

23-10-18

- Dos 시스템과 리눅스 시스템에서 파일 만들기
- Dos 시스템에서 파일 깃허브에 저장하는 방법

<리눅스 명령어>

- Sudo apt-get install 패키지명 //패키지명 설치
- Sudo apt-get install mariadb-server //mariadb-server 설치

1. apt-get => 설치목록
2. update => 목록
3. upgrade => 세부내용
4. remove => 삭제(패키지의 삭제)_설정파일 남김

나중에 다시 설치할 때 자료남기기 위해서 사용

5. purge => 삭제(패키지의 삭제)_설정파일안남김(같이제거)

나중에 다시 설치할 때 깔끔하게 제거되서 자료가 안남음

6. list => 조건목록

Q. 리눅스에서 사용할 때 삭제는 무엇을 사용할까요?

A. Purge 사용!

Remove를 사용하면 자료 기록이 남아서 손봐줘야 한다

Sudo apt-get install 패키지명
Sudo apt-get install mariadb-server

remove => 삭제 패키지 (설정 파일 남김)
purge => 삭제 패키지 (설정 파일 같이제거)
list => 조건 목록

- `sudo dpkg -l | grep mariadb` //마리아DB 사용하는 모든 패키지를 보여준다

```
you@dbserver: ~
you@dbserver:~$ sudo dpkg -l | grep mariadb
[sudo] password for you:
ii mariadb-client-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database client binaries
ii mariadb-client-core-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database core client binaries
ii mariadb-common 1:10.3.38-0ubuntu0.20.04.1 all
4 MariaDB common metapackage
ii mariadb-server 1:10.3.38-0ubuntu0.20.04.1 all
4 MariaDB database server (metapackage depending on the latest version)
ii mariadb-server-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database server binaries
ii mariadb-server-core-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database core server files
you@dbserver:~$
```

마리아디비 사용하는 모든 패키지이다
마리아 DB 총 패키지이다
우리가 사용한 마리아 DB 총 패키지이다
그 중에 우리는 2개 설치함

```
you@dbserver:~$ sudo apt-get remove mariadb-client-10.3
remove는 설정값 안지워진다
```

```
you@dbserver:~$ sudo apt-get purge mariadb-client-10.3
Reading package lists... Done
Building dependency tree
Reading state information... Done
purge는 다 지워진다
The following packages were automatically installed and are no longer required:
galera-3 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl
libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl
libhttp-date-perl libhttp-message-perl libio-html-perl
liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl
libtimedate-perl liburi-perl mariadb-client-core-10.3 mariadb-common
mariadb-server-core-10.3 mysql-common socat
Use 'sudo apt autoremove' to remove them.
The following packages will be REMOVED:
mariadb-client-10.3* mariadb-server* mariadb-server-10.3*
0 upgraded, 0 newly installed, 3 to remove and 0 not upgraded.
After this operation, 90.7 MB disk space will be freed.
Do you want to continue? [Y/n]
```

Purge를 사용하여 mariadb 모든 패키지를 지웠다.

```
you@dbserver:~$ sudo dpkg -l | grep mariadb
ii mariadb-client-core-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database core client binaries
ii mariadb-common 1:10.3.38-0ubuntu0.20.04.1 all
4 MariaDB common metapackage
ii mariadb-server-core-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64
4 MariaDB database core server files
you@dbserver:~$ s
```

client 10.3 만 지웠는데 6개에서 3개만 남는다

```
you@dbserver: ~  
you@dbserver:~$ sudo apt-get purge -y mariadb-client-core-10.3  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following packages were automatically installed and are no longer required:  
galera-3 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl  
libdbd-mysql-perl libdbi-perl libencode-locale-perl libfcgi-perl  
libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl  
libhttp-date-perl libhttp-message-perl libio-html-perl  
liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl  
libtimedate-perl liburi-perl mariadb-common mariadb-server-core-10.3  
mysql-common socat  
Use 'sudo apt autoremove' to remove them.  
The following packages will be REMOVED:  
mariadb-client-core-10.3*  
0 upgraded, 0 newly installed, 1 to remove and 0 not upgraded.  
After this operation, 28.3 MB disk space will be freed.  
(Reading database ... 73076 files and directories currently installed.)  
Removing mariadb-client-core-10.3 (1:10.3.38-0ubuntu0.20.04.1) ...  
Processing triggers for man-db (2.9.1-1) ...  
you@dbserver:~$ sudo dpkg -l | grep mariadb mariadb 모든 패키지 지웠다  
ii mariadb-common 1:10.3.38-0ubuntu0.20.04.1 all  
    MariaDB common metapackage  
ii mariadb-server-core-10.3 1:10.3.38-0ubuntu0.20.04.1 amd64  
    MariaDB database core server files  
you@dbserver:~$
```

- ls -al //모든 파일 검색

```
you@dbserver:~$ ls -al  
total 36  
drwxr-xr-x 4 you you 4096 Oct 17 11:46 .  
drwxr-xr-x 3 root root 4096 Oct 16 15:20 ..  
-rw-r----- 1 you you 1032 Oct 18 09:58 .bash_history DB 서버가 아래 나와 있다.  
-rw-r--r-- 1 you you 220 Feb 25 2020 .bash_logout . (점하나는) 나 자신을 말함  
-rw-r--r-- 1 you you 3771 Feb 25 2020 .bashrc .. (점두개는) 윗단계를 말함  
drwx----- 2 you you 4096 Oct 16 15:32 .cache  
-rw-r----- 1 you you 0 Oct 17 11:26 .mysql_history .cache (파란글자) 숨김파일이다  
-rw-r--r-- 1 you you 807 Feb 25 2020 .profile  
drwx----- 2 you you 4096 Oct 16 15:20 .ssh  
-rw-r--r-- 1 you you 0 Oct 16 15:38 .sudo_as_admin_successful  
-rw-r----- 1 you you 882 Oct 17 11:46 .viminfo  
you@dbserver:~$
```

- cd .ssh //ssh로 이동
- cd .. //상위 단계로 이동

```
you@dbserver:~$ cd .ssh  
you@dbserver:~/ssh$ ls -al  
total 8  
drwx----- 2 you you 4096 Oct 16 15:20 .  
drwxr-xr-x 4 you you 4096 Oct 17 11:46 ..  
-rw-r----- 1 you you 0 Oct 16 15:20 authorized_keys  
you@dbserver:~/ssh$ cd ..  
you@dbserver:~$
```

.ssh 로 이동한거고 그 안에 파일은 없고
cd.. 해서 상위 단계로 간것

- tree //

```
papa@dbserver:~/ssh$ ls -al
total 8
drwx----- 2 papa papa 4096 Oct 17 10:04 .
drwxr-xr-x 4 papa papa 4096 Oct 17 14:00 ..
-rw----- 1 papa papa 0 Oct 17 10:04 authorized_keys
papa@dbserver:~/ssh$ cd ..
papa@dbserver:~$ tree
```

snap을 사용하면 무엇이 좋은가?
리눅스 내에서 프로세스 단위에서 처리하는 것이 달
라짐

tree를 사용하고 싶다
아래 목록은 업데이트 했기 때문에 나옴

Command 'tree' not found, but can be installed with:

```
sudo snap install tree # version 1.8.0+pkg-3fd6, or
sudo apt install tree # version 1.8.0-1
```

