

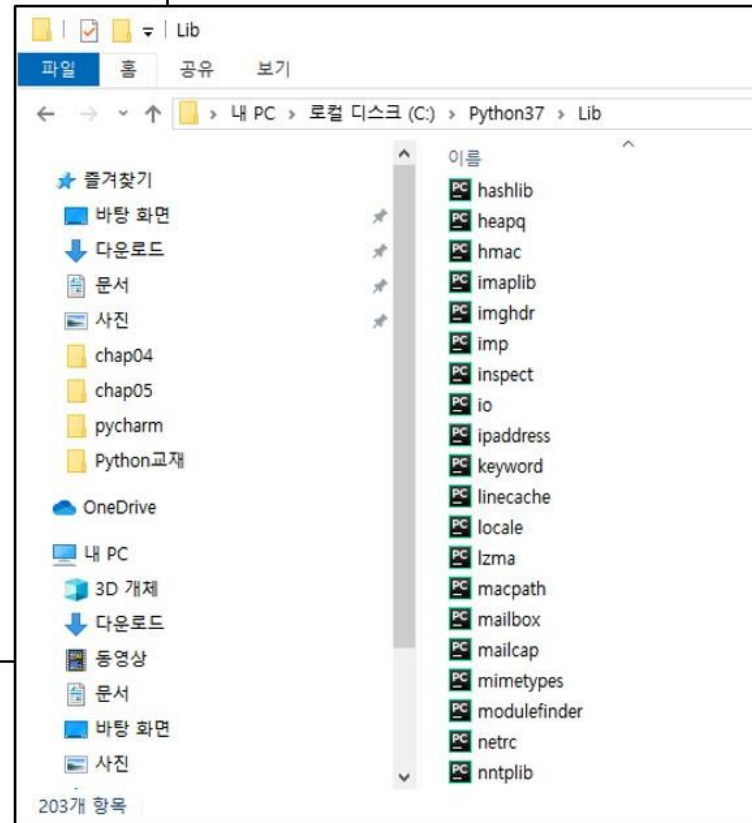
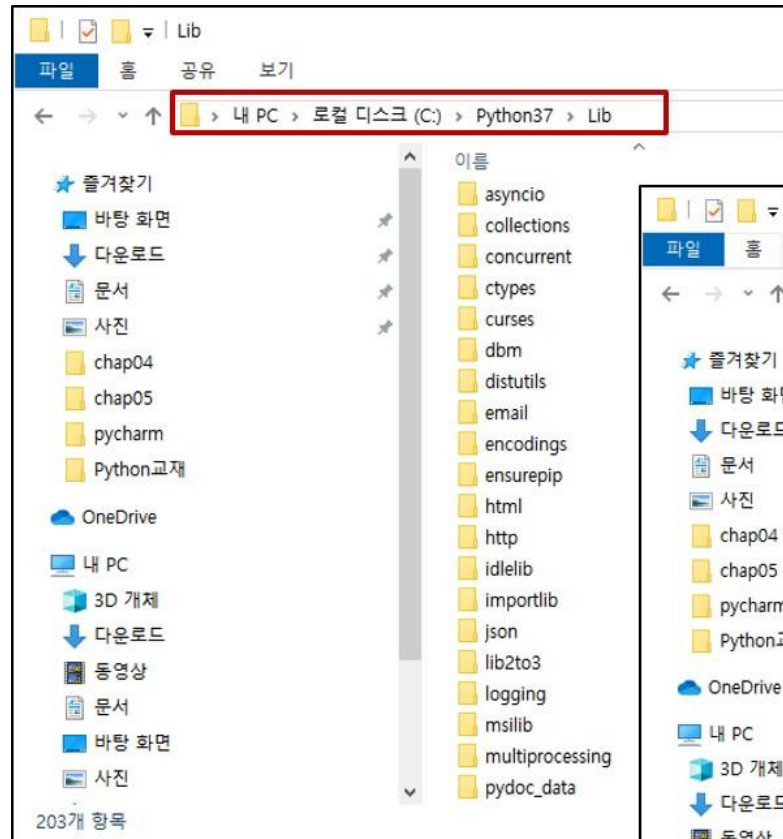
파이썬

