

Chapter 07

함수

함수 정의와 호출

예제 7-1. 함수의 정의와 호출

ex7-1.py

```
01 def hello() :  
02     print('안녕하세요.')  
03  
04 hello() → 함수 호출  
05 hello() → 함수 호출  
06 hello() → 함수 호출
```

} 함수 정의

✕ 실행 결과

안녕하세요.
안녕하세요.
안녕하세요.

파이썬의 내장 함수

내장 함수명	기능
print()	화면에 데이터 값을 출력함
input()	키보드를 통해 데이터를 입력받음
range()	정수의 범위를 설정함
list()	리스트를 생성함
abs()	숫자의 절댓값을 구함
len()	문자열, 리스트, 튜플, 딕셔너리 등의 길이를 구함
round()	소수점 이하 반올림 값을 구함. 예) round(2.5)는 3, round(5.2)는 5의 값을 가짐
int()	문자열이나 실수형 숫자를 정수형 숫자로 변환함
float()	문자열이나 정수형 숫자를 실수형 숫자로 변환함

매개 변수란?

서식

```
def 함수명(매개변수1, 매개변수2, ...):
```

```
    문장1
```

```
    문장2
```

```
    ....
```

```
함수명(입력값1, 입력값2, ....)
```

}

함수 정의



함수 호출

매개 변수가 1개인 경우

ex7-2.py

```
def even_odd(num) :  
    if num % 2 == 0 :  
        print('%d은(는) 짝수이다.' % num)  
    else :  
        print('%d은(는) 홀수이다.' % num)  
  
even_odd(7)  
even_odd(16)
```

:: 실행 결과

7은(는) 홀수이다.
16은(는) 짝수이다.

매개 변수가 여러 개인 경우

ex7-3.py

```
def favorite_color(name, color, amount) :  
    if (amount == 1) :  
        print('%s님은 %s을 좋아하지 않습니다.' % (name, color))  
    elif (amount == 2) :  
        print('%s님은 %s을 조금 좋아합니다.' % (name, color))  
    else :  
        print('%s님은 %s을 매우 좋아합니다.' % (name, color))
```

```
favorite_color('김지영', '빨강', 1)
```

```
favorite_color('홍채영', '노랑', 2)
```

```
favorite_color('진소진', '파랑', 3)
```

:: 실행 결과

김지영님은 빨강을 좋아하지 않습니다.
홍채영님은 노랑을 조금 좋아합니다.
진소진님은 파랑을 매우 좋아합니다.

매개 변수 *args

ex7-4.py

```
def average(*scores) :  
    sum = 0  
    for i in range(len(scores)) :  
        sum += scores[i]  
    avg = sum/len(scores)  
    print('%d과목의 평균 : %.2f' % (len(scores), avg))  
  
average(80, 90, 100)  
average(75, 80, 94, 78)  
average(80, 73, 76, 86, 82)
```

:: 실행 결과

```
3과목의 평균 : 90.00  
4과목의 평균 : 81.75  
5과목의 평균 : 79.40
```

파이썬의 값에 의한 호출

ex7-5.py

```
# 서브 루틴 : func() 함수
def func(x) :
    x = 100
    print('func() : x = ', x, ', id =', id(x))

# 메인 루틴
x = 10
print('메인 : x = ', x, ', id =', id(x))
func(x)
print('메인 : x = ', x, ', id =', id(x))
```

:: 실행 결과

```
메인 : x = 10 , id = 140719703827120
func() : x = 100 , id = 140719703830000
메인 : x = 10 , id = 140719703827120
```


파이썬의 레퍼런스에 의한 호출

ex7-6.py

```
# 서브 루틴 : func() 함수
def func(x) :
    x[0] = 100
    print('func() : x = ', x, ', id =', id(x))

# 메인 루틴
x = [1, 2, 3]
print('메인 : x = ', x, ', id =', id(x))
func(x)
print('메인 : x = ', x, ', id =', id(x))
```

:: 실행 결과

```
메인 : x = [1, 2, 3] , id = 2768370614984
func() : x = [100, 2, 3] , id = 2768370614984
메인 : x = [100, 2, 3] , id = 2768370614984
```

함수 값의 반환

ex7-7.py

```
def circle_area(r) :  
    area = r * r * 3.14  
    return area  
  
radius = int(input('원의 반지름을 입력하세요 : '))  
result = circle_area(radius)  
print('반지름 : %d, 원의 면적 : %.2f' % (radius, result))  
  
radius = int(input('원의 반지름을 입력하세요 : '))  
result = circle_area(radius)  
print('반지름 : %d, 원의 면적 : %.2f' % (radius, result))
```

:: 실행 결과

```
원의 반지름을 입력하세요 : 10  
반지름 : 10, 원의 면적 : 314.00  
원의 반지름을 입력하세요 : 20  
반지름 : 20, 원의 면적 : 1256.00
```

