



Fundamentos de Programación

Ayudantía 7

Alexander Inostroza
Felipe Zambrano

Cadenas / Strings



En programación, una **cadena** de caracteres, palabras, ristra de caracteres o frase (**string**, en inglés) es una secuencia ordenada (de longitud arbitraria, aunque finita) de elementos que pertenecen a un cierto lenguaje formal o alfabeto análogas a una fórmula o a una oración. En general, una **cadena** de caracteres es una **sucesión de caracteres** (letras, números u otros signos o símbolos). Si no se ponen restricciones al alfabeto, una cadena podrá estar formada por cualquier combinación finita de los caracteres disponibles (las letras de la 'a' a la 'z' y de la 'A' a la 'Z', los números del '0' al '9', el espacio en blanco ' ', símbolos diversos '!', '@', '%', etcétera).

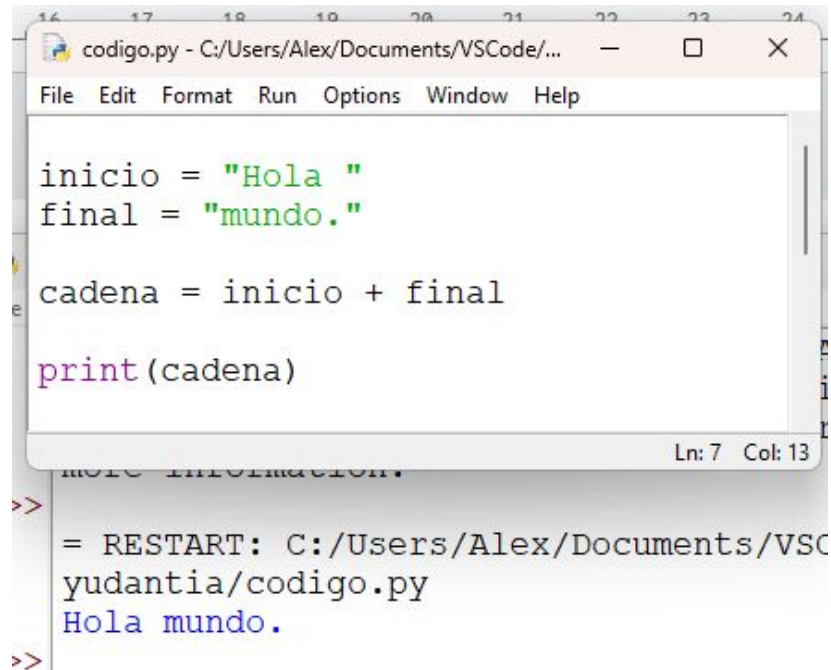
Tanto en PSeint como en Python se utilizan las comillas dobles (" ") para representar strings.

Concatenar Strings

```
1  Algoritmo sin_titulo
2
3      inicio ← "Hola "
4      final ← "mundo."
5
6      cadena ← Concatenar(inicio,final)
7
8      Escribir cadena
9
10 FinAlgoritmo
```

 PSeInt - Ejecutando proceso SIN_TITULO

```
*** Ejecución Iniciada. ***
Hola mundo.
*** Ejecución Finalizada. ***
```



The screenshot shows a VS Code window titled 'codigo.py - C:/Users/Alex/Documents/VSCo...'. The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code editor contains the following Python code:

```
inicio = "Hola "
final = "mundo."

cadena = inicio + final

print(cadena)
```

The status bar at the bottom right indicates 'Ln: 7 Col: 13'. Below the code editor, the output of the script is displayed:

```
= RESTART: C:/Users/Alex/Documents/VSCo
yudantia/codigo.py
Hola mundo.
```

Recorriendo Strings

```
frase = "CADENA"

for i in range(len(frase)):

    print(frase[i])
```

IDLE Shell 3.11.5

Edit Shell Debug Options Window Help

Python 3.11.5 (tags/v3.11.5:cce6ba9, A
:38:34) [MSC v.1936 64 bit (AMD64)] on
Type "help", "copyright", "credits" or
or more information.

