

# 기초 PYTHON 프로그래밍

## 12. 파이썬 모듈

1. 모듈 소개하기
2. 모듈 사용하기
3. 모듈 만들기
4. random 모듈

# 1. 모듈 소개하기

## ◆ 파이썬 구성 요소

modules + classes + built-in functions (내장함수)

### random module

random()  
shuffle()  
randint() ....

### math module

pow ()  
sin()  
cos() ....

.... module들

### 내장 함수들

abs() input() int() sum()  
type() print() list() pow() ....

### class list

append ()  
insert()  
sort() ....

### class str

upper ()  
lower()  
count() ....

.... class들

# 1. 모듈 소개하기

---

## ◆ 모듈 (module)

- 코드들을 한 단위로 묶어 사용할 수 있게 하는 하나의 단위.
- 모듈의 종류
  - ① 표준 모듈 - 파이썬 패키지 안에 포함된 모듈
  - ② 사용자 모듈 - 사용자가 만드는 모듈
  - ③ 써드 파티 (third party) 모듈 - 개인이 만들어서 제공하는 모듈
- 모듈 사용의 장점
  - ① 코드의 재사용성
  - ② 서로 다른 모듈에 같은 이름의 메소드가 있어도 충돌이 생기지 않음

## 2. 모듈 사용하기

- ◆ 모듈은 반드시 import 후에 사용할 수 있다

```
>>> import math
```

```
>>> math.pow(2,3) # 모듈 math 내의 pow 함수  
8.0
```

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

```
>>> dir(math)
```

```
['__doc__', '__loader__', '__name__', '__package__', '__spec__', 'acos',  
'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'copysign', 'cos',  
'cosh', 'degrees', 'e', 'erf', 'erfc', 'exp', 'expm1', 'fabs', 'factorial',  
'floor', 'fmod', 'frexp', 'fsum', 'gamma', 'hypot', 'isfinite', 'isinf',  
'isnan', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'modf', 'pi',  
'pow', 'radians', 'sin', 'sinh', 'sqrt', 'tan', 'tanh', 'trunc']
```

## 2. 모듈 사용하기

### ◆ 모듈 import 방법

#### ① import <모듈>

```
>>> import math
```

```
>>> math.pi          # math 모듈에 있는 pi 속성  
3.141592653589793
```

```
>>> math.pow(2,5)     # math 모듈에 있는 pow 함수  
32.0
```

```
>>> math.trunc(10.9)  # 소수점 아래를 버린다  
10
```

## 2. 모듈 사용하기

### ◆ 모듈 import 방법

② from <모듈> import <함수명>

```
>>> from math import pow
```

```
>>> pow(2,5) # 이 경우 함수명만 사용 가능하다  
32.0
```

```
>>> from math import pow, sqrt, trunc # 콤마로 여러 함수 사용 가능
```

```
>>> sqrt(9)
```

```
3.0
```

```
>>> trunc(1.5)
```

```
1
```

```
>>> pow(2,3)
```

```
8.0
```

```
>>> from math import * # math 모듈에 있는 모든 함수를 이름으로 사용
```

```
>>> log2(1024)
```

```
10.0
```

## 2. 모듈 사용하기

---

### ◆ 모듈 import 방법

③ import <모듈> as <alias>

<모듈명> 대신 <alias>를 모듈명으로 사용함.

```
>>> import math as mt
```

```
>>> mt.pow(3,5)
```

```
243.0
```

```
>>> mt.log2(1024)
```

```
10.0
```

### 3. 모듈 만들기

- ◆ 데이터, 함수들로 구성된 파일을 만든다.
- ◆ 파일명이 모듈명이 된다.

파일명 : **calc.py**

```
""" 연산에 필요한 함수들을  
모아 놓은 연산 모듈이다. """
```

```
data = 100
```

```
def add_all(a,b,c):  
    total = a + b + c  
    return total
```

```
def multiply_all(a,b,c):  
    result = a * b * c  
    return result
```

```
import calc
```

```
x = 10
```

```
y = 20
```

```
z = 5
```

```
ans1 = calc.add_all(x,y,z)
```

```
ans2 = calc.multiply_all(x,y,z)
```

```
print(ans1, ans2)
```

```
print('data :', calc.data)
```

```
35 1000  
data : 100
```



