

5주차: 함수

5-1. 함수 만들기

용어 정리

- 함수를 사용하는 것 → "함수를 호출한다."
- 함수 호출 시 넣는 여러가지 데이터 → 매개변수
- 함수를 호출해서 최종적으로 나오는 결과 → 리턴 값, 반환 값 등

함수의 기본

- 함수를 사용하는 이유
 - 반복적으로 사용되는 부분을 한 뭉치로 묶어서 사용
 - 프로그램의 흐름을 일목요연하게 파악 가능
- `def` 함수 이름():
 함수 내용
- `return` 을 통해 결과 반환

```
# 기본적인 함수
# 매개변수와 리턴값이 존재하지 않음
def print_3_times():
    print("안녕하세요")
    print("안녕하세요")
    print("안녕하세요")

# 함수 호출
print_3_times()
```

함수에 매개변수 만들기

```
def print_n_times(value, n):
    for i in range(n):
        print(value)
```

```
print_n_times("안녕하세요", 5)
```

가변 매개변수

- 매개변수를 원하는 만큼 입력하고 싶을 때
- 가변 매개변수 뒤에는 일반 매개변수가 올 수 없음
 - 그렇지 않으면 어디까지가 가변 매개변수고, 어디가 일반 매개변수인지 구분하기 힘들
- 가변 매개변수는 하나만 사용 가능
- 가변 매개변수의 타입은 '튜플'임
- `def 함수 이름(매개변수, 매개변수, ..., *가변 매개변수):`

```
# 가변 매개변수 함수
def print_n_times(n, *values):
    # n번 반복
    for i in range(n):
        # values는 리스트처럼 활용
        for value in values:
            print(value)
        print()

# 함수를 호출
print_n_times(3, "안녕하세요", "즐거운", "파이썬 프로그래밍")
```

기본 매개변수

- 매개변수를 입력하지 않았을 경우 매개변수에 들어가는 기본값
- 기본 매개변수 뒤에는 일반 매개변수가 올 수 없음

```
def print_n_times(value, n=2):
    # n번 반복
    for i in range(n):
        print(value)
```

```
# 함수를 호출합니다.  
print_n_times("안녕하세요")
```

키워드 매개변수

- 가변 매개변수와 기본 매개변수를 둘을 같이 쓴다면?

```
def print_n_times(n=2, *values):  
    # n번 반복  
    for i in range(n):  
        for value in values:  
            print(value)  
        print()
```

```
# 함수 호출, 그러나 "안녕하세요"가 n에 들어감  
print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍")
```

- 순서를 바꿔보자

```
def print_n_times(*values, n=2):  
    # n번 반복  
    for i in range(n):  
        for value in values:  
            print(value)  
        print()
```

```
# 함수 호출, values에 모든 값이 들어감. 가변 매개변수가 우선  
print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍", 3)
```

- 기본 매개변수는 가변 매개변수 앞에 써도 의미가 없다.
- 그러나, 두 가지를 함께 사용하기 위해 '키워드 매개변수' 사용
- 매개변수 이름을 지정해서 입력

```
def print_n_times(*values, n=2):  
    # n번 반복  
    for i in range(n):
```

```
for value in values:
    print(value)
print()
```

```
# 함수 호출, 그러나 "안녕하세요"가 n에 들어감
print_n_times("안녕하세요", "즐거운", "파이썬 프로그래밍", n=3)
```

리턴(반환)

- **return** : 함수를 실행했던 위치로 돌아가라는 뜻, 함수가 끝나는 위치

```
# 함수 정의
def return_test():
    print('A 위치입니다.')
    return
    print('B 위치입니다.')
```

```
# 함수 호출
return_test()
```

- 데이터 리턴하기

```
# 함수 정의
def return_test():
    return 100

# 함수 호출
value = return_test()
print(value)
```

- 아무것도 리턴하지 않는다면?

```
# 함수 정의
def return_test():
    return

# 함수 호출
```

```
value = return_test()
print(value) # None 출력
```

- `None` 은 파이썬에서 '없다'라는 의미

기본적인 함수의 활용

```
# 함수 정의
# 두 매개변수를 더하는 함수
# 두 매개변수를 받고 더한 결과를 리턴함
def add(a, b):
    return a + b

# add(3, 4)의 반환값 출력
# 함수를 호출할 때에는 지정한 매개변수와 같은 수의 매개변수를 입력
print(add(3, 4))
# add(3, 4)의 반환값을 다시 add()의 매개 변수로
print(add(1, add(2, 3)))
```

```
# 기본 매개변수와 키워드 매개변수를 활용해 범위의 정수를 더하는 함수
# 함수 선언
# start부터 end까지 step만큼 더하며 범위의 합을 구함
def sum_all(start=0, end=100, step=1):
    # 변수 선언
    output = 0
    # 반복문을 돌려 숫자를 더함
    for i in range(start, end + 1, step):
        output += i
    # 리턴
    return output

# 함수 호출
print("A. ", sum_all(0, 100, 10))
print("B. ", sum_all(end=50))
print("C. ", sum_all(end=40, step=5))
```