

파이썬 프로그래밍

24차시

함수 정의와 활용



⚠ 학습개요

- ... 함수 개요와 정의 구문
- ... 인자 전달과 반환
- ... 문서화 문자열

⚠ 학습목표

- ... 함수를 이해하고 특정 기능의 함수를 구현할 수 있다.
- ... 함수에 인자를 전달하고 받을 수 있다.
- ... 함수의 문서화 문자열을 만들 수 있다.

Chapter 1.

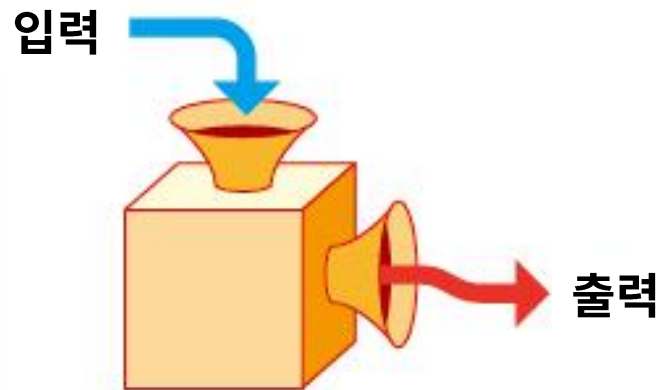
함수 개요와 정의 구문

P Y T H O N P R O G R A M M I N G

! 함수 개요 1

+ 특정한 기능을 수행하는 프로그램 단위인 함수

- 여러 입력을 받아 특정한 기능을 수행하고 결과값을 반환하는 코드
- 사용자가 직접 정의해 사용
- 파이썬에 설치된 다양한 함수를 활용



[그림24-1] 함수 개념: 커피머신과 믹서기

! 함수 개요

+ 사용자 정의 함수의 함수 정의와 함수 호출

사용자 정의 함수
(user-defined functions)

내장 함수
(built-in functions)

`print()`와 `input()`,
`str()`, `int()`, `float()`, `len()`

! 함수 개요

+ 사용자 정의 함수와 내장 함수의 이해

- 사용자 정의 함수(user-defined functions): 우리가 직접 만드는 함수

함수이름 인자

```
def fahr_to_celsius(temp):  
    return ((temp - 32) * (5/9))
```

반환 값

def keyword name parameter

```
def fahr_to_kelvin(temp):  
    return ((temp - 32) * (5/9)) + 273.15
```

return statement return value

! 함수 개요

+ 사용자 정의 함수와 내장 함수의 이해

- 내장 함수(built-in functions): 파이썬 설치와 함께 이미 만들어 놓은 함수

내장함수								
abs()	bytes()	dir()	frozenset()	int()	max()	pow()	setattr()	type()
all()	callable()	divmod()	getattr()	isinstance()	memoryview()	print()	slice()	vars()
any()	chr()	enumerate()	globals()	issubclass()	min()	property()	sorted()	zip()
ascii()	classmethod()	eval()	hasattr()	iter()	next()	range()	staticmethod()	__import__()
bin()	compile()	exec()	hash()	len()	object()	repr()	str()	
bool()	complex()	filerr()	help()	list()	oct()	reversed()	sum()	
breakpoint()	delattr()	float()	id()	locals()	open()	round()	super()	
bytearray()	dict()	format()	input()	map()	ord()	set()	tuple()	

Chapter 2.

