

1

셸(shell) 명령어 기초



2장 3절 ‘파이썬 프로그램 실행하기’에서 셸(shell)에 관해 간단히 설명하였다. 셸은 커널(kernel)이라는 운영체제의 내부와 사용자 사이의 인터페이스 역할을 한다. 셸은 사용자가 입력한 명령을 운영체제에 전달하고 그 처리 결과를 사용자에게 출력하는 시스템 프로그램이다. 셸은 명령어 인터페이스 또는 명령 줄 인터페이스(Command Line Interface, CLI)라 불리는 명령어 셸과 그래픽 사용자 인터페이스(Graphic User Interface, GUI)로 잘 알려져 있는 그래픽 셸로 구분된다.

명령어 셸은 텍스트 기반 터미널을 통해 사용자와 컴퓨터가 상호작용을 하는 방식이다. 사용자가 컴퓨터 키보드 등을 통해 문자 형태로 입력을 하며 컴퓨터로부터의 출력 또한 문자로 주어진다. 이 부록에서는 명령어 셸이 제공하는 일부 명령어를 설명할 것이다. 명령어 셸의 많은 명령어 중 디렉토리를 만들거나, 파일을 복사하거나, 디렉토리로 이동해서 파이썬 프로그램을 실행하는데 필요한 기초 명령어를 다룬다.

우선 널리 사용되는 윈도우(Windows) 운영체제가 제공하는 ‘명령 프롬프트’에서 파이썬을 실행하거나 파일과 디렉토리(폴더)를 관리하는 도스(DOS) 명령어를 설명한 후, 맥 OS(macOS) 운영체제가 제공하는 ‘터미널’에서의 명령어를 간단히 살펴보도록 한다. 한편 리눅스의 경우에는 맥 OS가 리눅스(Linux) 커널에 기반하기 때문에, 맥의 명령어가 리눅스의 명령어와 유사하다. 따라서 리눅스 사용자는 맥의 내용을 참조하면 된다.

디렉토리, 폴더, 경로



디렉토리(directory)는 운영체제 파일 시스템의 한 부분이다. 디렉토리는 파일(file)을 가질 수 있으며, 다른 디렉토리를 가질 수도 있다. 폴더(folder)와 같은 뜻으로 사용된다.

홈 디렉토리(home directory)란 시스템에 계정을 가진 사용자가 로그인하면 그 사용자에게 할당된 개인 작업 영역의 최상위 디렉토리를 말한다.

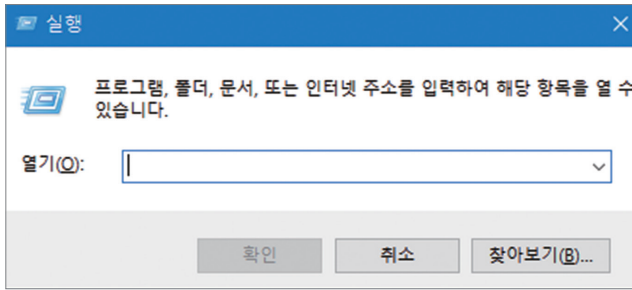
경로(path)는 디렉토리 트리 계층(tree hierarchy)으로 표현되는 파일 시스템에서 특정한 위치에 있는 파일이나 디렉토리를 구분자(/, \, ₩ 등)로 분리하여 문자로 표현한 것이다. 예를 들어, 'C:₩Users₩홍길동'이란 경로는 윈도우 파일 시스템의 C 드라이브에서 'Users' 디렉토리 아래 '홍길동'이란 디렉토리를 파일 계층 형식으로 나타낸 것이다. 맥 OS의 경우에는 앞의 경로가 '/Users/홍길동'으로 표현된다. 맥 OS 파일 시스템은 리눅스 커널 기반이라 C 또는 D 등의 드라이브 개념을 사용하지 않는다.

경로는 절대경로(absolute path)와 상대경로(relative path)로 구분된다. 절대경로란 현재 작업 디렉토리와 상관없이 최상위(루트) 디렉토리부터 시작하는 고유한 경로를 말한다. 예를 들어, 윈도우의 경우는 'C:₩Users₩홍길동₩pyprg₩homework', 맥 OS의 경우는 '/Users/홍길동/pyprg/homework'이 절대경로이다. 상대경로는 현재 작업 디렉토리를 기준으로 상대적인 위치에 있는 경로를 말한다. 예를 들어, 만약 현재 작업 디렉토리가 'Users'의 하위 디렉토리인 '홍길동'이라면 윈도우의 경우는 '.₩pyprg₩homework', 맥 OS의 경우는 './pyprg/homework'이 상대경로다. 파일 시스템에서 '.'은 현재 디렉토리를 뜻하며 '..'은 상위 디렉토리를 의미한다.

