

UNIT 44

모듈과 패키지 사용하기

44 모듈과 패키지 사용하기

» 모듈과 패키지 사용하기

- 모듈(module)은 각종 변수, 함수, 클래스를 담고 있는 파일이고, 패키지(package)는 여러 모듈을 묶은 것임
- 파이썬을 설치할 때 다양한 모듈과 패키지가 기본으로 설치됨
- 기본 모듈과 패키지로 부족하다면 다른 사람이 만든 유명 모듈과 패키지를 설치해서 쓸 수도 있음

44.1 import로 모듈 가져오기

» import로 모듈 가져오기

- 모듈은 import 키워드로 가져올 수 있음(모듈을 여러 개 가져올 때는 모듈을 콤마로 구분)

`import` 모듈

- `import` 모듈1, 모듈2
- 모듈.변수
- 모듈.함수()
- 모듈.클래스()

- 그럼 간단하게 파이썬 표준 라이브러리의 수학 모듈 `math`를 가져와서 원주율을 출력해보자

```
>>> import math
>>> math.pi
3.141592653589793
```

44.1 import로 모듈 가져오기

» import로 모듈 가져오기

- 이번에는 math 모듈의 제곱근 함수 sqrt를 사용해보자

```
>>> import math
>>> math.sqrt(4.0)
2.0
>>> math.sqrt(2.0)
1.4142135623730951
```

44.1 import로 모듈 가져오기

» import as로 모듈 이름 지정하기

- 모듈의 함수를 사용할 때 `math.sqrt`처럼 일일이 `math`를 입력하고 싶지 않을 때는 `import as`를 사용하여 모듈의 이름을 지정할 수 있음

- `import 모듈 as 이름`

- 이제 `math` 모듈을 `m`으로 줄여보자

```
>>> import math as m      # math 모듈을 가져오면서 이름을 m으로 지정
>>> m.sqrt(4.0)           # m으로 제공된 함수 사용
2.0
>>> m.sqrt(2.0)           # m으로 제공된 함수 사용
1.4142135623730951
```

44.1 import로 모듈 가져오기

» from import로 모듈의 일부만 가져오기

- from 모듈 import 변수

```
>>> from math import pi    # math 모듈에서 변수 pi만 가져옴
>>> pi                    # pi를 바로 사용하여 원주율 출력
3.141592653589793
```

44.1 import로 모듈 가져오기

» from import로 모듈의 일부만 가져오기

- from 모듈 import 함수
- from 모듈 import 클래스

```
>>> from math import sqrt      # math 모듈에서 sqrt 함수만 가져옴
>>> sqrt(4.0)                  # sqrt 함수를 바로 사용
2.0
>>> sqrt(2.0)                  # sqrt 함수를 바로 사용
1.4142135623730951
```

44.1 import로 모듈 가져오기

» from import로 모듈의 일부만 가져오기

- math 모듈에서 가져올 변수와 함수가 여러 개일때는 import 뒤에 가져올 변수, 함수, 클래스를 콤마로 구분하여 여러 개를 지정해주면 됨

• `from 모듈 import 변수, 함수, 클래스`

```
>>> from math import pi, sqrt    # math 모듈에서 pi, sqrt를 가져옴
>>> pi                          # pi로 원주율 출력
3.141592653589793
>>> sqrt(4.0)                   # sqrt 함수 사용
2.0
>>> sqrt(2.0)                   # sqrt 함수 사용
1.4142135623730951
```


44.1 import로 모듈 가져오기

» from import로 모듈의 일부만 가져오기

- from import는 모듈의 모든 변수, 함수, 클래스를 가져오는 기능도 있음

• from 모듈 import *

```
>>> from math import *      # math 모듈의 모든 변수, 함수, 클래스를 가져옴
>>> pi                      # pi로 원주율 출력
3.141592653589793
>>> sqrt(4.0)              # sqrt 함수 사용
2.0
>>> sqrt(2.0)              # sqrt 함수 사용
1.4142135623730951
```