Q7-1 함수로 정수 합계 구하기

:: 실행 결과

10 ~ 100의 정수 합계 : 5005
100 ~ 1000의 정수 합계 : 495550
1000 ~ 10000의 정수 합계 : 49505500

```
def sum(①_____, ②_____):  
    ③_____ = 0  
    for i in range(start, end+1):  
        ④_____ += i  
    print('%d ~ %d의 정수 합계 : %d' % (start, end, total))  
  
sum(10, 100)  
sum(100, 1000)  
sum(1000, 10000)
```

Q7-2 함수로 배수 합계 구하기

:: 실행 결과

시작 수를 입력하세요: 10
끝 수를 입력하세요: 100
합계를 구할 배수를 입력하세요: 5
10 ~ 100의 정수 중 5의 배수의 합 : 1045

```
def sum_besu(①_____, ②_____, num):  
    sum = 0  
    for i in range(n1, n2+1):  
        if i % ③_____ == 0:  
            sum += i  
    return sum
```

```
start = int(input('시작 수를 입력하세요: '))  
end = int(input('끝 수를 입력하세요: '))  
besu = int(input('합계를 구할 배수를 입력하세요: '))  
  
result = sum_besu(start, end, ④_____)  
  
print('%d ~ %d의 정수 중 %d의 배수의 합 : %d' % (start, end,  
besu, result))
```

Q7-3 함수로 최대공약수 구하기

:: 실행 결과

첫 번째 수를 입력하세요: 33
두 번째 수를 입력하세요: 44
33과(와) 44의 최대공약수 : 11

```
def computeMaxGong(①_____, ②_____):  
    if x > y:  
        small = y  
    else:  
        small = x
```

```
for i in range(1, small+1):
```

```
    if((x % ③_____ == 0) and (y % ③_____ == 0)):
```

```
        result = i    return ④_____
```

```
num1 = int(input("첫 번째 수를 입력하세요: "))
```

```
num2 = int(input("두 번째 수를 입력하세요: "))
```

```
max_gong = computeMaxGong(num1, num2)
```

```
print('%d과(와) %d의 최대공약수 : %d' % (num1, num2,  
⑤_____))
```

람다 함수

ex7-8.py

```
x = lambda a : a**2
```

```
print(x(5))
```

```
print(x(10))
```

:: 실행 결과

25

100

람다 함수 사용 예

ex7-9.py

```
f = lambda x, y, z : x + y + z
```

```
print(f(10, 20, 30))
```

```
def mul(n) :
```

```
    return lambda x : x * n
```

```
g = mul(3)
```

```
h = mul(5)
```

```
print(g(10))
```

```
print(h(10))
```

:: 실행 결과

60

30

50

지역 변수와 전역 변수

- 지역 변수(Local Variable) : 호출된 함수 내에서만 유효
- 전역 변수(Global Variable) : 하위의 모든 함수에서 유효

지역 변수 사용 시의 오류

ex7-10.py

```
def func() :  
    x = 10  
    print(x)  
  
func()  
print(x)
```

:: 실행 결과

10

```
-----  
-----  
NameError Traceback (most recent call last)  
<ipython-input-1-77d987d7788b> in  
<module>  
5  
6 func()  
----> 7 print(x)
```

NameError: name 'x' is not defined

전역 변수의 사용 예

ex7-11.py

```
def func() :  
    print(x)  
    print(id(x))  
  
x = 10  
print(x)  
print(id(x))  
func()
```

:: 실행 결과

```
10  
140719703827120  
10  
140719703827120
```

전역 변수의 값의 변경

ex7-12.py

```
def func() :  
    x = 100  
    print(x)  
    print(id(x))
```

```
x = 10  
print(x)  
print(id(x))
```

```
func()
```

```
print(x)
```

```
print(id(x))
```

:: 실행 결과

```
10  
140719703827120  
100  
140719703830000  
10  
140719703827120
```

키워드 global

ex7-13.py

```
def func() :  
    global x  
    x = 100  
    print(x)  
    print(id(x))
```

```
x = 10  
print(x)  
func()  
print(x)  
print(id(x))
```

:: 실행 결과

```
10  
100  
140719703830000  
100  
140719703830000
```

파일 쓰기

ex7-14.py

```
file = open('sample.txt', 'w', encoding='utf8')  
file.write('안녕하세요. 반갑습니다.')  
file.close()  
print('파일 쓰기 완료!')
```

:: 실행 결과

파일 쓰기 완료!