See 'snap info tree' for additional versions.

```
papa@dbserver:~$
```

```
you@dbserver:~$ sudo apt-get install tree
```

```
you@dbserver:~$ tree
.
0 directories, 0 files
you@dbserver:~$
```

파일이란 디렉토리 없다는 뜻
숨김파일은 안보여줌

- mkdir test01 //test01이라는 디렉토리 만든다는 것
- rmdir test01 //test01이라는 디렉토리 삭제하는 것

```
you@dbserver:~$ mkdir test01
you@dbserver:~$ ls -al
total 40
drwxr-xr-x 5 you you 4096 Oct 18 10:42 .
drwxr-xr-x 3 root root 4096 Oct 16 15:20 ..
-rw----- 1 you you 1032 Oct 18 09:58 .bash_history
-rw-r--r-- 1 you you 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 you you 3771 Feb 25 2020 .bashrc
drwx----- 2 you you 4096 Oct 16 15:32 .cache
-rw----- 1 you you 0 Oct 17 11:26 .mysql_history
-rw-r--r-- 1 you you 807 Feb 25 2020 .profile
drwx----- 2 you you 4096 Oct 16 15:20 .ssh
-rw-r--r-- 1 you you 0 Oct 16 15:38 .sudo_as_admin_successful
drwxrwxr-x 2 you you 4096 Oct 18 10:42 test01
-rw----- 1 you you 882 Oct 17 11:46 .viminfo
you@dbserver:~$ tree
.
└─ test01

1 directory, 0 files
you@dbserver:~$ rmdir test01
you@dbserver:~$ tree
.
0 directories, 0 files
you@dbserver:~$
```

리눅스에서 파일 만드는 명령어
mkdir test01

리눅스에서 파일 제거하는 방법
rmdir test01

- Touch test01.txt test02.txt test03.txt //여러 개 파일 한번에 만들 때 사용
- > test01.txt //파일 만드는데 명령어가 가장 짧다

Touch test01.txt test02.txt test03.txt 여러개 파일생성
 > test01.txt <가장 짧다.
 cat

```
you@dbserver:~/ex$ touch text01.txt
you@dbserver:~/ex$ ls -al
total 20
drwxrwxr-x 5 you you 4096 Oct 18 11:54 .
drwxr-xr-x 5 you you 4096 Oct 18 11:02 ..
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test01
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test02
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test03
-rw-rw-r-- 1 you you 0 Oct 18 11:54 text01.txt
you@dbserver:~/ex$ touch text01.txt text02.txt text03.txt
you@dbserver:~/ex$ ls -al
total 20
drwxrwxr-x 5 you you 4096 Oct 18 11:54 .
drwxr-xr-x 5 you you 4096 Oct 18 11:02 ..
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test01
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test02
drwxrwxr-x 2 you you 4096 Oct 18 11:00 test03
-rw-rw-r-- 1 you you 0 Oct 18 11:54 text01.txt
-rw-rw-r-- 1 you you 0 Oct 18 11:54 text02.txt
-rw-rw-r-- 1 you you 0 Oct 18 11:54 text03.txt
you@dbserver:~/ex$ text04.txt
text04.txt: command not found
you@dbserver:~/ex$ > text04.txt
you@dbserver:~/ex$
```

- Cat > text01.txt //내용 입력하고 파일추가
- Cat >> text01.txt //기존에 있던 내용에 추가
- Cat //cat의 줄임말로, 파일 내용을 보여주는 명령어

Cat의 리다일액션이 가장편하다 추가적인 것이 필요할 때 vim으로 간다

```
you@dbserver:~/ex$ cat text01.txt
you@dbserver:~/ex$ cat > text01.txt
1st
2st
3st
^C
you@dbserver:~/ex$ cat > text01.txt
1st
2st
3st
you@dbserver:~/ex$ cat text01.txt
1st
2st
3st
```

ctrl +c하면 원래 ctrl+d가 저장하고 나
 오는 명령어 인데
 c 도 가능하다 (안될 수도 있다)

```

you@dbserver:~/ex$ cat text01.txt
1st
2st
3st
4st
5st
6st
you@dbserver:~/ex$ cat >> text01.txt
7st
8st
9st^C
you@dbserver:~/ex$ cat text01.txt
1st
2st
3st
4st
5st
6st
7st
8st

```

기존에 있던 내용에 추가하는 것

- Vim text01.txt //새 파일창 열어 편집할 수 있다

```

you@dbserver:~/ex$ vim text01.txt
you@dbserver:~/ex$ sudo vim text01.txt

```

sudo 관리자 명령으로 들어갔다
환경설정 바꾸기 때문에
환경설정바꾸려면 sudo명령어로 들어가야함

지금 일반 저장파일은
sudo 해도되고 안해도됨

```

1st
2st
3st
~
~

```

- Cp *.* test01 //모든파일명 디렉토리 파일 모두 복사한다

```

you@dbserver:~/ex$ copy *.* test01

```

Command 'copy' not found, did you mean:

- command 'cozy' from snap cozy (1.2.1)
- command 'opy' from snap opy (latest)
- command 'copay' from snap copay (12.5.6)
- command 'hcopy' from deb hfsutils (3.2.6-14)
- command 'mcopy' from deb mtools (4.0.24-1)
- command 'copyq' from deb copyq (3.10.0-1)
- command 'fcopy' from deb fai-client (5.3.6ubuntu1)
- command 'bcopy' from deb bacula-sd (9.4.2-2ubuntu5)
- command 'rcopy' from deb rdmacm-utils (28.0-1ubuntu1)
- command 'rope' from deb libdisorder-tools (0.0.2+git20130809.8062ee1-1)

See 'snap info <snapname>' for additional versions.

```

you@dbserver:~/ex$ cp *.* test01
you@dbserver:~/ex$

```

copy는 dos 명령어라서 작동안될
cp가 리눅스에서 복사 명령어다!!

