

Strings en Python

Programación I

Strings en Python

En python las cadenas de caracteres se expresan mediante la clase **str**.

La misma posee métodos que me facilitan la vida cuando se refiere al manejo de strings.

¿Qué es un Método?

Dentro del paradigma orientado a objetos un método corresponde al comportamiento del objeto en sí.

Yendo a términos reales y sin profundizar tanto un método de un objeto persona podría ser caminar(), respirar(), comer(), entre otros.

Son similares a las funciones ya que son acciones, pero con la diferencia que pertenecen a un objeto en sí.

Métodos de string

Para trabajar con cadenas de texto en Python, se emplean una serie de métodos a las variables del tipo **str**.

Métodos de string

variable.strip(): El método strip eliminará todos los caracteres vacíos del principio y del final que pueda contener la variable

variable.lower(): El método lower convertirá a las letras en minúsculas.

Método: **strip**

El método **strip** eliminará todos los caracteres vacíos del principio y del final que pueda contener la variable

```
cadena = "    Hola Mundo    "  
cadena = cadena.strip()  
print(cadena) # Hola Mundo
```

Método: **lower**

El método **lower** convertirá a las letras en minúsculas.

```
cadena = "Hola Mundo"  
cadena = cadena.lower()  
print(cadena) # hola mundo
```

Método: **upper**

El método **upper** convertirá a las letras en mayúsculas.

```
cadena = "Hola Mundo"  
cadena = cadena.upper()  
print(cadena) # HOLA MUNDO
```


Método: **capitalize**

El método **capitalize** convertirá a la primera letra de la Strings en mayúscula y el resto en minúscula.

```
cadena = "hola Mundo"  
cadena = cadena.capitalize()  
print(cadena) # Hola mundo
```

Método: **replace**

El método **replace** remplazará un conjunto de caracteres por otro.

```
cadena = "Hola Mundo"  
cadena = cadena.replace("la", "@")  
print(cadena) # Ho@ Mundo
```

Método: **split**

El método **split** divide una cadena en subcadenas y las devuelve almacenadas en una lista.

```
cadena = "Python,Java,C"  
print(cadena.split(","))  
  
#['Python', 'Java', 'C']
```

Método: **join**

El método **join** devuelve la primera cadena unida a cada uno de los elementos de la lista que se le pasa como parámetro.

```
separador = "+"  
cadena = separador.join(["A", "B", "C"])  
print(cadena) # A+B+C
```

Método: **zfill**

El método **zfill** rellena la cadena con ceros a la izquierda hasta llegar a la longitud pasada como parámetro.

```
cadena = "314"  
print(cadena.zfill(6))  
#000314
```

Método: **isalpha**

El método **isalpha** devuelve True si todos los caracteres son alfabéticos, False de lo contrario.

```
cadena = "Hola Mundo"  
print(cadena.isalpha())  
# False -> por el espacio
```

```
cadena = "HolaMundo"  
print(cadena.isalpha())  
# True
```

Método: **isalnum**

El método **isalnum** devuelve True si todos los caracteres son alfanuméricos, False de lo contrario.

```
cadena = "Hola Mundo 123"  
print(cadena.isalnum())  
# False -> por el espacio  
  
cadena = "HolaMundo123"  
print(cadena.isalnum())  
# True
```

Método: **isnumeric**

El método **isnumeric** devuelve True si todos los caracteres son numéricos (incluido números romanos, fracciones, etc), False de lo contrario. Tampoco puede haber espacios

```
cadena = "josesito2020"  
print(cadena.isnumeric())  
# False -> por el 'josesito'  
  
cadena = "2020"  
print(cadena.isnumeric())  
# True
```


Método: **count**

El método count permite contar las veces que otra cadena se encuentra dentro de la primera.

```
cadena = "Hola Mundo Hola"  
print(cadena.count("la")) # 2
```

Método: **format**

En el método format las llaves, llamadas campos de formato, son reemplazadas con los valores de las variables pasadas.

```
nombre_usuario="JUAN"  
edad_usuario=35  
cadena = "Nombre: {1}, Edad: {0}"  
print(cadena.format(edad_usuario,nombre_usuario))  
#Nombre: JUAN, Edad: 35
```

Las cadenas literales o f-strings, permiten incrustar expresiones dentro de cadenas.

```
n_usuario="JUAN"
e_usuario=35
cadena = f"Nombre: {n_usuario}, Edad: {e_usuario}"
print(cadena)
#Nombre: JUAN, Edad: 35
```

Longitud: **len**

El método **len** indica la longitud de la cadena de texto dentro de la variable en ese momento.

```
cadena = "Hola Mundo"  
print(len(cadena)) # 10
```

Slice

Cuando se crea una **slice** (rebanada), el primer número es donde comienza (inclusivo), y el segundo número de índice es donde termina (exclusivo).

```
cadena = "Hola Mundo"  
print(cadena[5:10]) # Mundo  
print(cadena[5:])  # Mundo  
print(cadena[:5])  # Hola
```