Open() 함수의 파일모드

파일 모드	설명
r	읽기 모드 : 파일을 읽을 때 사용
w	쓰기 모드 : 파일에 내용을 쓸 때 사용 ※ 해당 파일이 존재하지 않으면 새로운 파일을 열고, 해당 파일이 존재 하면 파일을 쓸 때 기존 파일의 내용에 덮어씀
a	추가 모드 : 기존의 파일에 새로운 내용을 추가할 때 사용

리스트를 파일로 저장

ex7-15.py

```
scores = ['김소영 82 80 93 97 93 88',  
          '정예린 86 100 93 86 90 77',  
          '이세영 91 88 99 79 92 68',  
          '정수정 86 100 93 89 92 93',  
          '박지수 80 100 95 89 90 84']  
  
data = ""  
for item in scores :  
    data += item + '\n'  
  
# 화면 출력하기  
print(data)
```

```
# 파일(scores.txt)에 저장하기  
file = open('scores.txt', 'w', encoding='utf8' )  
file.write(data)  
file.close()
```

:: 실행 결과

```
김소영 82 80 93 97 93 88  
정예린 86 100 93 86 90 77  
이세영 91 88 99 79 92 68  
정수정 86 100 93 89 92 93  
박지수 80 100 95 89 90 84
```

파일 읽기

ex7-16.py

```
file = open('scores.txt', 'r', encoding='utf8')
lines = file.readlines()

print('scores.txt 파일의 내용 : ')
for line in lines :
    print(line, end=")

file.close()
```

:: 실행 결과

scores.txt 파일의 내용 :
김소영 82 80 93 97 93 88
정예린 86 100 93 86 90 77
이세영 91 88 99 79 92 68
정수정 86 100 93 89 92 93
박지수 80 100 95 89 90 84

Q7-4 함수로 만드는 영어단어 퀴즈

:: 실행 결과

오렌지에 맞는 영어 단어는? orange
맞습니다!
과자에 맞는 영어 단어는? cookie
맞습니다!
어머니에 맞는 영어 단어는? mother
틀렸습니다!
형제에 맞는 영어 단어는? brother
맞습니다!
파이썬에 맞는 영어 단어는? python
틀렸습니다!

```
def matchWord(①_____, answer) :
```

```
    if word == answer :
```

```
        msg = '맞습니다!'
```

```
    else :
```

```
        msg = '틀렸습니다!'
```

```
    return ②_____
```

```
eng_dict = {'orange':'오렌지', 'cookie':'과자', 'mother':'어머  
니', 'brother':'형제', 'python':'파이썬'}
```

```
for ③_____ in eng_dict :
```

```
    string = input(eng_dict[key] + '에 맞는 영어 단어는? ')
```

```
    result = matchWord(string, ④_____)
```

```
    print(result)
```

Q7-5 함수로 세 수 중 큰 수 찾기

:: 실행 결과

첫 번째 수를 입력하세요: 33
두 번째 수를 입력하세요: 77
세 번째 수를 입력하세요: -21
33, 77, -21 중 가장 큰 수 : 77

```
def maxTwo(i, j):
```

```
    if i > j:
```

```
        return ❶_____
```

```
    else :
```

```
        return ❷_____
```

```
def maxThree(x, y, z) :
```

```
    max1 = maxTwo(x, y)
```

```
    max2 = maxTwo(y, z)
```

```
    if max1 > max2 :
```

```
        largest = max1
```

```
    else :
```

```
        largest = max2    return ❸_____
```

```
a = int(input('첫 번째 수를 입력하세요: '))
```

```
b = int(input('두 번째 수를 입력하세요: '))
```

```
c = int(input('세 번째 수를 입력하세요: '))
```

```
max_num = ❹_____(a, b, c)
```

```
print('%d, %d, %d 중 가장 큰 수 : %d' % (a, b, c, max_num))
```

Q7-6 파일에서 합계/평균 구하기

scores.txt 파일

```
김소영 82 80 93 97 93 88
정예린 86 100 93 86 90 77
이세영 91 88 99 79 92 68
정수정 86 100 93 89 92 93
박지수 80 100 95 89 90 84
```

:: 실행 결과

```
-----
김소영
합계 : 445, 평균 : 89.00
```

```
-----
정예린
합계 : 455, 평균 : 91.00
```

```
-----
이세영
합계 : 449, 평균 : 89.80
```

```
-----
정수정
합계 : 460, 평균 : 92.00
```

```
-----
박지수
합계 : 454, 평균 : 90.80
-----
```

Q7-6 파일에서 합계/평균 구하기(계속)

```
file = ❶_____('scores.txt', 'r', encoding='utf8')
lines = file.readlines() file.close()
print('-' * 50)

for line in lines :
    student = line.split()

    i = 0
    sum = 0
```

```
while i < ❷_____:
    if i == 0 :
        print(❸_____)
    else :
        ❹_____ += int(student[i])
    i += 1
print('합계 : %d, 평균 : %.2f' % (sum, sum/5))
print('-' * 50)
```