윈도우의 명령 프롬프트

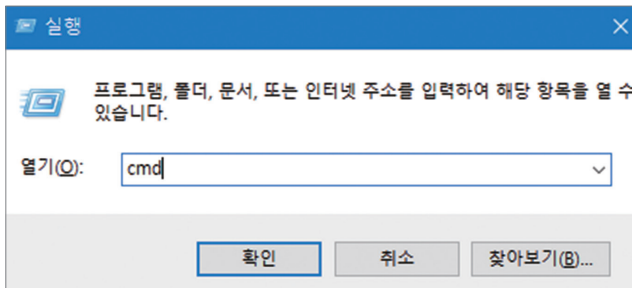
윈도우(Windows) 운영체제에서 명령어 셸은 '명령 프롬프트'다. '명령 프롬프트' 실행 파일 이름은 'cmd.exe'이고 윈도우에서 기본으로 제공되는 프로그램이다. 명령어는 MS-DOS와 호환되기 때문에 도스(DOS)를 사용해본 경험이 있다면 '명령 프롬프트'를 사용하는 데 큰 문제가 없을 것이다.

먼저 '명령 프롬프트' 창을 여는 방법을 알아보자. 여러가지 방법으로 '명령 프롬프트'를 실행할 수 있는데 이 중 가장 간단하면서도 빨리 실행할 수 있는 방법을 설명하도록 하겠다. 키보드의 하단에 위치한 **[윈도우 시작 버튼(■)]**(대부분 스페이스 바 좌측에 위치)을 누른 상태에서 **[R]**을 누르면 다음 그림 1과 같은 '실행' 창이 나타난다.



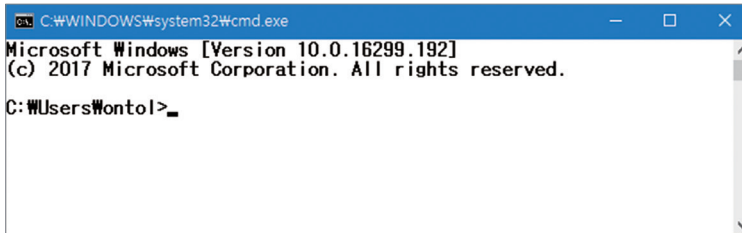
[그림 1] 실행 창

그림 2와 같이 ‘실행’ 창의 [열기] 우측에 ‘cmd’라고 입력을 한 후 [확인] 버튼을 클릭하거나 **Enter** 키를 누르면 ‘명령 프롬프트’ 프로그램이 실행된다.



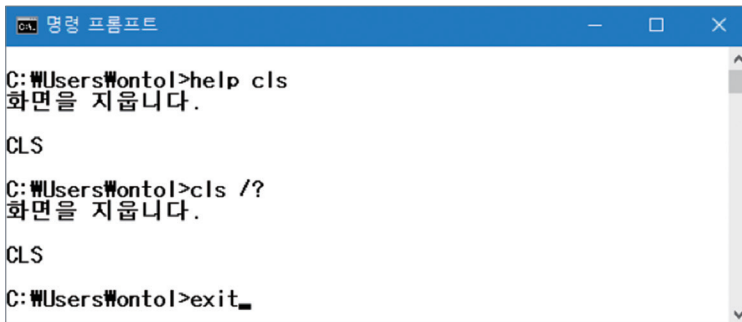
[그림 2] 실행 창에서 ‘명령 프롬프트’ 실행

‘명령 프롬프트’를 실행하면 대부분의 경우 사용자의 홈 디렉토리가 현재 작업 디렉토리로 설정되어 있다. 홈 디렉토리란 시스템에 계정을 가진 사용자가 로그인하면 그 사용자에게 할당되는 개인 작업 영역의 최상위 디렉토리이다. 여기에 파이썬 프로그램을 저장하고 실행하면 된다. 처음 ‘명령 프롬프트’ 창이 열리면 주로 사용자 아이디와 ‘>’ 표시가 함께 프롬프트로 표시된다. 프롬프트(prompt)는 컴퓨터가 사용자의 명령어를 받아들이기 위한 준비가 되어있다는 것을 알리기 위해 화면에 나타나는 표시다. 프롬프트는 운영체제마다 그리고 명령어 셸마다 다르게 표시되는데 보통 윈도우의 ‘명령 프롬프트’에서는 사용자 아이디와 ‘>’로 표시된다. 그림 3을 보면 사용자 아이디가 ‘ontol’인 것을 알 수 있다. 그리고 ‘>’ 표시 우측에 커서가 깜빡거리고 있는데 여기에 명령어를 입력하면 된다. 예를 들어, 프롬프트에서 help를 입력하면 ‘명령 프롬프트’에서 제공하는 명령어 목록을 볼 수 있다.



[그림 3] '명령 프롬프트' 창

만약 특정 명령어에 대한 자세한 내용을 알고 싶다면 그림 4처럼 'help [특정 명령어]' 또는 '[특정 명령어] /?'를 입력하고 **Enter** 키를 누르면 된다. cls는 화면에 나타난 모든 출력을 지우는 명령어고 exit은 '명령 프롬프트'를 종료하는 명령어다.



