

Programación con python

String

Str

El tipo de dato str representa una secuencia de caracteres.
En Python, comienzan y terminan con ", ' , "" o ""'.

Los siguientes son ejemplos de strings válidos en Python:

"hello"

"2.0"

'str'

""False""

""true""

Carácter de nueva línea - \n

El carácter de nueva línea es \n.

Este carácter es invisible cuando se imprime, pero le indica a la computadora que mueva el cursor a la siguiente línea.

```
print("hello\nworld\n!")
```

Produce la siguiente salida:

```
hello  
world  
!
```

índices (positivos y negativos)

```
s = "Python"  
print("String:", s)  
  
print("s[0]:", s[0])    # P  
print("s[3]:", s[3])    # h  
  
print("s[-1]:", s[-1])   # n  
print("s[-3]:", s[-3])
```

Slicing (rebanadas) opcional para mostrar rango

```
print("s[1:4]:", s[1:4]) # yth
```

len(): longitud del string

```
frase = "hola mundo"  
print("len(frase):", len(frase)) # 10 (incluye el espacio)
```

count(): cuántas veces aparece un substring

```
texto = "banana"  
print("texto:", texto)  
print("texto.count('a'):", texto.count('a')) # 3  
print("texto.count('na'):", texto.count('na')) # 2  
print("texto.count('x'):", texto.count('x')) # 0
```

count con rangos (inicio, fin) opcional

```
print("count('a', 2, 5):", texto.count('a', 2, 5)) # 1 (entre índices 2 y 4)
```

find(): posición de la primera ocurrencia

```
mensaje = "abracadabra"  
print("mensaje:", mensaje)
```

```
print("mensaje.find('bra'):", mensaje.find('bra')) # 1  
print("mensaje.find('ra'):", mensaje.find('ra')) # 2  
print("mensaje.find('z'):", mensaje.find('z')) # -1 (no encontrado)
```

IMPORTANTE

- find() devuelve -1 si no encuentra el substring.
- count() devuelve cuántas veces aparece (0 si no aparece).
- En strings no podemos reasignar un carácter: s[0] = 'X' daría error (inmutables).

upper(): convierte a mayúsculas

```
texto = "Hola Mundo"  
print("Texto original:", texto)
```

```
mayusculas = texto.upper()  
print("Texto en mayúsculas:", mayusculas) # HOLA MUNDO
```

lower(): convierte a minúsculas

```
frase = "PYTHON es GENIAL"  
print("Frase original:", frase)
```

```
minusculas = frase.lower()  
print("Frase en minúsculas:", minusculas) # python es genial
```

Uso combinado de upper() y lower()

```
palabra = "pYtHon"  
print("Palabra original:", palabra)  
print("Solo mayúsculas:", palabra.upper()) # PYTHON  
print("Solo minúsculas:", palabra.lower()) # python
```

Explicación de utilidad ejemplo de upper y lower

```
answer = input("What is my name? ")
```

```
if answer == "Luis":  
    print("Correct")  
else:  
    print("Incorrect")
```

Solución

```
answer = input("What is my name? ")
```

```
if answer.lower() == "luis":  
    print("Correct")  
else:  
    print("Incorrect")
```

capitalize(): primera letra en mayúscula

```
texto1 = "python es genial"
```

```
texto2 = "hOLa MUNDo"
```

```
print("Texto original:", texto1)
```

```
print("Con capitalize():", texto1.capitalize()) # Python es genial
```

```
print("Texto original:", texto2)
```

```
print("Con capitalize():", texto2.capitalize()) # Hola mundo
```

strip(): eliminar espacios al inicio y al final

```
texto = " hola mundo "
print("Texto original:", repr(texto)) # ' hola mundo '
print("Con strip():", repr(texto.strip())) # 'hola mundo'
```

```
# También se puede usar para quitar caracteres específicos
ejemplo = "----Python----"
print("Con strip('-'):", ejemplo.strip("-")) # Python
```

Uso de 'in' en strings

```
texto = "Python es genial"
```

```
# Revisar si un carácter está en el string
print("'P' in texto:", 'P' in texto)    # True
print("'z' in texto:", 'z' in texto)    # False
```

```
# Revisar si una palabra está en el string
print("'genial' in texto:", 'genial' in texto) # True
print("'java' in texto:", 'java' in texto)    # False
```

Uso de 'in' en strings

```
# También funciona con condiciones if  
if "Python" in texto:  
    print("La palabra 'Python' está en el texto")  
  
if "Java" not in texto:  
    print("La palabra 'Java' NO está en el texto")
```

isdigit(): True si todos los caracteres son dígitos

```
s = "19"
```

```
is_digit = s.isdigit()  
print(is_digit) # True
```

`isdigit()`: False si el string contiene algo distinto a dígitos

```
s = "19h"
```

```
is_digit = s.isdigit()  
print(is_digit) # False
```

isdigit() con un número decimal

```
s = "19.4"
```

```
is_digit = s.isdigit()  
print(is_digit) # False
```

