

# Git Bash와 친해져보기

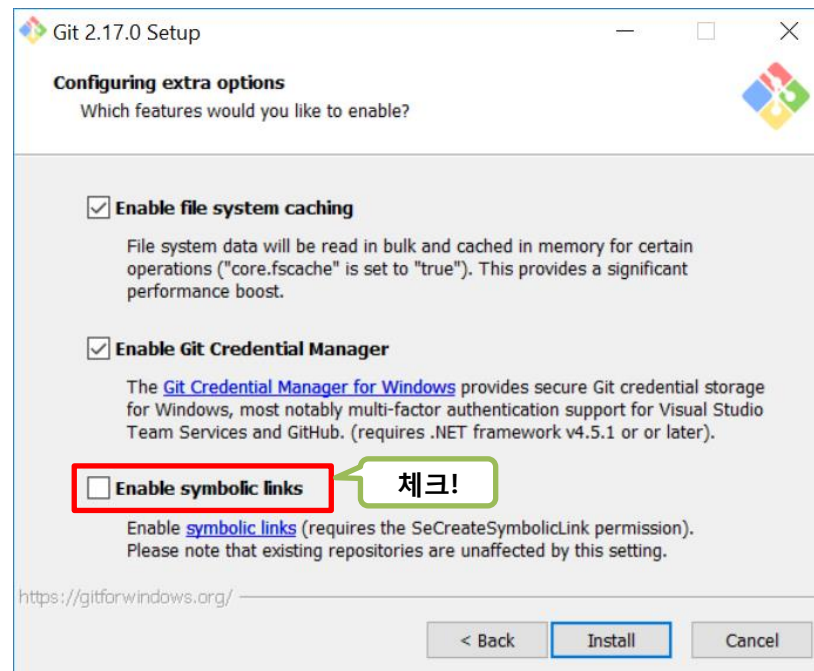
우리 친해질수 있을까?

# 따라해볼것들

- git-scm 설치
  - <https://git-scm.com/downloads>
  - 계속 next 버튼을 누르다가 (천천히 눌러야 실수가 적음) 마지막에 next 버튼이 finish로 바뀌는 시점이 있음. 이 화면을 주목해야함.

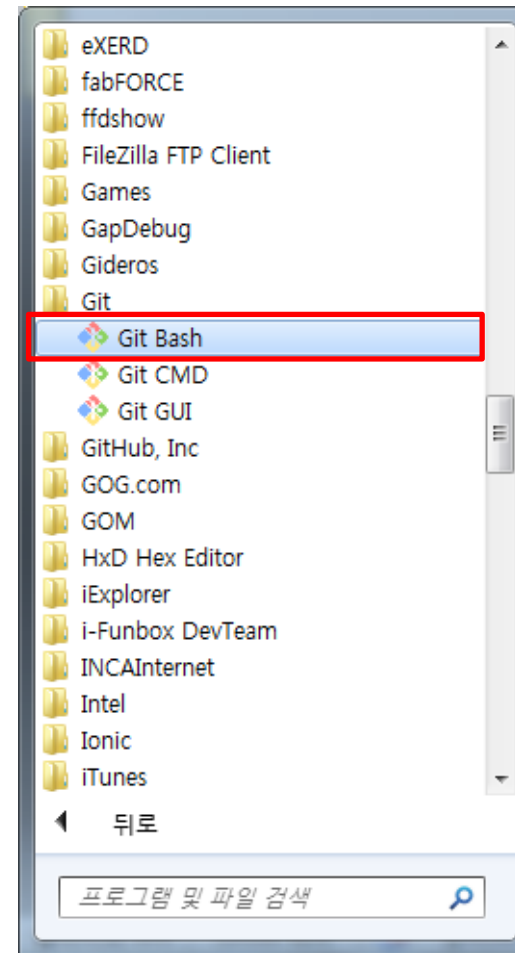
# 따라해볼것들

- Git Bash에서 심볼릭 링크 활성화
  - 'Enable symbolic links' 체크
    - 심볼릭 링크 : 윈도우의 바로가기와 비슷한 개념.