44.1 import로 모듈 가져오기

» from import로 모듈의 일부를 가져온 뒤 이름 지정하기

- from 모듈 import 변수 as 이름
- from 모듈 import 함수 as 이름
- from 모듈 import 클래스 as 이름

```
>>> from math import sqrt as s      # math 모듈에서 sqrt 함수를 가져오면서 이름을 s로 지정
>>> s(4.0)                          # s로 sqrt 함수 사용
2.0
>>> s(2.0)                          # s로 sqrt 함수 사용
1.4142135623730951
```

44.1 import로 모듈 가져오기

» from import로 모듈의 일부를 가져온 뒤 이름 지정하기

- 그럼 여러 개를 가져왔을 때 각각 이름을 지정할 때는 각 변수, 함수, 클래스 등을 콤마로 구분하여 as를 여러 개 지정하면 됨

• from 모듈 import 변수 as 이름1, 함수 as 이름2, 클래스 as 이름3

```
>>> from math import pi as p, sqrt as s
>>> p          # p로 원주율 출력
3.141592653589793
>>> s(4.0)     # s로 sqrt 함수 사용
2.0
>>> s(2.0)     # s로 sqrt 함수 사용
1.4142135623730951
```

44.2 import로 패키지 가져오기

» import로 패키지 가져오기

- 패키지는 특정 기능과 관련된 여러 모듈을 묶은 것인데, 패키지에 들어있는 모듈도 import를 사용하여 가져옴

- `import 패키지.모듈`
- `import 패키지.모듈1, 패키지.모듈2`
- `패키지.모듈.변수`
- `패키지.모듈.함수()`
- `패키지.모듈.클래스()`

- 여기서는 파이썬 표준 라이브러리에서 urllib 패키지의 request 모듈을 가져와보자(urllib은 URL 처리에 관련된 모듈을 모아 놓은 패키지임)

```
>>> import urllib.request
>>> response = urllib.request.urlopen('http://www.google.co.kr')
>>> response.status
200
```

- 패키지에 들어있는 모듈은 `import urllib.request`와 같이 `패키지.모듈` 형식으로 가져옴
- 모듈의 함수를 사용할 때도 `urllib.request.urlopen()`과 같이 `패키지.모듈.함수()` 형식으로 패키지 이름과 모듈 이름을 모두 입력해줌

44.2 import로 패키지 가져오기

» import as로 패키지 모듈 이름 지정하기

- `import 패키지.모듈 as 이름`

```
>>> import urllib.request as r    # urllib 패키지의 request 모듈을 가져오면서 이름을 r로 지정
>>> response = r.urlopen('http://www.google.co.kr')    # r로 urlopen 함수 사용
>>> response.status
200
```

44.2 import로 패키지 가져오기

» from import로 패키지의 모듈에서 일부만 가져오기

- from 패키지.모듈 import 변수
- from 패키지.모듈 import 함수
- from 패키지.모듈 import 클래스
- from 패키지.모듈 import 변수, 함수, 클래스

```
>>> from urllib.request import Request, urlopen    # urlopen 함수, Request 클래스를 가져옴
>>> req = Request('http://www.google.co.kr')      # Request 클래스를 사용하여 req 생성
>>> response = urlopen(req)                       # urlopen 함수 사용
>>> response.status
200
```

- 참고로 urlopen 함수에 URL을 바로 넣어도 되고,
Request('http://www.google.co.kr')와 같이 Request 클래스에 URL을 넣은 뒤에
req를 생성해서 urlopen 함수에 넣어도 됨

44.2 import로 패키지 가져오기

» from import로 패키지의 모듈에서 일부만 가져오기

- 패키지의 모듈에서 모든 변수, 함수, 클래스를 가져오는 방법은 다음과 같음

- `from 패키지.모듈 import *`

```
>>> from urllib.request import *      # urllib의 request 모듈에서 모든 변수, 함수, 클래스를 가져옴
>>> req = Request('http://www.google.co.kr')  # Request를 사용하여 req 생성
>>> response = urlopen(req)              # urlopen 함수 사용
>>> response.status
200
```