[그림 4] '명령 프롬프트'에서 셀 명령어 실행

이제부터 사용자 홈 디렉토리 내에서 파이썬 프로그램 파일 저장하고 실행하는 데 필요한 기본적인 명령어들을 살펴보기로 하자. 프롬프트가 사용자마다 다르게 표시되기 때문에 여기서는 그냥 간단히 '>' 기호로 표시하겠다. 내가 현재 어느 디렉토리에 있는지 확인하고 싶으면 cd('change directory'의 줄임말)라는 명령어를 입력하면 된다.

```
cd
```

다음과 같이 cd라는 명령어를 입력하고 **Enter** 키를 누르면 현재 작업 디렉토리가 'C:\Users\Wontol'인 것을 확인할 수 있다.

```
> cd  
C:\Users\Wontol
```

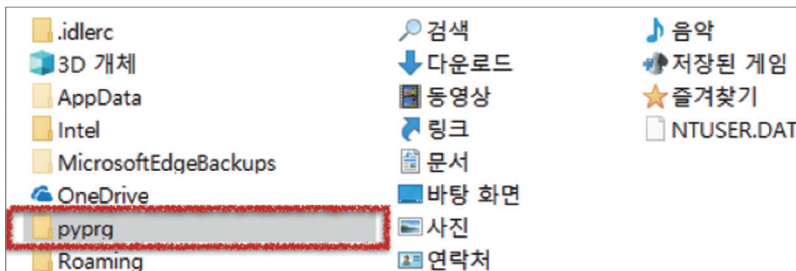
새로운 디렉토리를 만들기 위해서는 다음 명령어를 실행하면 된다.

```
mkdir [디렉토리 이름]
```

예를 들어, 앞으로 작업할 파이썬 프로그램을 저장하기 위해 사용자 홈 디렉토리에서 'pyprg'라는 이름의 디렉토리를 만들기 위해서는 다음과 같이 하면 된다.

```
> mkdir pyprg
```

디렉토리가 만들어졌는지 확인하기 위해 [파일 탐색기]를 열어 사용자 홈 디렉토리(여기서는 'C:\Users\Wontol')를 살펴보면 그림 5와 같이 'pyprg'라는 디렉토리가 만들어진 것을 확인할 수 있다.



[그림 5] 'pyprg' 폴더 생성 결과

다음은 현재 경로에서 하위 디렉토리로 작업 디렉토리를 변경하고 싶을 때 사용하는 명령어를 알아보자. 현재 경로의 하위 경로로 이동하고자 할 때는 다음 명령어를 사용하면 된다.

```
cd [디렉토리 이름]
```

cd는 'change directory'의 약자이며 현재 작업 디렉토리를 다른 디렉토리로 변경할 때 사용하는 명령어다. 동일한 명령어로는 chdir이 있다. 즉, cd와 chdir('change directory')은 같은 명령어다. 따라서 cd 대신 chdir을 사용해도 결과는 같다. 그럼 다음 예처럼 방금 만든 'pyprg' 디렉토리로 경로를 이동하고 제대로 이동했는지 확인해보자.

```
> cd pyprg
> cd
C:\Users\Wontol\pyprg
```

상위 디렉토리로 돌아가고 싶은 경우에는 어떻게 할까? 파일 시스템에는 두 개의 특별한 부호가 있는데, 마침표 2개(..)는 상위 디렉토리(parent directory)을 뜻하며 마침표 1개(.)는 현재 작업 디렉토리를 말한다. 따라서, 상위 디렉토리로 이동하려면 다음과 같이 하면 된다.

```
cd ..
```

만약 현재 경로의 하위 또는 상위 디렉토리가 아닌 다른 디렉토리로 작업 디렉토리를 변경하고 싶다면 cd 다음에 경로를 지정하면 되는데, 이 경우 최상위(루트) 디렉토리부터 시작하는 절대경로(absolute path)나 현재 작업 디렉토리 기준으로 상대적인 위치를 정하는 상대경로(relative path) 중 하나를 사용하면 된다.

```
cd [(절대 또는 상대)경로]
```

예를 들어, 현재 위치인 'C:\Users\Wontol'에서 C 드라이브의 최상위 바로 아래 있는 'Windows' 디렉토리의 하위 디렉토리인 'System'으로 이동한다면 다음 둘 중 하나를 실행하면 된다.

```
> cd C:\Windows\System
> cd ../../Windows/System
```

전자는 절대경로를 사용하였고 후자는 상대경로를 통해 디렉토리를 변경하는 예다. 그리고 다시 사용자 홈 디렉토리로 돌아올 경우 다음과 같이 하면 된다.

```
> cd C:\Users\Wontol
```

그런데 만약 사용자 홈 디렉토리의 경로 이름이 길다면 매번 긴 경로 이름을 입력하는 것이 성가신 일이 될 수도 있다. 이를 때 환경변수를 사용하면 편리하다. 환경변수는 시작과 끝이 %로 되어있는 시스템 환경 변수 이름인데 사용자 홈 디렉토리를 저장하는 환경변수는 %homepath%(대문자 %HOMEPATH%와 동일)이다. 다음과 같은 명령어를 실행하면 현재 작업 디렉토리 위치와 상관없이 손쉽게 나의 사용자 홈 디렉토리로 돌아올 수 있다.

```
cd %homepath%
```