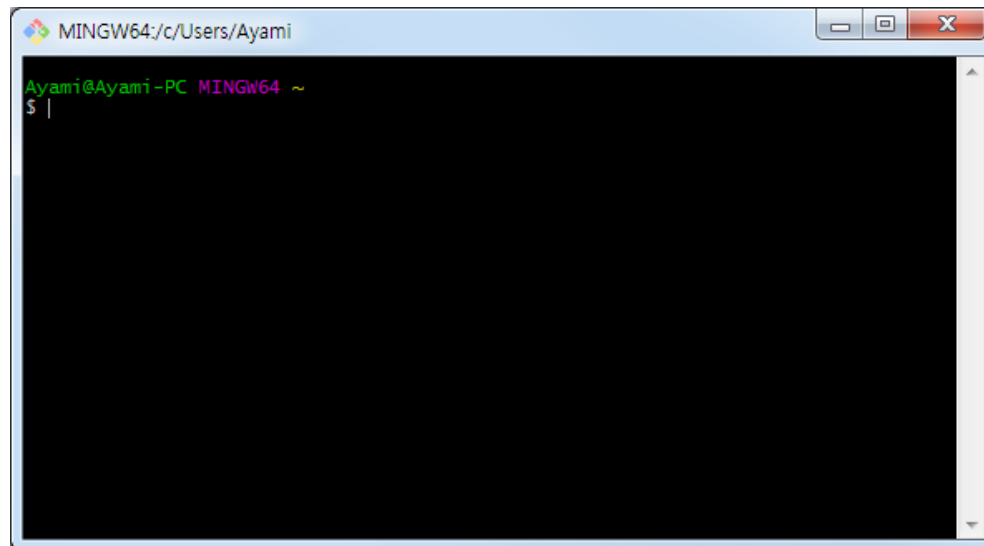
# Git Bash 실행

- Git Bash 실행
  - [시작] – [모든 프로그램 램] – [Git] – [Git Bash]



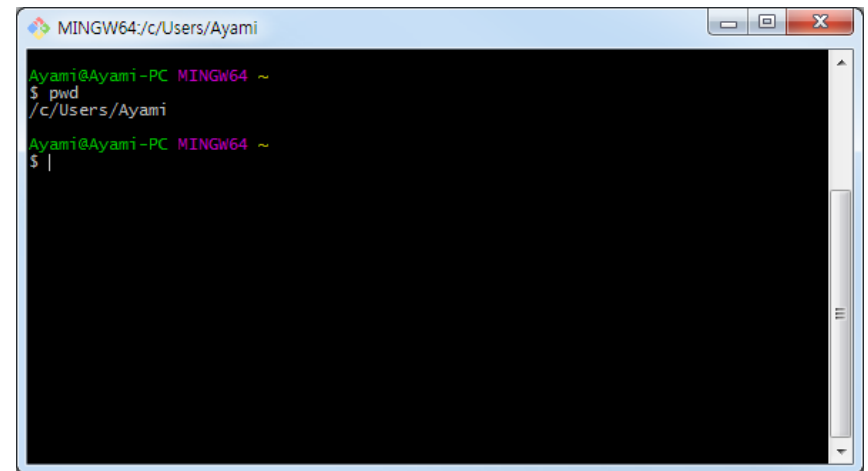
# Git Bash

- 까만것은 종이요 하얀것은 글씨니...
  - 쉘 화면이라고도 부름
  - 리눅스에서나 쓸법한 Bash 쉘을 윈도우에서 사용가능.



# Git Bash - 간단한 리눅스 명령어 따라해보기

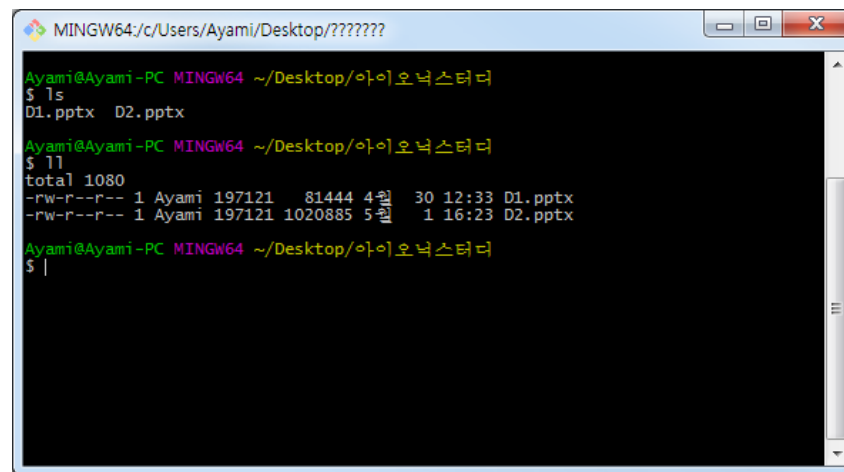
- pwd
  - 현재 작업중인 디렉토리 경로를 출력함.
  - Git Bash를 처음 실행하고 pwd를 입력하면 현재 윈도우에 로그인되어 있는 사용자 디렉토리(홈 디렉토리)의 경로를 출력함.
    - C:\Users\사용자명



```
MINGW64:/c/Users/Ayami
Ayami@Ayami-PC MINGW64 ~
$ pwd
/c/Users/Ayami
Ayami@Ayami-PC MINGW64 ~
$ |
```