인자 전달과 반환

P Y T H O N P R O G R A M M I N G

⚠ 파이썬 함수 정의 구문: 함수 이름과 인자, 반환 값

+ 함수 머리는 키워드 def

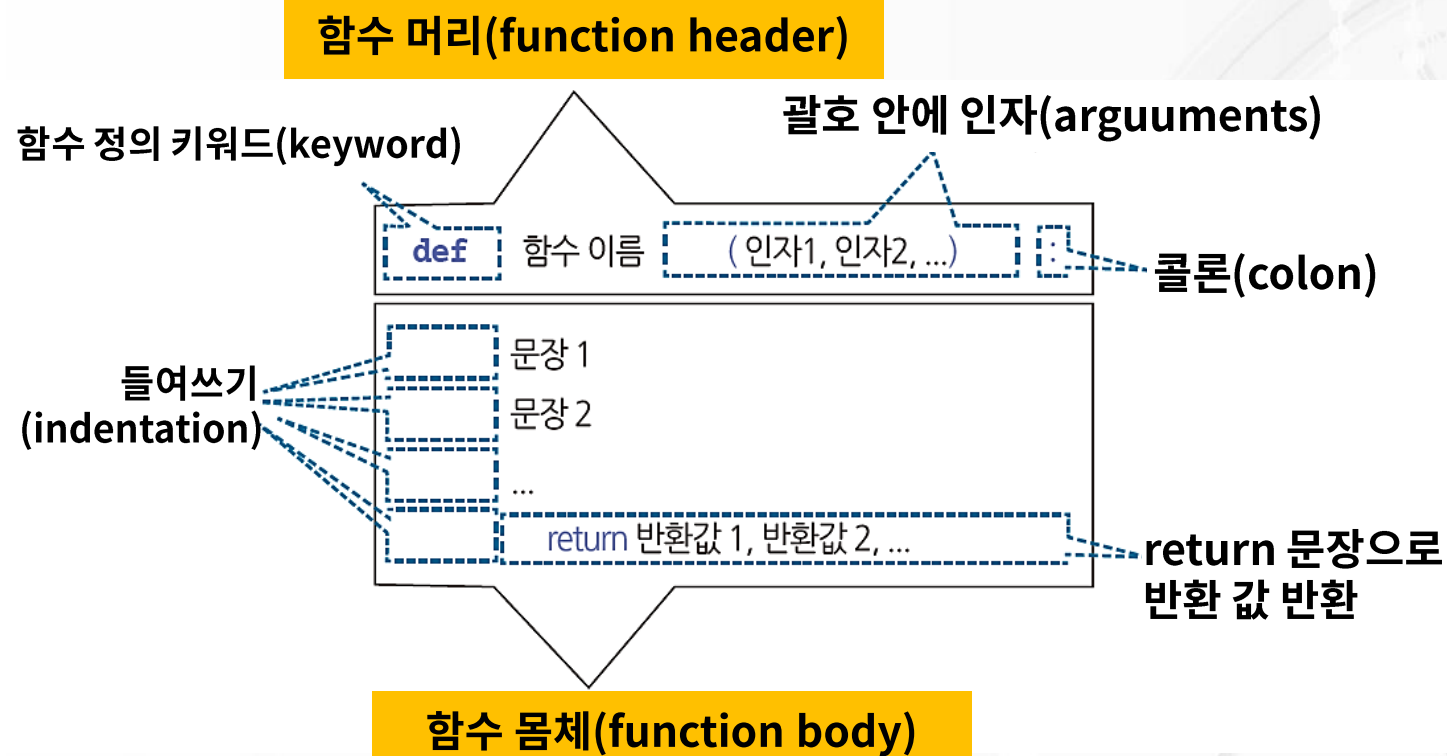
- 함수 이름과 괄호, 괄호 사이의 인자들
- 함수 정의의 몸체인 블록을 예고하는 콜론(:)이 필요

+ 함수 몸체는 들여쓰기로 시작

- 기능을 수행하는 문장들
- 마지막으로 키워드 return에 의한 반환 값을 돌려 주는 구문이 필요

⚠ 파이썬 함수 정의 구문: 함수 이름과 인자, 반환 값

✦ 인자(arguments)와 return 문장은 선택적(optional)



⚠ 함수 정의와 호출

+ 함수 호출은 함수 정의 이후

- 함수 호출 실행: 함수 호출은 함수 정의 이후에만 가능하다.

```
def hello():  
    print('Hello, Python!')
```

```
# 함수 정의(구현)  
def hello():  
    print('Hello, Python!')
```

```
hello() # 함수 호출
```

