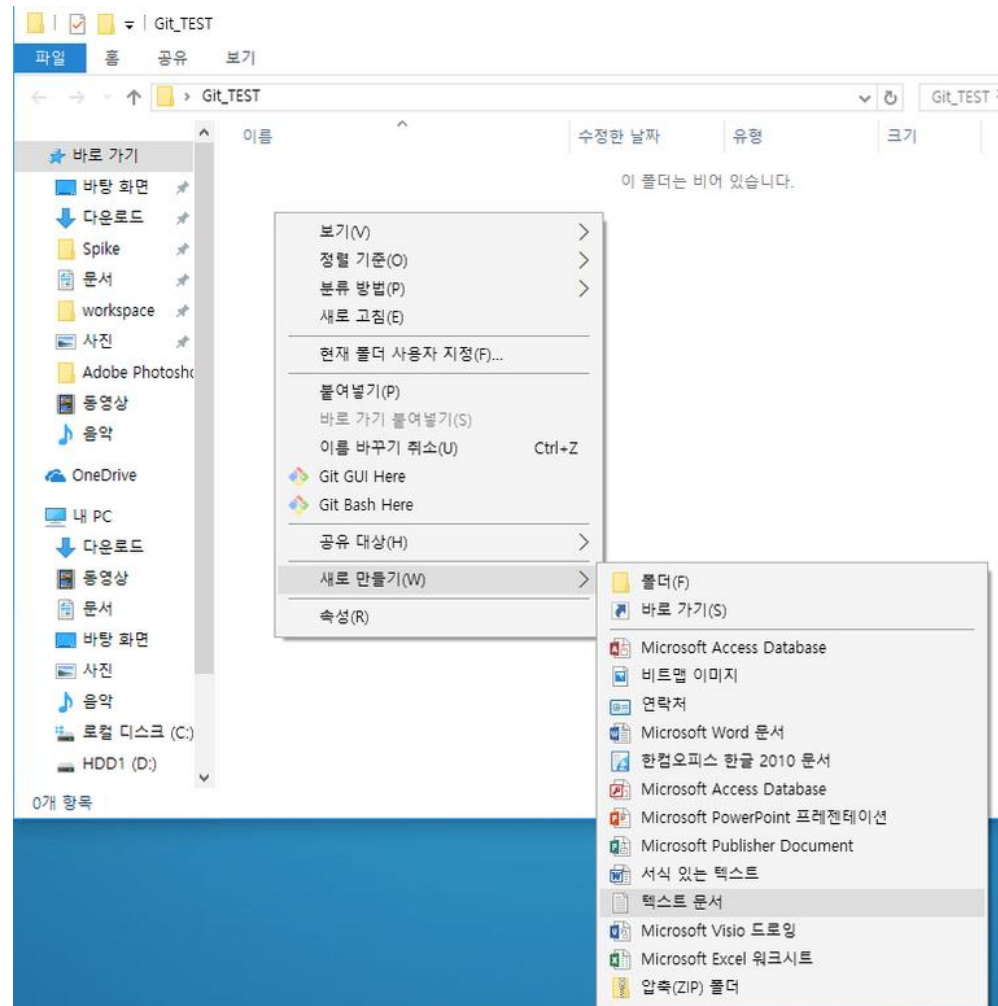


# Git init 명령어 실행

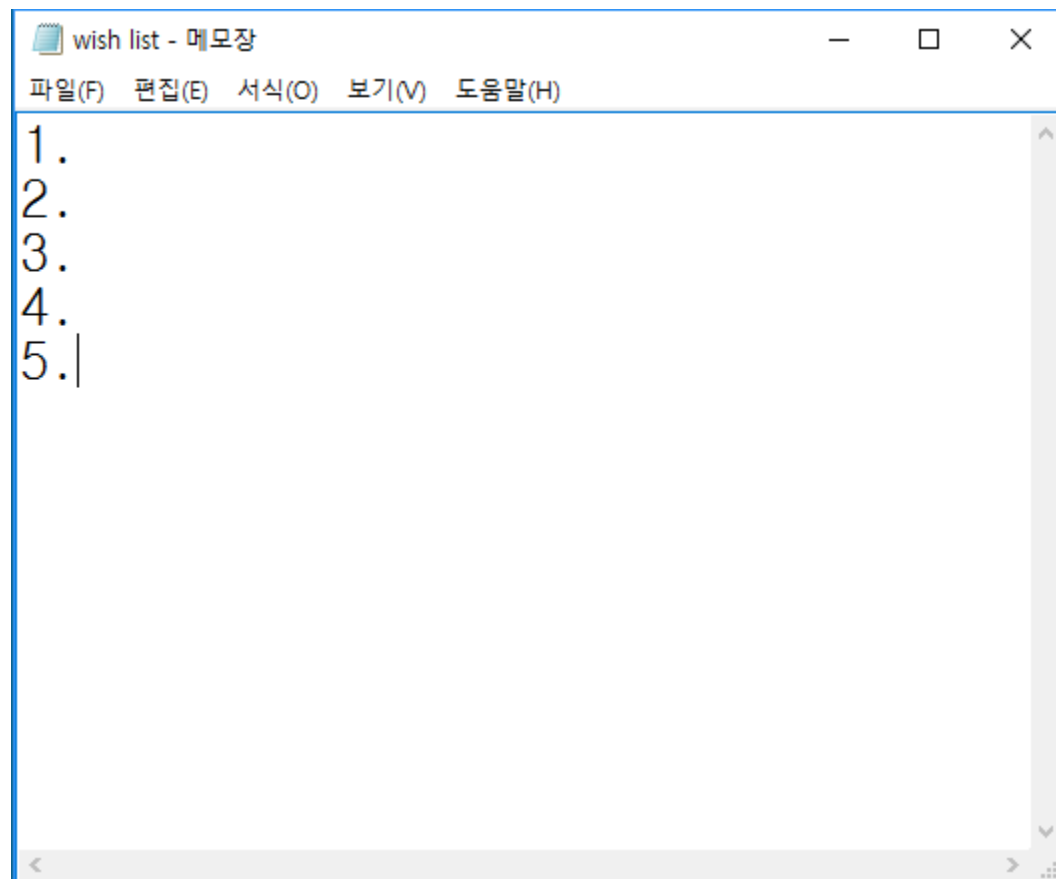


```
MINGW64:/c/Users/Spike/Desktop/Git_TEST
Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST
$ git init
Initialized empty Git repository in C:/Users/Spike/Desktop/Git_TEST/.git/
Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ |
```