패키지 설치, 권한 설정할 때 -> 최상위 명령어 sudo를 사용해야 함

파일이나 간단한 작업은 sudo 사용안해도 됨

Dos의 copy 대신 리눅스에서 cat 사용할 것이다

- Ps //프로세스
- Ps -e //내가 작업하는 모든 프로세스 나옴

```
papa@dbserver:~$ ps
  PID TTY          TIME CMD
 1443 pts/0        00:00:00 bash
 3722 pts/0        00:00:00 ps
papa@dbserver:~$ ps -e
  PID TTY          TIME CMD
   1 ?            00:00:01 systemd
   2 ?            00:00:00 kthreadd
   3 ?            00:00:00 rcu_gp
   4 ?            00:00:00 rcu_par_gp
   6 ?            00:00:00 kworker/0:0H-kblockd
   8 ?            00:00:00 mm_percpu_wq
   9 ?            00:00:00 ksoftirqd/0
  10 ?            00:00:03 rcu_sched
  11 ?            00:00:00 migration/0
  12 ?            00:00:00 idle_inject/0
  14 ?            00:00:00 cpuhp/0
  15 ?            00:00:00 cpuhp/1
  16 ?            00:00:00 idle_inject/1
  17 ?            00:00:00 migration/1
  18 ?            00:00:00 ksoftirqd/1
  20 ?            00:00:00 kworker/1:0H-kblockd
  21 ?            00:00:00 kdevtmpfs
  22 ?            00:00:00 netns
  23 ?            00:00:00 rcu_tasks_kthre
  24 ?            00:00:00 kauditd
  26 ?            00:00:00 khungtaskd
  27 ?            00:00:00 oom_reaper
  28 ?            00:00:00 writeback
  29 ?            00:00:00 kcompactd0
  30 ?            00:00:00 ksm
```

Bash는 다시 생길 수 밖에 없다.

Kill 해도 다시 나온다

PID를 죽일 수 없어서 컴퓨터를 끄고 다시 켤 수밖에 없다.

- 오류 생기면 ps로 PID 강제로 죽이고 다시 작업하기

```
you@dbserver:~$ ps
  PID TTY          TIME CMD
 3305 pts/1        00:00:00 bash
 3316 pts/1        00:00:00 ps
you@dbserver:~$ kill 3305
you@dbserver:~$ ps
  PID TTY          TIME CMD
 3305 pts/1        00:00:00 bash
 3317 pts/1        00:00:00 ps
you@dbserver:~$ kill 3317
-bash: kill: (3317) - No such process
you@dbserver:~$ ps
  PID TTY          TIME CMD
 3305 pts/1        00:00:00 bash
 3318 pts/1        00:00:00 ps
you@dbserver:~$
```

PID : proccer id로 작업하고 있는 아이디이다
원래 죽일 수 있는 아이가 아니다

bash를 kill 하면 kill되지 않는다

ps에서 찾아서 종료하면 종료한다
윈도우 운영체제보다 리눅스가 직관적이고 편하다

<명령 프롬프트 - Dos 명령어>

- md test01 //test01이라는 파일 만든다는 것
- rd test01 //test01이라는 파일 지운다는 것
- dir //디렉터리 보여주는 명령어

```
명령 프롬프트
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\you>cd\
도스 상에서 명령어 md (파일 만든다는 것)

C:\>md test01

C:\>dir /p
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4CD7-4908

C:\>디렉터리

2023-10-16 오전 11:35 <DIR> database
2023-10-06 오후 05:08 <DIR> ESD
2023-10-06 오후 03:37 <DIR> git-work
2023-10-06 오후 03:10 <DIR> Intel
2023-10-16 오후 02:17 <DIR> park
2023-10-07 오후 06:14 <DIR> PerfLogs
2023-10-16 오후 03:54 <DIR> Program Files
2023-10-16 오전 11:32 <DIR> Program Files (x86)
2023-10-18 오전 10:41 <DIR> test01
2023-10-06 오후 03:14 <DIR> Users
2023-10-17 오후 04:45 <DIR> Windows
2023-10-16 오전 11:43 <DIR> work
0개 파일 0 바이트
12개 디렉터리 63,064,727,552 바이트 남음
```

```
C:\>cd test01

C:\test01>tree
폴더 PATH의 목록입니다.
볼륨 일련 번호는 4CD7-4908입니다.
C:
    내 디렉토리 안에서 지우려고 하면 안되됨
에 하위 폴더가 없습니다. 내가 있는 곳 상위 폴더도 안되됨

C:\test01>rd test01
하위만 가능

cmd에서 파일만드는 것 md test01
파일 제거하는 것 rd test01
```

```
C:\test01>rd test01
지정된 파일을 찾을 수 없습니다.

C:\test01>cd ..

C:\>cd \
c:\> 안해도 되됨

C:\>rd c:\test01
어디에 있느냐에 따라 상대경로를 써야하는지
절대경로를 써야하는지 정해짐
```

- rd test01 /s //파일이 빈 파일이 아니라도 모두 지우겠다는 뜻이다

```
C:\>rd test01
디렉터리가 비어 있지 않습니다. rd test01 /s
/s 는 파일이 빈파일도 아니라도 모두 지우겠다는 뜻임

C:\>rd test01 /s
test01, 계속하시겠습니까(Y/N)? y
```


- rd /? //rd에 대한 자세한 정보를 보여달라는 명령, 위에 rd /s 가 무엇인지 알 수 있음

```
C:\Wex>rd /?
디렉터리를 지웁니다.

RMDIR [/S] [/Q] [드라이브:]경로
RD [/S] [/Q] [드라이브:]경로

/S      지정된 디렉터리 자체와, 그 안의 모든 디렉터리 및 파일을 지웁니다.
        디렉터리 트리를 지우는데 사용합니다.

/Q      조용한 모드로, /S로 디렉터리 트리를 지우는데 문제가 없으면 다시
        묻지 않습니다.
```

- copy 원본 대상
- copy con test01.txt //test01.txt 파일을 만든다 안에 내용 적을 수 있음

con : 콘솔 지금 층 에서는 키보드만 가능하다, 마우스 안됨

=>Dos 상에서 파일 만드는 첫번째 단계, 위에 입력하면 빈줄이 나옴 콘솔에서
입력한 것이 없으니까, 이후 ctrl + c 누르면 -> 종료가 된다

```
C:\Wex>copy con test01.txt
1st
2st
1개 파일이 복사되었습니다.

C:\Wex>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4CD7-4903

C:\Wex 디렉터리
2023-10-18 오전 11:03 <DIR> .
2023-10-18 오전 11:03 <DIR> ..
2023-10-18 오전 10:51 <DIR> test01
2023-10-18 오전 11:03 10 test01.txt
2023-10-18 오전 10:51 <DIR> test02
2023-10-18 오전 10:52 <DIR> test03
1개 파일 10 바이트
5개 디렉터리 63,075,663,872 바이트 남음
```

- copy test01.txt test02.txt //test01.txt 파일을 test02.txt 파일로 복사해라

원본이 있으니까 원본을 복사하여 test01를 만들었다

```
C:\Wex>copy test01.txt
같은 파일로 복사할 수 없습니다.
0개 파일이 복사되었습니다.
대상파일 있을때는
원본이 있으니까 copy con으로 안만들어도
원본이 있으니까 이름만 지정하면
새로 만들어짐

C:\Wex>copy test01.txt test02.txt
1개 파일이 복사되었습니다.

C:\Wex>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4CD7-4903

C:\Wex 디렉터리
2023-10-18 오전 11:11 <DIR> .
2023-10-18 오전 11:11 <DIR> ..
2023-10-18 오전 10:51 <DIR> test01
2023-10-18 오전 11:03 10 test01.txt
2023-10-18 오전 10:51 <DIR> test02
2023-10-18 오전 11:03 10 test02.txt
2023-10-18 오전 10:52 <DIR> test03
2개 파일 20 바이트
5개 디렉터리 63,078,510,592 바이트 남음
```