그러면 이제 디렉토리에 들어 있는 내용물 목록을 출력하는 명령어를 알아보자. 디렉토리에 들어있는 파일이나 하위 디렉토리의 목록을 출력하기 위해서는 다음 명령어를 사용한다.

```
dir
```

dir은 'directory'의 약자로 현재 디렉토리에 있는 파일과 하위 디렉토리 목록을 열람할 때 사용하는 명령어다. 다음 예는 'ontol' 사용자의 홈 디렉토리에 저장되어 있는 파일과 디렉토리 이름의 일부를 보여주고 있다.

```
> dir
C 드라이브의 볼륨: ontology
볼륨 일련 번호: 3E0A-201C

C:\Users\Wonto1 디렉터리

2018-01-08 오후 08:51 <DIR> .
2018-01-08 오후 08:51 <DIR> ..
2018-01-08 오후 12:18 <DIR> Contacts
2018-01-08 오후 12:18 <DIR> Desktop
2017-12-11 오후 06:28 <DIR> Documents
2018-01-08 오후 12:18 <DIR> Downloads
2018-01-10 오전 11:47 <DIR> Favorites
```

dir 명령어의 여러 가지 옵션을 알고 싶으면 다음과 같이 명령어를 실행하면 된다.

```
> dir /?
```

만약 디렉토리를 삭제하려면 다음 명령어 중 하나를 선택해서 사용하면 된다.

```
rmdir [디렉토리 이름]
rd [디렉토리 이름]
```

rmdir와 rd는 ‘remove directory’의 약자로 디렉토리를 삭제하는 명령어다. 하지만 만약 디렉토리가 비어 있지 않으면 그 디렉토리를 삭제할 수 없다. 다음 예는 현재 디렉토리 아래에 있는 ‘pyprg’라는 하위 디렉토리를 삭제한다.

```
> rmdir pyprg
```

디렉토리에 있는 파일을 삭제하려면 어떻게 하면 될까? 파일을 삭제하기 위해서는 다음 명령어 중 하나를 실행하면 된다. 둘 다 같은 결과를 가져온다.

```
> del [(경로)파일 이름]
> erase [(경로)파일 이름]
```


del은 'delete'의 약자로 뒤에 지정한 파일을 삭제한다. 만약 삭제하려는 파일이 해당 디렉토리에 존재하지 않으면 파일 삭제에 실패해서 에러 메시지가 출력된다. 다음 예는 현재 작업 디렉토리에 있는 'text.txt' 파일을 삭제하는 것을 보여준다. 이렇게 하면 'text.txt' 파일은 현재 작업 디렉토리에서 지워진다.

```
> del test.txt
```

파일을 복사하려면 다음 명령어를 사용하면 된다.

```
copy [원본 (경로)파일 이름] [대상 (경로)파일 이름]
```

파일을 복사할 때 다른 이름으로도 복사가 가능한데 그렇게 하려면 [대상 파일 이름]을 다른 파일 이름으로 하면 된다. 만약 복사하고자 하는 파일과 같은 파일 이름이 대상 디렉토리에 있을 경우 기존 파일을 덮어쓰게 되니 주의할 필요가 있다.

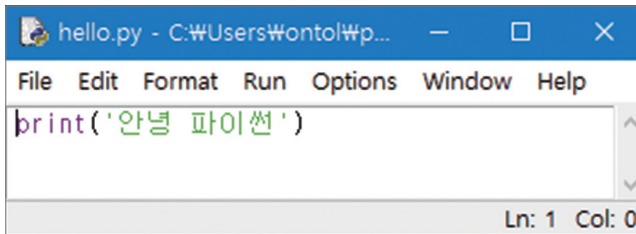
```
> copy test.txt test2.txt  
> copy test.txt C:\Temp\test.txt
```

위의 예 중 첫 번째 줄은 'test.txt'라는 파일을 같은 디렉토리 안에 있는 다른 파일 이름인 'test2.txt'로 복사한 것이고, 두 번째 줄은 C 드라이브의 'Temp' 디렉토리에 같은 이름으로 파일을 복사한 것이다. 만약 복사하려는 원본 파일이 경로에 존재하지 않을 경우 파일을 복사할 수 없기 때문에 에러 메시지를 출력한다.

마지막으로 미리 작성한 파이썬 프로그램을 인터프리터 모드에서 실행하려면 다음 명령어를 사용하면 된다.

```
python [(경로)파일 이름].py
```

예를 들어, 다음 화면과 같은 파이썬 프로그램이 있다고 가정하자. 파일을 직접 실행하고 싶으면 텍스트 편집기인 ‘메모장’이나 ‘IDLE’을 이용해서 다음과 똑같이 코드를 입력하고 ‘hello.py’라는 파일 이름으로 저장해보자.



