

# string methods 이해

7주차\_01\_03

한 동 대 학 교  
김경미 교수

# 학습목표

2

- ▶ 문자열 지원하는 다양한 메소드를 알기
- ▶ 문자열 메소드 활용하기

# 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>	리스트를 문자열을 변환한다
<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>	문자열을 리스트로 변환한다
<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>	대문자로 바꾼다

# String methods, .capitalize()

- ▶ 첫 글자만 대문자로 바꿈

```
>>> word = 'abcd'
>>> new_word = word.capitalize()
>>> print(new_word)
Abcd
```

# String methods, .count()

5

- ▶ 문자열 내부에 특정 문자가 몇 개 포함되어 있는지 카운팅
- ▶ 문자열.count('특정문자', 시작위치, 마지막위치)

```
>>> word = 'helloWorld'
>>> word.count('o')
2

>>> word.count('o', 1, 3)
0
```

# String methods, .find()

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

```
>>> word = 'banana'
>>> index = word.find('a')
>>> print(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, .join(), .split()

7

- ▶ join() : 문자열을 합쳐 줌
- ▶ split() : 문자열을 나누어 리스트로 만들어 줌

```
>>> a = ['1', '2', 'h', 'a', 'n', 'd', 'o', 'n', 'g']
>>> result1 = ''.join(a)
>>> print(result1)
12handong

>>> result2 = result1.split()
>>> print(result2)
['1', '2', 'h', 'a', 'n', 'd', 'o', 'n', 'g']
```

# String methods, .isalpha()

- ▶ 모든 요소가 알파벳이면 True

```
>>> a = 'A12'
>>> b = 'AB C'
>>> c = '123'
>>> for char in a:
>>>     print(char.isalpha())
True
False
False
>>> for char in b:
>>>     print(char.isalpha())
True
True
False
True
>>> for char in c:
>>>     print(char.isalpha())
False
False
False
```



# String methods, .isdigit()

- ▶ 모든 요소가 숫자면 True

```
>>> a = '010-1234'
>>> b = "a1f3"
>>> for char in a:
    print(char.isdigit())
```

```
True
True
True
False
True
True
True
True
```

```
>>> for char in b:
    print(char.isdigit())
```

```
False
True
False
True
```

# String methods, `.islower()`, `.isupper()`

10

- ▶ `islower()` : 모든 요소가 소문자이면 True
- ▶ `isupper()` : 모든 요소가 대문자이면 True

# String methods, .islower()

```
>>> word = "It is Python"
```

```
>>> for char in word:  
    print(char.islower())
```

```
False
```

```
True
```

```
False
```

```
True
```

```
True
```

```
False
```

```
False
```

```
True
```

```
True
```

```
True
```

```
True
```

```
True
```

# String methods, .isupper()

12

```
>>> word = "It is Python"
```

```
>>> for char in word:  
    print(char.isupper())
```

```
True
```

```
False
```

```
False
```

```
False
```

```
False
```

```
False
```

```
True
```

```
False
```

```
False
```

```
False
```

```
False
```

```
False
```

# String methods, .isspace()

13

- ▶ 모든 요소가 공백이면 True

```
>>> s1 = " Hello Python "
```

```
>>> for char in s1:  
    if char.isspace():  
        print("true")  
    else:  
        print("false")
```

```
False  
False  
False  
False  
False  
True  
False  
False  
False  
False  
False  
False
```

# String methods, .lower()

14

- ▶ lower() : 모든 문자를 소문자로 바꿈

```
>>> word = 'HELLO'  
>>> new_word = word.lower()  
>>> print(new_word)  
hello
```

# String methods, .replace()

15

- ▶ 해당문자를 변경 문자로 바꾸어줌
- ▶ 문자열.replace('검색문자', '변경문자')

```
>>> feel = "I'm so happy"
>>> result = feel.replace("so", "not")
>>> print(result)
I'm not happy
```

```
>>> str = "you are welcome"
>>> result = str.replace("e", "f")
>>> print(result)
You arf wflcdmf
```

# String methods, .swapcase()

16

- ▶ 대문자는 소문자로, 소문자는 대문자로 바꿔줌

```
>>> sentence = 'This is Python Class'
>>> result= sentence.swapcase()

>>> print(result)
tHis IS pYTHON cLASS
```



# String methods, .upper()

17

- ▶ 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()
>>> print(new_word)
BANANA
```

# in operator

18

```
>>> '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
>>>
```

# in operator 예제

19

```
fruits1=['apple', 'banana', 'lemon', 'grape', 'strawberry']  
fruits2=['cherry', 'watermelon','lemon', 'melon']
```

```
for fruit in fruits2:  
    if fruit in fruits1:  
        print(fruit,"is in","fruits1")
```

```
----- RESTART: -----  
lemon is in fruits1  
>>>
```

# .find 사용 예제

```
s = 'abcd cdef def gh'

if s.find('d')== -1:
    print("찾는 문자가 없음")
else:
    print(s.find('d'))

if s.find('f')== -1:
    print("찾는 문자가 없음")
else:
    print(s.find('f'))

if s.find('i')== -1:
    print("찾는 문자가 없음")
else:
    print(s.find('i'))
```

```
3
8
찾는 문자가 없음
>>>
```

# 강의 요약

21

<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>	리스트를 문자열을 변환한다
<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>	문자열을 리스트로 변환한다
<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>	대문자로 바꾼다

# 목표 달성 질문

22

- ▶ 다음 문자열 메소드 기능을 설명하시오
  - ▶ `.count()`
  - ▶ `.find()`
  - ▶ `.isupper()`
  - ▶ `.swapcase()`

# 감사합니다