# Git Bash - 간단한 리눅스 명령어 따라해보기

- ls
  - 현재 작업중인 디렉토리에 있는 파일과 디렉토리를 목록으로 보여줌.
  - 기본값으로 이름만 나열해줌
- ll
  - ls 명령이 파일과 디렉토리의 이름만 나열해준다면, ll 명령은 이름, 권한, 크기, 생성일등의 상세 정보를 알려줌
    - 사실 ll은 ls -l의 단축명임.
- ls 명령어는 Glob 패턴이란 단짝친구이다.
  - Glob 패턴 : 여러 파일 이름의 집합을 지정하기 위해 와일드카드 문자 (?, \*, [...])가 포함된 패턴.
  - ls \*.txt
    - 확장자가 txt인 파일들만 나열해라
    - test.txt, hello.txt등...
  - ls ?at.\*
    - 파일명 앞 글자가 임의의 문자이면서 모든 파일들을 나열해라
    - cat, Cat, bat, Bat등...



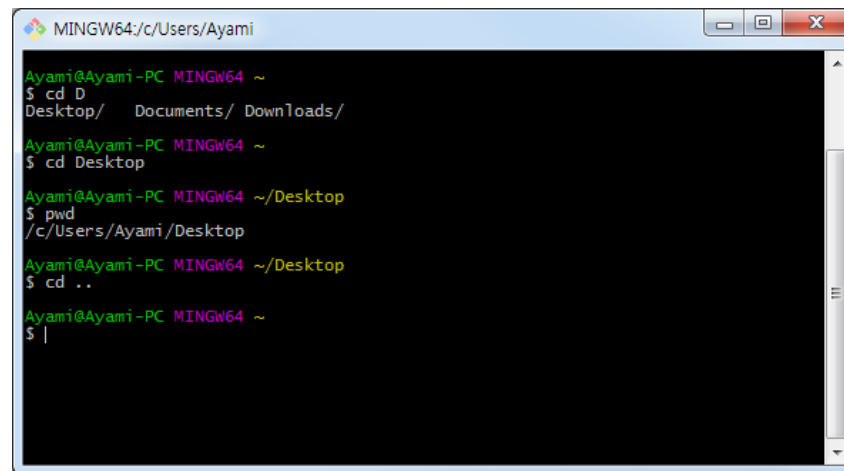
```
MINGW64/c/Users/Ayami/Desktop/?????????
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ls
D1.pptx  D2.pptx

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ll
total 1080
-rw-r--r-- 1 Ayami 197121  81444 4월 30 12:33 D1.pptx
-rw-r--r-- 1 Ayami 197121 1020885 5월 1 16:23 D2.pptx

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ |
```

