

11. String 의 여러가지 사용방법

11-1. string class 의 주요 메소드 (method)

대, 소문자 변환

`lower()`, `upper()`

문자열(string) 내의 특정 문자 갯수

`count()`

앞부분 / 뒷부분 화이트 스페이스(whitespace) 제거

Python 의 whitespace

' ' : space (공백)
'\n': newline (줄바꿈)
'\r': carriage return (
'\t': 수평탭 (horizontal tab)
'\v': 수직탭 (vertical tab)

`strip()` - 앞, 뒤의 whitespace 제거
`lstrip()` - 앞부분의 whitespace 제거
`rstrip()` - 뒷부분의 whitespace 제거

문자열(string) 내의 특정 문자, 문자열의 위치

`find()`, `index()`

문자열을 다른 문자열로 대치

`replace()`

문자열 쪼개기 / 합치기

`split()`
`join()`

built-in methods

In [1]:

```
1 s = "good morning"
```

In [2]:

```
1 dir(s)
```

Out[2]:

```
['__add__',  
 '__class__',  
 '__contains__',  
 '__delattr__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattr__',  
 '__getitem__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__iter__',  
 '__le__',  
 '__len__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rmod__',  
 '__rmul__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'capitalize',  
 'casefold',  
 'center',  
 'count',  
 'encode',  
 'endswith',  
 'expandtabs',  
 'find',  
 'format',  
 'format_map',  
 'index',  
 'isalnum',  
 'isalpha',  
 'isascii',  
 'isdecimal',  
 'isdigit',  
 'isidentifier',  
 'islower',  
 'isnumeric',  
 'isprintable',  
 'isspace',  
 'istitle',
```

```
'isupper',  
'join',  
'ljust',  
'lower',  
'lstrip',  
'maketrans',  
'partition',  
'replace',  
'rfind',  
'rindex',  
'rjust',  
'rpartition',  
'rsplit',  
'rstrip',  
'split',  
'splitlines',  
'startswith',  
'strip',  
'swapcase',  
'title',  
'translate',  
'upper',  
'zfill']
```

In [3]:

```
1 S = s.upper()  
2 print(S)
```

GOOD MORNING

In [4]:

```
1 S.lower()
```

Out[4]:

'good morning'

In [5]:

```
1 s.count('o')
```

Out[5]:

3

In [6]:

```
1 w = "  good morning  "
```

In [7]:

```
1 print(w.strip())
```

good morning

In [8]:

```
1 w = "  good \n morning  "
```

In [9]:

```
1 w.strip()
```

Out[9]:

```
'good \n morning'
```

In [10]:

```
1 w.lstrip()
```

Out[10]:

```
'good \n morning  '
```

In [11]:

```
1 w.rstrip()
```

Out[11]:

```
'  good \n morning'
```

In [12]:

```
1 s = "World Cup"
```

In [13]:

```
1 s.find("C")
```

Out[13]:

```
6
```

In [14]:

```
1 s.index("Cup")
```

Out[14]:

```
6
```

In [15]:

```
1 s.find("Cup")
```

Out[15]:

```
6
```

In [16]:

```
1 s.find("k")
```

Out[16]:

-1

In [17]:

```
1 s.index("k")
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-17-d0c044c34d00> in <module>  
----> 1 s.index("k")
```

ValueError: substring not found

In [20]:

```
1 try:  
2     s.index("k")  
3 except:  
4     print("not found")
```

not found

In [23]:

```
1 s.replace("World", "bad")
```

Out[23]:

'bad Cup'

In [24]:

```
1 slist = s.split()
```

In [25]:

```
1 print(slist)
```

['World', 'Cup']

In [26]:

```
1 s = 'World and Cup'  
2 s.split("and")
```

Out[26]:

['World ', ' Cup']

- 구분자를 이용한 list 의 단일 string 변환

In [32]:

```
1 ' '.join(slist)
```

Out[32]:

'World Cup'

In [33]:

```
1 '-'.join(["one", "to", "one"])
```

Out[33]:

'one-to-one'

11-2. 문자열 포매팅 (String Formatting)

- 문자열 포매팅이란 문자열 내에 어떤 값을 삽입하는 것
- {} 로 위치를 표시하고 format() method 로 mapping 시킨다.
- {{순서};[표시형식]} 지정 가능
- format() method 대신 % 를 이용할 수도 있다. → Python 2.7 방식

In [34]:

```
1 "Python is {}".format("powerful")
```

Out[34]:

'Python is powerful'

In [35]:

```
1 "Python is {} and {}".format("easy", "powerful")
```

Out[35]:

'Python is easy and powerful'

In [36]:

```
1 "Float : {}, List: {}".format(1.5, ['a','b','c'])
```

Out[36]:

"Float : 1.5, List: ['a', 'b', 'c']"

In [37]:

```
1 print("많이 팔리는 순서는 {0}, {1}, {2} 이고 {1} 는 품절입니다.".format("사과", "바나나", "망고"))
```

많이 팔리는 순서는 사과, 바나나, 망고 이고 바나나 는 품절입니다.