[그림 6] 윈도우의 IDLE에서 파이썬 코드 작성 예시

이어서 다음과 같이 ‘명령 프롬프트’에서 ‘hello.py’를 실행하면 다음과 같은 결과가 나온다.

```
> python hello.py
안녕 파이썬
```

지금까지 윈도우 운영체제에서 ‘명령 프롬프트’ 프로그램을 통해 이 책에서 필요로 하는 최소한의 기초 셸 명령어를 학습하였다. 파이썬 프로그램을 IDLE이나 다른 통합개발환경(integrated development environment, IDE)에서 실행하지 않고 명령어 셸에서 실행하는 습관을 처음부터 가지는 것은 프로그래밍 개발 환경을 이해할 때 매우 중요한 기초가 될 것이다. 이 부록에서는 명령어 셸에서 제공하는 명령어들을 활용해서 파일을 열람하고 경로를 자유자재로 이동하면서 파이썬 코드를 실행하는 방법을 설명하였다.

여기서 배운 명령어들을 다음 페이지의 표에 정리해 놓았으니 필요할 때 참고하면 된다.

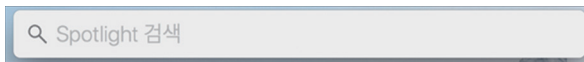
[표] 명령 프롬프트 명령어 목록

명령어	설명	예시
help [명령어] [명령어] /?	[명령어]에 관한 도움말을 제공한다.	help cd cd /?
cls	화면을 지운다.	cls
exit	'명령 프롬프트'를 종료한다.	exit
mkdir [디렉토리 이름]	새 디렉토리를 만든다.	mkdir mydirectory
cd	현재 디렉토리 위치를 보여준다.	cd
cd [경로]	[경로]로 현재 디렉토리를 이동한다.	cd mydirectory cd c:\Windows
cd ..	상위 디렉토리로 이동한다.	cd ..
cd %homepath%	사용자 홈 디렉토리로 복귀한다.	cd %homepath%
dir	디렉토리와 파일 목록을 보여준다.	dir
rmdir [디렉토리 이름] rm [디렉토리 이름]	디렉토리를 지운다.	rmdir mydirectory rm c:\tmp
del [(경로)파일 이름] erase [(경로)파일 이름]	파일을 지운다.	del myfile.txt erase c:\tmp\tmp.txt
copy [원본 (경로)파일 이름] [대상 (경로)파일 이름]	원본 파일을 대상 파일로 복사한다.	copy c:\prg\mypython.py c:\Users\홍길동\pyprg\mypython.py

맥 OS의 터미널

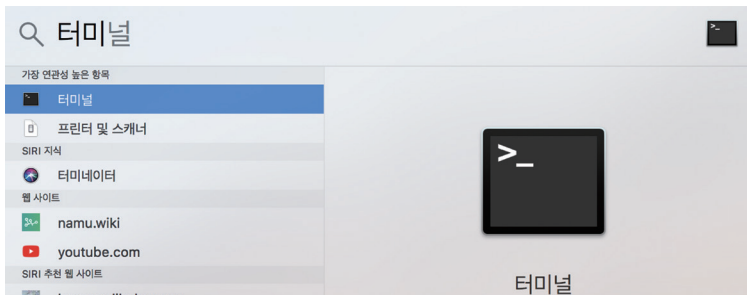
맥 OS(macOS) 운영체제에서 명령어 셸은 ‘터미널’이다. ‘터미널’은 맥 OS에서 기본적으로 제공되는 프로그램이다. 명령어는 리눅스(linux)와 호환되기 때문에 리눅스를 사용해본 경험이 있다면 ‘터미널’을 사용하는 데 큰 문제가 없을 것이다.

먼저 ‘터미널’을 실행하는 방법을 알아보자. 여러 가지 방법으로 ‘터미널’을 실행할 수 있는데 이 중 가장 간단하면서도 빨리 실행할 수 있는 방법은 다음과 같다. 키보드에서 **cmd** 와 **space bar** 또는 **ctrl** 과 **space bar**를 누르면 다음과 같이 **[스포트라이트(Spotlight) 검색]** 창이 뜬다.



[그림 7] 스포트라이트 검색 창

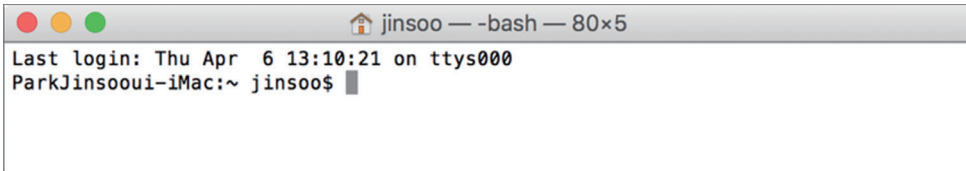
[스포트라이트(Spotlight) 검색] 창에 ‘터미널’이라고 입력한 후 **[터미널]** 항목을 더블 클릭하면 ‘터미널’이 실행된다.



