

8. range() 함수

Python 은 range() 함수를 built-in 으로 제공한다.

range() 함수는 파라미터 (parameter) 로 받은 숫자 범위의 연속된 정수를 반환함.

문법

```
range([start,] stop [, increment] )
```

list() 함수는 range() 함수의 반환값을 list 로 변환해 줌.

for-loop 문에서 range() 함수는 list 를 반환한다.

In [1]:

```
1 list(range(10))
```

Out[1]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [2]:

```
1 list(range(0, 10, 1))
```

Out[2]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

In [3]:

```
1 list(range(1, 10, 2))
```

Out[3]:

```
[1, 3, 5, 7, 9]
```

In [4]:

```
1 list(range(2, 10, 2))
```

Out[4]:

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

In [5]:

```
1 list(range(5, 5))
```

Out[5]:

```
[]
```

In [6]:

```
1 list(range(5, 0, -1))
```

Out[6]:

```
[5, 4, 3, 2, 1]
```

In [7]:

```
1 for i in range(5):  
2     print(i)
```

```
0  
1  
2  
3  
4
```

In [8]:

```
1 for i in range(5, -1, -1):  
2     print(i)
```

```
5  
4  
3  
2  
1  
0
```

In [9]:

```
1 order = ['1st', '2nd', '3rd', '4th', '5th']  
2 for i in range(len(order)):  
3     print(i, order[i])
```

```
0 1st  
1 2nd  
2 3rd  
3 4th  
4 5th
```

In [10]:

```
1 for i in range(len(order)-1, -1, -1):  
2     print(i, order[i])
```

```
4 5th  
3 4th  
2 3rd  
1 2nd  
0 1st
```

In [11]:

```
1 reversed(order)
```

Out[11]:

<list_reverseiterator at 0x2242c693ac8>

In [12]:

```
1 list(enumerate(reversed(order)))
```

Out[12]:

[(0, '5th'), (1, '4th'), (2, '3rd'), (3, '2nd'), (4, '1st')]

In [13]:

```
1 for i, ord in enumerate(reversed(order)):
2     print(i, ord)
```

0 5th
1 4th
2 3rd
3 2nd
4 1st

In [14]:

```
1 for i in range(0, len(order), 2):
2     print(i, order[i])
```

0 1st
2 3rd
4 5th

In [15]:

```
1 sum = 0
2 for i in range(1, 11):
3     sum = sum + i
4     print(sum)
```

55

In [16]:

```
1 for i in range(5):
2     print("Hello, world!")
```

Hello, world!
Hello, world!
Hello, world!
Hello, world!
Hello, world!

tuple 과 for-loop

In [17]:

```
1 xtupple = (1, 'two', 2)
```

In [18]:

```
1 for e in xtupple:  
2     print(e)
```

```
1  
two  
2
```

In [19]:

```
1 for i in range(len(xtupple)):  
2     print("t["+i+"] = ", xtupple[i])
```

```
t[ 0 ] = 1  
t[ 1 ] = two  
t[ 2 ] = 2
```

연습문제

1) 다음 문장 수행 후의 output 은 ?

```
xlist = [1, [1, 2], [1, 2, 3]]  
print(xlist[1][1] + 1)
```

2) 다음 문장 수행 후의 output 은 ?

```
s = ("a", "b", "c")  
for i in range(1, len(s) + 1):  
    sub = ""  
    for j in range(i):  
        sub = s[j] + sub  
    print(sub)
```

3) 다음 문장 수행 후의 output 은 ?

```
def sum_part(xlist, n):  
    sum = 0  
    for x in xlist[n]:  
        sum = sum + x  
    return sum  
  
ylist = [[1, 2], [3, 4], [5, 6], [7, 8]]  
x = sum_part(ylist, 2)  
print(x)
```

4) 숫자로 피라미드 만들기 : 홀수 숫자를 입력으로 받아서 좌우 대칭되는 피라미드형태로 출력 한다.

- python 의 for 문과 range 함수에 대한 활용 능력
- **(Hint)** pyramid 를 만들어야 하므로 입력 받은 숫자 만큼 for loop n 회 반복 → for i in range(n):
 - 좌우 대칭을 만들기 위해 처음 시작은 (n-i) 개의 space print
 - 줄 바꿈 없이 1 에서 i+1 까지 print
 - 줄 바꿈 없이 i 에서 1 까지 역순으로 print
 - 줄 바꿈

In [27]:

```
1 def pyramid(n):
2     for i in range(n):
3         print(" " * (n-i), end="")
4         for j in range(?, ?):
5             print(j, end="")
6         for j in range(?, ?, ?):
7             print(j, end="")
8         print()
9
10 pyramid(7)
```

```
      1
     121
    12321
   1234321
  123454321
 12345654321
1234567654321
```

5) 임의의 범위의 숫자를 모두 곱하는 함수를 작성하라.

ex) multiply(2,4) ==> 2 * 3 * 4 = 24

6) 숫자로 이루어진 list 의 평균을 구하는 함수를 작성하라. 단, built-in 함수를 이용하지 않고 for loop 을 이용하여 새로운 함수 작성.

ex) average([2,3,4,5,6]) => 4.0

7) x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] 을 for 문을 이용하여,

```
[1,2,3] , 4
[2,3,4] , 5
[3,4,5] , 6
[4,5,6] , 7
[5,6,7] , 8
[6,7,8] , 9
[7,8,9] , 10 을 출력하라
```

(Hint) [7, 8, 9], 10 을 만들기 위해 뒤에서 4 개를 남겨야 하므로 range 함수의 범위를 range(len(x) - 3) 으로 한다. 따라서,

```
for i in range(len(x)-3):  
    print(x[?: ?], ',', x[?])
```

도전문제

피보나치 수열(Fibonacci Sequence)을 계산하는 프로그램도 파이썬으로 간단히 작성할 수 있다.

피보나치 수열은 0 과 1 로 시작하고 다음의 숫자는 이전 숫자 두개를 더한 숫자들로 이루어 진다.
0, 1, 1, 2, 3, 5, 8, 13

$$F_n = \begin{cases} 0 & n = 0 \\ 1 & n = 1 \\ F_{n-1} + F_{n-2} & n > 1 \end{cases}$$

n 개의 숫자로 이루어진 피보나치 수열을 출력하는 함수를 작성하라.

(결과) fib(10) => [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

(Hint)

- old, new 두개의 변수 이용
- 초기값 : old = 0, new = 1
- next 숫자는 old + new
- new 값이 old 가 되고 next 숫자가 new 값이 된다.