파이선으로 배우는 데이터구조

Data Structures Learning with Python

김영훈

한양대학교 ERICA 인공지능학과

문자열

- 문자열 만들기: ‘’ 혹은 “”로 감싼 문자의 나열
- 불변형 자료구조 (immutable)
  - 즉, 데이터 수정이 불가

String

Method	Description
s.count(substring, [start,end])	선택적 시작 및 종료 매개 변수가 있는 부분 문자열의 발생 횟수를 계산합니다.
s.expandtabs([tabsize])	탭을 공백으로 바꿉니다.
s.find(substring, [start, end])	하위 문자열의 첫 번째 발생 인덱스를 반환하거나 하위 문자열을 찾을 수 없는 경우 -1을 반환합니다.
s.isalnum()	모든 문자가 영숫자인 경우 True를 반환하고, 그렇지 않으면 False를 반환합니다.
s.isalpha()	모든 문자가 알파벳이면 True를 반환하고, 그렇지 않으면 False를 반환합니다.
s.isdigit()	모든 문자가 숫자인 경우 True를 반환하고, 그렇지 않으면 False를 반환합니다.
s.join(t)	문자열을 시퀀스 t로 조인합니다.
s.lower()	문자열을 모두 소문자로 변환합니다.
s.replace(old, new [maxreplace])	부분 문자열을 새 부분 문자열로 바꿉니다.
s.strip([characters])	공백 또는 선택적 문자를 제거합니다.
s.split([separator], [maxsplit])	선택적 시작 및 종료 매개 변수가 있는 부분 문자열의 발생 횟수를 계산합니다.

가변? 불변!

String

- 불변값을 수정하여 값을 얻으려면 다른 변수를 새로 만들어야 합니다.

```
a = "test"
a.upper()

print(a)

test
```

String

- 불변값을 수정하여 값을 얻으려면 다른 변수를 새로 만들어야 합니다.

```
a = "test"
b = a.upper()

print(b)

TEST
```

String

- 불변값을 수정하여 값을 얻으려면 다른 변수를 새로 만들어야 합니다.

```
a = "test"
b = a.upper()

if id(a) == id(b):
    print(“a is mutable")
else:
    print("b is a new variable")


b is a new variable
```

String 1.

- 문자열의 길이를 계산하는 Python 프로그램을 작성하십시오.
- print(string_length('hanyang university erica'))

```
def string_length(str1):
    count = 0
    for char in str1:
        count += 1
    return count
print(string_length('hanyang university erica'))
```

String 2.

- 문자열이 주어졌을 때, 각 문자들(문자 빈도수)의 개수를 세는 Python 프로그램을 작성하십시오.
- print(char_frequency('google.com'))
- 예상 출력 : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}

```
def char_frequency(str1):
    dict = {}
    for n in str1:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
print(char_frequency('google.com'))
```

String 3.

- 주어진 문자열의 첫 두 문자와 마지막 두 문자로 만들어진 문자열을 반환하는 Python 프로그램을 작성하십시오. 만약 문자열의 길이가 2보다 작다면 빈 문자열을 반환하도록 하세요.
- print(string_both_ends(‘hanyang’)), 예상 출력 : ‘hang’
- print(string_both_ends('w’)), 예상 출력:

```
def string_both_ends(str):
  if len(str) < 2:
    return ''

  return str[0:2] + str[-2:]

print(string_both_ends('w3resource'))
print(string_both_ends('w'))
```

String 4.

- 주어진 문자열의 첫번째 문자가 문자열 내에 또 있다면 모두 ' $'로 바꾼 Python 프로그램을 작성하십시오. 단, 첫번째 문자는 그대로 두십시오.
- print(change_char('restart'))
- 예상 출력: 'resta$t'

```
def change_char(str1):
    ?

print(change_char('restart'))
```

String 5.

- 2개의 문자열이 주어졌을 때, 각 문자열의 첫 2개 문자가 뒤바뀌고 space로 나눠진 하나의 문자열을 출력하는 프로그램을 작성하시오.
- print(chars_mix_up('abc', 'xyz'))
- 예상 출력 : 'xyc abz'

```
def chars_mix_up(a, b):
                    ?
print(chars_mix_up('abc', 'xyz'))
```

String 6.

- 주어진 문자열의 n번째 문자열을 삭제하는 프로그램을 작성하시오.
- print(remove_char('Python', 3))
- 예상 출력: Pyton

```
def remove_char(str, n):
                    ?
print(remove_char('Python', 0))
print(remove_char('Python', 3))
print(remove_char('Python', 5))
```

String 7.

- 주어진 문자열의 홀수 번째 index의 문자를 출력하는 프로그램을 작성하시오.
- print(odd_values_string('abcdef'))
- 예상 출력: bdf

```
def odd_values_string(str):
    ?

print(odd_values_string('abcdef'))
print(odd_values_string('python'))
```

String 8.

