

## MÉTODOS STRINGS

1. ``len()``: Retorna la longitud (cantidad de caracteres) de un string.

```
mensaje = "Hola, mundo!"  
print(len(mensaje))    # Salida: 13
```

2. ``upper()``: Retorna una versión en mayúsculas del string original.

```
nombre = "juan"  
print(nombre.upper())  # Salida: JUAN
```

3. ``lower()``: Retorna una versión en minúsculas del string original.

```
nombre = "JUAN"  
print(nombre.lower())  # Salida: juan
```

4. ``capitalize()``: Retorna el string con el primer carácter en mayúscula y el resto en minúsculas.

```
nombre = "juan"  
print(nombre.capitalize())  # Salida: Juan
```

5. ``strip()``: Retorna una copia del string sin espacios en blanco al inicio y al final.

``rstrip()``: Retorna una copia del string sin espacios a la derecha.

``lstrip()``: Retorna una copia del string sin espacios a la izquierda.

```
mensaje = "  Hola, mundo!  "  
print(mensaje.strip())  # Salida: "Hola, mundo!"
```

6. `split()`: Divide el string en una lista de subcadenas utilizando un delimitador.

```
frase = "Python es un lenguaje de programación"
palabras = frase.split(" ")
print(palabras)    # Salida: ['Python', 'es', 'un', 'lenguaje', 'de', 'progra
```

7. `join()`: Concatena los elementos de una lista en un solo string utilizando un separador.

```
palabras = ['Python', 'es', 'un', 'lenguaje', 'de', 'programación']
frase = " ".join(palabras)
print(frase)    # Salida: "Python es un lenguaje de programación"
```

8. `replace()`: Reemplaza una subcadena con otra en el string.

```
mensaje = "Hola, mundo!"
nuevo_mensaje = mensaje.replace("mundo", "amigo")
print(nuevo_mensaje)    # Salida: "Hola, amigo!"
```

9. `startswith()`: Verifica si el string comienza con una subcadena especificada.

```
nombre = "Juan"
print(nombre.startswith("J"))    # Salida: True
print(nombre.startswith("A"))    # Salida: False
```

10. `endswith()`: Verifica si el string termina con una subcadena especificada.

```
nombre = "Juan"
print(nombre.endswith("n"))    # Salida: True
print(nombre.endswith("a"))    # Salida: False
```

11. ``isdigit()``: Verifica si todos los caracteres del string son dígitos.

```
numero = "12345"
print(numero.isdigit())    # Salida: True

texto = "Hola123"
print(texto.isdigit())     # Salida: False
```

12. ``isalpha()``: Verifica si todos los caracteres del string son letras.

```
texto = "Hola"
print(texto.isalpha())     # Salida: True

texto2 = "Hola123"
print(texto2.isalpha())    # Salida: False
```

13. ``isalnum()``: Verifica si todos los caracteres del string son alfanuméricos (letras o dígitos).

```
cadena1 = "Hola123"
print(cadena1.isalnum())   # Salida: True

cadena2 = "Hola!"
print(cadena2.isalnum())   # Salida: False
```

14. ``count()``: Retorna la cantidad de ocurrencias de una subcadena en el string.

```
mensaje = "Hola, hola, hola!"
print(mensaje.count("hola")) # Salida: 3
```

15. `find()`: Retorna la posición de la primera ocurrencia de una subcadena en el string.

```
frase = "Python es un lenguaje de programación"
print(frase.find("lenguaje")) # Salida: 11
```

16. `rfind()`: Retorna la posición de la última ocurrencia de una subcadena en el string.

```
frase = "Python es un lenguaje de programación"
print(frase.rfind("a")) # Salida: 28
```

17. `startswith()` y `endswith()` con múltiples opciones: Verifica si el string comienza o termina con una de varias subcadenas especificadas.

```
nombre = "Juan"
print(nombre.startswith(("J", "M"))) # Salida: True

nombre2 = "Pedro"
print(nombre2.endswith(("o", "a"))) # Salida: True
```

18. `title()`: Retorna una versión del string donde la primera letra de cada palabra está en mayúscula.

```
titulo = "aprendiendo python"
print(titulo.title()) # Salida: "Aprendiendo Python"
```

19. `isupper()` e `islower()`: Verifica si todos los caracteres del string están en mayúsculas o minúsculas, respectivamente.

```
texto1 = "HOLA"  
print(texto1.isupper())    # Salida: True  
  
texto2 = "hola"  
print(texto2.islower())    # Salida: True
```

20. ``swapcase()``: Retorna una versión del string donde las letras en mayúsculas se convierten en minúsculas y viceversa.

```
texto = "Hola, Mundo!"  
print(texto.swapcase())    # Salida: "hOLA, mUNDO!"
```

Estos son solo algunos de los métodos más comunes utilizados en los strings de Python. Hay muchos otros métodos disponibles. Puedes consultar la documentación oficial de Python para obtener más detalles sobre los métodos de strings:  
<https://docs.python.org/3/library/stdtypes.html#string-methods>)