

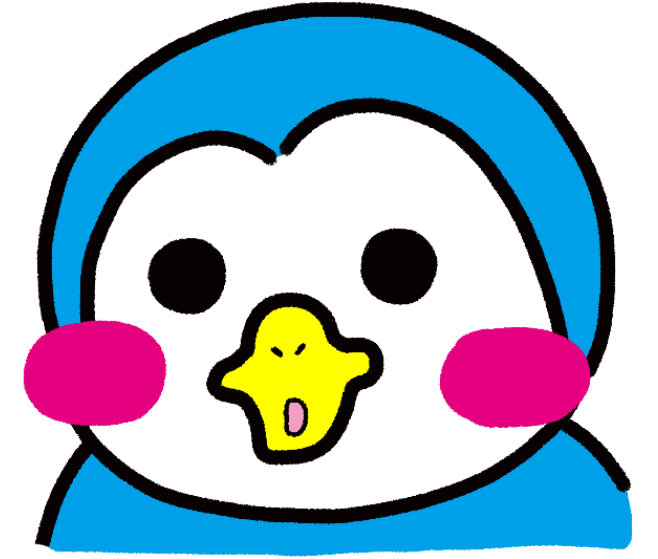
# 입문자를 위한 파이썬 기초



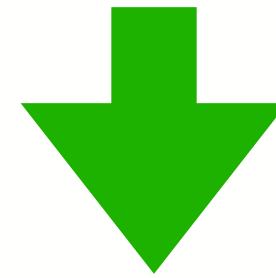
---

함수의 매개변수 그리고 반환값

# 함수에 데이터를 전달하세요



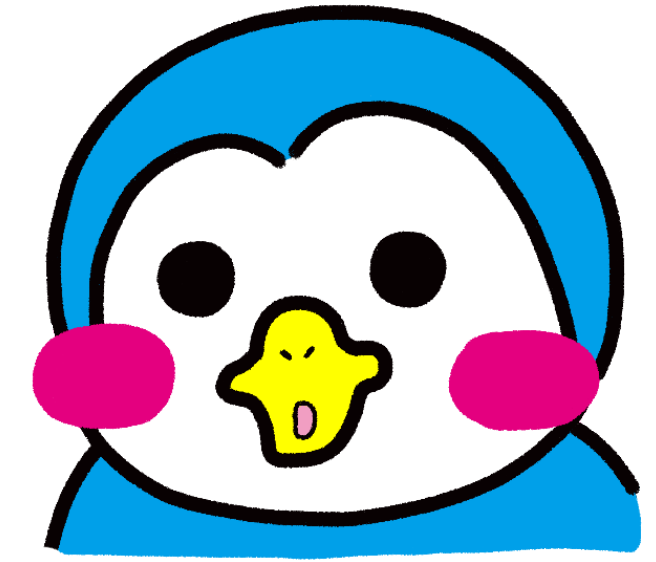
함수 호출문은 '함수명 + 소괄호'이다. 소괄호의 역할은, 함수가 실행될 때 사용할 데이터를 전달받는 것이다. 이렇게 전달되는 데이터를 가리켜 '인수'라 한다!



`yourFunc( 여기 )`

=> 소괄호 안에 데이터를 전달할지 말지 여부는 프로그래머의 선택 사항이다!