# Git Bash - 간단한 리눅스 명령어 따라해보기

- cd
  - 다른 경로로 이동함.
  - 경로를 나열하는 방법은 크게 2가지가 있음
    - 상대경로 : 현재 작업중인 디렉토리를 기준으로 하위 경로를 나열하는 방식 (application\views\index.php...)
    - 절대경로 : 루트 드라이브부터 차례대로 경로를 나열하는 방식 (C:\Windows\System32\...)
  - 윈도우에서는 경로를 구분할 때 백슬래시 (or 윈 기호)(\, ₩)를 사용함.
  - 리눅스와 유닉스(or Mac)에서는 슬래시 (/)를 이용함. 리눅스 환경을 흉내내는 Git Bash 또한 경로를 구분할 때 슬래시를 이용함.
  - 어느 디렉토리로 이동해야 하는데 이름이 기억안난다면 앞글자만 입력하고 [Tab] 키를 눌러서 힌트를 확인할 수 있음.
  - cd .. 를 입력하면 현재 작업중인 디렉토리의 '상위 디렉토리' 로 이동함.
    - ex: 현재 작업중인 디렉토리가 /c/Users라면 해당 위치에서 cd ..를 입력했을 때, 작업중인 디렉토리가 /c/로 변경됨.



```
Ayami@Ayami-PC MINGW64 ~
$ cd D
Desktop/ Documents/ Downloads/

Ayami@Ayami-PC MINGW64 ~
$ cd Desktop

Ayami@Ayami-PC MINGW64 ~/Desktop
$ pwd
/c/Users/Ayami/Desktop

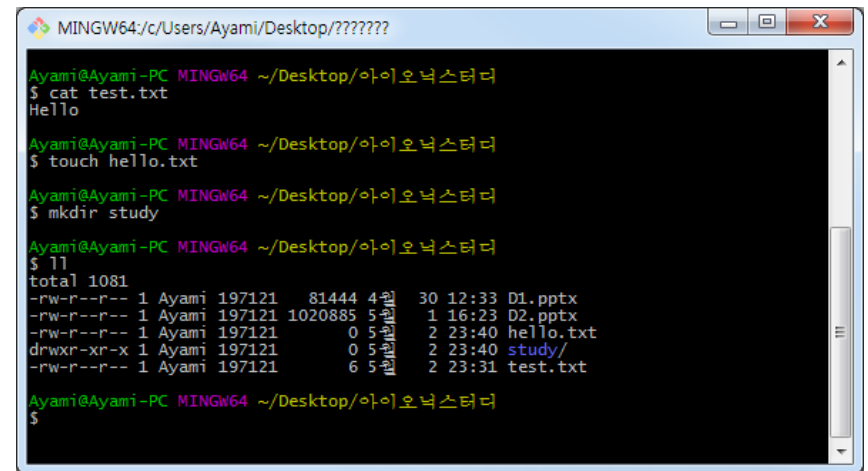
Ayami@Ayami-PC MINGW64 ~/Desktop
$ cd ..

Ayami@Ayami-PC MINGW64 ~
$ |
```



# Git Bash - 간단한 리눅스 명령어 따라해보기

- cat
  - 텍스트 파일 내용을 확인하는 명령어.
- touch
  - 0바이트짜리 빈 파일을 생성해줌
- mkdir
  - 새로운 디렉토리를 생성



```
MINGW64: c:/Users/Ayami/Desktop/???????
```

```
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ cat test.txt
Hello

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ touch hello.txt

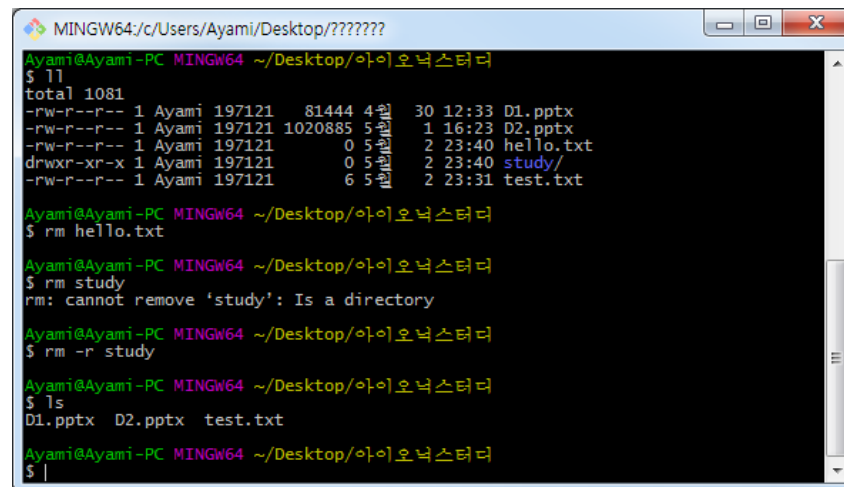
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ mkdir study

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ll
total 1081
-rw-r--r-- 1 Ayami 197121 81444 4월 30 12:33 D1.pptx
-rw-r--r-- 1 Ayami 197121 1020885 5월 1 16:23 D2.pptx
-rw-r--r-- 1 Ayami 197121 0 5월 2 23:40 hello.txt
drwxr-xr-x 1 Ayami 197121 0 5월 2 23:40 study/
-rw-r--r-- 1 Ayami 197121 6 5월 2 23:31 test.txt

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$
```

# Git Bash - 간단한 리눅스 명령어 따라해보기

- rm
  - 파일이나 디렉토리를 삭제하는 명령어.
  - 한번 삭제한 파일은 복구할 수 없으므로 사용에 신중을 기해야함.
  - rm hello.txt
    - hello.txt를 삭제
  - rm study
    - study는 디렉토리라 삭제 불가
    - rm -r study로 삭제해야 함.