Ejemplo práctico de isdigit()

```
num = input("Number: ")

if num.isdigit():
    print("It is a number")
    num = int(num)      # Convertimos el string a entero
    print(num + 5)      # Operación matemática
else:
    print("This is not an int")
```

split(): separar un string en una lista de palabras

```
s = "hello my name is luis"
```

```
words = s.split() # divide el string por espacios  
print(words)      # ['hello', 'my', 'name', 'is', 'luis']
```

split() con separador personalizado

```
s = "manzana,pera,uva,melon"
```

```
# Usamos split(",") para dividir por comas
frutas = s.split(",")
print(frutas) # ['manzana', 'pera', 'uva', 'melon']
```

split() con separador personalizado

Cuando no pasamos nada a split(), se usa el espacio como separador.

Si queremos usar otro separador (coma, punto, guion, etc.), lo ponemos entre comillas dentro de los paréntesis.

```
s = "rojo-azul-verde-amarillo"  
colores = s.split("-")  
print(colores) # ['rojo', 'azul', 'verde', 'amarillo']
```

replace(): reemplazar caracteres o cadenas en un string

```
s = "hello,my,name,is,,,luis"
```

```
# Reemplazar las comas por barras verticales  
s2 = s.replace(",","|")
```

```
print(s2) # hello|my|name|is|||luis  
print(s) # hello,my,name,is,,,luis (el string original no cambia)
```

replace(): reemplazar caracteres o cadenas en un string

```
texto = "python es divertido"  
nuevo = texto.replace("divertido", "poderoso")  
print(nuevo) # python es poderoso
```

f-strings: interpolación de variables en un string

```
# Variables
```

```
nombre = "Luis"
```

```
edad = 30
```

```
# Forma tradicional (concatenación)
```

```
print("Hola, mi nombre es " + nombre + " y tengo " + str(edad) + " años.")
```

```
# Usando f-strings (más limpio y legible)
```

```
print(f"Hola, mi nombre es {nombre} y tengo {edad} años.")
```

String multiplication (*): repetir cadenas

```
name = input("Name: ")  
  
# Repite el string 5 veces  
print(name * 5)
```

Strings multilínea con comillas triples

```
string = """hello my name is luis  
and this is a multiline string
```

```
!  
"""
```

```
print(string)
```

join(): unir elementos de una lista en un string

```
# Lista de palabras
```

```
palabras = ["Python", "es", "genial"]
```

```
# Unir con espacio
```

```
frase = " ".join(palabras)
```

```
print(frase) # Python es genial
```

```
# Unir con guiones
```

```
frase_guiones = "-".join(palabras)
```

```
print(frase_guiones) # Python-es-genial
```

```
# Unir con comas
```

```
frase_comas = ", ".join(palabras)
```

```
print(frase_comas) # Python, es, genial
```

join(): unir elementos de una lista en un string

.join() toma cada elemento de la lista (que debe ser de strings) y los une en un solo string.

Lo que pongas antes de .join() será el separador entre cada palabra.

Ejemplos:

" ".join(lista) → une con espacio.

"-".join(lista) → une con guion.

", ".join(lista) → une con coma y espacio.

Actividad

Pide al usuario que escriba una frase.

Muestra en pantalla el primer y el último carácter de esa frase.

Indica cuántos caracteres tiene la frase en total.

Pregunta al usuario un carácter y muestra cuántas veces aparece en la frase.

Pregunta al usuario si quiere ver la frase toda en mayúsculas, toda en minúsculas o con la primera letra en mayúscula, y muéstralas en ese formato.

Pide al usuario una palabra y muestra si está o no dentro de la frase.

Cambia los espacios de la frase por otro símbolo y muestra el resultado final.

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Ing. Luis Francisco Contreras González

Ejercicio 2

Pide al usuario que ingrese varias palabras separadas por un mismo símbolo (por ejemplo comas).

Convierte esa entrada en una lista de palabras.

Muestra cuántas palabras se capturaron.

Muestra la primera palabra y la última.

Pide al usuario un número que represente una posición de la lista y muestra la palabra que está ahí.

Repite esa palabra tres veces seguidas y muéstralas.

Une nuevamente todas las palabras de la lista en un solo texto, pero usando otro símbolo como separador, y muestra el resultado.

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS

Ing. Luis Francisco Contreras González

Ejercicio 3

Pide al usuario que ingrese un dato cualquiera.

Verifica si lo que ingresó es un número entero o no.

Si es un número entero, conviértelo y realiza una operación matemática sencilla con él (por ejemplo, sumarle 5).

Si no es un número entero, muestra un mensaje indicando que no es válido.

Pide al usuario que ingrese otra cadena de texto y elimina los espacios en blanco al inicio y al final.

Muestra un mensaje de varias líneas que incluya tanto el número procesado (si fue válido) como la segunda cadena ya limpia.

NO USE INTELIGENCIA ARTIFICIAL PARA SU TRABAJO
ENTREGUELA EN LA SECCION DE TAREAS