



파이썬기초

4주차 - 1교시
함수





학습내용

- 함수



학습목표

- 함수에 대한 이해와 파이썬에서 함수 정의하는 방법에 대해 알아보고 함수를 정의할 수 있다.



생각해 봅시다

반복적으로 사용되는 코드가 있다면
효과적으로 사용할 수 있는 방법에 대해
생각해봅시다.



01



함수



1 | 함수 기본

01 함수(function)

함수

코드의 집합으로, 값을 전달하면 전달 받은 값을
이용하여 약속된 처리 한 후 처리된 결과 또는 약속된
값을 반환해주는 기능

→ 함수를 만들어 놓고 프로그래밍 시에 가져다 사용함

1 | 함수 기본

01 함수(function)

- 파이썬에서의 함수의 구분

사용자 정의 함수

사용자가 프로그램을 작성하면서 필요로 되는 특정 기능을 함수로 작성하여 사용하는 것을 의미

람다함수

이름이 없는 익명함수

내장 함수

파이썬 인터프리터에서 기본적으로 포함하고 있는 함수로 프로그램에서 많이 사용되는 기능을 미리 만들어 놓은 함수를 의미

외장함수

외부의 라이브러리에서 제공하는 함수

1 | 함수 기본

02 함수는 왜 사용할까?

- 하나의 프로그램에서 반복 사용되는 부분을 함수로 작성함으로써, 코드의 크기를 줄이고 재사용할 수 있음
- 공통으로 사용하는 기능이나 반복되는 코드를 재사용할 수 있음

프로그램을 기능 단위로 분리하여 구조화

코드의 가독성을 높이고 유지보수가 용이함

1 | 함수 기본

03 함수의 기본 정의

- 프로그램의 규모가 커질 경우에는 프로그램을 작은 단위로 나누어 여러 개의 프로그램으로 작성하는 것이 원칙임

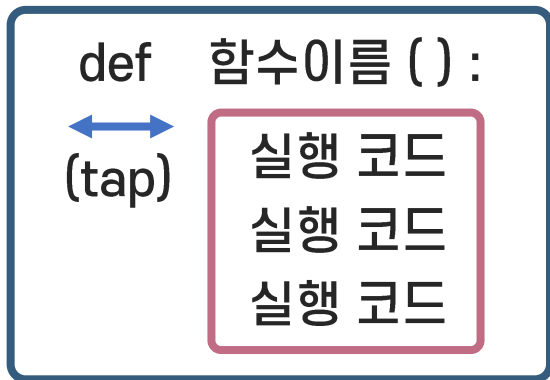
프로그램 작업을
작은 작업들로
분해해서 함수로 작성



이 함수들을 연결해서
프로그램 구성

1 | 함수 기본

04 함수의 정의



- 함수 호출 시 사용하는 식별자, 함수명 다음에는 괄호()가 붙여 사용됨
- 기능을 쉽게 연상할 수 있는 함수명 지정이 바람직함

1 | 함수 기본

04 함수의 정의

- 함수 정의 예제

```
In [3]: def fn():  
        print('fn() 호출 시 출력되는 문장입니다.')        print('함수에서 처리하는 문장들은 들여쓰기를 해줍니다.')  
        print('들여쓰기가 안된 문장은 함수 호출 시 실행되지 않습니다.')        print('종료...')
```

들여쓰기가 안된 문장은 함수 호출 시 실행되지 않습니다.
종료...

1 | 함수 기본

04 함수의 정의

- 함수 정의 예제

```
In [4]: # 함수 호출  
fn()
```

fn() 호출 시 출력되는 문장입니다.
함수에서 처리하는 문장들은 들여쓰기를 해줍니다.

2 | 함수의 매개변수

01 함수의 매개변수

```
def 함수이름 (매개변수 1, 매개변수 2, 매개변수 3, ... ) :  
    ↔ 실행 코드  
(tap) 실행 코드  
      실행 코드
```

- 함수가 호출될 때 전달되는 값들을 저장하는 변수임
- 함수명 뒤의 괄호에 나열해서 함수 내부의 변수들과 구별함
- 매개변수는 함수 호출 시에 값을 전달하게 되므로 함수 내부 내부에서는 바로 사용됨

2 | 함수의 매개변수

01 함수의 매개변수

- 함수 정의 예제

In [5]: #두 수를 받아 더하기 연산 후 결과를 출력하는 함수

```
def add(num1, num2):  
    result = num1 + num2  
    print('전달받은 두 수의 합 : ', result)
```

In [6]: add(100,200)

전달받은 두 수의 합 : 300

2 | 함수의 매개변수

02 매개변수의 종류와 활용



디폴트 매개변수

함수를 호출할 때 매개변수에 값을 입력해주지 않아도
매개변수에 기본값을 사용

- 매개변수 선언 시 “매개변수=값” 형식으로 사용

2 | 함수의 매개변수

02 매개변수의 종류와 활용

- 기본 매개변수 뒤에는 일반 매개변수가 올 수 없음

```
print_n_times(num=2, value) →  
print_n_times("Hello")로 호출
```

→ num이 매개변수인지, value가 매개변수인지
알 수 없음

2 | 함수의 매개변수

02 매개변수의 종류와 활용

- 디폴트 매개변수의 사용 예제

```
In [28]: # 기본 매개변수
def print_nth(str, num=2):
    for i in range(num):
        print(str) # end는 디폴트 매개변수
```

```
In [29]: print_nth('안녕하세요')
```

```
안녕하세요
안녕하세요
```