In [38]:

```
1 print("Friends are {s}, {b} and {j}.".format(j="Bill", b="John", s="Sean"))
```

Friends are Sean, John and Bill.

In [39]:

```
1 "{}-{}-{}".format(5, 10, 15)
```

Out[39]:

'5-10-15'

In [40]:

```
1 "{0:5}-{1:5}-{2:5}".format(5, 10, 15)
```

Out[40]:

' 5- 10- 15'

In [41]:

```
1 "{0:<10} left justify".format("LEFT")
```

Out[41]:

'LEFT left justify'

In [42]:

```
1 "{:>10} right justify".format("RIGHT")
```

Out[42]:

' RIGHT right justify'

In [43]:

```
1 "{:^10} center justify".format("CENTER")
```

Out[43]:

' CENTER center justify'

In [44]:

```
1 "{:*^10} filling with *".format("CENTER")
```

Out[44]:

'**CENTER** filling with *'

In [45]:

```
1 y = 3.141592
2 "{0} is {1:0.4f}".format("Pi", y)
```

Out[45]:

'Pi is 3.1416'

In [46]:

```
1 import math
2 r2 = math.sqrt(2)
3 print("square root 2 is {:.4f}".format(r2))
```

square root 2 is 1.4142

In [47]:

```
1 print("square root 2 is {:.10.4f}".format(r2))
```

square root 2 is 1.4142

In [48]:

```
1 print("square root 2 is [{:.10.4f}].format(r2))
```

square root 2 is [1.4142]

In [49]:

```
1 print("square root 2 is {{{:.10.4f}}}.format(r2))
```

square root 2 is { 1.4142}

11-3. 포맷문자열 리터럴 (f-문자열)

- f+string 내에 {expression} 사용

In [50]:

```
1 import math
2 print(f'The value of pi is approximately {math.pi:.3f}.')
```

The value of pi is approximately 3.142.

In [51]:

```
1 table = {'Sjoerd': 4127, 'Jack': 4098, 'Dcab': 7678}
2
3 for name, phone in table.items():
4     print(name, '==> {}'.format(phone))
5     print(f'{name:10} ==> {phone:10d}')
```

```
Sjoerd ==> 4127
Sjoerd  ==>    4127
Jack ==> 4098
Jack    ==>    4098
Dcab ==> 7678
Dcab    ==>    7678
```

Python 2.7 형식

In [52]:

```
1 "I have %s apples." % "three"
```

Out[52]:

```
'I have three apples.'
```

In [53]:

```
1 "I have %d apples" % 5
```

Out[53]:

```
'I have 5 apples'
```

In [54]:

```
1 "square root is %.4f" % r2
```

Out[54]:

```
'square root is 1.4142'
```

In [55]:

```
1 "% formatting is %s python %s" % ("old", "style")
```

Out[55]:

```
'% formatting is old python style'
```

연습문제

1) 다음의 print 결과는 ?

```
s = "Python is awesome"
print(s[1:3])
```

2) 다음 영어 문장에서 you 는 몇번 나오는가 ?

- (Hint) count() 메소드 사용

"You don't go to school on Saturday. I want you be happy. I miss you very much. You don't forget me"

3) "Spam and Eggs" 를 입력할 때 다음 프로그램의 출력 결과는 ?

```
def main():  
    msg = input("Enter a phrase: ")  
    for w in msg.split():  
        print(w[0], end="")
```

main()

(결과) SaE

4) 다음의 출력 결과는 ?

```
for x in "Mississippi".split("i"):  
    print(x, end="")
```

(결과) Msssspp

5) 다음 출력 결과는 ?

```
s = "Jane Doe"  
print(s[3 : 1 : -1])
```

(결과) en

6) s = "Hello, Python World" 을 알파벳 별로 몇개인지 계산 (단, 대소문자 무시)

- * hint : 1. ", " 와 " " 을 "" 로 replace
- 2. s.lower() 를 이용하여 소문자로 통일
- 3. dictionary 를 이용하여 알파벳 별 개수 누적

(결과)

```
d 1  
e 1  
h 2  
l 3  
n 1  
o 3  
p 1  
r 1  
t 1  
w 1  
y 1
```

7) parameter 로 받은 string 의 선두 character 가 뒤따르는 문자열에 나타나면 * 로 바꾸어 반환하는 함수 작성

```
def fix_starts(s):  
    front = s[0]    # 선두 character  
    back = s[1:]    # 뒤따르는 문자열  
    fixed_back = back.replace(front, '*')  
    return front + fixed_back  
  
print(fix_starts('babble'))    # ba**le  
print(fix_starts('google'))    # goo*le
```