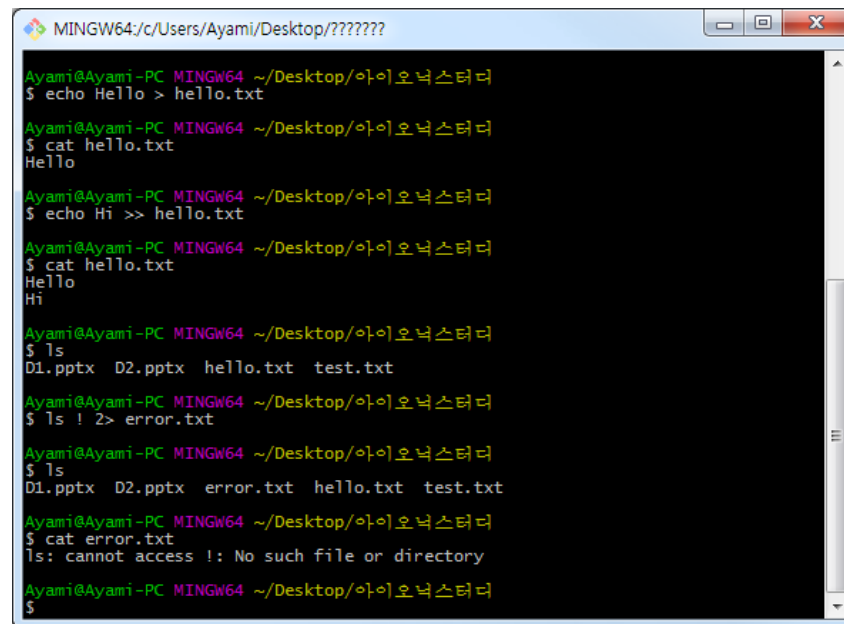


```
MINGW64/c/Users/Ayami/Desktop/???????
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ll
total 1081
-rw-r--r-- 1 Ayami 197121 81444 4월 30 12:33 D1.pptx
-rw-r--r-- 1 Ayami 197121 1020885 5월 1 16:23 D2.pptx
-rw-r--r-- 1 Ayami 197121 0 5월 2 23:40 hello.txt
drwxr-xr-x 1 Ayami 197121 0 5월 2 23:40 study/
-rw-r--r-- 1 Ayami 197121 6 5월 2 23:31 test.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ rm hello.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ rm study
rm: cannot remove 'study': Is a directory
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ rm -r study
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ls
D1.pptx D2.pptx test.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ |
```

