

# Métodos de Strings en Python

En Python, las cadenas (strings) cuentan con una variedad de métodos incorporados que permiten modificarlas, analizarlas y evaluarlas de forma sencilla.

Método	Descripción	Ejemplo
capitalize()	Devuelve la cadena con la primera letra en mayúscula.	'hola'.capitalize() → 'Hola'
casefold()	Convierte la cadena a minúsculas más agresivamente.	'MAYÚSCULA'.casefold() → 'mayúscula'
center(n)	Centra el texto en un campo de n caracteres.	'hola'.center(10) → ' hola '
ljust(n)	Alinea a la izquierda en un campo de n caracteres.	'hola'.ljust(10) → 'hola '
rjust(n)	Alinea a la derecha en un campo de n caracteres.	'hola'.rjust(10) → ' hola'
zfill(n)	Rellena con ceros por la izquierda.	'42'.zfill(5) → '00042'
find(sub)	Devuelve el índice de la primera aparición o -1 si no existe.	'hola'.find('l') → 2
rfind(sub)	Índice de la última aparición o -1.	'hola'.rfind('l') → 2
index(sub)	Como find(), pero lanza error si no encuentra.	'hola'.index('o') → 1
rindex(sub)	Índice de la última aparición o error.	'hola'.rindex('l') → 2
count(sub)	Cuenta cuántas veces aparece un subcadena.	'banana'.count('a') → 3
isalnum()	¿ Todos los caracteres son alfanuméricos?	'abc123'.isalnum() → True
isalpha()	¿ Solo contiene letras?	'abc'.isalpha() → True
isdigit()	¿ Solo contiene dígitos?	'123'.isdigit() → True
islower()	¿ Todas las letras son minúsculas?	'abc'.islower() → True
isupper()	¿ Todas las letras son mayúsculas?	'ABC'.isupper() → True
isspace()	¿ Solo contiene espacios?	' '.isspace() → True
istitle()	¿ Está en formato título?	'Hola Mundo'.istitle() → True
lower()	Convierte a minúsculas.	'HOLA'.lower() → 'hola'
upper()	Convierte a mayúsculas.	'hola'.upper() → 'HOLA'
swapcase()	Cambia mayúsculas por	'HoLa'.swapcase() → 'hOlA'

	minúsculas y viceversa.	
title()	Convierte a formato título.	'hola mundo'.title() → 'Hola Mundo'
replace(a, b)	Reemplaza a por b.	'hola mundo'.replace('hola', 'adiós') → 'adiós mundo'
split()	Divide en lista usando espacios u otro separador.	'uno dos'.split() → ['uno', 'dos']
rsplit()	Divide desde la derecha.	'uno dos'.rsplit() → ['uno', 'dos']
splitlines()	Divide por saltos de línea.	'a\nb'.splitlines() → ['a', 'b']
join(lista)	Une elementos con un separador.	'-'.join(['a', 'b']) → 'a-b'
partition(sep)	Divide en tres partes: antes, sep y después.	'a-b'.partition('-') → ('a', '-', 'b')
rpartition(sep)	Como partition, pero desde la derecha.	'a-b-c'.rpartition('-') → ('a-b', '-', 'c')
strip()	Elimina espacios en ambos extremos.	' hola '.strip() → 'hola'
lstrip()	Elimina espacios a la izquierda.	' hola'.lstrip() → 'hola'
rstrip()	Elimina espacios a la derecha.	'hola '.rstrip() → 'hola'
startswith()	¿Empieza con cierto prefijo?	'hola'.startswith('ho') → True
endswith()	¿Termina con cierto sufijo?	'archivo.txt'.endswith('.txt') → True

## Tipos de Datos en Python con Ejemplos

### 1. Entero (int):

```
numero = 10
print(type(numero)) # <class 'int'>
```

### 2. Decimal (float)

```
decimal = 3.14
print(type(decimal)) # <class 'float'>
```

### 3. Cadena de texto (str)

```
mensaje = "Hola, Jhoan"
print(type(mensaje)) # <class 'str'>
```

### 4. Booleano (bool)

```
es_mayor = True
print(type(es_mayor)) # <class 'bool'>
```

### 5. Carácter (str de un solo carácter)

```
letra = 'A'
print(type(letra)) # <class 'str'> (en Python no hay tipo 'char', se usa 'str')
```

**6. Nulo (NoneType)**

```
vacio = None  
print(type(vacio)) # <class 'NoneType'>
```

**7. Lista (list)**

```
frutas = ["manzana", "pera", "uva"]  
print(type(frutas)) # <class 'list'>
```

**8. Tupla (tuple)**

```
coordenadas = (10.5, 20.3)  
print(type(coordenadas)) # <class 'tuple'>
```

**9. Diccionario (dict)**

```
persona = {"nombre": "Jhoan", "edad": 22}  
print(type(persona)) # <class 'dict'>
```

**10. Conjunto (set)**

```
numeros = {1, 2, 3, 4}  
print(type(numeros)) # <class 'set'>
```