

모듈과 패키지

모듈(Module)

- 사전적 의미 : 독자적인 기능을 갖는 구성 요소
- 파이썬에서는 각각의 소스 파일을 일컬음
- 제공자 기준
 - 표준 모듈 : 파이썬과 함께 제공되는 모듈
 - 사용자 생성 모듈 : 프로그래머가 직접 작성한 모듈
 - 서드 파티 모듈 : 다른 프로그래머 or 업체에서 제공한 모듈

모듈(Module) – import에 대해

- 다른 모듈을 현재 모듈로 불러오는 기능
- '다른 모듈 내의 코드에 대한 접근'을 가능
- 접근 가능 코드는 변수/함수/클래스 모두 포함

모듈(Module) – import에 대해

- 사용 방법

- import 모듈명 # 모듈의 실제 파일명은 “모듈.py”
- from 모듈명 import 변수 또는 함수

from 모듈명	from 모듈명 import 변수 또는 함수
<pre>import calculator print(calculator.plus(10, 5)) print(calculator.minus(10, 5)) print(calculator.multiply(10, 5)) print(calculator.divide(10, 5))</pre>	<pre>from calculator import plus from calculator import minus from calculator import multiply from calculator import divide print(plus(10, 5)) print(minus(10, 5)) print(multiply(10, 5)) print(divide(10, 5))</pre>

모듈(Module) 경로 탐색 규칙

- import문을 만나면
 1. 파이썬 인터프리터 내장 모듈 검색
 - `sys.builtin_module_names`
 2. `sys.path`에 정의되어 있는 디렉토리 검색
 - 파이썬 모듈이 실행되고 있는 현재 디렉토리
 - `PYTHONPATH` 환경변수에 정의되어 있는 디렉토리
 - 파이썬과 함께 설치된 기본 라이브러리

메인 모듈(Main Module)

- 별도의 메인 함수가 따로 없음.
- '최상위 수준(Top Level)'에서 실행되는 스크립트만 있을 뿐.
- 최상위 수준 실행 -> 명령 프롬프트 창이나 탐색기를 이용하여 파이썬 모듈을 실행하는 것을 말함.
- 메인 모듈 : 최상위 수준으로 실행되는 스크립트.
- `__name__`
 - 내장 전역 변수
 - 모듈이 최상위 수준으로 실행 될 때 '`__main__`'으로 지정됨.

하위 모듈(Sub Module)

- 메인 모듈이 import 문을 이용하여 불러오는 모듈
- `__name__`
 - 모듈 자체의 이름을 담고 있음.

패키지(Package)

- 모듈을 모아놓는 디렉토리
- 해당 디렉토리가 파이썬의 패키지로 인식되려면 `__init__.py` 파일을 그 경로에 갖고 있어야 함.
- `__init__.py`
 - `__all__` 변수 조정
 - : 패키지로부터 반입할 모듈의 목록을 정의하기 위해 사용

패키지(Package) / site-packages

- 파이썬의 기본 라이브러리 패키지 외에 추가적인 패키지를 설치하는 디렉토리

```
>>>import sys
```

```
>>>sys.path
```

```
['C:\\workspaces\\sources\\python\\chap07\\main_sub2',
```

```
...
```

```
, 'C:\\Anaconda3\\envs\\venvModel\\lib\\site-packages',
```

```
...
```

```
, 'C:\\Anaconda3\\envs\\venvModel\\lib\\site-packages\\Pythonwin']
```

```
>>>from my_package import my_module
```

```
>>>my_module.info()
```