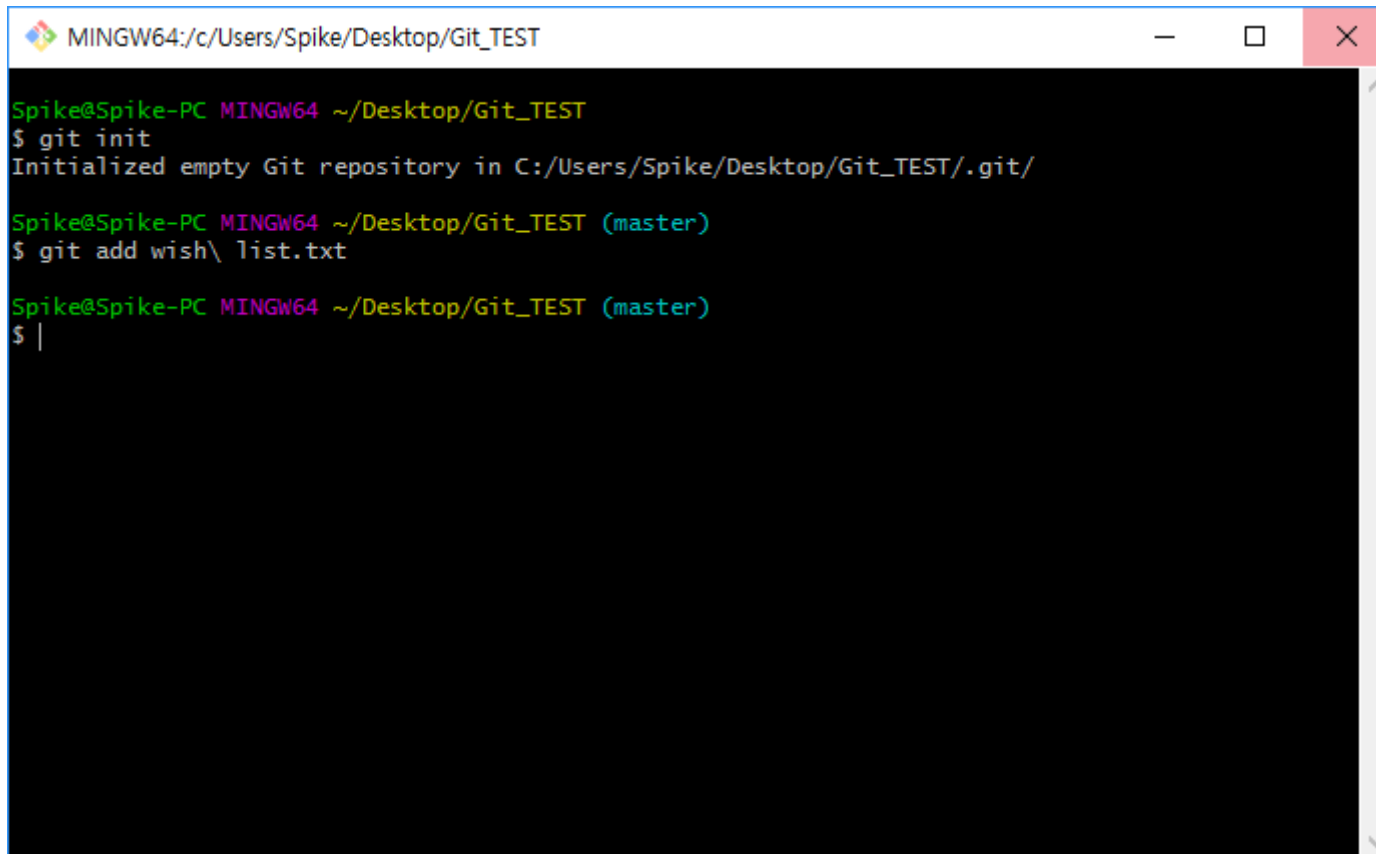
# txt 파일 생성



# 파일 작성



# 파일 추가



```
MINGW64:/c/Users/Spike/Desktop/Git_TEST

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST
$ git init
Initialized empty Git repository in C:/Users/Spike/Desktop/Git_TEST/.git/

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git add wish\ list.txt

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ |
```



# commit

```
MINGW64:/c/Users/Spike/Desktop/Git_TEST
Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST
$ git init
Initialized empty Git repository in C:/Users/Spike/Desktop/Git_TEST/.git/

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git add wish\ list.txt

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git commit -m '최초 커밋'
[master (root-commit) d51ffa7] 최초 커밋
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 wish list.txt

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ |
```

# 확인

```
MINGW64:/c/Users/Spike/Desktop/Git_TEST
Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST
$ git init
Initialized empty Git repository in C:/Users/Spike/Desktop/Git_TEST/.git/

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git add wish\ list.txt

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git commit -m '최초 커밋'
[master (root-commit) d51ffa7] 최초 커밋
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 wish list.txt

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ git log
commit d51ffa79b1eb732a9dd440293f806dba143d3647 (HEAD -> master)
Author: kai1945 <rkdtkddl@gmail.com>
Date: Mon Sep 11 21:27:27 2017 +0900

    최초 커밋

Spike@Spike-PC MINGW64 ~/Desktop/Git_TEST (master)
$ |
```

# 정리

**commit**

수정 내역을 사용자 기준의 의미로 기록한다

**init**

지정된 폴더의 변경 내용을 추적하기 위한 준비

**add**

commit 할 파일들을 추가

**log**

commit 한 내용을 보기위한 로그