파이썬 프로그래밍

25차시

함수 인자의 다양한 활용

⚠ 학습개요

… 지역 변수와 전역 변수

… 위치 인자와 키워드 인자

… 가변 인자와 사전 형식의 키워드 인자

⚠ 학습목표

… 지역과 전역 변수를 이해하고 활용할 수 있다.

… 위치와 키워드 인자를 이해하고 활용할 수 있다.

… 가변과 키워드 인자를 이해하고 활용할 수 있다.

Chapter 1.

지역 변수와 전역 변수

PYTHON PROGRAMMING

⚠ 지역 변수와 전역 변수

+ 지역(local)과 전역(global) 변수

- 파이썬에서는 변수가 메모리에 생성되고 유지되는 범위(scope)

+ 지역 변수(local variables)

- 함수 **내부**에서 대입돼 생성된 변수
- 지역 변수는 함수 외부에서 절대 사용될 수 없음

+ i: 지역 변수

```
def addone():
    # 대입에 사용되는 변수는 지역 변수
    i = 30 # 지역 변수 생성
    i += 1 # 지역 변수 수정
    print('\t 지역 변수 i:', i) # 지역 변수 참조
```

⚠ 지역 변수와 전역 변수

+ 전역 변수(global variables)

- 함수 **외부**에서 대입돼 생성된 변수

+ i: 전역 변수

```
i = 20 # 전역변수
print('i:', i) # 전역 변수
addone() # 함수 호출
print('i:', i) # 전역 변수
```

실행 결과

```
i: 20
        지역 변수 i: 31
i: 20
```

⚠ 함수 내부에서 대입이 있는 변수는 지역 변수로 인식

[코딩실습] 함수 내부에서 대입이 있는 변수는 지역 변수로 인식 난이도 기본

```
1. def addone():
2.     '''함수에서 대입에 사용되는 변수는 지역 변수'''
3.     i = 30 # 지역 변수 생성
4.     i += 1 # 지역 변수 수정
5.     print('\t지역 변수 i:', i) # 지역 변수 참조
6.
7. i = 20 # 전역 변수
8. print('i:', i) # 전역 변수
9. addone() # 함수 호출
10.print('i:', i) # 전역 변수
```

결과

```
i = 20
        지역 변수 i: 31
i = 20
```

⚠ 함수 내부에서 대입 없이 참조만 하면 전역 변수로 인식

[코딩실습] 함수 내부에서 대입이 없는 변수는 전역 변수로 인식 난이도 기본

```
1. def addone():
2.     '''대입이 없는 변수가 참조되면 전역 변수로 인식'''
3.     print('\t 전역 변수 i 읽기, i + 1: ', i + 1)
4.
5. i = 20 # 전역 변수
6. print('i =', i)
7. addone() # 함수 호출
8. print('i =', i)
```

결과

```
i = 20
        전역 변수 i 읽기, i + 1: 21
i = 20
```

⚠ 함수 내부에서 대입이 있는 변수는 지역 변수로 인식

```
def addone():
    # 대입이 없는 변수가 참조되면 전역 변수로 인식
    # 그러나 전역 변수에 i가 없다면 오류 발생
    print('\t전역 변수 i 읽기, i + 1: ', i + 1)


addone() # 함수 호출
```

실행 결과

```
    print('\t전역 변수 i 읽기, i + 1: ', i + 1)
NameError: name 'i' is not defined
```

⚠ 함수에서 값을 대입하지 않고 지역 변수를 참조하면 오류

+ 생성되지 않은 변수 참조는 오류 발생

```
>>> value += 1 # value = value + 1
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'value' is not defined
```

⚠ 함수에서 값을 대입하지 않고 지역 변수를 참조하면 오류

+ 함수에서 주의

```
def addone():
    '''대입이 있는 변수는 지역 변수로 인식'''
    # "지역 변수 i에 대입하기도 전에 참조했다"라는 오류 메시지 출력
    print('\t 지역 변수 i 읽기, i + 1:', i + 1)
    i += 1 # 수정이 있으므로 i는 지역 변수로 인지
```

```
UnboundLocalError: local variable 'i' referenced before assignment
```

```
i = 20 # 전역 변수
print('i =', i)
addone() # 함수 호출로 함수 내부 4번 줄에서 오류
print('i =', i)
```

⚠ 함수에서 global 문장으로 변수를 전역 변수로 지정

[코딩실습] 키워드 global로 전역 변수를 명시적으로 지정 난이도 응용

```
1. def addone():
2.     '''명시적으로 변수를 전역 변수로 지정'''
3.     global i # i를 명시적으로 전역 변수로 인식
4.     print('\t 전역 변수 i읽기, i + 1:', i + 1)
5.     i += 1 # 전역 변수 i 수정
6.
7. i = 20 # 전역 변수
8. print('i =', i)
9. addone()
10.print('i =', i)
```

결과

```
i = 20
        전역 변수 i 읽기, i + 1: 21
i = 21
```

끝

Chapter 2.

