

Python을 활용한 데이터 분석 강의

함수

함수

- 반복되는 부분이 있을 경우

→ ‘반복적으로 사용되는 가치 있는 부분’을 한 뭉치로 묶어서 함수로 작성

```
def 함수명(입력 인수):  
    수행할 문장1  
    수행할 문장2  
    .....
```

```
def sum(a, b):  
    return a+b
```



함수명은 sum이고 입력 인수로 2개의 값을 받으며
결과값은 두 개의 입력값을 더한 값이다

print(), len(), range() ...

함수

- 함수의 형태

```
def sum(a, b):  
    result = a + b  
    return result
```

```
a = sum(3, 4)  
print(a)
```

7

```
def sum(a, b):  
    print('{} , {}의 합은 {}입니다.'.format(a, b, a+b))
```

```
sum(3, 4)
```

3, 4의 합은 7입니다.

```
def say():  
    return 'Hi'
```

```
a = say()  
print(a)
```

Hi

```
def say():  
    print('Hi')
```

```
say()
```

Hi

함수

- 입력값이 여러 개일 때?

```
def 함수명(*입력 인수):  
    수행할 문장1  
    수행할 문장2  
    .....
```

```
def sum_many(*args):  
    sum = 0  
    for i in args:  
        sum = sum + i  
    return sum
```

```
result = sum_many(1, 2, 3)  
print(result)
```

6

```
result = sum_many(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)  
print(result)
```

55

```
def sum_multi(choice, *args):  
    if choice == 'sum':  
        result = 0  
        for i in args:  
            result = result + i  
    elif choice == 'mul':  
        result = 1  
        for i in args:  
            result = result * i  
    return result
```

```
result = sum_multi('sum', 1, 2, 3, 4, 5)  
print(result)
```

15

```
result = sum_multi('mul', 1, 2, 3, 4, 5)  
print(result)
```

120

Practice 3-1 🧐

● 계산기

- 연산자와 정수 두 개를 입력 받아 간단히 수치 연산을 수행하는 함수 `calculator()`를 구현
- 함수를 정의할 때 세 개의 입력 인수를 아래 순서로 정의
 - `operator`, `integer1`, `integer2`
 - `operator`가 'add'이면, 두 정수의 더하기 연산을 수행
 - `operator`가 'sub'이면, 두 정수의 빼기 연산을 수행
 - `operator`가 'mul'이면, 두 정수의 곱하기 연산을 수행
 - `operator`가 'div'이면, 두 정수의 나누기 연산을 수행

```
>>> calculator('add', 3, 5)
8
>>> calculator('mul', 11, 2)
22
```

Practice 3-2 🧐

- 영어 모음 개수를 계산하는 함수
 - 영어 단어(또는 문장) 한 개를 입력 받아 단어(또는 문장) 내에 속한 모음 알파벳(a, e, i, o, u)의 개수를 출력하는 함수 vowel()을 구현

```
>>> vowel('apples')
2
>>> vowel('Hello Mobis')
4
```

Practice 3-3 🧐

- 두 직사각형 면적의 비율을 계산하는 함수
 - 직사각형의 가로 세로 길이를 정수로 입력 받아 직사각형 넓이를 계산한 후, 가로의 길이는 5 증가시키고 세로의 길이는 두 배로 확장한 직사각형의 넓이를 계산한 후에 원래 넓이에서 확장한 후의 넓이를 나눈 값을 소수점 두 자리까지 출력하는 함수 `expand_rectangle()`을 구현
 - 원래 입력한 직사각형의 넓이를 확장한 직사각형의 넓이로 나눈 비율을 출력하기 전에 확장한 가로와 세로의 길이를 먼저 출력

```
>>> expand_rectangle(5, 10)
Width = 10
Length = 20
Area Ratio = 0.25
```