# 인수를 전달할 수 있으려면

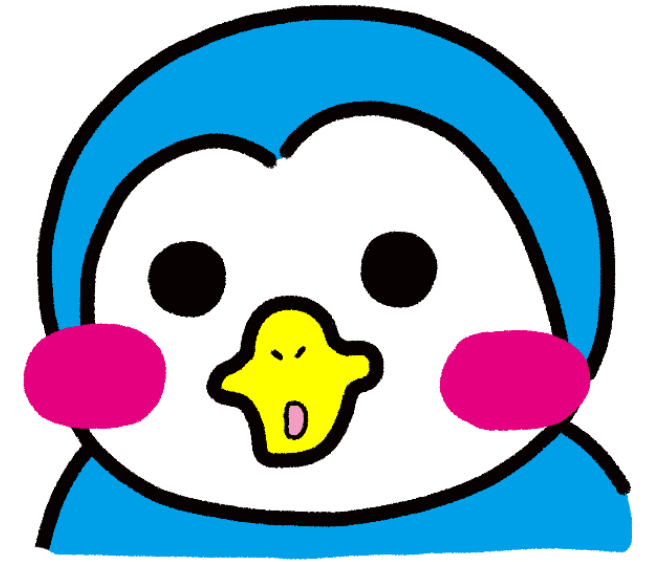


인수를 전달받아 기능을 수행할 수 있는 함수를 만들기 위해서는 함수 정의 시 매개변수를 추가해야 한다. 이는 함수의 기능을 다양화하는 유용한 도구다!

```
# 매개변수 greet는 함수가 호출될 때 전달되는  
# 인수를 대입하기 위해 사용하는 변수이다.  
def sayHello(greet) :  
    print(greet)
```

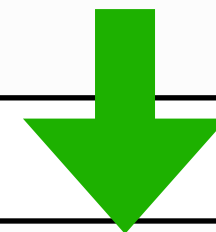
=> 매개변수는 변수의 일종이므로, 이름 규칙은 일반 변수와 동일하다!

# 인수를 전달할 수 있으면



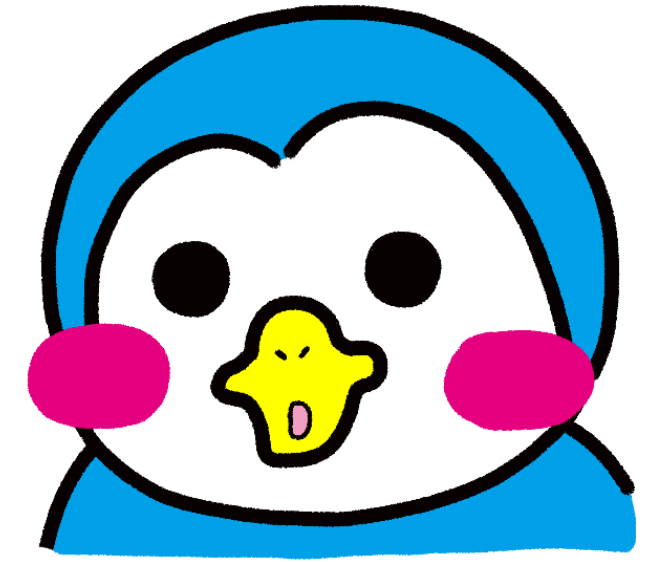
인수를 전달받아 기능을 수행할 수 있는 함수를 만들기 위해서는 함수 정의 시 매개변수를 추가해야 한다. 이는 함수의 기능을 다양화하는 유용한 도구다!

```
# 매개변수 greet는 함수가 호출될 때 전달되는  
# 인수를 대입하기 위해 사용하는 변수이다.  
def sayHello(greet) :  
    print(greet)
```



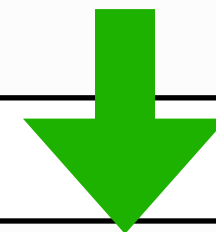
```
sayHello("안녕하세요") # greet = "안녕하세요"  
sayHello("Hello") # greet = "Hello"
```

# 인수를 전달할 수 있으면

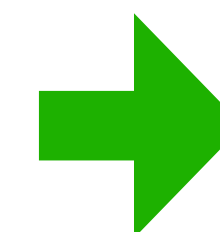


인수를 전달받아 기능을 수행할 수 있는 함수를 만들기 위해서는 함수 정의 시 매개변수를 추가해야 한다. 이는 함수의 기능을 다양화하는 유용한 도구다!

```
# 매개변수 greet는 함수가 호출될 때 전달되는  
# 인수를 대입하기 위해 사용하는 변수이다.  
def sayHello(greet) :  
    print(greet)
```



