



Data Analytics Based Python

SECT10. 모듈과 패키지 (Module & Package)

Innovation Growth Intensive Training
Kim Jin Soo



- ◆ 모듈 이해하기
- ◆ 모듈 정의 및 불러오기
- ◆ 모듈을 파이썬 환경 밖에서 실행할 때
- ◆ `dir()` 함수를 통한 모듈 속 들여다 보기
- ◆ 모듈들의 집합, 패키지(Packages)



❖ 모듈의 필요성

- 파이썬 셸 상에서 실행된 모든 함수와 변수들은 셸을 종료하는 순간 다시 사용할 수 없다.
- 매번 필요할 때마다 Copy & Paste를 하는 것은 비효율적
- 재사용을 하고자 하는 변수나 함수의 정의문들의 필요성 대두

❖ 모듈, Module

- 파이썬으로 정의한 소스 코드를 담고 있는 파일
- py 확장자로 끝나야 하며 파일명은 곧 모듈명
- 특정 파일이나 셸 환경에서 사용하기 위해서는, 미리 호출하여 해당 모듈을 실행환경에 탑재(import)해야 함

❖ 패키지, Package

- 모듈들을 특정 기준에 따라 모아 놓은 단위



❖ import 한 모듈 파일이 파이썬 실행 도중에 변경이 되었을 때

- 파이썬은 모듈에 대한 효율적인 호출을 위하여 import문 수행 시에만 모듈을 호출한다.
- 모듈을 사용할 때마다 불러 오면 성능이 떨어지기 때문이다.
- 그렇기 때문에, 만약에 import 문 수행 이후에 해당 모듈 파일의 변경이 생긴다면 파이썬 번역기를 다시 시작해야 한다.
- 만약, 현재 상태에서 해당 모듈을 다시 로딩하고 싶다면 `imp.reload()` 함수를 활용
 - 인자값에 해당모듈명을 대입하면 된다.

```
>>> import imp  
>>> imp.reload(module_name)
```

dir() 함수를 통한 모듈 속 들여다 보기



- ❖ 이미 만들어진 모듈의 안을 들여다 보는 방법
- ❖ dir() 함수는 파이썬의 내장 함수로 인자값에 모듈을 집어 넣는 경우 모듈을 구성하고 있는 변수나 함수의 이름들을 리스트로 반환

```
>>> import fibo  
>>> dir(fibo)  
>>> [ '__name__', 'fib', 'fib2' ]
```

모듈들의 집합, 패키지(Packages)



- ❖ 모듈들을 그룹핑 할 수 있는 방법
- ❖ 직접 만든 모듈들을 동일한 폴더 아래에 해당모듈을 모아 놓은 것을 패키지라고 함
 - 예를 들어, 위에서 만든 fibo 라는 함수를 내가 직접 만든 모듈임을 표기하기 위해 hkbda 라는 폴더에 옮겼을 경우

```
>>> from hkbda.fibo import fib
>>> fib(100)
>>> 1 1 2 3 5 8 13 21 34 55 89
```

from문과 import문 사용시 주의 사항



❖ from문과 import문이 함께 사용되는 경우

- import문 뒤에 항목은
 - 패키지의 서브 모듈,
 - 서브 패키지,
 - 패키지 내에 구현 되어 있는 함수나 클래스, 변수 등이 올 수 있다.

❖ from문 없이 import문만 사용하는 경우

- 마지막 항목이 모듈이나 패키지가 되어야 한다.
- 클래스나 함수 혹은 변수가 올 수 없으니 이 또한 주의해야 한다.



- ❖ 모듈의 개념에 대해 이해
- ❖ 직접 모듈을 만들어 보고 호출
- ❖ 파이썬 내부가 아닌 외부 환경(프롬프트 상)에서 직접 호출
- ❖ sys 모듈을 통해 외부 인자를 파이썬 내부로 집어 넣기
- ❖ dir() 함수를 통해 모듈 내부를 들여다 보기
- ❖ 모듈의 상위 개념인 패키지의 개념 및 импорт 방법