

# 중요사항 정리

## 함수

함수는 우리가 반복적으로 사용하는 코드를 묶은 것. 코드의 덩어리  
객체는 코드 중에서 독립적인 단위로 분리할 수 있는 조각이다.  
모듈은 프로그램의 일부를 가지고 있는 독립적인 파일이다.

## def 예약어

def 함수이름(매개변수):

실행내용

함수이름(인자) - 호출 시

이런식으로 함수를 작성하고 함수를 호출할 때는 함수의 이름을 사용하여 호출하면 된다.

함수를 작성해놓으며 필요할 때 언제든지 불러 사용가능하다.

함수 호출시 전달되는 실제 값을 인자라고 하고, 함수 내에서 전달받는 변수를 매개변수라고 한다. 여러개의 인자와 매개변수를 동시에 사용할 때는 매개변수의 개수와 인자의 개수가 정확히 일치하여야 한다.

키워드 인자를 사용하면 여러개의 인자를 입력할 때 혼돈을 방지할 수 있다. 인자는 키워드 인자를 사용하기 전에 사용해야 오류를 방지 할 수 있다.

## 전역변수 지역변수

전역변수: 함수 외에서 선언된 함수로 전체에서 사용된다.

지역변수: 함수 내에서 선언된 함수로 함수 내에서 사용된다.

## 재귀함수

함수 내부에서 자기 자신을 호출하는 함수

## 모듈

모듈이란 파이썬 함수나 변수 또는 클래스들을 모아놓은 스크립트 파일을 말한다.

## 자기점검

6.7문제에서 재귀함수를 사용하지 않고 for문을 사용하여 n!을 구하는 방법을 모르겠습니다.

## 심화문제 풀이

### 6.1

```
In [2]: def square(n):
```

```
    return n**2
print('3의 제곱은 : ',square(3))
print('4의 제곱은 : ',square(4))
```

3의 제곱은 : 9  
4의 제곱은 : 16

## 6.2

In [7]:

```
def max2(m, n):
    if m > n:
        return m
    else:
        return n

def min2(m, n):
    if m < n:
        return m
    else:
        return n

print('100과 200중 큰 수는: ', max2(100, 200))
print('100과 200중 작은 수는: ', min2(100, 200))
```

100과 200중 큰 수는: 200  
100과 200중 작은 수는: 100

## 6.3

In [13]:

```
n1, n2, n3 = map(int, input('3 수를 입력하시오 :').split())

def max3(n1, n2, n3):
    if n1 >= n2:
        if n2 >= n3:
            return n1
        else:
            if n1 < n3:
                return n3
    else:
        if n1 >= n3:
            return n2
        else:
            if n2 < n3:
                return n3

def min3(n1, n2, n3):
    if n1 <= n2:
        if n2 <= n3:
            return n1
        else:
            if n1 > n3:
                return n3
    else:
        if n1 <= n3:
            return n2
        else:
            if n2 > n3:
                return n3

Max = max3(n1, n2, n3)
Min = min3(n1, n2, n3)
```

```
print('가장 큰 수 :', Max)
print('가장 작은 수 :', Min)
```

가장 큰 수 : 66  
가장 작은 수 : 20

## 6.4

```
In [35]: n1, n2, n3 = map(int, input("3 수를 입력하시오 : ").split())

def max_and_min(n1, n2, n3) :
    if n1 > n2 and n1 > n3 :
        if n2 > n3 :
            return n1, n3
        else :
            return n1, n2

    elif n2 > n1 and n2 > n3 :
        if n1 > n3 :
            return n2, n3
        else :
            return n2, n1

    else :
        if n1 > n2 :
            return n3, n2
        else :
            return n3, n1

Max, Min = max_and_min(n1, n2, n3)
print("가장 큰 수 : ", Max)
print("가장 작은 수 : ", Min)
```

가장 큰 수 : 66  
가장 작은 수 : 20

## 6.5

```
In [19]: def cel2fah(ce1):
    fah = (9/5) * ce1 + 32
    return fah

for cel in range(0,51,10):
    temp = cel2fah(ce1)
    print(temp)
```

32.0  
50.0  
68.0  
86.0  
104.0  
122.0

## 6.6

```
In [37]: def is_prime(x) :
    for i in range(2, x) :
        if num % i == 0 :
            return False
```

```

        else :
            continue
    return True

num = int(input('소수 여부를 검사할 정수를 입력하시오 : '))
print('소수인가요? :', is_prime(num))

```

소수인가요? : True

## 6.7

In [34]:

```

def factorial(n):
    for i in range(n,2,-1):
        i = i * (i+1)
    return i
print(factorial(5))
print(factorial(7))
print(factorial(10))

```

12  
12  
12

## 6.8

In [36]:

```

def distance(x1, y1, x2, y2) :
    distance = ((x1 - x2) ** 2 + (y1 - y2) ** 2) ** (1/2)
    return distance

x1 = float(input('x1 : '))
y1 = float(input('y1 : '))
x2 = float(input('x2 : '))
y2 = float(input('y2 : '))

A = (x1, y1)
B = (x2, y2)
print(A, '과', B, '사이의 거리는', distance(x1, y1, x2, y2))

```

(4.0, 2.0) 과 (2.0, 1.0) 사이의 거리는 2.23606797749979