# Git Bash - 간단한 리눅스 명령어 따라해보기

- echo

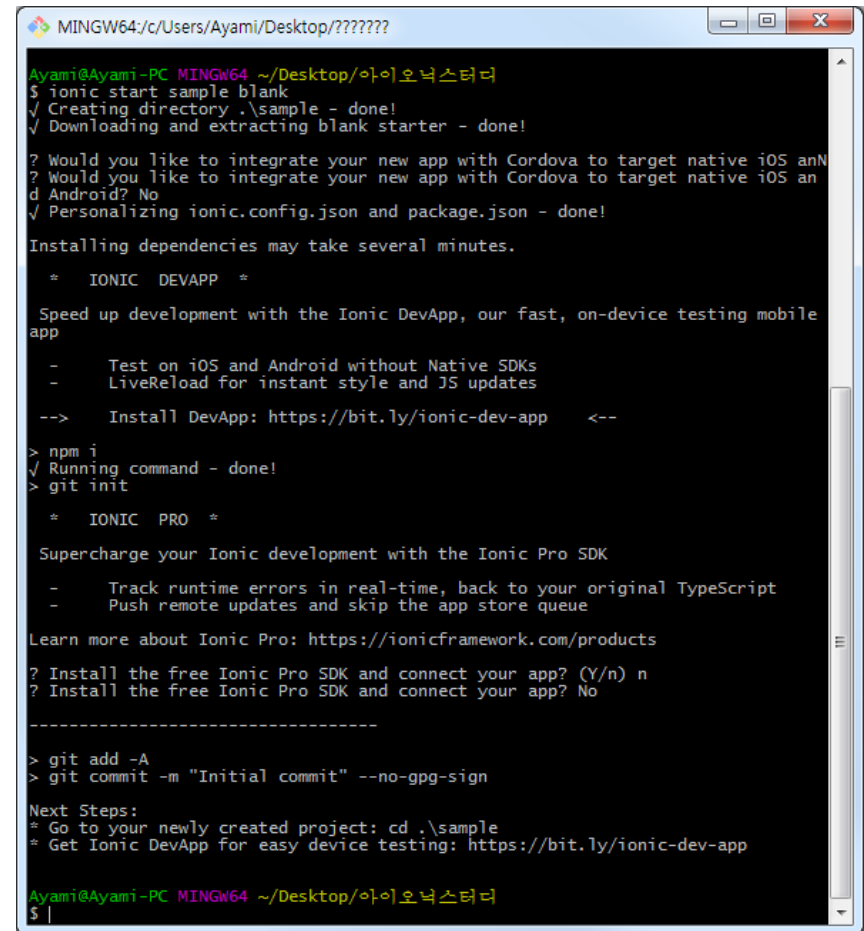
- 메시지를 출력하는 명령.
- 단일로 사용되기도 하지만, 출력 재지정과 함께 사용할 수도 있음.
  - 출력 재지정 : 출력되는 위치를 화면, 파일등 여러 장치 중 하나로 지정하는 것.
- echo Hello > hello.txt
  - Hello라는 내용을 hello.txt로 출력
  - 다른 내용을 동일한 파일로 출력 시도시 덮어쓰워짐
- echo Hi >> hello.txt
  - Hi라는 내용을 hello.txt로 출력하는데 기존에 내용이 존재한다면 해당 내용을 유지한 상태로 출력함.
- ls ! 2> error.txt
  - 2> 앞의 명령이 오류를 발생시킨다면 해당 오류의 내용을 error.txt로 출력.
  - ls ! 라는 명령어는 'No such file or directory' 라는 오류를 발생시키므로, 해당 오류의 내용이 error.txt에 출력됨.



```
MINGW64/c/Users/Ayami/Desktop/???????
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ echo Hello > hello.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ cat hello.txt
Hello
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ echo Hi >> hello.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ cat hello.txt
Hello
Hi
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ls
D1.pptx  D2.pptx  hello.txt  test.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ls ! 2> error.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ls
D1.pptx  D2.pptx  error.txt  hello.txt  test.txt
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ cat error.txt
ls: cannot access !: No such file or directory
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$
```