⚠ 함수 정의와 호출

+ 함수 호출은 함수 정의 이후

- 함수 호출 실행: 함수 호출은 함수 정의 이후에만 가능하다.

```
def hello():  
    print('Hello, Python!')
```

```
# 함수 정의(구현)  
def hello():  
    print('Hello, Python!')
```

```
hello() # 함수 호출
```

[함수 정의 후 함수 호출 가능]

```
hello() # 함수 호출
```

NameError: name 'hello' is not defined

```
# 함수 정의(구현)  
def hello():  
    print('Hello, Python!')
```

⚠ 함수 정의와 호출

+ 함수 호출은 함수 정의 이후

- 함수 호출 실행: 함수 호출은 함수 정의 이후에만 가능하다.

```
1 def helloint(): # 함수 정의
2     print('\t', 77)
3
4 print(1)
5 print(helloint) # 메모리에 저장된 함수 확인
6 helloint() # 함수 호출
7 print(3)
```

실행
결과

```
1
<function helloint at 0x0000028D2271B510>
77
3
```



[예제] 첫 함수 정의와 호출

[코딩실습] 첫 함수 정의와 호출

난이도 기본

```
1 def hello(): # 함수 정의
2     print('\tHello, Python!')
3
4 def helloint(): # 함수 정의
5     print('\t', 77)
6
7 print('처음 하는 함수 호출 hello()')
8 hello() # 함수 호출
9 print('함수 호출 hello() 이후')
10
11 print(1)
12 print(helloint) # 메모리에 저장된 함수 확인
13 helloint() # 함수 호출
14 print(3)
```




[예제] 첫 함수 정의와 호출

[코딩실습] 첫 함수 정의와 호출

난이도 기본

결과

처음 하는 함수 호출 `hello()`

 Hello, Python!

함수 호출 `hello()` 이후

1

<function helloint at 0x0000028D226D6840>

77

3

⚠ 함수로 필요한 값을 전달하는 인자

[코딩실습] 함수 정의에서 인자 사용

난이도 기본

```
1 # 함수 정의
2 def hello(name):
3     # 전달받은 인자 name을 함수 정의에서 사용
4     print('안녕, {}!'.format(name)) # 인자 name에게 인사
5
6 hello('수빈') # 함수 호출, 수빈에게 인사
7 sname = '현수'
8 hello(sname) # 함수 호출, 현수에게 인사
```

결과

안녕, 수빈!
안녕, 현수!

⚠ 함수로 필요한 값을 전달하는 인자

[코딩실습] 인자의 기본값 활용

난이도 기본

```
1 # 인자 name의 기본 값을 '여러분'으로 지정
2 def hello(name = '여러분'):
3     print('안녕, {}'.format(name))
4
5 hello() # name의 기본값 사용
6 hello('현철') # name으로 지정된 '현철' 사용
```

결과

안녕, 여러분!
안녕, 현철!

⚠ return문으로 함수 기능을 수행한 후 반환 값을 전달

```
def hello(name):  
    print('안녕, {}!'.format(name))  
    # 변환 값 전달  
    return '{}에게 인사'.format(name)
```

```
>>> print(hello('희정')) # 희정에게 인사 출력  
안녕, 희정!  
희정에게 인사
```

[다른 실행 결과 1]

```
def hello(name):  
    print('안녕, {}!'.format(name))  
  
>>> print(hello('헤림')) # None 출력  
안녕, 헤림!  
None
```

[다른 실행 결과 2]

```
def hello(name):  
    print('안녕, {}!'.format(name))  
    return  
  
>>> print hello('헤림') # None 출력  
안녕, 헤림!  
None
```

⚠ 인자인 두 수의 합을 반환하는 함수 mysum(x, y)

[코딩실습] 두 수 인자의 합을 변환하는 함수

난이도 기본

```
1 def mysum(x, y = 0): # 인자가 2개
2     """두 수의 합 반환 함수"""
3     return x + y # 두 인자의 합으로 반환
4
5 hap = mysum(5) # y는 기본 값이 0으로
6 print(hap)
7
8 hap = mysum(5, 10)
9 print(hap)
```

결과

5
15

⚠ 인자와 같이 반환 값도 여러 개 가능

```
def getprime():  
    return 2, 3, 5, 7
```

```
>>> a, b, c, d = getprime()  
>>> print(a, b, c, d)  
2 3 5 7  
>>>  
>>> prime = getprime()  
>>> print(prime)  
(2, 3, 5, 7)
```


Chapter 3.