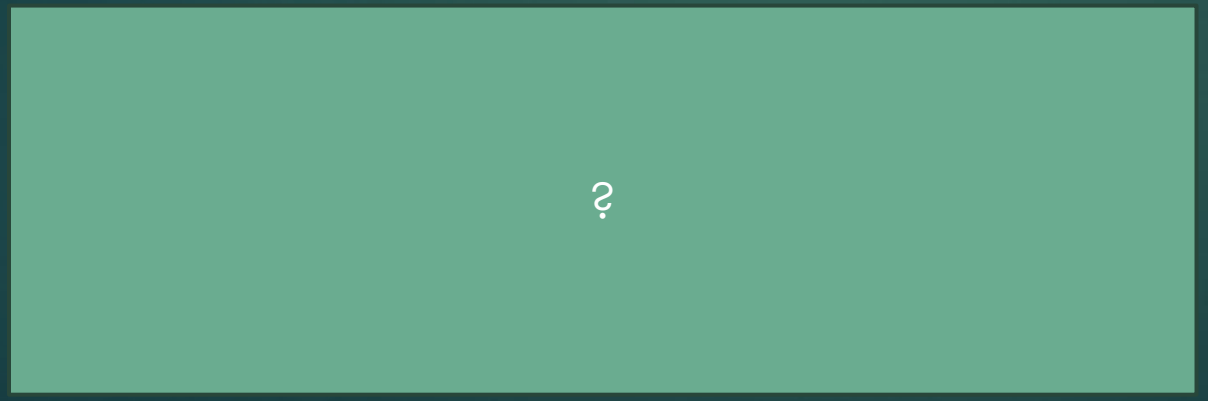
- 주어진 문자열의 첫 3개의 문자를 출력하는 프로그램을 작성하시오. 문자열의 길이가 3보다 작다면 , 원래 문자열을 반환하시오.
- print(first_three('python')), 예상 출력 : pyt
- Print(first_three('py')) , 예상 출력: py

```
def first_three(str):
                    ?
print(first_three('python'))
print(first_three('py'))
```

String 9.

- 문자열이 주어졌을 때, 특정 문자가 나오기 전 까지의 문자열을 출력하는 프로그램을 작성하시오.
- 문자열 샘플 : 'https://www.hanyang.ac.kr/' , 특정 문자 : 'c'
- 예상 출력 : https://www.hanyang.a

```
str1 = 'https://www.hanyang.ac.kr/'
char = 'c'
```



String 10.

- 문자열이 주어졌을 때, 첫 4개 문자들에 적어도 2개의 대문자가 있다면, 모든 문자를 대문자로 바꾸는 프로그램을 작성하시오.
- print(to_uppercase('Python')) , 예상 출력: Python
- print(to_uppercase('PyThon')) , PYTHON

```
def to_uppercase(str1):
    ?

print(to_uppercase('Python'))
print(to_uppercase('PyThon'))
```

String 11.

- 주어진 문자열을 사전식으로 sort 하는 프로그램을 작성하시오.
- print(lexicographi_sort('hanyang'))
- 예상 출력 : ['a', 'a', 'g', 'h', 'n', 'n', 'y']

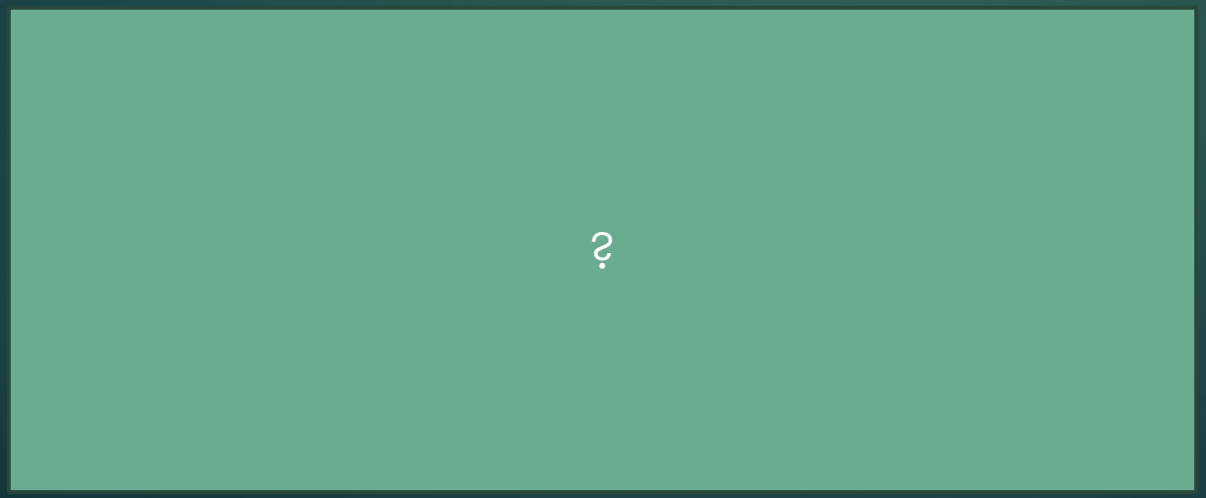
```
def lexicographi_sort(s):
    ?

print(lexicographi_sort('hanyang'))
```

String 12.

