

함수 만들기

들어가기

반복적으로 실행되어야 하는 코드의 덩어리를 우리가 쉽게 호출할 수 있는 이름으로 정의하여 한 번의 호출로 함수 내의 실행 코드를 순차적으로 실행하고 싶을 때 우리는 함수를 사용한다.

지금부터는 함수에 입력값과 출력값에 대해서 살펴 보도록 하자.

학습목표

함수의 개념을 이해하고, 나만의 함수를 만들 수 있다.

매개변수를 사용하는 정의된 함수를 호출할 때, 인자를 전달하여 원하는 결과를 볼 수 있다.

핵심 키워드

- 함수
- 인자 `Argument`
- 매개변수 `Parameter`

학습내용

우리만의 함수 만들기

함수를 만드는 과정에서 가장 중요한 것은 `저장과 호출에 대한 이해`이다.

앞서 살펴본 것처럼 함수를 저장하기 위해서는 `def` 라는 예약어를 사용한다.

```
def greeting():  
    print("Hello World")
```

`:` (콜론) 뒤에 우리가 실행하고자 하는 실행코드를 입력하는 것으로 우리가 원하는 결과를 기대할 수 는 없다.

여기까지는 함수를 정의하는 단계이다.

지금부터는 호출 방법이다.

우리가 원하는 결과를, 즉, Hello World 가 실행되기를 바란다면, 우리가 정의한 이름으로 저장된 함수를 호출해야 한다.

여기서는 `greeting()` 이라는 이름으로 호출하면 된다.

```
def greeting():  
    print("Hello World")  
  
greeting()  
# Hello World 가 출력된다.
```

인자 (Argument)

인자 란 함수를 호출할 때 전달하는 값을 말한다.

넘겨받는 수 또는 값이라고 생각하면 쉽게 이해할 수 있다.

앞서 사용했던 `print()` 함수에 들어가는 문자열도 인자이다.

매개변수 (Parameter)

매개변수 는 함수가 정의된 곳에서, 변수처럼 사용하는 것을 말한다.

```
def greeting(lang):  
    print(lang)  
  
greeting("Hello World")  
  
# Hello World 가 출력된다.
```

반환값 (Return Value)

종종 함수는 함수가 정의된 곳에서 전달받은 매개변수를 이용해 프로그래머가 의도한 코드를 실행한 뒤, 계산 결과인 값을 반환할 수도 있다.

이와 같은 상황이라면 당연히 함수를 다른 함수의 인자로 사용할 수도 있다.

```
def greet():  
    return "Hello"  
  
print(greet(), "Connect")  
print(greet(), "python")  
  
# Hello Connect으로 출력됩니다.  
# Hello Python으로 출력됩니다.
```

Multiple 매개변수 / 인자

여러 개의 매개변수를 받는 함수를 만들 수도 있다.

더하기 함수를 한번 만들어 보자.

```
def add(left, right):  
    return left + right  
  
print(add(1,2))  
  
# 3
```