[그림 8] ‘터미널’ 검색 결과

‘터미널’을 실행하면 보통 사용자의 홈 디렉토리가 현재 작업 디렉토리로 설정되어 있다. 홈 디렉토리란 시스템에 계정을 가진 사용자가 로그인하면 그 사용자에게 할당되는 개인 작업 영역의 최상위 디렉토리다. 여기에 파이썬 프로그램을 저장하고 실행하면 된다. 처음 ‘터미널’ 창이 열리면 사용자 아이디와 ‘\$’ 표시가 함께 프롬프트로 표시된다. 프롬프트(prompt)는 컴퓨터가 사용자의 명령어를 받아들이기 위한 준비가 되어 있다는 것을 알리기 위해 화면에 나타나는 표시다. 프롬프트는 운영체제마다 그리고

명령어 셀마다 다르게 표시되는데 맥 OS ‘터미널’의 경우는 주로 컴퓨터 이름과 사용자 아이디, 그리고 ‘\$’로 표시된다. 그림 9의 경우는 사용자 아이디가 ‘jinsoo’인 것을 알 수 있다. 그리고 ‘\$’ 표시 우측에 커서가 깜빡거리고 있는데 여기에 명령어를 입력하면 된다.



[그림 9] ‘터미널’ 창

만약 특정 명령어에 대한 자세한 내용을 알고 싶다면 man 명령어를 사용하면 된다. man 명령어는 특정 명령어에 대한 도움말을 제공하는 명령어로서, man은 ‘online manual page’의 약자다. 프롬프트에서 ‘man [특정 명령어]’를 입력하고 **return** 키를 누르면 특정 명령어에 대한 자세한 내용을 알 수 있다. 예를 들어, 만약 clear라는 명령어를 자세히 알고 싶다면 다음과 같이 입력하고 **return** 키를 누른다.

```
$ man clear
```

그러면 다음 화면과 같이 clear 명령어에 대한 도움말이 화면 전체에 나타난다. 이전 화면으로 돌아가려면 키보드에서 **Q**를 누르면 된다. 참고로 clear는 화면에 나타난 모든 출력을 지우는 명령어이고 exit은 ‘터미널’에서 실행되고 있는 프로세스를 종료한다.

```
jinsoo — less · man clear — 80×22
clear(1) clear(1)
NAME
  clear - clear the terminal screen
SYNOPSIS
  clear
DESCRIPTION
  clear clears your screen if this is possible. It looks in the environ-
  ment for the terminal type and then in the terminfo database to figure
  out how to clear the screen.

  clear ignores any command-line parameters that may be present.
SEE ALSO
  tput(1), terminfo(5)

  This describes ncurses version 5.7 (patch 20081102).
clear(1)
(END)
```

[그림 10] '터미널'에서 셸 명령어 실행 예시

이제부터 사용자 홈 디렉토리 내에서 파이썬 프로그램 파일을 저장하고 실행하는 데 필요한 기본적인 명령어들을 살펴보기로 하자. 프롬프트가 사용자마다 다르게 표시되기 때문에 여기서는 그냥 간단히 '\$' 기호로 표시하겠다. 내가 현재 어느 디렉토리에 있는지 확인하고 싶으면 pwd라는 명령어를 입력하면 된다. pwd는 'print working directory'의 약자이며 현재 작업 중인 디렉토리를 화면에 출력하라는 명령어다.

```
pwd
```

다음과 같이 pwd라는 명령어를 입력하고 **return** 키를 누르면 현재 작업 디렉토리가 '/Users/jinsoo'인 것을 확인할 수 있다.

```
$ pwd
/Users/jinsoo
```

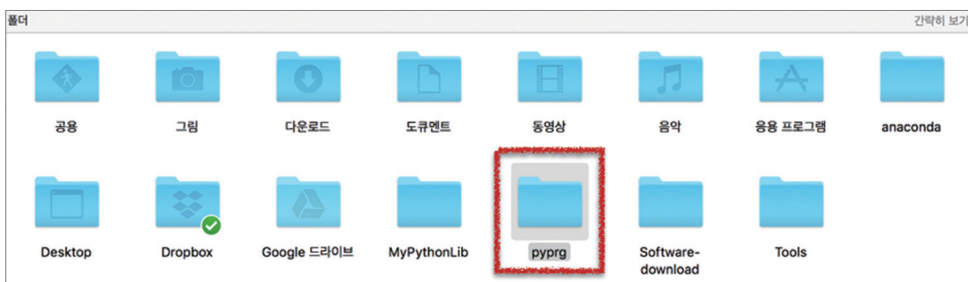
새로운 디렉토리를 만들기 위해서는 다음 명령어를 실행하면 된다.

```
mkdir [디렉토리 이름]
```

예를 들어, 앞으로 작업할 파이썬 프로그램을 저장하기 위해 사용자 홈 디렉토리에서 'pyprg'라는 이름의 디렉토리를 만들기 위해서는 다음과 같이 하면 된다.

```
$ mkdir pyprg
```

디렉토리가 만들어졌는지 확인하기 위해 [Finder]를 열어 사용자 홈 디렉토리(여기서는 '/Users/jinsoo')를 살펴보면 그림 11과 같이 'pyprg'라는 디렉토리가 만들어진 것을 확인할 수 있다.