## 4. random 모듈

### ◆ 임의의 값을 선택하는 함수들로 구성된 모듈

```
>>> import random
```

```
>>> dir(random)
```

```
['BPF', 'LOG4', 'NV_MAGICCONST', 'RECIP_BPF', 'Random', 'SG_MAGICCONST',  
'SystemRandom', 'TWOPI', '_BuiltinMethodType', '_MethodType', '_Sequence',  
'_Set', '__all__', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__',  
'__name__', '__package__', '__spec__', '_acos', '_ceil', '_cos', '_e', '_exp', '_inst',  
'_log', '_pi', '_random', '_sha512', '_sin', '_sqrt', '_test', '_test_generator',  
'_urandom', '_warn', 'betavariate', 'choice', 'expovariate', 'gammavariate',  
'gauss', 'getrandbits', 'getstate', 'lognormvariate', 'normalvariate',  
'paretovariate', 'randint', 'random', 'randrange', 'sample', 'seed', 'setstate',  
'shuffle', 'triangular', 'uniform', 'vonmisesvariate', 'weibullvariate']
```

## 4. random 모듈

### ◆ 임의의 값을 선택하는 함수들로 구성된 모듈

함수		설명
정수 선택	<code>randint(a,b)</code>	$a \leq N \leq b$ 사이의 임의의 정수 N 선택
	<code>randrange(,,)</code>	<code>range()</code> 결과 중에서 임의의 값 선택
실수 선택	<code>random()</code>	$0.0 \leq F < 1.0$ 사이의 임의의 실수 F 선택
	<code>uniform(a,b)</code>	a와 b 사이의 임의의 실수 선택
군집 자료형 에서 선택	<code>choice(X)</code>	군집자료형 X에서 임의의 원소 선택
	<code>sample(X,k)</code>	군집자료형 X에서 k개의 원소를 임의로 중복없이 선택
	<code>shuffle(X)</code>	군집자료형 X를 섞는다. immutable 자료형에는 적용할 수 없다.

## 4. random 모듈

---

### ◆ 임의의 정수 생성 (randint())

```
>>> import random
```

```
>>> random.randint(10,20)    # 10 ≤ N ≤ 20 사이에서 임의의 수 선택  
20
```

```
>>> random.randint(11,13)  
12
```

```
>>> random.randint(1,100)  
1
```

```
>>> random.randint(1,100)  
90
```

## 4. random 모듈

### ◆ 임의의 정수 생성 (randrange())

```
>>> import random
>>> random.randrange(10)      # range(10)에서 임의의 수 반환
3
>>> random.randrange(5,10)    # range(5,10)에서 임의의 수 반환
9
>>> random.randrange(5,10)
5
>>> random.randrange(5,15,3)  # range(5,15,3)에서 임의의 수 반환
5
>>> random.randrange(5,15,3)
11
>>> random.randrange(5,15,3)
8
```

## 4. random 모듈

### ◆ 임의의 정수 여러 개 생성

```
>>> import random
```

```
>>> [random.randrange(20) for i in range(10)]    # 중복 허용
```

```
[8, 6, 12, 8, 8, 0, 12, 11, 6, 8]
```

```
>>> [random.randrange(0,20,3) for i in range(5)]
```

```
[3, 9, 0, 0, 18]
```

```
>>> random.sample(range(20),10)                # 중복 허용하지 않음
```

```
[12, 8, 17, 1, 3, 6, 9, 0, 11, 10]
```

## 4. random 모듈

### ◆ 임의의 실수 생성 (random(), uniform())

- random() : 0.0에서 1.0 사이의 임의의 float 값 반환
- uniform(a,b) : 두 인수 a와 b 사이의 임의의 float 값 반환

```
>>> from random import random, uniform
```

```
>>> random()  
0.2832728368301739
```

```
>>> random()  
0.4033935667377505
```

```
>>> uniform(3,4)  
3.757279572878299
```

```
>>> uniform(100, 200)  
188.10693319776624
```

## 4. random 모듈

### ◆ 리스트에 random 모듈 이용하기

```
>>> L = list(range(10))
>>> L
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> random.choice(L)
6
>>> [random.choice(L) for i in range(3)]
[1, 6, 4]
>>> random.sample(L,3)
[5, 0, 4]
>>> L
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> random.shuffle(L)
>>> L
[1, 4, 0, 6, 3, 7, 5, 2, 8, 9]
```

## 4. random 모듈

---

### ◆ 튜플에서 random 모듈 이용하기

```
>>> T = tuple(range(1,10,2))
```

```
>>> print(T)
```

```
(1, 3, 5, 7, 9)
```

```
>>> random.choice(T)
```

```
7
```

```
>>> random.sample(T,2)
```

```
[1, 7]
```

```
>>> random.shuffle(T)      # 에러
```



## 4. random 모듈

---

### ◆ 문자열에서 random 모듈 이용하기

```
>>> w = 'abcdefghijklmn'
```

```
>>> random.choice(w)
```

```
'b'
```

```
>>> random.choice(w)
```

```
'l'
```

```
>>> random.sample(w,3)
```

```
['n', 'c', 'a']
```

```
>>> random.shuffle(w)      # 예러
```