15강. 모듈과 패키지 및 함수의 유형

1. 모듈과 패키지

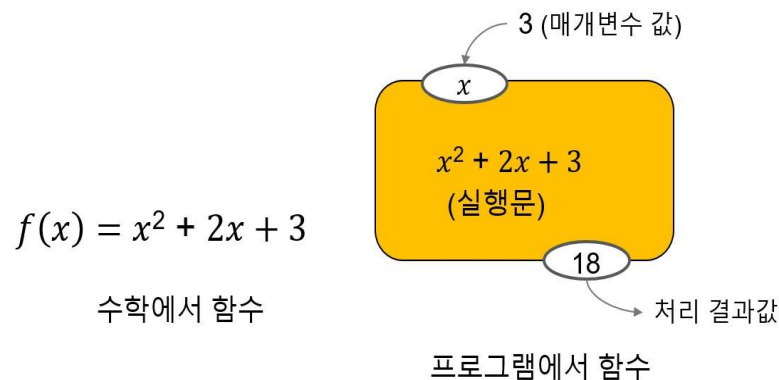
- 파이썬은 모듈(module)과 패키지(package) 형태로 라이브러리11)를 제공한다. 모듈은 다수의 함수 나 클래스(class)를 묶어서 파일 형식(*.py)으로 제공하며, 패키지는 비슷한 유형의 모듈이 많은 경우 폴더 형태로 묶어서 꾸러미로 제공한다.
- 라이브러리로 제공되는 함수나 클래스를 사용하기 위 해서는 관련 모듈이나 패키지를 먼저 가져와야한다. 파이썬 설치 과정에서 만들어 진 파이썬 라이브러리이다

1. 모듈과 패키지



2. 함수의 유형

- 일반적으로 함수는 외부에서 들어오는 값을 매개변수로 받아서 함수내의 실행문을 통해서 매개변수를 처리하고, 처리 결과를 반환하는 형태를 갖는다.
- 수학에서 $f(x)$ 함수를 프로그램의 함수로 표현하는 예이다. 여기서 x 는 외부에서 들어오는 값을 받는 매개변수이고, $f(x)$ 함수의 수식은 함수 내부에서 실행문으로 이용된다.
- 실행문은 매개변수 x 를 처리하는 역할을 하며, 실행문의 처리 결과는 외부로 반환된다.



2. 함수의 유형

- 프로그램에서 함수는 라이브러리 형식으로 제공되는 내장함수와 사용자가 직접 함수 내에 필요한 코드 를 작성하고, 필요한 경우 함수를 호출하는 사용자정의 함수로 분류된다.
- 파이썬에서는 풍부한 라이브러리를 통해서 수많은 함수를 제공하고, 이를 이용하여 프로그래머는 자신 이 원하는 애플리케이션을 구현할 수 있다.
- 하지만 솔루션을 개발하는데 있어서 자신에게 딱 맞는 함수 가 없는 경우의 사례는 빈번하게 발생된다.
- 이러한 경우에는 개발자가 직접 함수를 구현하여 문제를 해결해야한다.
- 이절에서는 라이브러리에서 제공하는 함수를 사용하는 방법과 사용자가 직접 정의하는 사용자 정의함수의 작성 방법에 대해서 알아본다.

3. 내장함수

- 모듈이나 패키지에서 제공되는 함수를 이용하기 위해서는 먼저 import 명령어를 이용하여 해당 모듈이나 패키지를 작성중인 파일에 포함시켜야 한다. 하지만 사용자의 편의를 위해서 import를 하지 않아도 사용할 수 있는 내장함수도 있다. 이러한 함수들은 builtins 이라는 모듈에 의해서 제공된다.
- builtins 모듈이 아닌 다른 모듈의 함수를 이용하기 위해서는 반드시 다음과 같은 형식으로 해당 모듈을 포함시켜야한다.
- 형식1)은 해당 모듈에서 제공하는 함수 또는 클래스를 가져온다.
- 형식2)는 해당 모듈에서 특정 함수만 가져온다.

```
1)      import 모듈명 ↵  
2)      from 모듈명 import 함수명1, 함수명2, ... ↵
```

3. 내장함수

- import 방식 : 해당 패키지 또는 모듈이 포함하고 있는 모든 구성요소를 포함시키는 방식이다. 명령 문 형식은 간결하지만 구성요소가 많은 경우에는 모든 구성요소가 메모리에 올라가기 때문에 처리속 도가 느려지고, 메모리의 소모가 많아진다.
- from 방식 : 해당 패키지 또는 모듈이 포함하고 있는 구성요소 중에서 특정 요소만 포함시키는 방식 이다. 명령문 형식은 다소 복잡하지만 특정 요소만 메모리에 올라가기 때문에 속도가 빠르고, 메모리 의 효율성 이 높다.