- 주어진 문자열이 특정 문자열로 시작하는지 확인하는 프로그램을 작성하시오. (힌트: for와 zip 사용)
- 샘플 문자열 : “hanyang.ac.kr” , 특정 문자열 : “han”
- 예상 출력: True

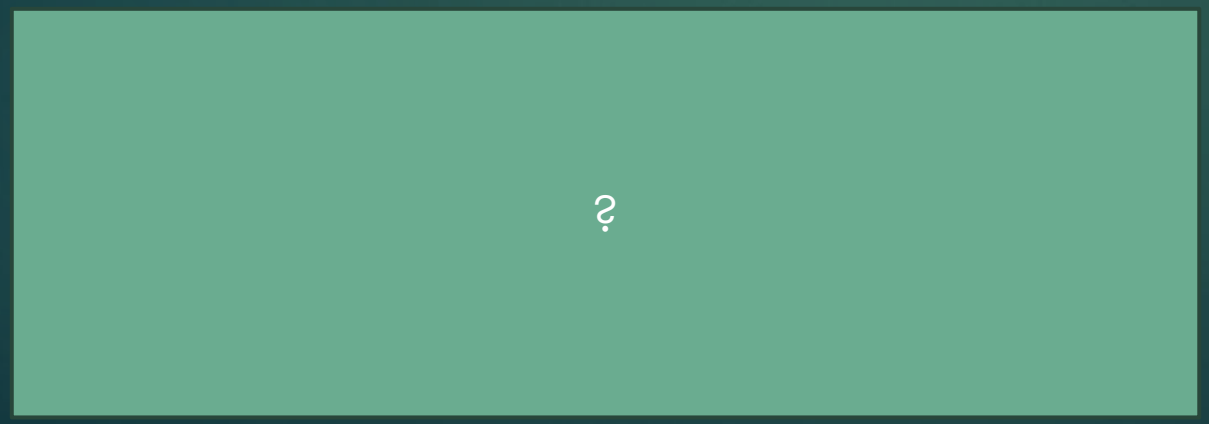
```
string =  " hanyang.ac.kr"
```



String 13.

- 주어진 숫자를 소수점 2째자리까지 출력하는 프로그램을 작성하시오.
- 샘플 숫자 : x = 3.141592
- 예상 출력 : 3.14

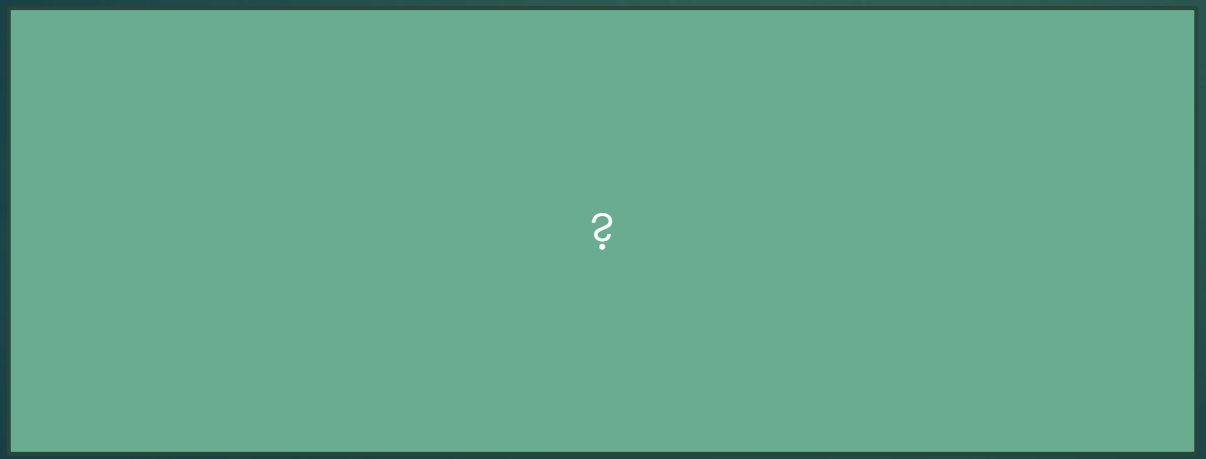
```
x = 3.1415926
y = 12.9999
```



String 14.