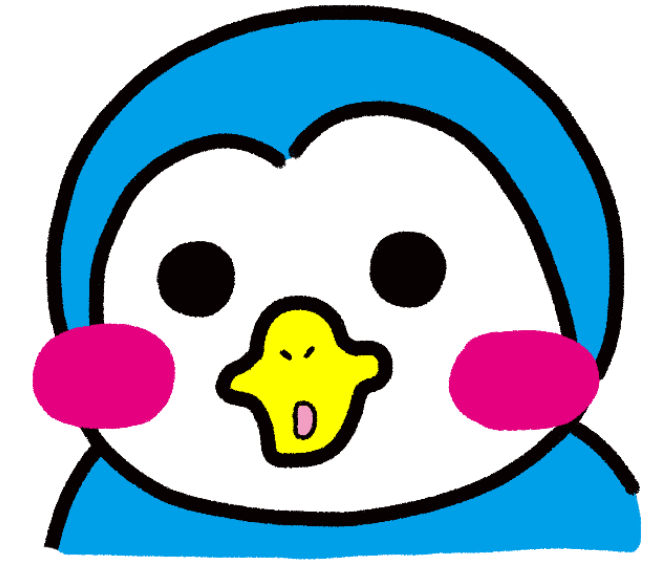
```
sayHello("안녕하세요") # greet = "안녕하세요"  
sayHello("Hello") # greet = "Hello"
```



안녕하세요  
Hello

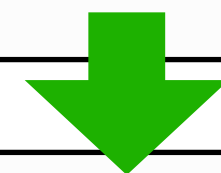


# 원하는 만큼 써도 돼요

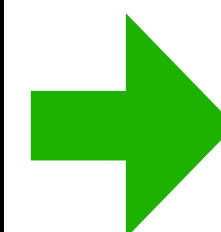


함수 정의 시에는 매개변수를 '원하는 만큼' 추가할 수 있다. 단 두 개 이상의 매개변수를 사용할 경우, 각각을 심표로 구분해주어야 한다.

```
def info(date, weather) :  
    form = "%s의 날씨 : %s"  
    script = form % (date, weather)  
    print(script)
```

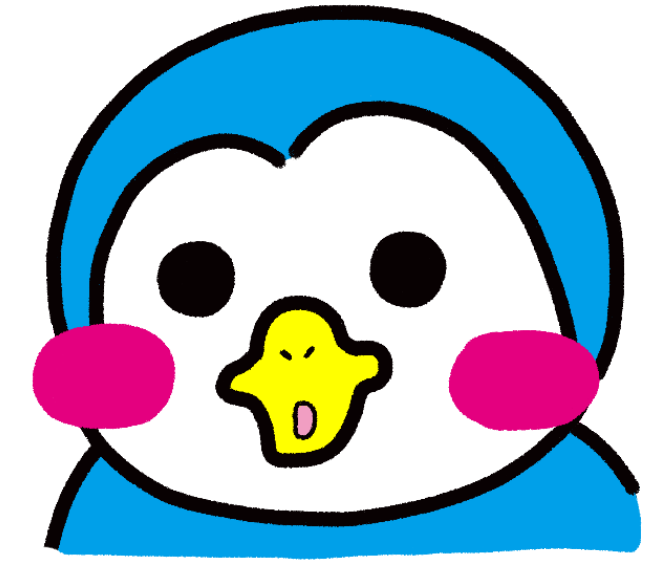


```
info("12월 25일", "흐리고 눈")  
info("05월 05일", "맑음")
```



```
12월 25일 의 날씨 : 흐리고 눈  
05월 05일 의 날씨 : 맑음
```

# 원하는 만큼 써도 돼요



함수 정의 시에는 매개변수를 ‘원하는 만큼’ 추가할 수 있다. 단 두 개 이상의 매개변수를 사용할 경우, 각각을 심표로 구분해주어야 한다.

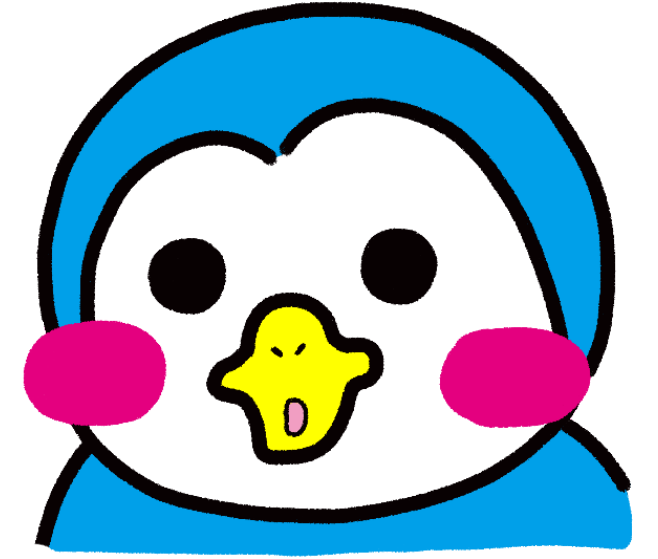
```
def info(date, weather) :  
    form = "%s의 날씨 : %s"  
    script = form % (date, weather)  
    print(script)
```