- copy test01.txt c:\Wex\test01 //복사할거야 test01.txt 파일을 c:\Wex\test01로(절대경로)
- copy test01.txt test01 //test01.txt 파일을 test01디렉토리로 복사할거야(상대경로)

```
C:\Wex>copy test01.txt c:\Wex\test01
1개 파일이 복사되었습니다.

C:\Wex>copy test01.txt test01
test01\test01.txt을(를) 덮어쓰시겠습니까? (Yes/No/All): y
1개 파일이 복사되었습니다.
```

```
C:\Wex>cd test01

C:\Wex\test01>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4CD7-4903

C:\Wex\test01 디렉터리
2023-10-18 오전 11:20 <DIR>
2023-10-18 오전 11:20 <DIR>
2023-10-18 오전 11:03 10 test01.txt
1개 파일 10 바이트
2개 디렉터리 63,068,237,824 바이트 남음
```

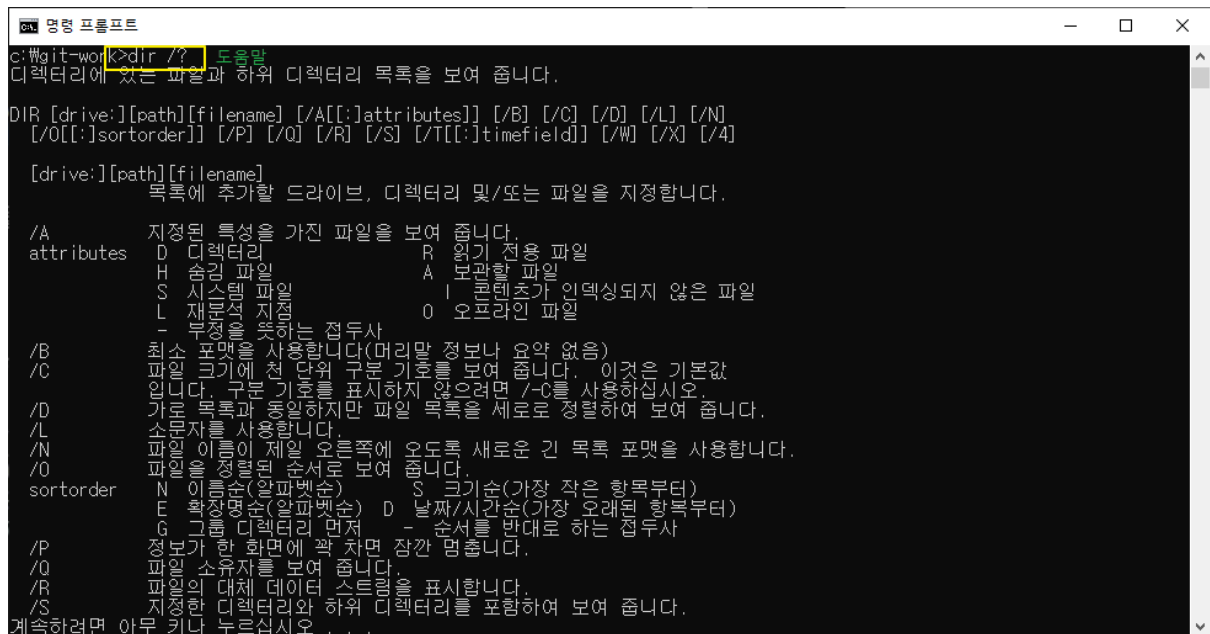
- copy *.txt test01 //ex디렉토리에 있는 모든 파일을 test01디렉토리로 복사함

```
C:\Wex>copy *.txt test01
test01.txt
test01\test01.txt을(를) 덮어쓰시겠습니까? (Yes/No/All): y
test02.txt
test01\test02.txt을(를) 덮어쓰시겠습니까? (Yes/No/All): y
2개 파일이 복사되었습니다.
```

copy *.txt test01
=> ex 디렉토리에 있는 모든 파일을 test01디렉토리로 복사
아래 덮어쓰시겠습니까에 y 하면 하나씩 물어봄
모두 하려면 a 누르기 (all)

- cls //화면에 쓰고 있던 것 지워짐
- * //다. 모두. All
- t* //t로 시작하는 모든 파일명을 말한다
- *t //t로 끝나는 모든 파일명을 말한다
- T? //T로 시작하는 두글자만 가능(T를 포함한다)
- ?T //T로 끝나는 두글자(T나옴 T앞에 아무것도 없을 수 있다)

- dir /? //dir의 도움말



```
c:\git-work>dir /?
DIR [drive:][path][filename] [/A[:attributes]] [/B] [/C] [/D] [/L] [/N]
[/O[:sortorder]] [/P] [/Q] [/R] [/S] [/T[:timefield]] [/W] [/X] [/4]

[drive:][path][filename]
    목록에 추가할 드라이브, 디렉터리 및/또는 파일을 지정합니다.

/A      지정된 특성을 가진 파일을 보여 줍니다.
attributes
    D 디렉터리                      R 읽기 전용 파일
    H 숨김 파일                      A 보관할 파일
    S 시스템 파일                  I 콘텐츠가 인덱싱되지 않은 파일
    L 재부식 지점                  O 오프라인 파일
    - 부정을 뜻하는 접두사

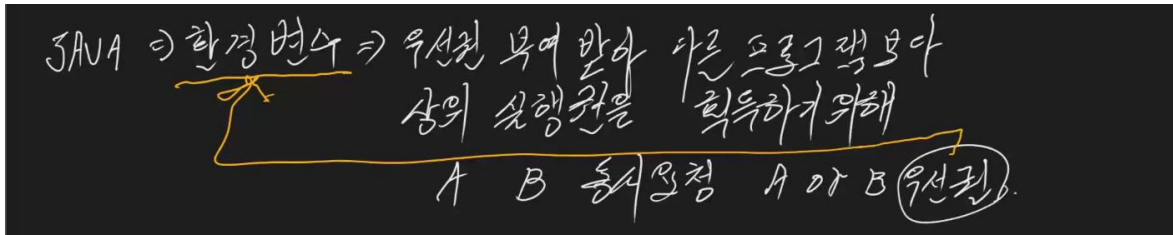
/B      최소 포맷을 사용합니다(머리말 정보나 요약 없음)
/C      파일 크기에 천 단위 구분 기호를 보여 줍니다. 이것은 기본값
        입니다. 구분 기호를 표시하지 않으려면 /-C를 사용하십시오.
/D      가로 목록과 동일하지만 파일 목록을 세로로 정렬하여 보여 줍니다.
/L      소문자를 사용합니다.
/N      파일 이름이 제일 오른쪽에 오도록 새로운 긴 목록 포맷을 사용합니다.
/O      파일을 정렬된 순서로 보여 줍니다.
sortorder
    N 이름순(알파벳순)              S 크기순(가장 작은 항목부터)
    E 확장명순(알파벳순)            D 날짜/시간순(가장 오래된 항목부터)
    G 그룹 디렉터리 먼저          - 순서를 반대로 하는 접두사

/P      정보가 한 화면에 꼭 차면 잠깐 멈춥니다.
/Q      파일 소유자를 보여 줍니다.
/R      파일의 대체 데이터 스트림을 표시합니다.
/S      지정한 디렉터리와 하위 디렉터를 포함하여 보여 줍니다.
        계속하려면 아무 키나 누르십시오 . . .
```