- 주어진 숫자를 퍼센테이지로 출력하는 프로그램을 작성하시오.
- 샘플 숫자 : 0.25
- 예상 출력 : 25.00%

```
x = 0.25
```



String 15.

- 주어진 문자열을 단어들의 list로 변환하는 프로그램을 작성하시오.
- 샘플 문자열 : "The-quick-brown-fox-jumps-over-the-lazy-dog."
- 예상 출력 : ['The', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog.']

```
str1 = "The-quick-brown-fox-jumps-over-the-lazy-dog."
print(str1.split('-'))
```

String 16.

- 주어진 문자열에 특정 substring이 포함되어 있는 개수를 세는 프로그램을 작성하시오. (힌트: for와 dictionary, split()을 사용)
- 샘플 문자열: str1 = 'The quick brown fox jumps over the lazy dog.'
  특정 substring : "fox"
- 예상 출력 : 1

```
str1 = 'The quick brown fox jumps over the lazy dog.'
```

```
?
```

String 17.

- 주어진 문자열을 reverse하는 프로그램을 작성하시오. (힌트: for 사용 가능, 혹은 reverse()란 함수 사용법을 검색해 사용, ''.join()을 사용해야 함)
- print(reverse_string("abcdef"))
- 예상 출력 : fedcba

```
def reverse_string(str1):
    ?
print(reverse_string("abcdef"))
```

String 18.

- 주어진 문자열에서 특정 문자 set를 strip하는 프로그램을 작성하시오. (힌트: in / not in)
- print(strip_chars(" The quick brown fox jumps over the lazy dog. " , " aeiou "))
- 예상 출력 : Th qck brwn fx jmps vr th lzy dg.

```
def strip_chars(str, chars):
    ?

print(strip_chars("The quick brown fox jumps over the lazy dog.",
"aeiou"))
```

String 19.

- 주어진 문자열에서 반복되는 문자를 세는 프로그램을 작성하시오. (collection 라이브러리 사용)
- 샘플 문자열 : 'thequickbrownfoxjumpsoverthelazydog'

```
str1 = 'thequickbrownfoxjumpingoverthelazydog’
d = dict()

                    ?

for c in sorted(d, key=d.get, reverse=True):
  if d[c] > 1:
      print('%s %d' % (c, d[c]))
```

String 21.

- 주어진 문자열의 첫 n개의 문자를 소문자로 바꾸는 프로그램을 작성하시오.
- 샘플 문자열 : ‘HANYANG.AC.KR’ , n= 4
- 예상 출력 : hanyANG.AC.KR

```
str1 = ‘HANYANG.AC.KR'
```

?

String 22.

- 주어진 문자열에서 쉼표와 마침표를 swap하는 프로그램을 작성하시오. (힌트: for와 if 사용)
- 샘플 문자열 : "32.054,23"
- 예상 출력 : "32,054.23"

```
str = "32.054,23"
```

```
?
```

String 23.

- 주어진 문자열에 있는 모음과 모음의 개수를 출력하는 프로그램을 작성하시오. (힌트: 모든 모음의 문자열 vowels = "aeiuoAEIOU"를 정의해 놓고 in vowels 연산을 이용해 모음의 개수를 세면 됨)
- vowel('Hanyang University')
- 예상 출력 :
  - ['e', 'o', 'u', 'e']
  - 4

```
def vowel(text):
                                    ?

vowel('Hanyang University');
```

String 24.

- 주어진 문자열에 있는 대문자, 소문자, 특수기호, 숫자의 개수를 순서대로 출력하는 프로그램을 작성하시오.
- print(count_chars("@Hanyang.University.ERICA"))
- 예상 출력 : (7, 15, 1, 0)

```
def count_chars(str):
    ?

print(count_chars("@Hanyang.University.ERICA"))
```

String 25.

- 주어진 2개의 문자열에 있는 공통 문자를 출력하는 프로그램을 작성하시오.
- print(intersection_of_two_string('Python3', 'Pyton2.7'))
- 예상 출력 : Pyton

```
def intersection_of_two_string(str1, str2):
    
    ?

str1 = 'Python3'
str2 = 'Pyton2.7'
print(intersection_of_two_string(str1, str2))
```

String 26.

- 주어진 문자열의 끝에 ' ing'을 추가하는 프로그램을 작성하시오. 만약 주어진 문자열이 이미 ' ing ' 으로 끝난다면 ' ly ' 를 추가하시오.
- print(add_string('abc')) -> 예상 출력 : abcing
- print(add_string('string')) -> 예상 출력 : stringly

```
def add_string(str1):
    ?
    return str1
print(add_string('abc'))
print(add_string('string'))
```