**Manipulación de String en Python**.

Prueba las siguientes instrucciones en el Shell de Python y ve documentando en la tabla qué hace cada una de las instrucciones o métodos de manipulación de strings.

|  |  |
| --- | --- |
| “hola”\*3 | holaholahola |
| palabra=”soy un string” | soy un string |
| print(len(palabra)) | 13 |
| print(palabra[0]) | s |
| print(palabra[-1]) | g |
| print(palabra[2]) | y |
| print(palabra[1:5]) | oy u |
| print(palabra[4:]) | un string |
| print(palabra[ :3] | soy |
| print(palabra[0:6:2]) | syu |
| print(palabra[:6:2]) | syu |
| print(palabra[0::2]) | syu tig |
| print(palabra[::-1]) | gnirts nu yos |
| print(palabra.capitalize()) | Soy un string |
| print(palabra) | Soy un string |
| print(palabra.count('o')) | 1 |
| print(palabra.endswith("tring")) | True |
| print(palabra.endswith("a")) | False |
| "tr" in palabra | True |
| print(palabra.find(“y”)) | 2 |
| print(palabra.find("ai")) | -1 |
| print(palabra.lower()) | soy un string |
| otra=palabra.upper() | --- |
| print(otra) | SOY UN STRING |
| print(otra.islower()) | False |
| print(otra.isupper()) | True |
| print(palabra.replace(“s”, “c”)) | coy un ctring |
| print(palabra) | soy un string |
| unaMas=palabra.replace(“s”, “c”) | --- |
| print(unaMas) | coy un ctring |
| palabra.startswith("s") | True |
| palabra.startswith("c") | False |
| modificada=palabra.capitalize() | ---- |
| print(modificada) | Soy un string |
| print(modificada.swapcase()) | sOY UN STRING |
| print(palabra.title()) | Soy Un String |
| lista\_letras=list(palabra) | ---- |
| print(lista\_letras) | ['s', 'o', 'y', ' ', 'u', 'n', ' ', 's', 't', 'r', 'i', 'n', 'g'] |

Prueba las instrucciones *in y not* *in* como lo hiciste en listas, por ejemplo:

>>> palabra=”corazón”

>>>’o’ in palabra

>>>’x’ not in palabra

Recuerda: Los strings son secuencias inmutables ¿qué significa que sean inmutables?

|  |  |
| --- | --- |
|  |  |

*NOTA: Muchas de las instrucciones nos dan como resultado un nuevo string pero modificado de acuerdo a lo que solicita cada instrucción. El string original no se modificará.*

1. ¿Qué utilizaríamos para obtener un string sin espacios? Por ejemplo si tenemos el string “La casa azul”, ¿Qué utilizarías para crear otro string que no tuviera los espacios: “Lacasaazul”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. ¿Qué instrucción pondrías para “esconder” una palabra, es decir, tener otra con la misma cantidad, pero de asteriscos en vez de letras? Por ejemplo: Si la palabra es “fotosíntesis”, obtener su versión “escondida” como \*\*\*\*\*\*\*\*\*\*\*\* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. ¿Qué instrucción de Python utilizarías para saber si un carácter es una vocal? Es una sola línea de código.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. a=”hipopótamo” ¿Por qué a[3]=”s” nos marcaría un error? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

¿Por qué no podemos cambiar caracteres de un string? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. ¿Qué instrucción utilizarías para obtener sólo los caracteres de índice impar de un string?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. ¿Qué instrucción utilizarías para obtener los últimos 3 caracteres de un string pero en sentido inverso? Por ejemplo, si el string es “girasol”, obtener “los” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. ¿Qué instrucción utilizarías para obtener la cantidad de vocales que tiene un string?”

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. En un programa, le pedimos al usuario que nos teclee la respuesta si o no, ¿qué instrucción utilizarías para que no importando si el usuario usó mayúsculas, minúsculas o una combinación de ellas para escribir si o no, el programa siempre identifique las respuestas?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Pensemos que estamos procesando un archivo de texto, ¿qué función de Python para manipular strings te serviría si quieres identificar palabras que comienzan con la letra “C”?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. ¿Qué otras cosas se te ocurren que puedes hacer manipulando strings?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_