함수 info는 두 개의 매개변수가 있기에 호출 시 인수를 두 개 받을 수 있다!  
함수 info는 전달받은 인수를 가지고 문자열 포매팅 및 출력을 수행한다!

```
info("12월 25일", "흐리고 눈")  
info("05월 05일", "맑음")
```

```
12월 25일 의 날씨 : 흐리고 눈  
05월 05일 의 날씨 : 맑음
```

# 함수의 값 반환



프로그래머는 값을 반환하는 함수를 만들 수 있다. 함수가 값을 반환한다는 것은, 함수 호출문이 값을 대체할 수 있다는 의미다. 그리고 이를 위해서는 파이썬의 `return` 키워드를 사용해야 한다.

`input(x)`

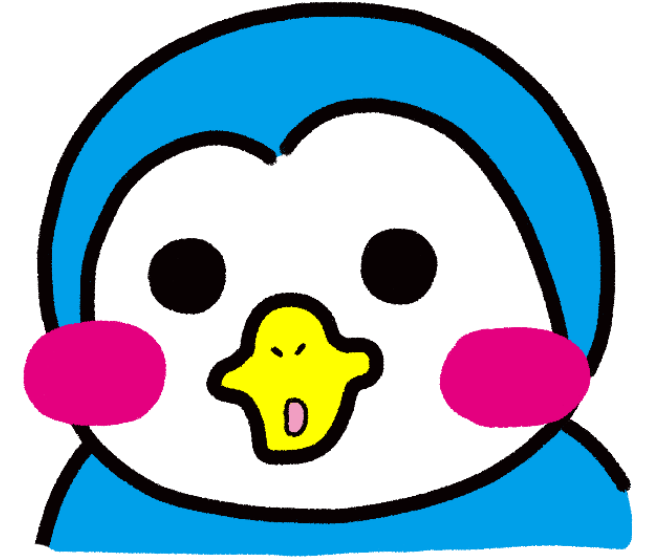
주어진 인수 `x`를 보여준 다음  
사용자 입력을 처리한다!

=> 값을 반환해 주는 함수의 대표적인 예!



# return이 없을 때

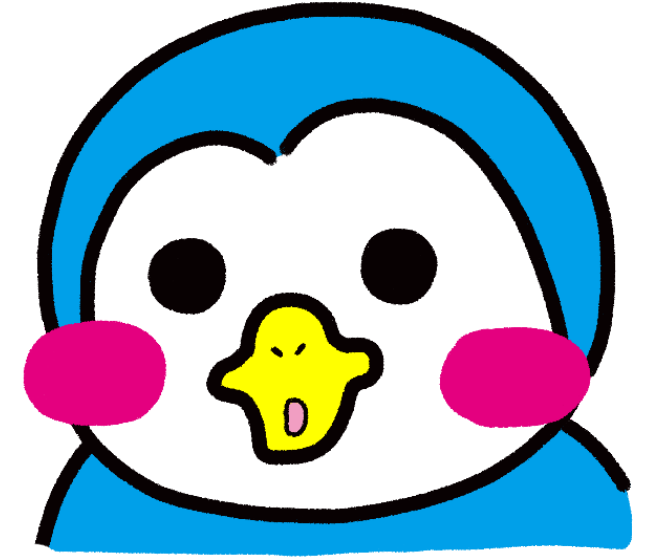
---



```
def sayHello() :  
    print( "안녕하세요!" )
```

