



Computación 2022 Verano

Lab. 5.3. Métodos de String de Python

Febrero 02, 2022

Cree una carpeta (folder) en el disco D, nómbrela con su apellido paterno seguido de su código. Ejemplo: LOPEZ12345

Tabla 1.1 Operaciones de Secuencias aplicables a String

Nombre	Operación	Significado para S="¡Hola Como Estas!"
Length	len(str)	len(S) ---> 17
Select	S[i]	S[6] ---> 'C'
Slice	S[i:j]	S[11:16] ---> "Estas" (desde i hasta j-1)
	S[i:j:k]	S[1:12:5] ---> "HCE"
Count	S.count(chr)	S.count('o') ---> 3
Index	S.index(chr)	S.index('E') ---> 11
Membership	'chr' in S	'p' in S ---> False
	'chr' not in S	'p' not in S ---> True
Concatenation	S + W	S + "!!" ---> "¡Hola Como Estas!!!"
	n*S (S*n)	2*S --> "¡Hola Como Estas!¡Hola Como Estas!"
Minimum Value	min(S)	min(S) ---> " " (espacio)
Maximum Value	max(S)	max(S) ---> 't'
Comparison	S == W	S == "Hola" ---> False

Tabla 1.2 Principales Métodos de String

Métodos para verificar el contenido de un String

str.isalpha()	Devuelve True si str contiene solo letras	S="Hello"	S.isalpha() --> True
		S="Hello!"	S.isalpha() --> False
str.isdigit()	Devuelve True si str contiene solo dígitos	S="789"	S.isdigit() --> True
		S="789W"	S.isdigit() --> False
str.islower() str.isupper()	Devuelve True si str contiene solo letras minúsculas (mayúsculas)	S="hello"	S.islower() --> True
		S="Hello"	S.isupper() --> False
str.lower() str.upper()	Devuelve la versión minúsculas (mayúsculas) de str	S="Hello!"	S.lower() --> "hello!"
		S="hello!"	S.upper() --> "HELLO!"

Método de búsqueda en un String

str.find(w)	Retorna el índice de la primera ocurrencia de w en str. Retorna -1 si no lo encuentra	S="Hello!"	S.find('l') --> 2
		S="Goodbye"	S.find('l') --> -1

Método para reemplazar el contenido de un String

str.replace(w,t)	Retorna una copia de str con todas las ocurrencias de w reemplazadas con t	S="Hello!"	S.replace('H','J') --> "Jello"
		S="Hello"	S.replace('ll','r') --> "Hero"

Método para remover el contenido de un String

str.strip(w)	Retorna una copia de str con todos los caracteres iniciales y finales que aparecen en w eliminados	S=" Hello! "	S.strip(' !') --> "Hello"
		S="Hello\n"	S.strip('\n') --> "Hello"

Método para dividir el contenido de un String

str.split(w)	Devuelve una lista que contiene todas las string de str delimitado por w	S="Lu,Chao"	S.split(',') --> ["Lu", "Chao"]
--------------	--	-------------	---------------------------------

Cómo extraer subcadenas (substring) de una cadena de caracteres (string)

Python no posee una función específica para extraer subcadenas (substring) de un texto, sin embargo, podemos usar la técnica de **cortar (rebanar)** una cadena, en inglés **slicing**.

La sintaxis para rebanar una cadena (string slicing) es:

```
string_name[start:end[:step]]
```

Mediante string slicing podemos acceder a una secuencia de caracteres especificando un rango de índices separados por dos puntos (:). String slicing retorna una secuencia de caracteres comprendidos entre start y end pero sin incluir end. El tercer parámetro de string slicing es opcional, se usa cuando el incremento es diferente a 1 (el incremento es step).

Ejemplo:

```
name = "myfile.txt"

name[0:]          # todo el string
'myfile.txt'
name[0:1]         # el primer carácter
'm'
name[0,2]         # dos primeros caracteres
'my'
name[-3:]         # los últimos tres caracteres
'txt'
name[7:]          # los últimos tres caracteres
'txt'
name[2:6]         # para extraer "file"
'file'
```

Comprobando si un substring se encuentra en un string, usando los operadores in; not in

La sintaxis del operador in es:

```
target_string in search_string
```

El operador **in** devuelve True si target_string se encuentra en algún lugar en search_string, de lo contrario devuelve False.