문서화 문자열

P Y T H O N P R O G R A M M I N G



좀 더 알아봅시다!

+ 함수의 해설을 담는 문서화 문자열(docstring)

- 문서화 문자열(docstring)은 함수의 문서화에 도움이 되는 문자열을 말한다.
- 여러 줄에 걸친 문서 내용이 필요하므로 설명 앞뒤에 따옴표 3개를 사용해 문서 내용을 작성한다.
(주로 큰따옴표 3개를 사용)
- 참고로 문서화 문자열은 함수뿐 아니라 다음에 배울 파이썬 모듈, 클래스 및 메소드와 문서를 연결하는 방법도 제공한다.
- 함수 구현에서는 함수 정의 블록 시작 첫 줄에서 문서 내용을 작성한다.
- 다음은 함수 hello() 구현에서 간단히 문서 내용을 작성한 예로, 시각적으로 필요하다면 중간중간에 빈 줄을 삽입한다.

```
def hello(name):
```

```
    """
```

```
    친구 name에게 하는 간단한 인사말 출력
```

```
    인자:
```

```
        name: 인사말을 하는 상대 이름
```

```
    """
```

```
    print('안녕, {}'.format(name))
```



좀 더 알아봅시다!

+ 함수의 해설을 담은 문서화 문자열(docstring)

- 문서화 문자열은 주석의 일환으로 출력 결과에 영향을 미치지 않는다.
- 이전과 같이 문서화 문자열이 있는 함수는 다음과 같이 `help()` 함수를 사용해 그 내용을 확인할 수 있다. 또한 함수의 속성 `__doc__`에도 저장돼 참조할 수 있다.

```
>>> help(hello)
Help on function hello in module __main__:
```

```
hello(name)
  친구 name에게 하는 간단한 인사말 출력
인자 :
      name: 인사말을 하는 상대 이름
```

```
>>> print(hello.__doc__)
친구 name에게 하는 간단한 인사말 출력
인자 :
      name: 인사말을 하는 상대 이름
```

⚠ 섭씨 온도와 화씨 온도의 관계와 변환 함수

[코딩실습] 섭씨 온도와 화씨 온도 간의 변환 함수

난이도 기본

```
1 def ctofahrenheit(cels):
2     """ 인자인 섭씨온도 cels을 화씨 온도로 변환하여 반환 """
3     return cels * 9/5 + 32
4
5 def ftocelsius(fahr):
6     """ 인자인 화씨온도 fahr을 섭씨 온도로 변환하여 반환 """
7     return (fahr - 32) * 5/9
8
9 for cels in range(28, 35, 2):
10    print('섭씨: {}, 화씨: {:.2f}'.format(cels, ctofahrenheit(cels)))
11
12 for fahr in range(90, 103, 3):
13    print('섭씨: {:.2f}, 화씨: {}'.format(ftocelsius(fahr), fahr))
```

⚠ 섭씨 온도와 화씨 온도의 관계와 변환 함수

[코딩실습] 섭씨 온도와 화씨 온도 간의 변환 함수

난이도 기본

결과

섭씨: 28, 화씨: 82.40
섭씨: 30, 화씨: 86.00
섭씨: 32, 화씨: 89.60
섭씨: 34, 화씨: 93.20
섭씨: 32.22, 화씨: 90
섭씨: 33.89, 화씨: 93
섭씨: 35.56, 화씨: 96
섭씨: 37.22, 화씨: 99
섭씨: 38.89, 화씨: 102

SUMMARY



⚠ 함수 개요와 정의 구문

... 함수 정의와 호출: **def**

... 함수 인자

⚠ 인자 전달과 반환

... 문장 **return**: 여러 개 반환도 가능

⚠ 문서화 문자열