

# Cadenas

Semestre 02, 2025

## Definición

---

Las cadenas son listas de caracteres que están unidos y pueden estar compuestas de letras, números, símbolos y espacios.

Son objetos **inmutables** lo que significa que al ser creadas ya no se modifica su valor (a menos que se reasigne la variable). Es posible crear nuevas cadenas a partir de cadenas existentes.

```
1  cadena1 = "Hola"
2
3
4  cadena2 = "Mundo"
5
6  cadena3 = cadena1 + " " + cadena2
7
8
9
10 print(cadena1)
11
12 print(cadena2)
13
14 print(cadena3)
15
```

Al momento de concatenar la cadena1 y la cadena2 no modificaron su valor.

## Funciones disponibles

---

<https://docs.python.org/3/library/stdtypes.html#string-methods>

Debido a que las cadenas son objetos, tienen disponibles funciones que se pueden utilizar, algunos de los más comunes son:

## len()

---

Retorna el largo (cantidad de caracteres) de una cadena.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  largo = len(cadena)
7
8
9
10 print(largo) # 10
11
```

## .isalpha()

---

Devuelve `True` si la cadena contiene solo letras (sin espacios, números ni símbolos).

```
1
2  cadena = "HolaMundo"
3
4
5
6  print(cadena.isalpha()) # True
7
8
9
10 cadena2 = "Hola Mundo"
11
12 print(cadena2.isalpha()) # False (porque tiene un espacio)
13
```

## .isdigit()

---

Devuelve `True` si la cadena contiene únicamente dígitos.

```
1
```

```
2  cadena = "12345"
3
4
5
6  print(cadena.isdigit()) # True
7
8
9
10 cadena2 = "123a"
11
12 print(cadena2.isdigit()) # False
13
```

## .upper()

---

Convierte todos los caracteres de la cadena a mayúsculas.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena.upper()) # "HOLA MUNDO"
7
```

## .lower()

---

Convierte todos los caracteres de la cadena a minúsculas.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena.lower()) # "hola mundo"
7
```

## .title()

---

Convierte la cadena a **formato de título** (cada palabra inicia con mayúscula).

```
1
2  cadena = "hola mundo desde python"
3
4
5
6  print(cadena.title()) # "Hola Mundo Desde Python"
7
```

## .count()

---

Cuenta cuántas veces aparece un carácter o subcadena en la cadena.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena.count("o")) # 2
7
8  print(cadena.count("M")) # 1
9
```

## .replace()

---

Reemplaza una subcadena por otra.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  nueva = cadena.replace("Mundo", "Python")
7
8
9
10 print(nueva) # "Hola Python"
11
```

## .find()

---

Devuelve la **posición (índice)** de la primera aparición de una subcadena. Si no la encuentra, retorna **-1**.

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena.find("M")) # 5
7
8  print(cadena.find("z")) # -1
9
```

## .rfind()

---

Similar a **.find()**, pero busca desde la **derecha** (final de la cadena).

```
1
2  cadena = "Hola Hola Mundo"
3
4
5
6  # 5 (última ocurrencia de "Hola")
7
8  print(cadena.rfind("Hola"))
9
```

# Índices

---

Los índices ayudan a ubicar los caracteres dentro de una cadena. Funcionan como lo hace el índice de un libro en el que se guarda la posición de cierto elemento.

Los índices **siempre** inician en **0** (como todo en computación).

Para las cadenas los índices están definidos desde **0** hasta el largo de la cadena **-1**.

Se pueden utilizar los índices para imprimir valores de una cadena dada su posición:

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena[0]) # imprime el primer elemento
7
8
9
10 print(cadena[1]) # imprime el segundo elemento
11
```

Es posible utilizar índices negativos para imprimir las posiciones en orden inverso:

```
1
2  cadena = "Hola Mundo"
3
4
5
6  print(cadena[-1]) # imprime el ultimo elemento
7
8  print(cadena[-2]) # imprime el penultimo elemento
9
```

## Recorrer cadenas

---

Debido a que las cadenas son listas de caracteres, esto significa que son iterables (se pueden recorrer mediante un ciclo).

Sin usar el índice:

```
1
2  cadena = "Hola Mundo"
3
4
5
6  for letra in cadena:
7
```

```
8     print(letra)
9
```

Utilizando el índice:

```
1
2     cadena = "Hola Mundo"
3
4
5
6     indice = 0
7
8     while indice < len(cadena):
9
10    print(cadena[indice])
11
12    indice += 1
13
```