44.2 import로 패키지 가져오기

» from import로 패키지의 모듈의 일부를 가져온 뒤 이름 지정하기

- `from 패키지.모듈 import 변수 as 이름`
- `from 패키지.모듈 import 변수 as 이름, 함수 as 이름, 클래스 as 이름`

```
>>> from urllib.request import Request as r, urlopen as u
>>> req = r('http://www.google.co.kr')    # r로 Request 클래스 사용
>>> response = u(req)                      # u로 urlopen 함수 사용
>>> response.status
200
```


44.3 파이썬 패키지 인덱스에서 패키지 설치하기

» 파이썬 패키지 인덱스에서 패키지 설치하기

- 파이썬은 파이썬 표준 라이브러리(Python Standard Library, PSL) 이외에도 파이썬 패키지 인덱스(Python Package Index, PyPI)를 통해 다양한 패키지를 사용할 수 있음
- 명령만 입력하면 원하는 패키지를 인터넷에서 다운로드하여 설치해줄 뿐만 아니라 관련된 패키지(의존성)까지 자동으로 설치해주므로 매우 편리함

44.3 파이썬 패키지 인덱스에서 패키지 설치하기

» pip 설치하기

- pip는 파이썬 패키지 인덱스의 패키지 관리 명령어이며 Windows용 파이썬에는 기본으로 내장되어 있음
- 리눅스와 macOS에서는 콘솔(터미널)에서 다음과 같은 방법으로 설치하면 됨

리눅스, macOS

```
$ curl -O https://bootstrap.pypa.io/get-pip.py  
$ sudo python3 get-pip.py
```

44.3 파이썬 패키지 인덱스에서 패키지 설치하기

» pip로 패키지 설치하기

- `pip install 패키지`

- Windows에서는 명령 프롬프트를 실행(윈도우 키+R을 누른 뒤 cmd를 입력)하고, 리눅스와 macOS에서는 콘솔(터미널)을 실행한 뒤 `pip install requests` 명령을 입력함 (pip 명령은 파이썬 셸 >>>에 입력하면 안 됨. 반드시 명령 프롬프트, 콘솔, 터미널에 입력)
- 참고로 requests는 파이썬 표준 라이브러리의 `urllib.request`와 비슷한 역할을 하는 패키지인데 좀 더 기능이 많고 편리함

Windows

```
C:\Users\dojang>pip install requests
```

- 리눅스, macOS에서는 앞에 `sudo`를 붙여서 관리자 권한으로 실행함

리눅스, macOS

```
$ sudo pip install requests
```

44.3 파이썬 패키지 인덱스에서 패키지 설치하기

» pip로 패키지 설치하기

- python에 -m 옵션을 지정해서 pip를 실행할 수도 있음
- m 옵션은 모듈을 실행하는 옵션이며 pip도 모듈임

Windows

```
C:\Users\dojang>python -m pip install requests
```

- 리눅스, macOS에서는 python3으로 실행하고, 앞에 sudo를 붙여서 관리자 권한으로 실행함

리눅스, macOS

```
$ sudo python3 -m pip install requests
```

- 명령을 입력하면 패키지 다운로드 및 설치 상황이 표시되는데, 다음과 같이 출력되면 정상적으로 설치된 것임(패키지는 계속 업데이트되므로 버전은 책과 다를 수 있음)

```
Collecting requests
  Downloading requests-2.9.1-py2.py3-none-any.whl (501kB)
    100% |#####| 503kB 974kB/s
Installing collected packages: requests
Successfully installed requests-2.9.1
```

44.3 파이썬 패키지 인덱스에서 패키지 설치하기

» import로 패키지 가져오기

- import 패키지

```
>>> import requests                                # pip로 설치한 requests 패키지를 가져옴
>>> r = requests.get('http://www.google.co.kr')    # requests.get 함수 사용
>>> r.status_code
200
```

- 보통 pip install 명령으로 설치한 패키지는 import 패키지 또는 import 패키지.모듈 형식으로 사용하면 됨