

## 10. 스트링 (String)

### 10-1. 스트링 (String) 자료형과 string 함수

Python 의 string 자료형은 text 형태의 문자열을 말함.

문자열은 " ", ' ' 혹은 """ """ 로 둘러싸인 문자들의 집합임.

```
"Life is short"
'a'
"123"
"""String is Python's text type."""
```

**string 자료는 수정할 수 없음 (immutable)**

- 수정할 수 있는 자료형 (mutable) list
- 수정할 수 없는 자료형 (immutable) string, tuple

**string 간의 +, \* 연산 가능**

**str( ) 내장함수로 string type 변환**

**string 은 iterable 한 특성을 가지므로 list 와 같이 for-loop, len( ), indexing, slicing 가능**

**string 과 list 비교**

In [1]:

```
1 xlist = ['g','o','o','d']
2 xlist[1] = 'x'
```

In [2]:

```
1 xlist
```

Out[2]:

```
['g', 'x', 'o', 'd']
```

In [3]:

```
1 s1 = 'good'
2 s1[1] = 'x'
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-3-6ef7a15e612e> in <module>
      1 s1 = 'good'
----> 2 s1[1] = 'x'
```

**TypeError:** 'str' object does not support item assignment

## string operation

In [4]:

```
1 s2 = "Good" + " morning"
2 print(s2)
```

Good morning

In [5]:

```
1 s3 = "good" * 3
2 print(s3)
```

goodgoodgood

In [6]:

```
1 s4 = "Number" + 3
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-6-27bcd2bebc18> in <module>
----> 1 s4 = "Number" + 3
```

**TypeError:** can only concatenate str (not "int") to str

In [7]:

```
1 s4 = "Number" + str(3)
2 s4
```

Out[7]:

'Number3'

In [8]:

```
1 s5 = "Python is powerful"
2 print(s5[0], s5[-1], s5[:3])
```

P I Pyt

## slicing

In [9]:

```
1 s5[-1:-4:-1]
```

Out[9]:

'luf'

In [10]:

```
1 s5[::-1]
```

Out[10]:

'lufrewop si nohtyP'

In [11]:

```
1 len(s5)
```

Out[11]:

18

## string and for-loop

In [12]:

```
1 for ch in s5:
2     print(ch, end="")
```

Python is powerful

In [13]:

```
1 s1 + s2 + s3 + s4 + s5
```

Out[13]:

'goodGood morninggoodgoodgoodNumber3Python is powerful'

In [14]:

```
1 s1 = """String is Python's "text" type."""
```

In [15]:

```
1 print(s1)
```

String is Python's "text" type.

In [16]:

```
1 my_string = "aeiou"
2 if "a" in my_string and "u" in my_string:
3     print("Both conditions are True")
```

Both conditions are True

## 10-2. Escape Sequence

"\" 로 시작하며 "\"에 뒤따르는 문자에 특별한 의미를 부여하기 위해 사용

예) \n : new line  
\\ : backslash ('\')  
\\', \": 따옴표 (quotation mark)  
\\t : tab

In [17]:

```
1 print("First Line \nSecond Line")
```

First Line  
Second Line

In [18]:

```
1 print("Seoul-Busan \\non-stop")
```

Seoul-Busan \non-stop

In [19]:

```
1 print('He\'s my firend')
```

He's my firend

In [20]:

```
1 print("He's my friend")
```

He's my friend

In [21]:

```
1 print("She said \"I love him.\"")
```

She said "I love him."

In [22]:

```
1 print("tap character \tmakes spaces")
```

tap character      makes spaces

In [23]:

```
1 print("tap character \t\tmakes spaces")
```

tap character            makes spaces

In [24]:

```
1 s1 = """I didn't know what was "it"."""
2 print(s1)
```

I didn't know what was "it".

In [25]:

```
1 s = "I was bo\nrn in Seoul and I liv\ted in Seoul."
2 print(s)
```

I was bo  
rn in Seoul and I liv    ed in Seoul.

## 연습문제

1) parameter 로 받은 string 의 양 끝단 2 글자를 붙여서 반환하는 함수 작성

- string 의 slicing 및 concatenation 이용

```
def both_ends(s):
    if len(s) < 2:
        return ""
    first2 = s[:2]
    last2 = s[-2:]
    return first2 + last2

print(both_ends('spring')) #spng
print(both_ends('Hello')) #Helo
```

2) 두개의 문자열을 parameter 로 받아 서로의 첫번째 두 글자를 교환한 후 중간에 한칸 띄우고 반환하는 함수 작성

```
def mix_up(a, b):
    a_swapped = b[:2] + a[2:]
    b_swapped = a[:2] + b[2:]
    return ? + ' ' + ?

print(mix_up('frog', 'dinner')) # diog frnner
```

3) 두개의 parameter 를 받아서 두개의 시작이 같으면 SAME 다르면 Different 를 반환하는 함수 작성

```
test('a', 'a')    # SAME  
test('aba', 'aca') # Different
```