
문자열의 이해

7주차_01

한 동 대 학 교
김경미 교수

문자열 (string)

- 문자열은 글자들(characters)의 나열(sequence)
- 문자열은 구성되는 각 글자를 첨자 표현으로 따로 나누어서 활용 가능

```
>>> name = 'apple'
```

```
>>> print(name[0])
```

```
'a'
```

```
>>> school = '한동대학교'
```

```
>>> address = '경북 포항시 북구 흥해읍'
```

문자열의 구성

변수명	저장된 값		
str	“apple”	“lemon”	“한동대학교”
str[0]	“a”	“l”	“한”
str[1]	“p”	“e”	“동”
str[2]	“p”	“m”	“대”
str[3]	“l”	“o”	“학”
str[4]	“e”	“n”	“교”

문자열에서 사용하는 연산자(1)

- 산술연산자: +, *

```
>>> s1 = 'Cutty'
```

```
>>> s2 = 'Cat'
```

```
>>> s3 = s1 + s2
```

```
>>> s3
```

```
'CuttyCat'
```

```
>>> s1*3
```

```
'CuttyCuttyCutty'
```

```
>>> '@' * 10
```

```
'@@@@@@@@@@@'
```

문자열에서 사용하는 연산자(2)

- 관계연산자: >, >=, <, <=, ==, !=

```
>>> s1 = 'cat'
```

```
>>> s2 = 'Cat'
```

```
>>> s1 == s2
```

```
False
```

```
>>> s1 == 'cat'
```

```
True
```

```
>>> s1 > 'bird'
```

```
True
```

문자열(string), len()

- len() 은

- 내장된 함수로서 문자열을 구성하는 글자수를 반환

```
>>> fruit = 'banana'
>>> len(fruit)
6
>>> length = len(fruit)
>>> last = fruit[length]
IndexError: string index out of range

>>> last = fruit[length-1]
>>> last
a
```

반복문으로 문자열 처리

```
fruit="apple"
```

```
index = 0
```

```
while index < len(fruit) :
```

```
    letter = fruit[index]
```

```
    print(letter)
```

```
    index = index + 1
```

```
>>>
===== RESTART:
py =====
a
p
p
l
e
>>>
```

```
fruit="apple"
```

```
for ch in fruit :
```

```
    print(ch)
```

For문이 간략

문자열 자르기 (String Slice)

- 문자열의 일부분을 조각(slice) 이라고 함
- 조각을 고르는 것은 글자(character)를 고르는 것과 유사

```
>>> s = 'Monty Python'
>>> s[0:5]           # index 0,1,2,3,4
Monty
>>> s[6:12]         # index 6,7,8,9,10,11
Python
>>> s[1:3]          # index 1,2
on
>>> fruit = 'banana'
>>> fruit[:3]       # index 0,1,2
'ban'
>>> fruit[3:]       # index from 3 to last
'ana'
```

연습문제 1

- **문자열을 입력 받는다**
 - 만약 입력된 값이 'apple'이라고 하면..
- **반복문과 함수 len()사용한다**
 - 입력한 문자열을 다음과 같이 출력되도록 코딩한다
 - 'a'
 - 'ap'
 - 'app'
 - 'appl'
 - 'apple'

연습문제 1, 코드와 결과

```
input_str = input("문자열 입력: ")
l = len(input_str)
index = 0
```

```
while index < l :
    print("s[0:", index+1, "]=", input_str[0 : index+1])
    index = index + 1
```

```
>>>
===== RESTART: E:/1_Works/
py =====
문자열 입력: strawberry
s[0: 1 ]= s
s[0: 2 ]= st
s[0: 3 ]= str
s[0: 4 ]= stra
s[0: 5 ]= straw
s[0: 6 ]= strawb
s[0: 7 ]= strawbe
s[0: 8 ]= strawber
s[0: 9 ]= strawberry
s[0: 10 ]= strawberry
>>>
```

String methods, `.upper()`

- **메소드(method)는 인수들을 받고 값을 반환**
 - 이 점에서 함수(function)와 유사
 - 사용하는 문법(syntax)은 다름
 - 변수명을 쓰고, 메소드를 사용
- **메소드 문법 `.upper()` 를 사용**
 - 모든 문자를 대문자로 변환

```
>>> word = 'banana'  
>>> new_word = word.upper()  
>>> new_word  
BANANA
```