Python Console ↴

```
help(len) # built-in function
Help on built-in function len
in module builtins:
# [1, 2, 3, 4, 5]

len= 5
sum= 15
max= 5
min= 1

평균= 3
중위수= 3
표본 분산= 2.5
표본 표준편차= 1.58113883
```


3. 내장함수

(1) builtins 함수

- builtins 모듈에서 제공되는 함수는 별도의 import 없이 바로 사용할 수 있는 내장함수이다. `help(len)` 명령문으로 `len()` 함수의 도움말을 확인해 보면 'in module builtins:' 문장에서 builtins 모듈에 의해서 제공되는 함수임을 알 수 있다. dataset은 5개의 원소를 갖는 리스트 객체를 참조하는 변수이다. `len()`, `sum()`, `max()`, `min()`의 4개 함수에 각각 dataset을 인수로 넣고 실행할 경우 원소의 길이, 합계, 최댓값, 최솟값 등의 기초 통계량이 계산된다. 이와 같은 함수들은 모두 builtins 모듈에 소속된 함수들이다.

3. 내장함수

(2) import 함수

- import 명령어를 이용하여 외부 모듈을 포함시켜야 사용할 수 있는 함수를 말한다. 대푯값으로 합계는 구했지만 평균은 내장함수가 아니기 때문에 평균을 구할 수 없다. 따라서 수학/통계 관련 함수를 제공하는 statistics 모듈을 import해야 한다. (방법1)은 statistics 모듈에서 제공하는 모든 멤버들을 가져온다는 의미이고, (방법2)는 멤버 중에서 분산과 표준편차를 계산하는 variance, stdev 함수만 가져온다는 의미이다. dataset을 대상으로 평균과 중위수는 (방법1)에 의해서 '모듈 명.함수명()' 형식으로 함수를 호출했으며, 분산과 표준편차는 (방법2)에 의해서 '함수명()' 형식 으로 호출한다.

4. builtins 모듈

- builtins 모듈에서 제공하는 주요 내장함수와 내장클래스는 아래와 같이 `dir(x)` 내장함수를 이용 해서 목록을 확인할 수 있다.
- `import builtins`
- `dir(builtins)` # 내장클래스, 내장함수 목록 보기
- 다음 <표> 5-1은 builtins 모듈에서 제공하는 주요 내장함수와 내장클래스의 기능을 정리한 표이다.

4. builtins 모듈

함수명/클래스명 ↵	기능 ↵	구분 ↵
abs(x) ↵	x의 절대값을 반환한다. ↵	내장함수 ↵
all(iterable) ↵	모든 요소가 True 일 때 True를 반환한다. ↵	내장함수 ↵
any(iterable) ↵	하나 이상의 요소가 True 일 때 True를 반환한다. ↵	내장함수 ↵
bin(number) ↵	10진수 정수를 2진수로 반환한다. ↵	내장함수 ↵
bool(x) ↵	x를 논리형으로 반환한다. ↵	내장클래스 ↵
complex(x) ↵	x를 복소수형으로 반환한다. ↵	내장클래스 ↵
dict(iterable) ↵	사전형 자료구조 형으로 반환한다. ↵	내장클래스 ↵
dir(x) ↵	변수, 내장함수, 내장클래스의 목록을 반환한다. ↵	내장함수 ↵
enumerate(iterable) ↵	열거형 자료를 순회하여 색인과 값을 반환한다. ↵	내장클래스 ↵
eval(expr) ↵	문자열 수식을 계산 가능한 수식으로 반환한다. ↵	내장함수 ↵
float(x) ↵	실수로 형 변환하여 반환한다. ↵	내장함수 ↵
format(value) ↵	value에 양식을 적용한다. ↵	내장함수 ↵
help(x) ↵	함수 또는 클래스의 도움말을 반환한다. ↵	내장함수 ↵
hex(number) ↵	10진수 정수를 16진수 값을 반환한다. ↵	내장함수 ↵