>
= RESTART: C:/Users/Alex/Documents/VSC
s_ayudantia/codigo.py

C
A
D
E
N
A

```
1  Algoritmo sin_titulo
2
3      frase ← "CADENA"
4
5      Para i←0 hasta Longitud(frase) Hacer
6          |
7              Escribir Subcadena(frase,i,i)
8          |
9      FinPara
10
11  FinAlgoritmo
12
```

PSelnt - Ejecutando proceso SIN_TITULO

*** Ejecución Iniciada. ***

C
A
D
E
N
A

*** Ejecución Finalizada. ***

Funciones

```
1 SubAlgoritmo resultado ← Suma(a,b)
2   resultado ← a+b
3 FinSubAlgoritmo
4
5 Algoritmo Principal
6   Escribir Suma(7,5)
7 FinAlgoritmo
8
```

PSelnt - Ejecutando proceso PRINCIPAL

*** Ejecución Iniciada. ***

12

*** Ejecución Finalizada. ***

```
def suma(a,b):
    resultado = a+b
    return resultado
```

```
print(suma(7,5))
```

```
>>>
```

```
License ( ) for more i
= RESTART: C:/Users/A
e/fundamentos_ayudant
12
```

```
>>>
```

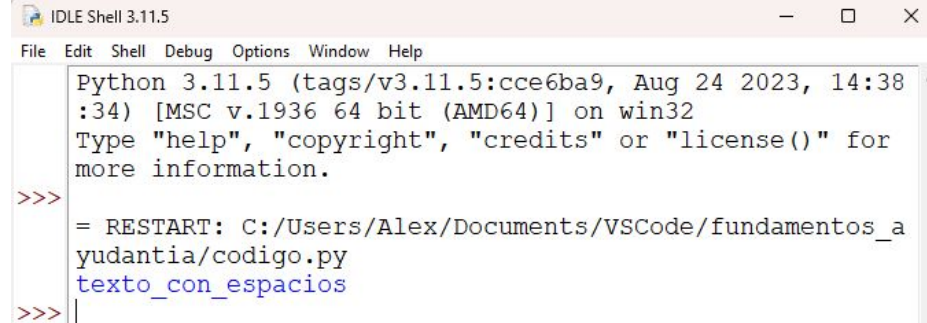
Ejemplo: quitar los espacios en un string

```
SubAlgoritmo sin_espacios ← quitar_espacios(entrada)
    sin_espacios ← ""
    para i ← 0 Hasta Longitud(entrada) Hacer
        letra ← Subcadena(entrada,i,i)
        Si letra ≠ " " Entonces
            sin_espacios ← Concatenar(sin_espacios, letra)
        FinSi
    FinPara
FinSubAlgoritmo
```

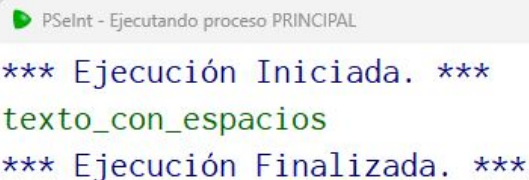
```
Algoritmo Principal
    texto_de_entrada ← " t e x t o _ c o n _ e s p a c i o s "
    Escribir quitar_espacios(texto_de_entrada)
FinAlgoritmo
```

```
def quitar_espacios(entrada):
    sin_espacios = ""
    for i in range(len(entrada)):
        letra = entrada[i]
        if letra != " ":
            sin_espacios = sin_espacios + letra
    return sin_espacios
```

```
texto_de_entrada = " t e x t o _ c o n _ e s p a c i o s "
print(quitar_espacios(texto_de_entrada))
```



```
IDLE Shell 3.11.5
File Edit Shell Debug Options Window Help
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=> RESTART: C:/Users/Alex/Documents/VSCode/fundamentos_yudantia/codigo.py
texto_con_espacios
>>>
```



```
PSeInt - Ejecutando proceso PRINCIPAL
*** Ejecución Iniciada. ***
texto_con_espacios
*** Ejecución Finalizada. ***
```

Ejercicio 1



Invertir un string: Escribe una función que tome un string como entrada y devuelva el string invertido.

Ejemplo: `invertir("hola")` debe devolver "aloh".


Ejercicio 2



Palíndromo: Escribe una función que determine si un string es un palíndromo (se lee igual al derecho y al revés, ignorando espacios y mayúsculas/minúsculas).

Ejemplo: `es_palindromo("Anita lava la tina")` debe devolver `True`.

Ejercicio 3



Contar palabras: Escribe una función que cuente el número de palabras en un string. Considera que las palabras están separadas por espacios.