String methods, `.find()`

- 문자열 내에 존재하는 문자를 찾음
- 찾는 문자가 존재하는 위치(index)를 알려줌

```
>>> word = 'banana'
>>> index = word.find('a')
>>> index
1
>>> word.find('na')
2
>>> word.find('na', 3) #두번째 숫자는 찾기 시작하는 index number
4
>>> name = 'bob'
>>> name.find('b', 1, 2) # 세번째 숫자는 어디까지 찾을지 지정하는 index number
-1
```

String Methods

<code>.capitalize()</code>	첫 글자만 대문자로 바꾼다	<code>.isupper()</code>	모든 요소가 대문자이면 참, 아니면 거짓
<code>.count()</code>	지정 구간에서 글자수를 센다	<code>.join()</code>	문자열을 합쳐준다
<code>.find()</code>	찾는 문자 또는 문자열이 시작하는 첨자를 찾아준다	<code>.lower()</code>	소문자로 바꾼다
<code>.isalpha()</code>	모든 요소가 알파벳이면 참, 아니면 거짓	<code>.replace()</code>	<code>.replace(old, new [,count])</code> 다른 문자열로 바꿔준다
<code>.isdigit()</code>	모든 요소가 숫자이면 참, 아니면 거짓	<code>.split()</code>	
<code>.islower()</code>	모든 요소가 소문자이면 참, 아니면 거짓	<code>.swapcase()</code>	대문자는 소문자로, 소문자는 대문자로 바꾼다
<code>.isspace()</code>	모든 요소가 공백이면 참, 아니면 거짓	<code>.upper()</code>	소문자로 바꾼다

in operator (오퍼레이터)

```
>>> 'a' in 'banana'
True
>>> 'seed' in 'banana'
False
>>> 'ana' in 'banana'
True
```

#함수 사용

```
def in_both(word1, word2):
    for letter in word1:
        if letter in word2:
            print(letter)
```

```
in_both('apples', 'oranges')
```

```
>>> in_both('apples', 'oranges')
|
| a
| e
| s
>>>
```

연습문제 2

- 문자열을 입력 받는다
- 대문자로 바꿀 문자 또는 문자열을 입력 받는다
- 입력 받은 문자를 대문자로 바꾼 후 전체 문자열을 출력한다
- 예시
 - 입력 문자열 “apple”
 - 바꿀 문자열 “le”
 - 출력결과 “appLE”

연습문제 2, 코드와 결과

```
Input_Str = input("문자열 입력: ")
Input_find = input("찾아 바꿀 문자: ")
find_Length = len(Input_find)
```

```
Index = Input_Str.find(Input_find)
```

```
if Index == -1 :
    print(" 바꿀 문자가 없네요!")
```

```
else:
```

```
    beforeStr = Input_Str[0 : Index]
```

```
    changeStr = Input_Str[Index : Index+find_Length].upper()
```

```
    afterStr = Input_Str[Index+find_Length : ]
```

```
    result = beforeStr + changeStr + afterStr
```

```
    print(result)
```

```
>>>
===== RESTART: E:/1_Works/2
py =====
문자열 입력: JavaScript
찾아 바꿀 문자: ava
JavaScript
>>>
===== RESTART: E:/1_Works/2
py =====
문자열 입력: Java
찾아 바꿀 문자: JA
바꿀 문자가 없네요!
>>>
```

연습문제 3

- 문자열을 입력 받는다
- 대문자는 소문자로, 소문자는 대문자로 바꾸어서 저장하고 출력한다
- 바꾼 후에 대문자 개수, 소문자 개수, 대소 문자가 아닌 문자 개수를 출력한다

연습문제 3, 코드와 결과

```
str = input("문자열 입력: ")
uppercount = 0
lowercount = 0
etccount = 0

swapstr = str.swapcase()
print("대소문자를 바꾼 결과 = ", swapstr)

for ch in swapstr :
    if ch.isupper() :
        uppercount +=1
    elif ch.islower() :
        lowercount +=1
    else :
        etccount +=1

print("대문자 갯수는 ", uppercount)
print("소문자 갯수는 ", lowercount)
print("대소문자 아닌 문자 갯수는 ", etccount)
```

```
>>>
===== RESTART: E:/1_Works/2017Work/KMooC:
py =====
문자열 입력: I love PYTHON!!
대소문자를 바꾼 결과 = i LOVE python!!
대문자 갯수는 4
소문자 갯수는 7
대소문자 아닌 문자 갯수는 4
>>> |
```

숙제

- 연습문제 1, 2, 3 코드와
- 실행결과 캡처한 사진을 게시판에 올려주세요!

요약

- 문자열을 이해한다
- 문자열에서 사용하는 연산자를 이해한다
- 문자열 지원하는 메소드를 알고, 활용한다

감사합니다

7주차_01 문자열의 이해