4. builtins 모듈

<code>id(obj)</code> ↵	객체의 주소를 반환한다. ↵	내장함수 ↵
<code>input([prompt])</code> ↵	키보드로 입력한 문자열을 반환한다. ↵	내장함수 ↵
<code>int(x)</code> ↵	x를 정수형으로 반환한다. ↵	내장클래스 ↵
<code>len(obj)</code> ↵	전체 원소의 길이를 반환한다. ↵	내장함수 ↵
<code>list((iterable))</code> ↵	list 자료구조로 형 변환한다. ↵	내장클래스 ↵
<code>max(iterable)</code> ↵	최댓값을 반환한다. ↵	내장함수 ↵
<code>min(iterable)</code> ↵	최솟값을 반환한다. ↵	내장함수 ↵
<code>oct(number)</code> ↵	10진수 정수를 8진수 값으로 반환한다. ↵	내장함수 ↵
<code>open(file, mode)</code> ↵	파일 읽기와 쓰기 ↵	내장함수 ↵
<code>ord(character)</code> ↵	character를 아스키 값으로 반환한다. ↵	내장함수 ↵
<code>pow(x, y)</code> ↵	x에 대한 y의 제곱을 계산하여 반환한다. ↵	내장함수 ↵
<code>print(value)</code> ↵	value를 콘솔에 출력한다. ↵	내장함수 ↵

4. builtins 모듈

<code>range(n)</code> ↗	0에서 $n-1$ 사이의 정수를 반환한다. ↗	내장클래스 ↗
<code>round(number)</code> ↗	실수 <code>number</code> 를 대상으로 반올림을 수행한다. ↗	내장함수 ↗
<code>set(iterable)</code> ↗	셋 자료구조로 형 변환한다. ↗	내장클래스 ↗
<code>sorted(iterable)</code> ↗	반복 가능한 원소들을 대상으로 정렬한다. ↗	내장함수 ↗
<code>str(object)</code> ↗	문자형 자료형으로 변환한다. ↗	내장클래스 ↗
<code>sum(iterable)</code> ↗	숫자들의 합을 구한다. ↗	내장함수 ↗
<code>tuple(iterable)</code> ↗	튜플 자료구조로 형 변환한다. ↗	내장클래스 ↗
<code>type(x)</code> ↗	<code>x</code> 의 자료형을 반환한다. ↗	내장클래스 ↗
<code>zip(iterable*)</code> ↗	<code>iterable</code> 의 원소를 묶어서 반환한다. ↗	내장함수 ↗

5. builtins 내장함수

- builtins 모듈에서 제공하는 주요 내장함수들을 대상으로 프롬프트 (>>>)에서 간단한 예문으로 내장함수의 기능을 알아본다.
- (기존에 학습한 내용과 앞으로 학습할 내용은 제외한다.)
- (1) abs(x)
- abs(x) 함수는 인수 x를 대상으로 절대값을 반환하는 함수이다.
- >>>abs(10) 10
- >>>abs(-10) 10
- >>>abs(-123.45) 123.45

5. builtins 내장함수

- (2) `all(iterable)`
- 모든 요소가 True 일 때 True를 반환한다. 영(0)이 아닌 숫자는 True로 해석한다.
- ```
>>>all([1, True, 10, -15.2])
```
- `True`
- ```
>>>all([1, True, 0, -15.2])
```
- `False`
- ```
>>>all([1, False, 0, -15.2])
```
- `False`



## 5. builtins 내장함수

- (3) `any(iterable)`
- 하나 이상의 요소가 True 일 때 True를 반환한다. 영(0)이 False로 해석한다.
- ```
>>> any([1, False, 0, -15.2])
```
- `True`
- ```
>>> any([False, 0, 0])
```
- `False`
- (4) `bin(number)`
- 10진수 정수를 2진수로 반환한다. 2진수는 '0b' 문자열로 시작한다.
- ```
>>> bin(10) '0b1010'
```
- ```
>>> bin(15) '0b1111'
```

## 5. builtins 내장함수

- (5) `dir(x)`
- 객체 `x`에서 제공하는 변수, 내장함수, 내장클래스의 목록을 반환한다. 다음 예문은 리스트 객체 `x`에서 호출할 수 있는 멤버(변수, 클래스, 함수)들의 목록을 제공하고 있다.
- `x`의 멤버들은 '`x.멤버`' 형식으로 호출한다. 예를 들면 `append()` 함수를 호출하기 위해서는 `x.append(object)` 형식으로 명령문을 작성한다.