[그림 11] 'pyprg' 폴더 생성 결과

다음은 현재 경로에서 하위 디렉토리로 작업 디렉토리를 변경하고 싶을 때 사용하는 명령어를 알아보자. 현재 경로의 하위 경로로 이동하고자 할 때는 다음 명령어를 사용하면 된다.

```
cd [디렉토리 이름]
```

cd는 'change directory'의 약자이며 현재 작업 디렉토리를 다른 디렉토리로 변경할 때 사용하는 명령어다. 그럼 다음 예처럼 방금 만든 'pyprg' 디렉토리로 경로를 이동하고 제대로 이동했는지 확인해보자.

```
$ cd pyprg
$ pwd
/Users/jinsoo
```

상위 디렉토리로 돌아가고 싶은 경우에는 어떻게 할까? 파일 시스템에는 두 개의 특별한 부호가 있는데, 마침표 2개(..)는 상위 디렉토리(parent directory)을 뜻하며 마침표 1개(.)는 현재 작업 디렉토리를 말한다. 따라서 상위 디렉토리로 이동하려면 다음과 같이 하면 된다.

```
cd ..
```

만약 현재 경로의 하위 또는 상위 디렉토리가 아닌 다른 디렉토리로 작업 디렉토리를 변경하고 싶다면 cd 다음에 경로를 지정하면 되는데, 이 경우 최상위(루트) 디렉토리부터 시작하는 절대경로(absolute path)나 현재 작업 디렉토리 기준으로 상대적인 위치를 정하는 상대경로(relative path) 중 하나를 사용하면 된다.

```
cd [(절대 또는 상대)경로]
```

예를 들어, 현재 위치인 '/Users/jinsoo'에서 루트(root) 디렉토리 아래 있는 'usr' 디렉토리의 하위 디렉토리인 'local'로 이동한다면 다음 둘 중 하나를 실행하면 된다.

```
$ cd /usr/local  
$ cd ../../usr/local
```

전자는 절대경로를 사용하였고 후자는 상대경로를 통해 디렉토리를 변경하는 예다. 그리고 다시 사용자 홈 디렉토리로 돌아올 경우 다음과 같이 하면 된다.

```
$ cd /Users/jinsoo
```

그런데 만약 사용자 홈 디렉토리의 경로 이름이 길다면 매번 긴 경로 이름을 입력하는 것이 성가신 일이 될 수도 있다. 이런 경우 다음 명령어를 사용하면 편리하다.

```
cd ~
```


이렇게 하면 현재 작업 디렉토리가 어디에 있는 상관없이 손쉽게 나의 사용자 홈 디렉토리로 돌아올 수 있다. 주의할 점은 cd와 ~ 사이에 반드시 공백이 하나 들어가 있어야 한다.

그러면 이제 디렉토리에 들어 있는 내용물 목록을 출력하는 명령어를 알아보자. 디렉토리에 들어있는 파일이나 하위 디렉토리의 목록을 출력하기 위해서는 다음 명령어를 사용한다.

```
ls
```

ls는 'list'의 약자로, 현재 디렉토리에 있는 파일과 하위 디렉토리 목록을 열람할 때 사용하는 명령어다. 다음 예는 'jinsoo' 사용자의 홈 디렉토리에 저장되어 있는 파일과 디렉토리 이름을 보여준다.

```
$ ls
Applications  Documents    Library      Music         Public       pyprg
Desktop        Downloads    Movies        Pictures       Tools
```

ls 명령어의 여러가지 옵션을 알고 싶으면, 다음과 같이 man 명령어를 실행하면 된다.

```
$ man ls
```

만약 디렉토리를 삭제하려면 다음 명령어를 사용한다.

```
rmdir [디렉토리 이름]
```

rmdir은 'remove directory'의 약자로 디렉토리를 삭제하는 명령어다. 하지만 만약 디렉토리가 비어 있지 않으면 그 디렉토리를 삭제할 수 없다. 다음 예는 현재 디렉토리 아래에 있는 'pyprg'라는 하위 디렉토리를 삭제한다.

```
$ rmdir pyprg
```

비어 있지 않은 디렉토리를 삭제하려면 `rm` 명령어에 `-r` 옵션을 사용하면 된다. 하지만 이 경우 하위 디렉토리와 파일 모두를 삭제하기 때문에 주의해서 사용해야 한다.

```
rm -r [디렉토리 이름]
```

디렉토리에 있는 파일을 삭제하려면 어떻게 하면 될까? 파일을 삭제하기 위해서는 다음 명령어를 실행하면 된다.

```
rm [(경로)파일 이름]
```

`rm`은 'remove'의 약자로, 뒤에 지정한 파일을 삭제한다. 만약 삭제하려는 파일이 해당 디렉토리에 존재하지 않는다면 파일 삭제를 할 수 없어서 에러 메시지를 출력한다. 다음 예는 현재 작업 디렉토리에 있는 'text.txt' 파일을 삭제하는 것을 보여준다. 이렇게 하면 'text.txt' 파일은 현재 작업 디렉토리에서 지워진다. `-r` 옵션을 사용할 경우 모든 하위 디렉토리와 이들 디렉토리에 있는 모든 파일도 반복해서(recursive) 모두 삭제하기 때문에 주의해서 사용해야 한다.

```
$ rm test.txt
```