# Git Bash – 아이오닉 프로젝트 생성해보기

- ionic start sample blank
  - blank 템플릿으로 sample 프로젝트를 생성
  - 인내심을 가지고 기다리다보면 프로젝트가 생성되어있음.
  - 주의할 점
    - Would you like to intergrate... : n 입력 후 엔터 – 당장 스마트폰 하드웨어를 쓸 일이 없으므로...
    - Install the free Ionic Pro SDK... : n 입력 후 엔터 – 아이오닉 프로를 쓸 일이 없으므로... (만약 Y를 누르게 된다면 마무리 단계에서 로그인하라고 나오고 튕김)



```
MINGW64/c/Users/Ayami/Desktop/???????
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ ionic start sample blank
✓ Creating directory .\sample - done!
✓ Downloading and extracting blank starter - done!

? Would you like to integrate your new app with Cordova to target native iOS and Android? No
✓ Personalizing ionic.config.json and package.json - done!

Installing dependencies may take several minutes.

* IONIC  DEVAPP *

Speed up development with the Ionic DevApp, our fast, on-device testing mobile app
- Test on iOS and Android without Native SDKs
- LiveReload for instant style and JS updates

--> Install DevApp: https://bit.ly/ionic-dev-app <--

> npm i
✓ Running command - done!
> git init

* IONIC  PRO *

Supercharge your Ionic development with the Ionic Pro SDK
- Track runtime errors in real-time, back to your original TypeScript
- Push remote updates and skip the app store queue

Learn more about Ionic Pro: https://ionicframework.com/products

? Install the free Ionic Pro SDK and connect your app? (Y/n) n
? Install the free Ionic Pro SDK and connect your app? No

-----

> git add -A
> git commit -m "Initial commit" --no-gpg-sign

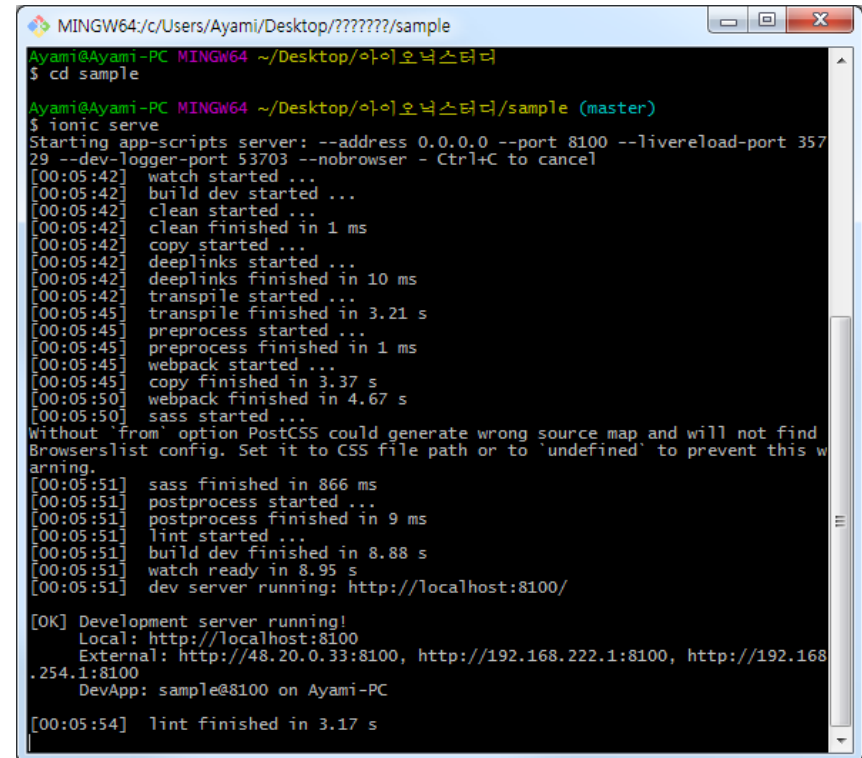
Next Steps:
* Go to your newly created project: cd .\sample
* Get Ionic DevApp for easy device testing: https://bit.ly/ionic-dev-app

Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ |
```

# Git Bash – 아이오닉 프로젝트 생성해보기

- ionic serve

- cd sample
  - sample 디렉토리로 이동
- ionic serve
  - 내장 브라우저로 ionic 프로젝트를 미리 보기 함
- ionic serve 실행 후, 화면이 얼어 있음
  - 사실 포그라운드로 node.js 서버를 돌리는 중임.
- 중단하려면?
  - Ctrl + C로 종료 가능.



```
MINGW64/c/Users/Ayami/Desktop/???????/sample
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디
$ cd sample
Ayami@Ayami-PC MINGW64 ~/Desktop/아이오닉스터디/sample (master)
$ ionic serve
Starting app-scripts server: --address 0.0.0.0 --port 8100 --livereload-port 357
29 --dev-logger-port 53703 --nobrowser - Ctrl+C to cancel
[00:05:42] watch started ...
[00:05:42] build dev started ...
[00:05:42] clean started ...
[00:05:42] clean finished in 1 ms
[00:05:42] copy started ...
[00:05:42] deeplinks started ...
[00:05:42] deeplinks finished in 10 ms
[00:05:42] transpile started ...
[00:05:45] transpile finished in 3.21 s
[00:05:45] preprocess started ...
[00:05:45] preprocess finished in 1 ms
[00:05:45] webpack started ...
[00:05:45] copy finished in 3.37 s
[00:05:50] webpack finished in 4.67 s
[00:05:50] sass started ...
Without 'from' option PostCSS could generate wrong source map and will not find
Browserslist config. Set it to CSS file path or to 'undefined' to prevent this w
arning.
[00:05:51] sass finished in 866 ms
[00:05:51] postprocess started ...
[00:05:51] postprocess finished in 9 ms
[00:05:51] lint started ...
[00:05:51] build dev finished in 8.88 s
[00:05:51] watch ready in 8.95 s
[00:05:51] dev server running: http://localhost:8100/
[OK] Development server running!
Local: http://localhost:8100
External: http://48.20.0.33:8100, http://192.168.222.1:8100, http://192.168
.254.1:8100
DevApp: sample@8100 on Ayami-PC
[00:05:54] lint finished in 3.17 s
```