**Métodos más utilizados en cadenas:**

1. **upper()**: Convierte la cadena a mayúsculas.

Código:

```
cadena = "Hola, mundo!"
cadena_mayusculas = cadena.upper() print(cadena_mayusculas)
# Resultado: "HOLA, MUNDO!"
```

2. **lower()**: Convierte la cadena a minúsculas.

Código:

```
cadena = "Hola, mundo!"
cadena_minusculas = cadena.lower() print(cadena_minusculas)
# Resultado: "hola, mundo!"
```

3. **strip()**: Elimina los espacios en blanco al inicio y al final de la cadena.

Código:

```
cadena = " Hola, mundo! "
cadena_sin_espacios = cadena.strip() print(cadena_sin_espacios)
# Resultado: "Hola,  mundo!"
```

4. **split()**: Divide la cadena en una lista de subcadenas utilizando un separador.

Código:

```
cadena = "Hola, mundo!"
subcadenas = cadena.split(", ") print(subcadenas)
# Resultado: ['Hola', 'mundo!']
```

5. **join()**: Une una lista de cadenas utilizando la cadena actual como separador.

Código:

```
subcadenas = ['Hola', 'mundo!']

cadena_unida = ", ".join(subcadenas) print(cadena_unida)

# Resultado: "Hola, mundo!"
```

6. **replace()**: Reemplaza todas las ocurrencias de una subcadena por otra subcadena.

Código:

```
cadena = "Hola, mundo!"

cadena_reemplazada = cadena.replace("mundo", "amigo") print(cadena_reemplazada)
# Resultado: "Hola, amigo!"
```

7. **startswith()**: Verifica si la cadena comienza con una subcadena específica.

Código:

```
cadena = "Hola, mundo!"

comienza_con_hola = cadena.startswith("Hola") print(comienza_con_hola)

# Resultado: True
```

8. **endswith()**: Verifica si la cadena termina con una subcadena específica.

Código:

```
cadena = "Hola, mundo!"

termina_con_exclamacion = cadena.endswith("!") print(termina_con_exclamacion)

#   Resultado: True
```

9. **find()**: Encuentra la posición de la primera aparición de una subcadena.

Código:

```
cadena = "Hola, mundo!"

posicion_mundo = cadena.find("mundo") print(posicion_mundo)

# Resultado: 6
```

10. **count()**: Cuenta el número de ocurrencias de una subcadena.

Código:

```
cadena = "Hola, mundo!"

ocurrencias_o = cadena.count("o") print(ocurrencias_o) # Resultado: 2
```

**Métodos más utilizados en números:**

1. **abs()**: Devuelve el valor absoluto de un número.
2. **round()**: Redondea un número al entero más cercano.
3. **int()**: Convierte un valor a un número entero.
4. **float()**: Convierte un valor a un número de punto flotante.
5. **str()**: Convierte un valor a una cadena de texto.
6. **max()**: Devuelve el valor máximo de una lista de números.
7. **min():** Devuelve el valor mínimo de una lista de números.
8. **sum():** Devuelve la suma de todos los elementos en una lista de números.
9. **sorted()**: Ordena una lista de números en orden ascendente.
10. **len()**: Devuelve la cantidad de elementos en una lista de números.

**Métodos más utilizados en listas:**

1. **append()**: Agrega un elemento al final de la lista.
2. **extend()**: Extiende la lista agregando todos los elementos de otra lista.
3. **insert()**: Inserta un elemento en una posición específica de la lista.
4. **remove()**: Elimina la primera aparición de un elemento de la lista.
5. **pop()**: Elimina y devuelve el último elemento de la lista.
6. **index()**: Devuelve la posición de la primera aparición de un elemento en la lista.
7. **count()**: Devuelve el número de veces que aparece un elemento en la lista.
8. **sort()**: Ordena la lista en orden ascendente.
9. **reverse()**: Invierte el orden de los elementos en la lista.

10. **len()**: Devuelve la cantidad de elementos en la lista.

Estos son solo algunos de los métodos más utilizados en Python para cadenas, números y listas. Hay muchos más métodos disponibles que puedes explorar en la documentación oficial de Python.

**Función type():**

En Python, puedes utilizar la función type() para determinar el tipo de una variable. Aquí tienes un ejemplo:

```
x = 5

y = "Hola, mundo!"

z = [1, 2, 3]


print(type(x))  # Resultado: <class 'int'>

print(type(y))  # Resultado: <class 'str'>

print(type(z))  # Resultado: <class 'list'>
```

En este ejemplo, creamos tres variables: x que contiene un número entero, y que contiene una cadena de texto, y z que contiene una lista de números. Luego, utilizamos la función type() para mostrar el tipo de cada variable. El resultado será <class 'tipo'>, donde 'tipo' representa el tipo de variable correspondiente.

Observarás que el tipo de x es **int** (entero), el tipo de y es **str** (cadena de texto) y el tipo de z es **list** (lista).

Esta es una forma útil de verificar el tipo de una variable en Python, lo cual es especialmente útil cuando necesitas asegurarte de que estás trabajando con el tipo de dato correcto en tu código.