2 | 함수의 매개변수

02 매개변수의 종류와 활용



키워드 매개변수

함수를 호출할 때 함수 내에 있는 매개변수의 명칭을
직접 지정해서 값을 입력해줄 때 사용

2 | 함수의 매개변수

02 매개변수의 종류와 활용

- 키워드 매개변수의 사용예제

```
In [28]: # 키워드 맵개변수  
print_nth('안녕하세요', num=5)
```

안녕하세요
안녕하세요
안녕하세요
안녕하세요
안녕하세요

2 | 함수의 매개변수

02 매개변수의 종류와 활용

가변 매개변수

매개변수의 개수가 명확하지 않거나 여러 개를 사용해야 하는 경우 사용

```
def 함수이름(매개변수, 매개변수, ... , *가변매개변수):  
    코드...
```

- 가변 매개변수 뒤에는 일반 매개변수를 사용할 수 없음
- 가변 매개변수는 함수에 하나만 사용할 수 있음

2 | 함수의 매개변수

02 매개변수의 종류와 활용

- 가변 매개변수의 사용예제

```
In [18]: # 가변 매개변수
def print_nth_str(num, *strs):
    for i in range(num):
        # 매개변수는 리스트형식으로 사용
        for value in strs:
            print(value, end=' ') # end는 디폴트 매개변수
        # 줄바꿈
        print()
```

2 | 함수의 매개변수

02 매개변수의 종류와 활용

- 가변 매개변수의 사용예제

```
In [19]: print_nth_str(3, 'first', 'coding', 'Python')
```

```
first coding Python  
first coding Python  
first coding Python
```

3 | 함수의 리턴

01 함수의 리턴

```
def 함수이름 (매개변수 1, 매개변수 2, 매개변수 3, ... ) :  
    ↔ 실행 코드  
(tap) 실행 코드  
      실행 코드  
      return(반환 값 또는 연산 식 또는 공백)
```

- return 키워드를 이용하여 함수가 종료 될 때 값을 반환
→ 반환 값 기술 없이 return을 쓰는 경우 값을 반환하지 않고 함수를 종료시킴

3 | 함수의 리턴

02 Return 예제 1

```
In [7]: #두 수를 받아 더하기 연산 후 결과를 출력하는 함수
def add2(num1, num2):
    result = num1 + num2
    return result
```

```
In [8]: add2(100,200)
```

Out [8]: 300

3 | 함수의 리턴

03 Return 예제 2

```
In [9]: def checkAge(age) :  
        if age < 20 :  
            print('성인만 이용 가능한 콘텐츠입니다.')  
            print('프로그램을 종료합니다.')  
            return  
        print('금융상품 정보 게시판을 이용합니다.')  
        print('Q&A 게시판을 이용합니다.')
```


3 | 함수의 리턴

03 Return 예제 2

```
In [10]: checkAge(10)
```

성인만 이용 가능한 콘텐츠입니다.
프로그램을 종료합니다.

```
In [11]: checkAge(30)
```

금융상품 정보 게시판을 이용합니다.
Q&A 게시판을 이용합니다.

4 | 함수에서의 변수활용

01 변수의 사용 범위

변수의 사용 범위(Scoping Rule)

변수가 코드에서 사용되는 범위

4 | 함수에서의 변수활용

02 변수의 사용 범위에 따른 구분



지역 변수(Local Variable)

함수 내부에서 선언되어 함수 내부에서만 사용

- 정상 범위에서 사용 예제

```
In [33]: def fn_local_var1():  
          local_var = "지역 변수"  
          print(local_var)  
          fn_local_var1()
```

지역 변수

4 | 함수에서의 변수활용

02 변수의 사용 범위에 따른 구분



지역 변수(Local Variable)

함수 내부에서 선언되어 함수 내부에서만 사용

- 잘못된 사용 예제

```
In [33]: def fn_local_var1():  
         local_var = "지역 변수"  
         print(local_var)  
         print(local_var)
```

4 | 함수에서의 변수 활용

02 변수의 사용 범위에 따른 구분



지역 변수(Local Variable)

함수 내부에서 선언되어 함수 내부에서만 사용

NameError Traceback (most recent call last)

Cell In[34], line 4

```
2    local_var = "지역 변수"
```

```
3    print(local_var)
```

```
----> 4 print(local_var)
```

NameError: name 'local_var' is not defined

4 | 함수에서의 변수활용

02 변수의 사용 범위에 따른 구분



전역 변수(Global Variable)

함수 외부에서 선언한 변수로 프로그램 전체에서
사용

→ 함수에서 global 키워드로 재 선언하면 함수에서
사용 가능

4 | 함수에서의 변수활용

02 변수의 사용 범위에 따른 구분

- 전역 변수 사용 예제

```
In [47]: def func():  
         global global_var  
         global_var = 'Global Var'  
         local_var = "지역 변수"  
         print(local_var)  
         print(global_var)
```

4 | 함수에서의 변수활용

02 변수의 사용 범위에 따른 구분

- 전역 변수 사용 예제

```
In [47]: global_var = "전역 변수"  
func()  
print(global_var)
```

지역 변수

Global Var

Global Var



파이썬기초

NEXT

람다함수와 예외처리