## 5. builtins 내장함수

- >>>x = [1,2,3,4,5]
- >>>dir(x)
- [' add ', ' class ', ' contains ', ' delattr ', ' delitem ', ' dir ', ' doc ', ' eq ', ' format ', ' ge ', ' getattribute ', '\_
- \_getitem ', ' gt ', ' hash ', ' iadd ', ' imul ', ' init ', ' i
- nit\_subclass ', ' iter ', ' le ', ' len ', ' lt ', ' mul ', ' ne ', ' new ', ' reduce ', ' reduce\_ex ', ' repr ', ' reversed ', ' rmul ', ' setattr ', ' setitem ', ' sizeof ', ' str ', ' su
- bclasshook ', 'append', 'clear', 'copy', 'count', 'extend', 'index', 'insert', 'pop', 'remove', 'reverse', 'sort']
- >>>x.append(6) # 원소 추가
- >>>x
- [1, 2, 3, 4, 5, 6]

## 5. builtins 내장함수

- (6) `eval(expr)`
- 문자열 수식을 인수로 받아서 계산 가능한 파이썬 수식으로 변환한다.
- `>>>eval("10 + 20")`
- 30
- `>>>eval(10 + "20 * 30")` # 오류 발생
- Traceback (most recent call last): File "<input>", line 1, in <module>
- TypeError: unsupported operand type(s) for +: 'int' and 'str'
- `>>>eval("20 * 30") + 10`
- 610

## 5. builtins 내장함수

- (7) `hex(number)`
- 10진수 정수를 16진수로 반환한다. 16진수는 '0x' 문자열로 시작한다.
- `>>> hex(10) '0xa'`
- `>>> hex(15) '0xf'`
- `>>> hex(64) '0x40'`
- (8) `oct(number)`
- 10진수 정수를 8진수로 반환한다. 8진수는 '0o' 문자열로 시작한다.
- `>>> oct(10) '0o12'`
- `>>> oct(7) '0o7'`
- `>>> oct(8) '0o10'`

## 5. builtins 내장함수

- (9) ord(charactor)
- character를 아스키 값으로 반환한다. 숫자 영(0)은 48, 영문자 대문자 A는 65, 영문자 소문자 a는 97이다.
- >>> ord('0') 48
- >>> ord('9') 57
- >>> ord('A') 65
- >>> ord('Z') 90
- >>> ord('a') 97
- >>> ord('z') 122

## 5. builtins 내장함수

- (10) pow(x, y)
- x에 대한 y의 제곱을 계산하여 반환한다.
- >>>pow(10, 2)
- 100
- >>>pow(10, 3)
- 1000
- >>>pow(10, -1)
- 0.1
- >>>pow(10, -2)
- 0.01

## 5. builtins 내장함수

- (11) round(number)
- 실수를 인수로 하여 반올림을 수행하는 결과를 반환한다.
- >>>round(3.14159) 3
- >>>round(3.14159, 3) # 소숫점 3자리
- 3.142
- (12) sorted(iterable)
- 반복 가능한 원소들을 대상으로 오름차순 또는 내림차순 정렬한 결과를 반환한다.
- >>>sorted([3,2,1,5])
- [1, 2, 3, 5]
- >>>sorted([3,2,1,5], reverse=True) [5, 3, 2, 1]



## 5. builtins 내장함수

- (13) zip(iterable\*)
- 반복 가능한 객체와 객체 간의 원소들을 묶어서 튜플로 반환한다. zip() 함수에서 반환된 결과를 확인하기 위해서 list 클래스를 이용하여 리스트 자료구조로 변환해야 한다.
- >>> zip([1,3,5], [2,4,6])
- <zip object at 0x04030B70>
- >>> list(zip([1,3,5], [2,4,6]))
- [(1, 2), (3, 4), (5, 6)]

**THANK YOU**