chapter 11. 자료형 - set, tuple & 모듈 불러오기

1) 자료형 - set

① 1, 2, 3, 4로 구성된 집합 set1을 만들어 출력하고 자료형을 확인해보시오.

- 힌트: type(집합)

In [1]:

```
set1 = {1, 2, 3, 4}
print(set1)
print(type(set1))
```

```
{1, 2, 3, 4}
<class 'set'>
```

② 위에서 생성한 set1에 새로운 값 5를 추가하고 결과를 확인한 뒤,
또 다시 새로운 값 3을 추가하고 결과를 확인하시오.

- 힌트: 집합.add(추가할 값)

In [2]:

```
set1.add(5)
print(set1)

set1.add(3)
print(set1)
```

```
{1, 2, 3, 4, 5}
{1, 2, 3, 4, 5}
```

③ 'h', 'e', 'l', 'l', 'o'로 구성된 집합 set2을 만들어 출력하고 자료형을 확인해보시오.

In [3]:

```
set2 = {'h', 'e', 'l', 'l', 'o'}
print(set2)
print(type(set2))
```

```
{'l', 'h', 'o', 'e'}
<class 'set'>
```

④ 위에서 생성한 set2에 새로운 값 'w', 'o', 'r', 'l', 'd'를 추가하고 결과를 확인하시오.

In [4]:

```
set2.add('w')
set2.add('o')
set2.add('r')
set2.add('l')
set2.add('d')

print(set2)
```

```
{'o', 'r', 'l', 'h', 'e', 'w', 'd'}
```

2) 자료형 - tuple

① 23.5, 10.1로 구성된 튜플 t1과 1, 2, 3, 4, 5로 구성된 튜플 t2를 각각 만들어 출력하고 자료형을 확인하시오.

- 힌트: type(튜플)

In [5]:

```
t1 = (23.5, 10.1)
t2 = (1, 2, 3, 4, 5)

print(t1)
print(type(t1))
print(t2)
print(type(t2))
```

```
(23.5, 10.1)
<class 'tuple'>
(1, 2, 3, 4, 5)
<class 'tuple'>
```

② 1, 2, 3, 4, 5로 구성된 튜플 t2에서 인덱스를 사용하여 첫 번째 값과 세 번째 값을 조회하시오.

In [6]:

```
t2 = (1, 2, 3, 4, 5)

print(t2[0])
print(t2[2])
```

```
1
3
```

③ empty라는 이름으로 비어 있는 튜플을 만들고 튜플의 자료형을 확인하시오.

In [7]:

```
empty = ()
print(type(empty))
```

```
<class 'tuple'>
```

④ 튜플 t3가 1, 3, 5, 7, 9로 구성되어 있다고 할 때, 인덱스를 사용하여 튜플의 세 번째부터 마지막 값을 출력하시오.

In [8]:

```
t3 = (1, 3, 5, 7, 9)
print(t3[2:])
```

```
(5, 7, 9)
```

3) 라이브러리(패키지), 모듈, 함수

① import로 math 라이브러리를 불러와 봅시다. 불러온 라이브러리를 활용하여 팩토리얼 계산을 해 봅시다.
팩토리얼 계산 함수는 n!을 계산할 때 factorial(n)이다.

In [9]:

```
import math

math.factorial(3)
```

Out[9]:

```
6
```

② 이번에는 import로 불러온 라이브러리에 별칭을 붙여 봅시다.
math 라이브러리를 불러오며 m이라는 별칭을 붙여 주고,
팩토리얼 계산을 해 봅시다.

In [10]:

```
import math as m

m.factorial(3)
```

Out[10]:

```
6
```

③ 이번에는 math 라이브러리에서 factorial 함수만 불러와서 팩토리얼 계산을 해 봅시다.

In [11]:

```
from math import factorial

factorial(3)
```

Out[11]:

```
6
```

④ random 라이브러리 사용하기

- random 라이브러리를 rd 별칭으로 불러옵시다.
- randint 함수를 사용해 봅시다.

- randint(a,b) : a부터 b 사이의 임의의 숫자를 반환합니다.

In [3]:

```
import random as rd

rd.randint(1,10)
```

Out[3]:

5

⑤ [심화] 주사위를 던져서 나온 숫자들의 총합을 구하는 코드를 작성해 봅시다.

- 주사위를 던질 횟수를 받는 변수 count(숫자)
- count 만큼 주사위를 던져서 나온 모든 숫자들의 합 구해 봅시다.
- 주사위의 각 면에는 1 ~ 6까지의 눈이 새겨져 있고, 각 숫자는 무작위로 나옵니다.(randint 함수 사용)
- 힌트 : 주사위를 던져 나온 수들을 리스트에 저장하고, sum 함수를 이용해 봅시다.

In [1]:

```
import random as rd

count = 10
nums = []
for i in range(count) :
    n = rd.randint(1, 6)   # 1 ~ 6 사이에서 임의의 숫자 저장
    nums.append(n)

print(sum(nums))
```

Out[1]:

32

⑥ [심화] 주사위 두개를 한꺼번에 던져서 나온 두 숫자의 합을 구하고, 이를 여러번 시행했을 때의 평균 값을 구해 봅시다.

- 주사위 2개를 던질 횟수 변수 count
- 주사위의 각 면에는 1 ~ 6까지의 눈이 새겨져 있고, 각 숫자는 무작위로 나옵니다.(randint 함수 사용)
- n번 시행후, 두 주사위 숫자 합의 평균을 구해 봅시다.
- 힌트 : 두 주사위를 던져 나온 수의 합을 각각 리스트에 저장하고, sum 함수와 len 함수를 이용하여 평균을 구해 봅시다.

In [18]:

```
count = 10
nums = []
for i in range(count) :
    n1 = rd.randint(1, 6)   # 1 ~ 6 사이에서 임의의 숫자 저장
    n2 = rd.randint(1, 6)
    nums.append(n1+n2)

print(sum(nums)/count)
```

7.0

In [ ]:

```

```