파일을 복사하려면 다음 명령어를 사용하면 된다.

```
cp [원본 (경로)파일 이름] [대상 (경로)파일 이름]
```

`cp`는 'copy'의 약자다. 만약 복사하려는 원본 파일이 경로에 존재하지 않을 경우 에러 메시지를 출력하고 파일을 복사하지 못한다. 파일을 복사할 때 다른 이름으로도 복사가 가능한데 [대상 파일 이름]을 다른 파일 이름으로 하면 된다. 만약 복사하고자 하는 파일과 같은 파일 이름이 대상 디렉토리에 있을 경우 기존 파일을 덮어쓰게 되니 주의할 필요가 있다.

```
$ cp test.txt test2.txt  
$ cp test.txt /tmp/test.txt
```

위의 예 중 첫 번째 줄은 'test.txt'라는 파일을 같은 디렉토리 내에 다른 파일 이름인 'test2.txt'로 복사를 하였고, 두 번째 줄은 루트(root) 디렉토리 아래 있는 'tmp' 디렉토리에 같은 이름으로 파일을 복사하였다.

마지막으로 미리 작성한 파이썬 프로그램을 인터프리터 모드에서 실행하려면 다음 명령어를 사용하면 된다.

```
python [(경로)파일 이름].py
```

예를 들어, 그림 12와 같은 파이썬 프로그램이 있다고 가정하자. 파일을 직접 실행하고 싶으면 '텍스트 편집기'나 'IDLE'을 이용해서 다음 그림 12와 똑같이 코드를 입력하고 'hello.py'라는 파일 이름으로 저장해보자.



[그림 12] 맥 OS의 IDLE에서 파이썬 코드 작성 예시

이어서 '터미널'에서 'hello.py'를 실행하면 다음과 같은 결과가 나온다.

```
$ python hello.py
안녕 파이썬
```

위와 같이 했는데 만약 에러가 나면 'python' 명령어가 파이썬 3이 아니라 파이썬 2의 실행 명령어일 수 있다. 리눅스와 마찬가지로 애플 컴퓨터의 경우 파이썬 2는 이미 설치되어 있는 경우가 대부분이다. 이 경우 'python' 명령어를 입력할 때 실행되는 것은 파이썬 2다. 파이썬의 버전을 확인하려면 다음 명령어를 실행하면 된다.

```
python --version (또는 python -V)
```

만약 다음처럼 결과가 출력되면 파이썬 2 시리즈다.

```
$ python --version
Python 2.7.10
```

이처럼 파이썬 2가 미리 설치되어 있는 애플 컴퓨터의 경우 파이썬 3을 설치하면 실행 파일 이름이 'python3'으로 설치된다. 따라서, 이런 경우에는 다음처럼 'python' 명령어 대신에 'python3' 명령어로 'hello.py'을 실행하면 문제없이 실행될 것이다. 참고로 이 책에서는 'python'을 파이썬 3을 실행하는 명령어로 사용한다.

```
$ python3 hello.py
안녕 파이썬
```

지금까지 맥 OS 운영체제에서 '터미널' 프로그램을 통해 이 책에서 필요로 하는 최소한의 기초 셸 명령어를 학습하였다. 파이썬 프로그램을 'IDLE'이나 다른 통합개발환경(integrated development environment, IDE)에서 실행하지 않고 명령어 셸에서 실행하는 습관을 처음부터 가지는 것은 프로그래밍 개발 환경을 이해할 때 매우 중요한 기초가 될 것이다. 이 부록에서는 명령어 셸에서 제공하는 명령어들을 활용해서 파일을 열람하고 경로를 자유자재로 이동하면서 파이썬 코드를 실행하는 방법을 설명하였다.

여기서 배운 명령어들을 다음 페이지의 표에 정리해 놓았으니 필요할 때 참고하면 된다.

[표] 터미널 명령어 목록

명령어	설명	예시
man [명령어]	[명령어]에 관한 도움말을 제공한다.	help cd cd /?
clear	화면을 지운다.	clear
exit	'터미널' 프로세스를 종료한다.	exit
pwd	현재 디렉토리 위치를 보여준다.	pwd
mkdir [디렉토리 이름]	새 디렉토리를 만든다.	mkdir mydirectory
cd [경로]	[경로]로 현재 디렉토리를 이동한다.	cd mydirectory cd /usr/bin
cd ..	상위 디렉토리로 이동한다.	cd ..
cd ~	사용자 홈 디렉토리로 복귀한다.	cd ~
ls	디렉토리와 파일 목록을 보여준다.	ls
rmdir [디렉토리 이름]	디렉토리를 지운다.	rmdir mydirectory
rm [(경로)파일 이름]	파일을 지운다.	rm /lib/myfile.txt
cp [원본 (경로)파일 이름] [대상 (경로)파일 이름]	원본 파일을 대상 파일로 복사한다.	cp /lib/mypython.py / Users/홍길동/pyprg/ mypython.py