# return이 없을 때

---

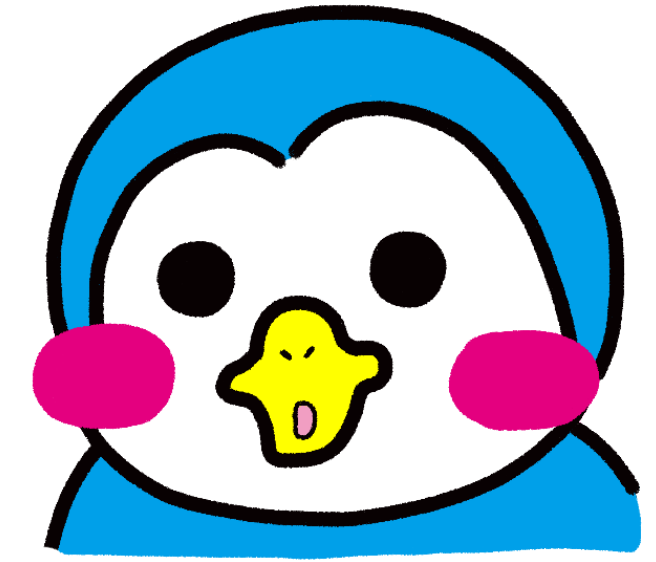


```
def sayHello() :  
    print("안녕하세요!")
```



```
...  
# 가지고 있는 코드를 실행시키고 사라진다!  
sayHello()
```

# return이 없을 때



```
def sayHello() :  
    print("안녕하세요!")
```

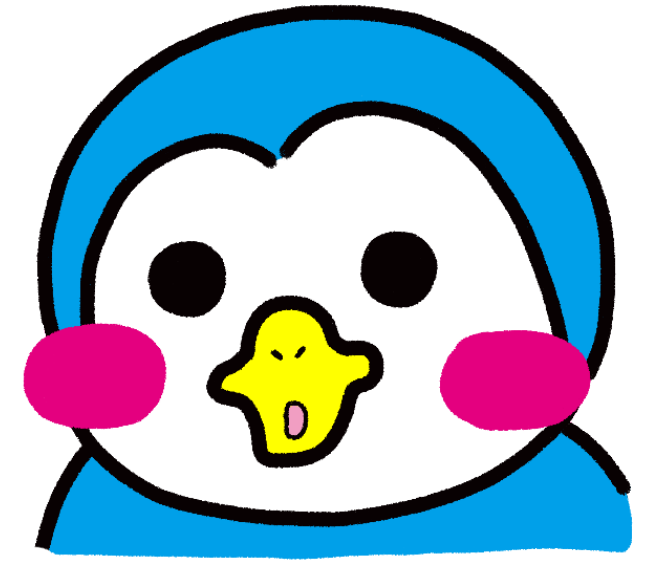


```
...  
# 가지고 있는 코드를 실행시키고 사라진다!  
sayHello()
```



```
check = sayHello()  
print(check) # 아무것도 없다!!
```

# return이 없을 때



```
def sayHello() :  
    print("안녕하세요!")
```

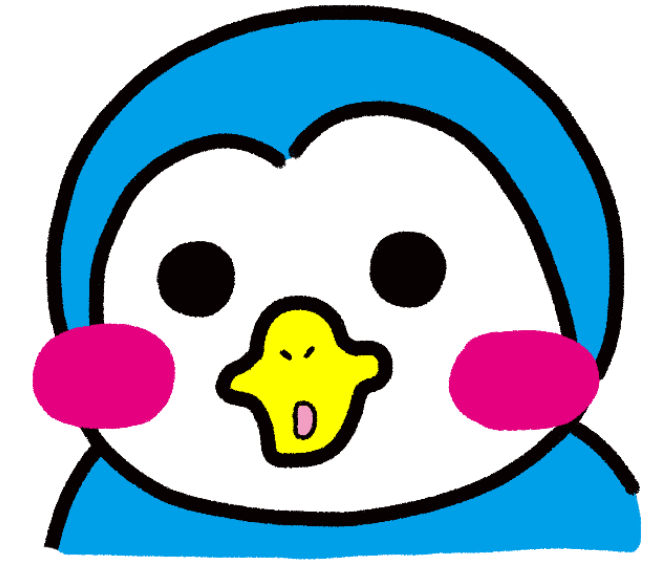
```
...  
# 가지고 있는 코드를 실행시키고 사라진다!  
sayHello()
```

```
check = sayHello()  
print(check) # 아무것도 없다!!
```

