

사물인터넷융합기술(51) 10월 9일 수업 과제

제출자: 경영학부 2018312021 유창연

```
# p. 330 중간 점검
# 1. 함수란 특정 작업을 수행하는 명령어들의 모음에 이름 붙인 것.
# 2. 함수를 이용하면 반복이 필요한 코드를 재할용할 수 있다.
```

```
# p. 340 중간 점검
# 1. 함수에 전달하는 값을 인수라고 한다.
# 2. 함수 안에서 전달되는 값을 받는 변수를 매개변수라고 한다.
def get_input():
    x = int(input('첫 번째 정수'))
    y = int(input('두 번째 정수'))
    return x, y

get_input()
(1, 2)
```

```
# p. 343 중간 점검
def get_rect_area(w, h):
    return w*h

def main():
    w = 2
    h = 3
    result = get_rect_area(w, h)
    print(f"너비가 {w}, 높이가 {h}인 사각형의 면적: {result}")

main()
너비가 2, 높이가 3인 사각형의 면적: 6
```

```
# p. 344
def main() :
    print("20cm 피자 2 개의 면적:", get_area(20)+get_area(20))
    print("30cm 피자 1 개의 면적:", get_area(30))

def get_area(radius):
    if radius > 0:
        area = 3.14*radius**2
    else:
```

```

        area = 0
    return area

main()

20cm 피자 2개의 면적: 2512.0
30cm 피자 1개의 면적: 2826.0

```

p. 358 중간 점검

1. 인수는 호출 프로그램에 의하여 함수에 실제로 전달되는 값이다. 매개 변수는 이 값을 전달받은 변수이다.

2. 디폴트 인수는 기본값을 넣어주는 메커니즘과 같다. `axis = 1`

3. 키워드 인수는 인수의 이름을 명시적으로 지정해서 값을 매개변수로 전달하는 방법이다.

4. 매개변수 앞의 `*`는 가변 길이 인수를 함수에 전달할 수 있다.

```

# p. 359
def display(msg, count=1):
    for k in range(count):
        print(msg)

display("환영합니다.", 5)

```

```

환영합니다.
환영합니다.
환영합니다.
환영합니다.
환영합니다.

```

```

# p. 361
def f(x):
    return(x**2-x-1)

def bisection_method(a, b, error):
    if f(a)*f(b) > 0:
        print("구간에서 근 찾을 수 없다.")
    else:
        while (b-a)/2.0 > error:
            midpoint = (a+b)/2
            if f(midpoint) == 0:
                return(midpoint)
            elif f(a)*f(midpoint) < 0:
                b = midpoint
            else: a = midpoint

```

```

        return(midpoint)

ans = bisection_method(1, 2, 0.0001)

print("x**2-x-1의 근:", ans)
x**2-x-1의 근: 1.6180419921875

```

```

# p. 363
def weeklyPay(rate, hour):
    money = 0
    if (hour > 30):
        money = rate*30 + 1.5*rate*(hour-30)
    else:
        money = rate*hour
    return money

rate = int(input("시급 입력:"))
hour = int(input("근무 시간 입력:"))
print("주급:", str(weeklyPay(rate, hour)))
주급: 420000.0

```

```

# p. 367 중간 점검
# 1. 값을 반환하는 return
def addsub(x, y):
    pl = x+y
    mi = x-y

    return pl, mi

addsub(3, 1)
(4, 2)

```

```

# p. 368
def get_info():
    name = input("이름")
    age = int(input("나이:"))
    return name, age

st_name, st_age = get_info()
print("이름:", st_name, " 나이:", st_age)