<내용정리>

Java 설치후 => 환경변수 설정했다

- ⇒ 우선권 부여 받아서 다른 프로그램 보다 상위 실행권을 획득하기 위해 A B 동시요청하는 경우가 생긴다
- ⇒ 데드락(맞부딪힐 때) 컴퓨터 누구한테 먼저 작동할 것인가
- ⇒ 우선권 부여할 것 -> 환경변수에서 획득하는 것

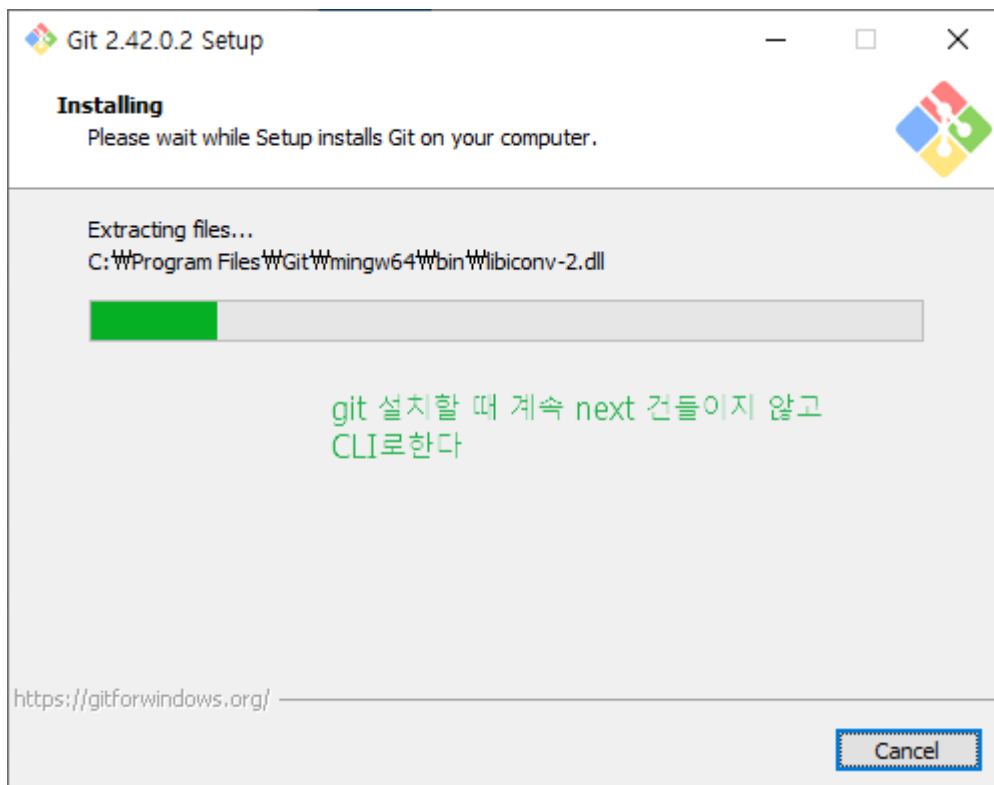
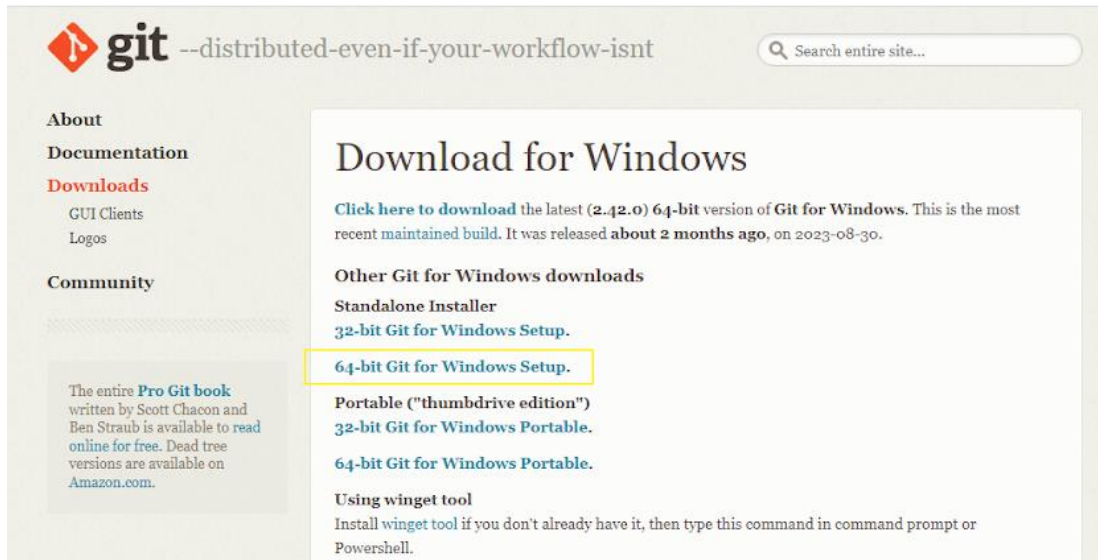


명령 프롬프트는 window에서 바뀐 내용들 바로 받지 못한다

Git 설치하고 안된 것은 바로 반영이 되지 않았기 때문 -> 껴다가 다시 켜니까 작동함

<git 과 git-hub>

Git 설치(64-bit Git for windows setup 으로 설치하기)



```

선택 명령 프롬프트
rm      Remove files from the working tree and from the index

examine the history and state (see also: git help revisions)
bisect  Use binary search to find the commit that introduced a bug
diff    Show changes between commits, commit and working tree, etc
grep    Print lines matching a pattern
log     Show commit logs
show    Show various types of objects
status  Show the working tree status      git의 모든 기본 명령어이다

grow, mark and tweak your common history
branch  List, create, or delete branches
commit  Record changes to the repository
merge   Join two or more development histories together
rebase  Reapply commits on top of another base tip
reset   Reset current HEAD to the specified state
switch  Switch branches
tag     Create, list, delete or verify a tag object signed with GPG

collaborate (see also: git help workflows)
fetch    Download objects and refs from another repository
pull     Fetch from and integrate with another repository or a local branch
push     Update remote refs along with associated objects

'git help -a' and 'git help -g' list available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
See 'git help git' for an overview of the system.

C:\Users\you>

```

- Git config --list //git이 설치되어 있는 최초의 환경설정 내용

```

명령 프롬프트
c:\git-work>git config --list      git이 설치되어 있는 최초의 환경설정 내용이다.
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt 위치
core.autocrlf=true
core.fscache=true      설정파일 들어가는 것이 모두 etc 이다.
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master      master라고 되어 있는 아이가 내가 만들 때
                                master라고 나올 것.

c:\git-work>

디렉토리 내pc에서 gitconfig 파일 열면(검색하면)
master로 해서 나올거다????

error: key does not contain a section: you
c:\git-work>git init
Initialized empty Git repository in C:/git-work/.git/
. .git 이라고 되어 있음 숨김파일

```

Git 이 있는 곳이 내가 저장하는 처음공간이다.