안녕하세요!  
안녕하세요!  
None

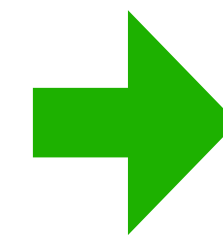


# return이 있을 때



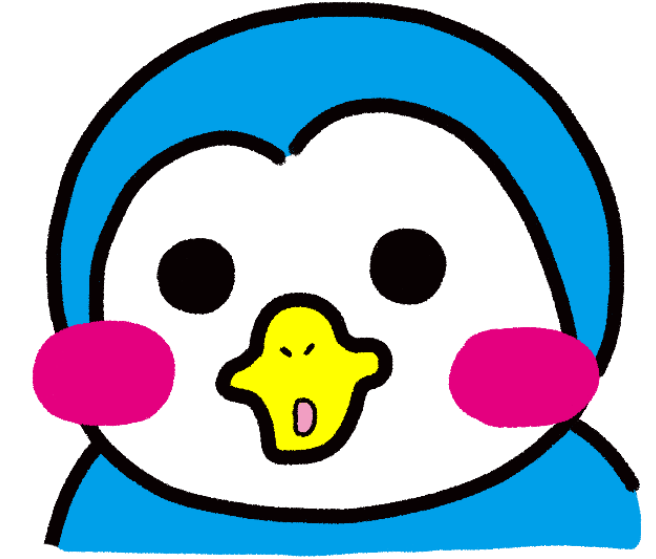
함수 내에서 사용된 return은 뒤에 붙은 데이터를 반환한다. 이때 데이터는 자료형과 무관하게 딱 하나씩만 반환할 수 있다.

```
def returnHello() :  
    print("인삿말을 반환합니다")  
    return "안녕하세요!"  
  
# 함수 returnHello의 호출문은  
# 기능 수행 후 반환값을 남기고 사라진다  
value = returnHello()  
print(value)
```



인삿말을 반환합니다  
안녕하세요!

# 데이터를 반환한다



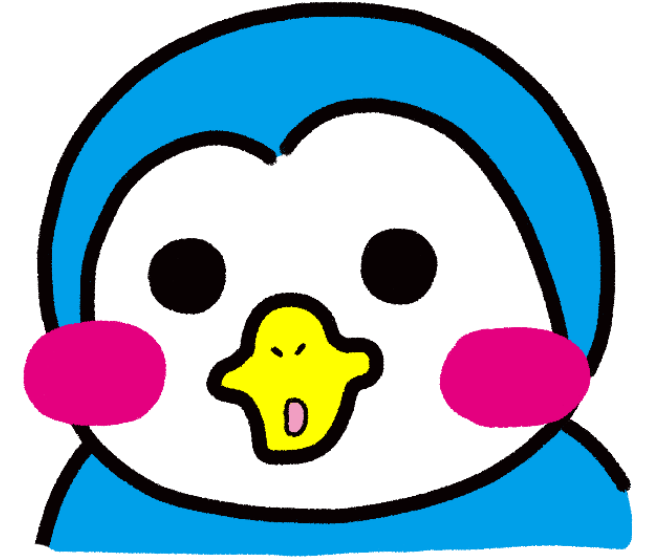
함수 내에서 사용된 return은 뒤에 붙은 데이터를 반환한다. 이때 데이터는 자료형과 무관하게 딱 하나씩만 반환할 수 있다.

```
def returnHello() :  
    print("인삿말을 반환합니다")  
    return "안녕하세요!"  
  
# 함수 returnHello의 호출문은  
# 기능 수행 후 반환값을 남기고 사라진다  
value = returnHello()  
print(value)
```

인삿말을 반환합니다  
안녕하세요!

# 내용 정리

---



- 함수 정의 시 매개변수를 추가하면 이를 이용해 함수 호출 시 데이터를 전달할 수 있다. 이때 전달되는 데이터를 가리켜 인수라 한다.
- 매개변수의 개수는 0개부터 n개까지, 원하는 만큼 자유롭게 추가할 수 있다.
- 함수 정의 시 return 키워드를 사용해 함수가 값을 반환하게 만들 수 있다.
- 값을 반환하는 함수가 호출되면, 함수 호출문은 기능을 수행한 후에 반환값으로 대체된다.
- return을 이용한 값 반환은 자료형과 무관하게 딱 하나씩만 가능하다.