```

이름: 유창연 나이: 23

```
# p. 371
def menu():
    print("1. 섭씨온도를 화씨온도로")
    print("2. 화씨온도를 섭씨온도로")
    print("3. 종료")
    selection = int(input("메뉴를 선택하세요: "))
    return selection

def ctof(c):
    temp = c*9/5+32
    return temp

def ftoc(f):
    temp = (f-32)*5/9
    return temp

def input_f():
    f = float(input("화씨온도 입력"))
    return f

def input_c():
    c = float(input("섭씨온도 입력"))
    return c

def main():
    while True:
        index = menu()
        if index == 1:
            t = input_c()
            t2 = ctof(t)
            print("화씨온도 =", t2, " ")
        elif index == 2:
            t = input_f()
            t2 = ftoc(t)
            print("섭씨온도 =", t2, " ")
        else:
            break
    main()
```

```
1. 섭씨온도를 화씨온도로
2. 화씨온도를 섭씨온도로
3. 종료
화씨온도 = 98.6
1. 섭씨온도를 화씨온도로
2. 화씨온도를 섭씨온도로
3. 종료
```

```
# p. 382 중간 점검
# 1. 지역변수는 함수마다 동일한 이름을 사용할 수 있다. 전역변수는 함수 외부에서
  생성된 변수이다.
# 2. 읽을 수 있다.
# 3. global 을 사용하지 않는 이상 전역변수 값은 변하지 않는다.
```

```
# p. 410
stdnts = 5
list = []
count = 0

for i in range(stdnts):
    value = int(input("성적 입력"))
    list.append(value)

print("성정평균 =", sum(list)/len(list))
print("최대점수 =", max(list))
print("최소점수 =", min(list))

for score in list:
    if score >= 80:
        count += 1
print("80 점 이상:", count)

성정평균 = 48.0
최대점수 = 80
최소점수 = 10
80점 이상: 1
```

```
# p. 412
list = [1, 2, 3, 4, 15, 99]

list.sort()

print("두 번째로 큰 수:", list[-2])
```

두 번째로 큰 수: 15

```
# p. 414
point = [10, 9, 8.3, 7.1, 3, 9]
print("제거전", point)
point.remove(max(point))
point.remove(min(point))
print("제거후", point)
```

제거전 [10, 9, 8.3, 7.1, 3, 9]
제거후 [9, 8.3, 7.1, 9]

```
# p. 416
stack = []

for i in range(3):
    f = input("과일 입력")
    stack.append(f)
```

```
for i in range(3):
    print(stack.pop())

grape
orange
apple
```

```
# p. 418
menu = 0
friends = []
while menu != 9:
    print("-----")
    print("1. 친구 리스트 출력")
    print("2. 친구추가")
    print("3. 친구삭제")
    print("4. 이름변경")
    print("9. 종료")
    menu = int(input("메뉴를 선택하시오: "))

    if menu == 1:
        print(friends)
    elif menu == 2:
        name = input("이름을 입력하시오: ")
    elif menu == 3:
```

```

del_name = input("삭제할 이름을 입력하시오: ")
if del_name in friends:
    friends.remove(del_name)
else:
    print("이름이 발견되지 않음")
elif menu == 4:
    old_name = input("변경할 이름을 입력하시오")
    if old_name in friends:
        index = friends.index(old_name)
        new_name = input("새로운 이름 입력")
        friends[index] = new_name
    else:
        print("이름이 발견되지 않음")

```

```

-----
1. 친구 리스트 출력
2. 친구추가
3. 친구삭제
4. 이름변경
9. 종료

```

```

-----
1. 친구 리스트 출력
2. 친구추가
3. 친구삭제
4. 이름변경
9. 종료
이름이 발견되지 않음

```

```

-----
1. 친구 리스트 출력
2. 친구추가
3. 친구삭제
4. 이름변경
9. 종료

```

```

# p. 434 중간 점검
# 1. 함수란 특정 작업을 수행하는 명령어들의 모음에 이름 붙인 것.
# 2. 함수를 이용하면 반복이 필요한 코드를 재할용할 수 있다.

```

```

# p. 436
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
reversed = numbers[::-2]
print(reversed)

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
numbers[1:] = []
print(numbers)

```

```
[10, 8, 6, 4, 2]
[1]
```

```
# p. 442
salaries = [200, 250, 300, 280, 500]

def modify(values, factor):
    for i in range(len(values)):
        values[i] = values[i]*factor

print("인상전", salaries)
modify(salaries, 1.3)
print("인상후", salaries)

인상전 [200, 250, 300, 280, 500]
인상후 [260.0, 325.0, 390.0, 364.0, 650.0]
```

```
# p. 447
nums = [x for x in range(100) if x % 2 == 0 and x % 3 == 0]
print(nums)

[0, 6, 12, 18, 24, 30, 36, 42, 48, 54, 60, 66, 72, 78, 84, 90, 96]
```

```
# p. 449
list1 = [10, 20, 30, 40, 50]
list2 = [sum(list1[0:x+1]) for x in range(0, len(list1))]

print("원래 리스트", list1)
print("새로운 리스트", list2)

원래 리스트 [10, 20, 30, 40, 50]
새로운 리스트 [10, 30, 60, 100, 150]
```

```
# p. 451
[(x, y, z) for x in range(1, 30) for y in range(x, 30) for z in range(y, 30)
if x**y + y**2 == z**2]

[(2, 6, 10), (3, 3, 6)]
```