위치 인자와 키워드 인자

PYTHON PROGRAMMING

⚠ 위치 인자와 키워드 인자

+ 위치 인자 사용

```
def mysubtract(pre, post):
    '''두 수의 차 pre - post 반환 함수'''
    return pre - post

print(mysubtract(10, 5))
```

+ 키워드 인자 사용

```
print(mysubtract(post = 5, pre = 10))
```

⚠ 위치 인자와 키워드 인자

✚ 키워드 인자는 모든 위치 인자 이후에 사용

```
print(mysubtract(10, post = 5))


>>> print(mysubtract(post = 5, 10))
  File "<stdin>", line 1
SyntaxError: positional argument follows keyword argument


>>> print(mysubtract(10, pre = 5))
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: mysubtract() got multiple values for argument 'pre'
```

Chapter 3.

가변 인자와 사전 형식의 키워드 인자

PYTHON PROGRAMMING

⚠ 가변 인자 *args

✚ 인자 *args, 가변 인자로 전달하고 처리

```
>>> print(1, 2, 3)
1 2 3
>>> print('python', 'program')
python program
```

⚠ 가변 인자 *args

+ *리스트

- 별표는 리스트나 튜플을 풀어(unpacking) **가변 인자 형태**로 변환한다는 의미

[코딩실습] 여러 친구에게 인사하는 함수 구현 난이도 응용

```
1. def hello(*names):
2.     """가변 인자 name 활용"""
3.     for each in names:
4.         print('안녕, {}!'.formet(each))
5.
6. hello('나타샤')
7. hello('수빈', '현수', '지효')
8. hello(*['방탄소년단', '여자친구'])
```

⚠ 사전 형식의 키워드 가변 인자 **kwargs

+ 키워드 인자 형태의 가변 인자로 전달하고 처리

[코딩실습] 키워드 형태 가변 인자 함구 구현 난이도 응용

```
1. def mykeyprint(**kwargs):
2.     for key in kwargs:
3.         print('{}: {}'.format(key, kwargs[key]), end = '')
4.     print()
5.
6. mykeyprint(여자친구=6, 마마무=4)
7. mykeyprint(엑소=9, 트와이스=9, 블래핑크=4, 방탄소년단=7)
8.
9. coffeeprice = {'에스프레소': 2500, '아메리카노': 2800, '카페라테': 3200}
10.mycar = {"brand": "현대", "model": "제네시스", "year": 2016}
11.mykeyprint(**coffeeprice)
12.mykeyprint(**mycar)
```

⚠ 사전 형식의 키워드 가변 인자 **kwargs

+ 키워드 인자 형태의 가변 인자로 전달하고 처리

결과

```
여자친구: 6  마마무: 4
엑소: 9  트와이스: 9  블랙핑크: 4  방탄소년단: 7
에스프레소: 2500  아메리카노: 2800  카페라테: 3200
brand: 현대  model: 제네시스  year: 2016
```

⚠ 위치 인자와 키워드 형태 가변 인자 함수 구현

[코딩실습] 위치 인자와 키워드 형태 가변 인자 함수 구현 난이도 응용

```
1. def sumvalue(value, **kwargs):
2.     hap = value
3.     for key in kwargs:
4.         hap += kwargs[key]
5.     return hap
6.
7. coffeeprice = {'에스프레소': 2000, '아메리카노': 2800, '카페라테': 3200}
8. print(sumvalue(1000, **coffeeprice))
9. print(sumvalue(value = 2000, **coffeeprice))
10.print(sumvalue(**coffeeprice, value = 2000)) # 둘다 키워드 인자
11.
12.# print(sumvalue(**coffeeprice)) # 오류 발생
13.# print(sumvalue(**coffeeprice, 2000)) # 오류 발생
```

결과

```
9000
10000
10000
```

⚠ 가변 인자 *args와 키워드 형태의 인자 **kwargs을 함께 사용

[코딩실습] 가변 인자와 키워드 형태 가변 인자 함수 구현 난이도 응용

```
1. def sumvalue(*values, **kwargs):
2.     hap = 0
3.     # 가변 인자 합 구하기
4.     for v in values:
5.         hap += v
6.     # 키워드 인자 합 구하기
7.     for key in kwargs:
8.         hap += kwargs[key]
9.     return hap
10.
11.coffeeprice = {'value': 2000, '에스프레소': 2000, '아메리카노': 2800, '카페라테': 3200}
12.print(sumvalue(1, 2, 3, **coffeeprice))
13.print(sumvalue(*[2, 3, 4], **coffeeprice))
14.
15.# print(sumvalue(**coffeeprice, *[2, 3, 4])) # 오류 발생
```

⚠ 가변 인자 *args와 키워드 형태의 인자 **kwargs을 함께 사용

결과

```
10006
10009
```

⚠지역 변수와 전역 변수

지역변수

… 함수 내부에서 대입이 발생하는 변수

전역변수

… 함수 외부(main)에서 대입이 발생하는 변수
… 키워드 global로 정의된 문서

⚠위치 인자와 키워드 인자

… 키워드 인자는 모든 위치 인자 이후에 사용

⚠가변 인자와 사전 형식의 키워드 인자

… 가변 인자: *args
… 사전형식의 키워드 인자: **kwargs