그래서 git-work안에 git이 있는 것보다 각 레파지토리(저장소)에 git이 있어야 한다. 그렇지 않으면 레파지토리를 기억도 하고 그 상위 폴더에서 변경된 사항도 다 저장하고 있는 것이다. 처음부터 다시 해야 한다면 git.폴더를 지우고 다시 생성하면 된다

```

c:\git-work>rd .git /s
.git, 계속하시겠습니까(Y/N)? y

c:\git-work>dir /ah
C: 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 4CD7-4903

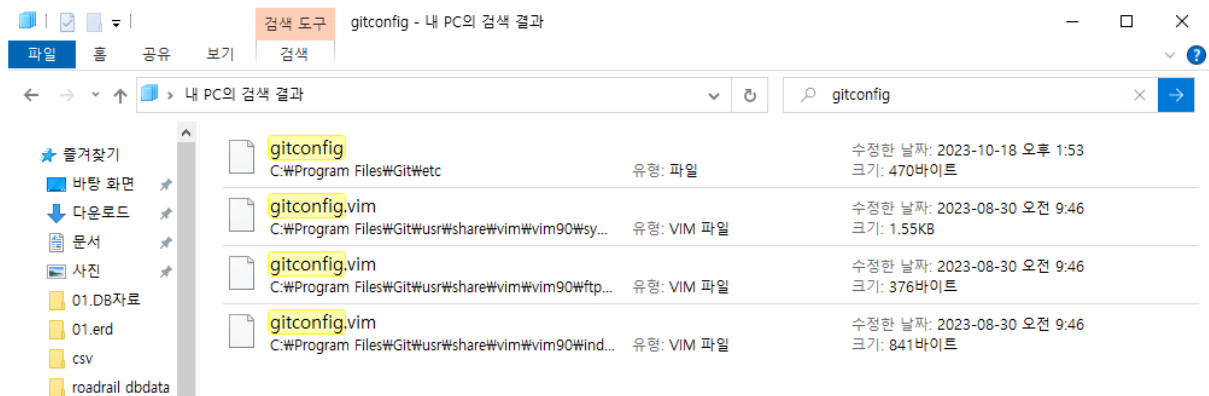
c:\git-work 디렉터리
레파지토리 : 저장소

git init 하고 왜 지우라고 했냐면
루트자리에서 init하면
하위 파일 생기는 순간 업데이트 해야됨

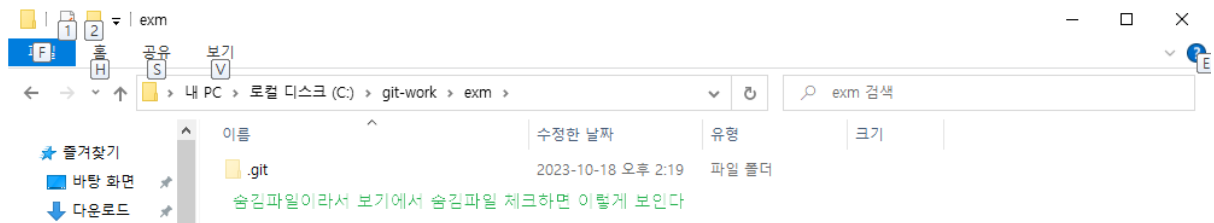
혼란일지 팀 따로 저장하고 싶다
디렉토리 명 만들고 그 밑에서 git을 해야됨

```

Gitconfig를 검색하면 이런 파일이 나오고 intelliJ로 보면 이 파일에 대한 정보가 나온다



Git-work안에 exm을 만들고 git에 대한 명령어 수행했을 때 git이 아래와 같이 만들어짐



window에서 git 설치, 명령프롬포트(cmd)에서 git 찾아냈다(숨김파일)

<cmd로 github에 파일 올리는 방법>

<https://github.com/youjiyou82/TDN.git>

git init // local git 의 시작을 알림, github 레파지토리와 연결되는 곳
(다음에 들어올 때 할필요없다)

git config --global user.name "you" //사용자 이름 확인하기 위해 처음만 하면됨
이름으로 형상관리 시작하겠다

git config --global user.email "youjiyou82@gmail.com"

git add README.md // 한번만 하면 된다

git status // 매번 해줘야 함 상태 알기 위해

git add . // 추가하는 것

git commit -m "first commit" //10-18 (코드가 되면 -01 적어주기) 커밋 message :
왜 이거 작업하는지 이유가 적혀야 함
"10-18-01 중복코드제거"

git branch -M main

git remote add origin <https://github.com/youjiyou82/TDN.git>

// git init 과 같다 깃허브의 주소 최초 연결

(위에는 youjiyou82 의 TDN 에 저장하겠다는 것)

git push -u origin main // 위어가 main 이니까 아래도 main 인 것, Push 하겠다
해당하는 데이터 올릴 것이다

* 깃에 대해서 올리는 것이 끝이다

* 어떠한 형태든 상관 없다 100mb 넘지 않으면 된다

- Git config --global user.name 과 email 설정은 사용자설정으로 처음에 필수로 해야하는 것

```
c:\git-work\wexm>git config --global user.name "you"
c:\git-work\wexm>git config --global user.email "youjiyou82@gmail.com"
error: key does not contain a section: global
c:\git-work\wexm>git config --global user.email "youjiyou82@gmail.com"
c:\git-work\wexm>git config --global user.name
you
c:\git-work\wexm>git config --global user.email
youjiyou82@gmail.com
```

- Git status 를 해주는 것은 상태변화를 확인하는 것인데 이 때 변경된 것이 있으면 변경된 파일이 빨간색으로 표시된다.
- Git add . // .은 모든 것을 추가해주겠다는 것

Git add . 를 해주면 변경된 사항을 추가해주겠다는 것이고, 다시 git status 를 해주면 반영이 완료되었다는 것을 초록색으로 표시해준다

```
c:\git-work\wexm>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  1a.txt

nothing added to commit but untracked files present (use "git add" to track)
c:\git-work\wexm>git add .
c:\git-work\wexm>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   1a.txt
```