Ejemplo:

```
a = "Learning Python is super fun!"
"super" in a
True

"super" not in a
False

S = input("Ingrese un texto:")

if "Python" in S:
    print("Python", "is in", S)
else:
    print("Python", "is not in", S)
```

1. Supongamos que `s1`, `s2`, `s3` y `s4` son cuatro String con los valores siguientes:

```
s1 = "Welcome to Python"
s2 = s1
s3 = "Welcome to Python"
s4 = "to"
```

¿Cuál es el resultado de las siguientes expresiones?

- | | |
|----------------------------------|------------------------------------|
| a. <code>s1 == s2</code> | l. <code>4 * s4</code> |
| b. <code>s2.count('o')</code> | m. <code>s1[::-2]</code> |
| c. <code>id(s1) == id(s2)</code> | n. <code>max(s1)</code> |
| d. <code>id(s1) == id(s3)</code> | o. <code>min(s1)</code> |
| e. <code>s1 <= s4</code> | p. <code>s1[-4]</code> |
| f. <code>s2 >= s4</code> | q. <code>s1.lower()</code> |
| g. <code>s1 != s4</code> | r. <code>s1.rfind('o')</code> |
| h. <code>s1.upper()</code> | s. <code>s1.startswith("o")</code> |
| i. <code>s1.find(s4)</code> | t. <code>s1.endswith("o")</code> |
| j. <code>s1[:7]</code> | u. <code>s1.isalpha()</code> |
| k. <code>s1[4 : 8]</code> | v. <code>s1 + s1</code> |

Para cada uno de los siguientes ejercicios implemente el programa Python correspondiente. Guarde vuestros programas en vuestra carpeta de trabajo.

- Algunos sitios Web imponen ciertas reglas para los password. Escriba una función que verifique si un string es un password válido. Suponga que las reglas para los password son:
 - Un password debe de tener al menos 8 caracteres
 - Un password debe de consistir solo de letras y dígitos
 - Un password debe de contener al menos dos dígitosEscriba un programa que solicite el ingreso de un password e imprima "password válido" si cumple las reglas o "password inválido" de lo contrario.
- Escriba una función que reciba una frase y determine el número de palabras, de dígitos, de letras mayúsculas y letras minúsculas. Use las funciones `isspace()`, `isdigit()`, `isupper()`, `islower()`.
- Escriba una función que reciba una frase cuyas palabras estén separadas por comas y devuelva la frase con las palabras separadas por guiones y en orden ascendente. Use las funciones `split()`, `sort()`, `joint()`.
- Dado un texto, que puede contener espacios tanto al inicio como al final y cuyas palabras pueden estar separadas por más de un espacio. Realizar las siguientes transformaciones. Genere un nuevo string para cada caso.
 - Colocar en mayúsculas la primera letra de cada palabra
 - Toda letra "e" reemplazar por la letra "E".
 - Verificar si la frase inicia con la palabra "Hola" y termina con "saludos".
 - Colocar en mayúscula solo la primera letra del texto.
 - Colocar en mayúsculas las letras que aparecen en minúsculas y viceversa.
 - Eliminar los espacios del inicio y del final del texto, más no los del medio.
 - Eliminar todos los espacios del texto.
 - Eliminar todos los espacios duplicados
- Dado un texto con las palabras separadas por un espacio:
 - Ordenar las palabras de la cadena en orden lexicográfico
 - Ordenar las palabras de la cadena por tamaño de las palabras
 - Eliminar las palabras duplicadas

OPCIONAL

6. Dado una secuencia de nombres de archivos con sus respectivas extensiones, imprimir solo aquellos archivos .txt.

Guarde todos vuestros programas y vuestra hoja de respuestas en una carpeta con el nombre su **Apellido** paterno seguido de vuestro **DNI**, luego comprima esta carpeta. Envíe este archivo a victor.melchor.e@upch.pe , especificando como asunto **Lab5.3**.