Practice 3-4 🧐

- 정수의 합과 평균을 동시에 계산하는 함수
 - 한 개 이상의 임의의 정수를 입력 받아 그 합과 평균을 반환하는 calculator() 함수를 구현
 - 정수의 합과 평균을 (합, 평균) 튜플로 반환
 - 평균은 소수점만 반올림한 후 정수만 출력
 - 정수를 입력하는 중 0이 입력되면 0 앞까지 입력된 합과 평균만 구함
 - if-else문 활용

```
>>> calculator(1, 2, 3, 4, 5)
(15, 3)
>>> calculator(1, 3, 5, 7, 9)
(25, 5)
>>> calculator(-3, 2, 0, 4, 5)
(-1, 0)
```


예외 처리

- 에러
 - 파이썬의 구문(문법)이 잘못되었을 때 : SyntaxError ~
 - 실행 중에 데이터를 처리할 수 없을 때
- 예외 처리 : 예외가 발생할만한 곳이나 실제 예외가 발생한 위치에 예외가 발생했을 때 어떻게 처리해야할 지 기술

```
try :  
    처리A (에러가 발생할 수 있는 처리)  
except :  
    처리B
```

```
try:  
    prin( '예외가 발생하는 처리' )  
except:  
    print( '예외를 잡았다!' )
```

예외를 잡았다!

Practice 3-5 🧐

- 리스트(튜플) 내의 객체를 찾아주는 함수
 - 리스트나 튜플 내에 특정 객체가 들어 있으면 그 인덱스를 반환하고, 들어 있지 않으면 False를 반환하는 함수 search()를 구현
 - 두 개의 입력 인수 (리스트, 찾고자하는 항목)

```
>>> mylist = [1, 2, 3]
>>> search(mylist, 1)
0
>>> search(mylist, 10)
False
```

Practice 3-6 🧐

- 문자열 검색 함수
 - 단어(혹은 문장)와 검색할 문자 한 개를 입력 받은 후 입력 받은 단어(또는 문장)에서 검색할 문자 한 개와 같은 문자를 (좌에서 우로) 찾아서 첫 번째 검색된 문자의 인덱스를 반환하는 함수 `search_char()`를 구현
 - 두 개의 입력 인수 (단어 혹은 문장, 검색할 문자)
 - 문자가 단어 내에 없을 경우 -1을 반환

```
>>> search_char('python', 'o')
4
>>> search_char('mobis', 'm')
0
>>> search_char('바나나', '나')
1
>>> search_char('사과', '오')
-1
```

Practice 3-7 🧐

- 선택적으로 문자열을 뒤집어 출력하는 함수
 - 한 개 이상의 단어(또는 문장)를 입력 받아 만약 해당 단어(또는 문장)의 길이가 홀수이면 거꾸로 뒤집어 출력하고 문자열의 길이가 짝수이면 그대로 출력하는 함수 `reverse_text()`를 구현
 - 모든 입력된 문자열을 출력한 후, 거꾸로 뒤집은 횟수(즉, 홀수 길이를 가진 문자열의 수)를 반환

```
odd_count = reverse_text('드럼', '베이스', '기타', '키보드', '보컬')
```

```
드럼  
스이베  
기타  
드보키  
보컬
```

```
odd_count
```

```
2
```

Practice 3-8 🧐

- 점수를 정렬하는 함수
 - 한 개 이상의 점수를 받아 가장 높은 점수부터 가장 낮은 점수까지 차례대로 출력하는 함수 `sort_scores()`를 구현

```
sort_scores(100, 27, 65, 88, 47, 97, 74, 53)
```

```
100  
97  
88  
74  
65  
53  
47  
27
```

Practice 3-9 🧐

- 단어 길이로 정렬하는 함수
 - 한 개 이상의 단어를 입력 받아서 단어의 길이가 긴 순서부터 정렬해서 리스트로 반환하는 함수 `sort_by_word_length()`를 구현
 - 길이가 같다면 단어를 알파벳 오름차순으로 정렬

```
colors = ['red', 'blue', 'green', 'brown', 'gray']  
sort_by_word_length(colors)
```

```
['green', 'brown', 'gray', 'blue', 'red']
```

```
instruments = ['피아노', '바이올린', '첼로', '기타', '드럼']  
sort_by_word_length(instruments)
```

```
['바이올린', '피아노', '첼로', '드럼', '기타']
```