이렇게 녹색이 된 상태는
git 저장소에 저장할 준비가 되었다는 뜻

```
(use "git rm --cached <file>..." to unstage)
new file:   1a.txt

c:\git-work\wexm>copy con 2a.txt
2
1개 파일이 복사되었습니다.

c:\git-work\wexm>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   1a.txt

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  2a.txt
```

1a는 git add 해줬기 때문에 초록색이고
2a는 추적되지 않은 파일로 뜬다
변경되었는데 올리지 않을 파일이란 것

새로 생겼어도 add 안해줬기 때문에 녹색으로 안바뀌었다
이내용 git 이 저장하지 않는다

```
c:\git-work\wexm>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
  new file:   1a.txt
  new file:   2a.txt
```

깃에 올린다 = 깃에 저장시킨다

- Git commit -m "10-18" //message 적는 곳 지금은 "10-18"로 적었다

Commit 해주는 순간 git local 에 저장됨

```
c:\git-work\wexm>git commit -m "10-18"
[master (root-commit) af21085] 10-18
2 files changed, 2 insertions(+)
create mode 100644 1a.txt
create mode 100644 2a.txt
```

git과 함께 삭제하지 않는한
계속 기록에 남는다 변경되는 내용마다
저 파일이 오류가 되면 복구가 되는 명령이 존재함

- 파일만들고 commit 까지 한번에 했다

명령 프롬프트

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   1a.txt
    new file:   2a.txt

c:\git-work\wexm>git commit -m "10-18"
[master (root-commit) af21085] 10-18
2 files changed, 2 insertions(+)
create mode 100644 1a.txt
create mode 100644 2a.txt
```

혼자서 해본 부분
파일 3,4를 만들라고 하셨다
1. copy con 3a.txt
2. git status (빨간색으로 나온다)
3. git add .
4. git status (초록색으로 변경된다)
5. git commit -m "10-18" (git에서 사용할 수 있게 커밋하기)

```
c:\git-work\wexm>copy con 3a.txt
3
1개 파일이 복사되었습니다.

c:\git-work\wexm>copy con 4a.txt
4
1개 파일이 복사되었습니다.

c:\git-work\wexm>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    3a.txt
    4a.txt

nothing added to commit but untracked files present (use "git add" to track)

c:\git-work\wexm>git add .

c:\git-work\wexm>git add .

c:\git-work\wexm>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   3a.txt
    new file:   4a.txt

c:\git-work\wexm>git commit -m "10-18"
git: 'commint' is not a git command. See 'git --help'.

The most similar command is
    commit

c:\git-work\wexm>git commit -m "10-18"
[master 0e24133] 10-18
2 files changed, 2 insertions(+)
create mode 100644 3a.txt
create mode 100644 4a.txt

c:\git-work\wexm>git status
On branch master
nothing to commit, working tree clean

c:\git-work\wexm>
```

```
c:\git-work\wexm>git status
On branch master
nothing to commit, working tree clean
```

- rd 폴더명 /s //폴더명입력한 폴더를 비어있지않아도 삭제한다는 명령어

```
c:\Wgit-work\exm>rd network_2022_2학기 /s
network_2022_2학기, 계속하시겠습니까(Y/N)? y
```

- 지금까지 작업한 훈련일지를 직접 git-hub TDN 레파지토리에 올리기
1. Git-work 안에 폴더 만들고 그곳에 업로드할 파일들을 넣어놓는다.
 2. Git 생성(사용자이름,init,status,add . 등)
 3. Commit 메시지 지정
 4. Branch & push 하면 올라가진다

아래가 에러난 것은 TDN 폴더가 아니라 exm 의 폴더로 지정해줬기 때문 !

```
선택 명령 프롬프트
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    23-10-17 class_daily.docx

nothing added to commit but untracked files present (use "git add" to track)
c:\Wgit-work\TDN>git add .
c:\Wgit-work\TDN>git status
On branch master
No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   23-10-17 class_daily.docx

c:\Wgit-work\TDN>git commit -m "TDN 10-18"
[master (root-commit) 0765134] TDN 10-18
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 23-10-17 class_daily.docx

c:\Wgit-work\TDN>git status
On branch master
nothing to commit, working tree clean

c:\Wgit-work\TDN>git branch -M main
c:\Wgit-work\TDN>git remote add origin https://github.com/youjiyou82/exm.git
c:\Wgit-work\TDN>git push -u origin main
To https://github.com/youjiyou82/exm.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/youjiyou82/exm.git'
hint: Updates were rejected because the remote contains work that you do not
hint: have locally. This is usually caused by another repository pushing to
hint: the same ref. If you want to integrate the remote changes, use
hint: 'git pull' before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.
```

```
git config global user.email "papawon01@gmail"
```

git init ==> local git 시작 github 레파지토리와 연결되는 곳. 두 번 아니다

```
git add README.md
```

```
git status
git add .
git status

git commit -m "10-18-01 커밋메세지 이유 적기"
```

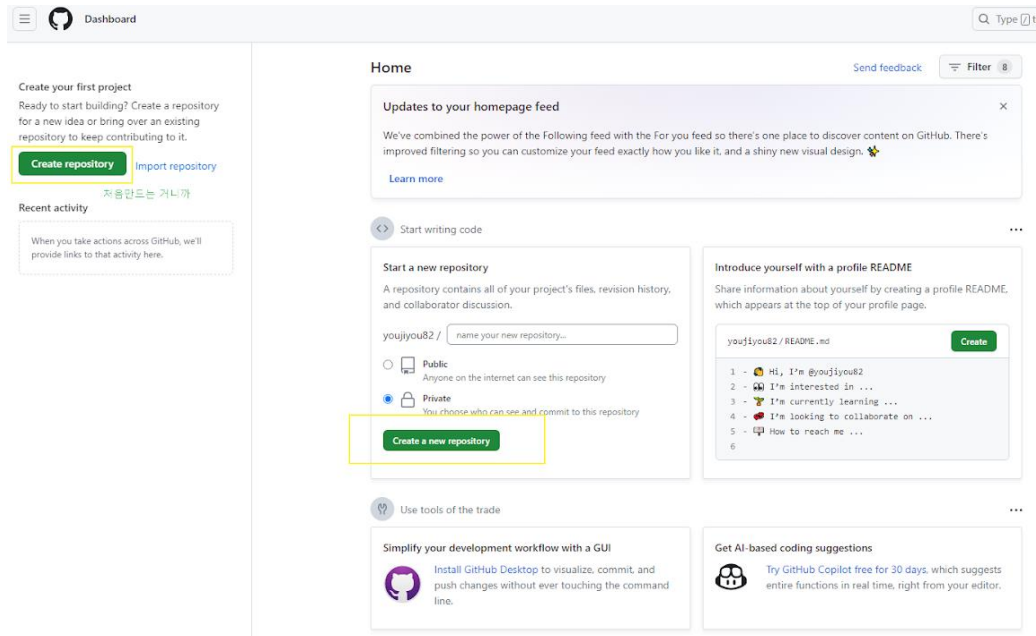
여기 부분 계속 반복됨

```
git branch -M main
git remote add origin https://github.com/papawon/exm.git
git push -u origin main
```

<git-hub>

Git-hub 를 구글아이디로 로그인 하고 처음 레파지토리를 만든다

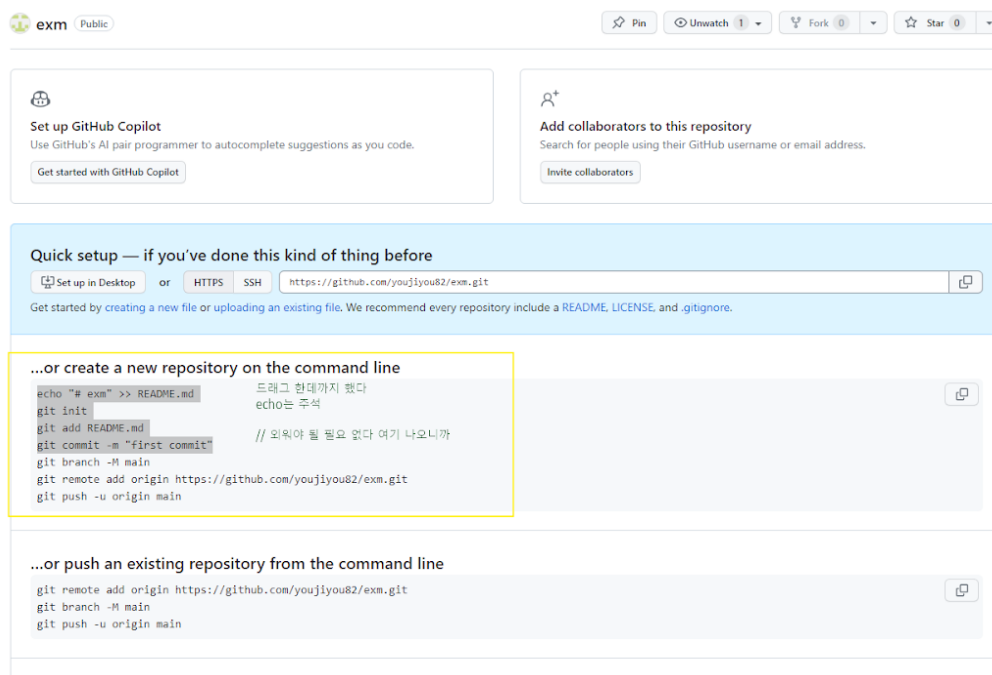
위에 cmd 로 파일들 올리는 방법은 git-hub 에 올리는 방법이다



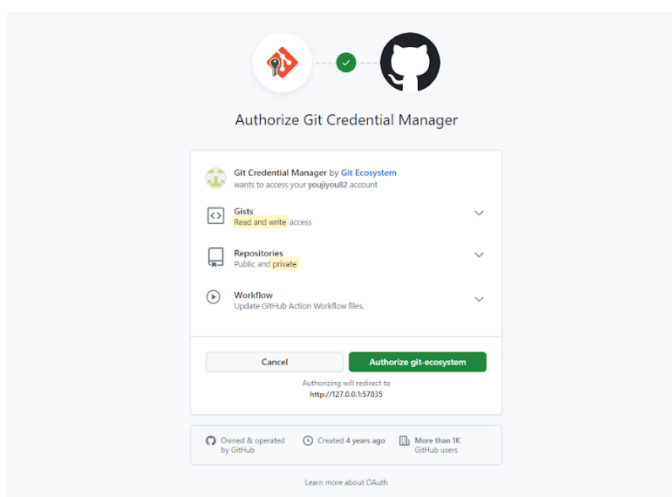
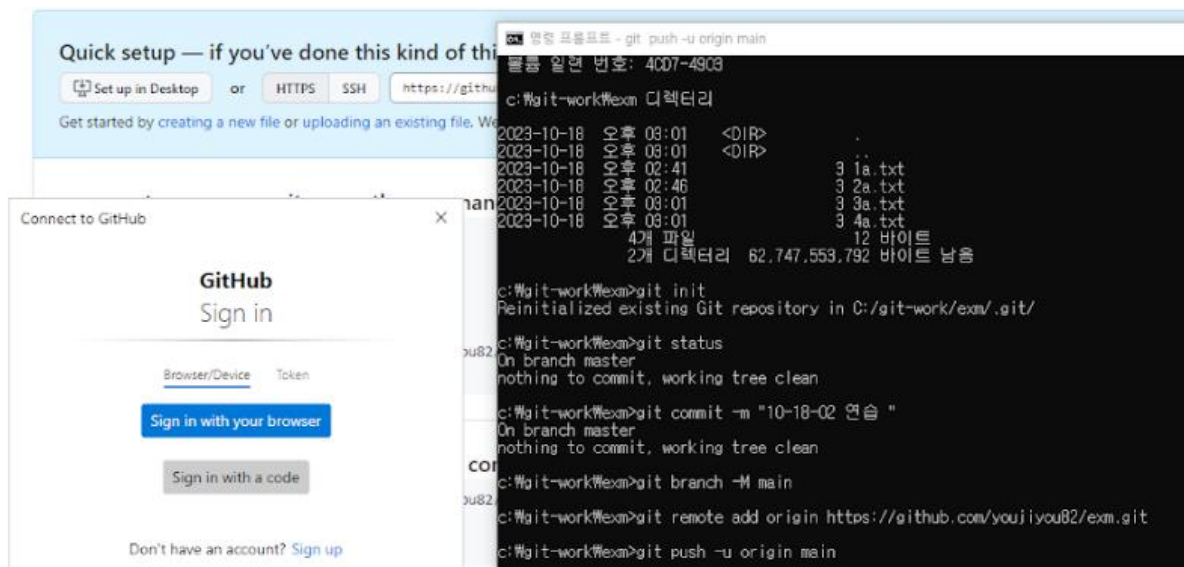
처음 git-hub 의 레파지토리(저장소)를 만들면 아래와 같이 코드들을 참고할 수 있다

아래 드래그 한 부분은 commit 에서 해주면 된다

드래그 안된 부분이 github 로 올리는 코드



드래그 안된 부분들 입력을 하면 아래와 같이 로그인 하는 창이 뜨고, 연결을 하면 commit 한 파일들이 git-hub 에 올라가져 있는 것을 확인 할 수 있다



github 에 올리는 과정 (사용자지정하는 부분 제외한 전체부분)

```
c:\git-work\exam>git init
Reinitialized existing Git repository in C:/git-work/exam/.git/

c:\git-work\exam>git status
On branch master
nothing to commit, working tree clean

c:\git-work\exam>git commit -m "10-18-02 연습 "
On branch master
nothing to commit, working tree clean

c:\git-work\exam>git branch -M main

c:\git-work\exam>git remote add origin https://github.com/youjiyou82/exm.git

c:\git-work\exam>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 474 bytes | 474.00 KiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/youjiyou82/exm.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```