

Programación 9163

String

Prof. Antonio Carrillo Ledesma

Ayudantes:

- Diego Alberto Magallanes Ramírez
- Adriana Esmeralda Corona Mundo
- Orlando Alain Apipilhuasco Rosas

Como vimos al inicio del curso, las cadenas o strings son un tipo de dato soportado en Python y es uno de los tipos de datos más utilizados en la programación ya que nos permite comunicarnos con el usuario y obtener datos. Por la misma razón, es importante saber como trabajar con ellas. Por ejemplo, sabemos que con el operador '+' podemos concatenar dos cadenas, pero que podemos hacer si queremos convertir una cadena a mayúsculas o a minúsculas, quitar los espacios, contar el número de repeticiones de un carácter, remplazar una parte de la cadena por otra, etc.

Todas las acciones anteriores se pueden definir dentro de funciones. Comencemos con una función para pasar toda una cadena a mayúsculas.

```
In [1]: # cadena - Es la cadena a convertir.
def aMayusculas(cadena):
    new_cadena = ''

    for i in cadena:
        valor_decimal = ord(i) # ord obtiene el valor numerico de un caracter
        if valor_decimal >= 97 and valor_decimal < 123:
            i = chr(valor_decimal - 32) # chr convierte un valor entero a caracter
        new_cadena = new_cadena + i

    return new_cadena # Regresamos la nueva cadena

# Probamos la funcion:
print(aMayusculas('Programacion'))
```

PROGRAMACION

Ahora veamos el caso contrario, como convertir toda una cadena a minúsculas.

```
In [2]: # cadena - Es la cadena a convertir.
def aMinusculas(cadena):
    new_cadena = ''

    for i in cadena:
        valor_decimal = ord(i) # ord obtiene el valor numerico de un caracter
        if valor_decimal >= 65 and valor_decimal < 133:
            i = chr(valor_decimal + 32) # chr convierte un valor entero a caracter
        new_cadena = new_cadena + i

    return new_cadena # Regresamos la nueva cadena

# Probamos la funcion:
print(aMinusculas('PROGRAMACION :V'))
```

programacion :v

Pero no debemos preocuparnos por estar creando funciones para cadenas, ya que Python cuenta con una gran variedad de funciones para trabajar con ellas. La documentación de esas funciones pueden encontrarla [aquí](#).

No es necesario aprender a utilizar todas las funciones en este momento, con el tiempo iremos utilizandolas poco a poco. Algunas de las más utilizadas son:

- count
- format
- index
- lower
- replace
- upper

Ejemplos

Para los siguientes ejemplos usaremos la cadena:

```
In [3]: mi_cadena = 'Los Humanos No Pueden Vivir Sin Recuerdos... Pero Tampoco Pueden Vivir Sólo De Recuerdos'

print(mi_cadena)
```

Los Humanos No Pueden Vivir Sin Recuerdos... Pero Tampoco Pueden Vivir Sólo De Recuerdos

Ejemplo 1: Convertir una cadena a mayúsculas

```
In [4]: en_mayusculas = mi_cadena.upper()

print(en_mayusculas)
```

LOS HUMANOS NO PUEDEN VIVIR SIN RECUERDOS... PERO TAMPOCO PUEDEN VIVIR SÓLO DE RECUERDOS

Del ejemplo anterior podemos notar 2 cosas:

1. Para invocar la función es necesario hacerlo a través de una variable/objeto usando la notación punto. Al no ser funciones generales como print, input, etc. Se debe indicar de donde provienen. En este caso, de la **clase string**.
2. La cadena original no se modifica. Esto es porque la función **upper** trabaja sobre una copia de la cadena que manda a llamar al método.

Ejemplo 2: Convertir una cadena a minúsculas

```
In [5]: en_minusculas = mi_cadena.lower()

print(en_minusculas)
```

los humanos no pueden vivir sin recuerdos... pero tampoco pueden vivir sólo de recuerdos

Ejemplo 3: Quitar los espacios de una cadena

```
In [6]: sin_espacios = mi_cadena.replace(' ', '')

print(sin_espacios)
```

LosHumanosNoPuedenVivirSinRecuerdos...PeroTampocoPuedenVivirSóloDeRecuerdos

Ejemplo 4: Contar la cantidad de apariciones de una subcadena

```
In [7]: subcadena = 'e'
apariciones = mi_cadena.count('e')

print('La subcadena \'', subcadena, '\' aparece: ', apariciones, ' veces', sep='')
```

La subcadena 'e' aparece: 10 veces

```
In [8]: subcadena = 'vivir'
apariciones = mi_cadena.count('a')

print('La subcadena \'', subcadena, '\' aparece: ', apariciones, ' veces', sep='')
```

La subcadena 'vivir' aparece: 2 veces

Para la función **format** hay muchos usos. Por ejemplo:

Ejemplo 5: Concatenar una cadena

```
In [9]: print('My name is {fname}, I\'m {age}'.format(fname = "John", age = 36))

My name is John, I'm 36
```

```
In [10]: print('My name is {1}, I\'m {0}'.format("John", 36))

My name is 36, I'm John
```

```
In [11]: print('My name is {}, I\'m {}'.format("John", 36))

My name is John, I'm 36
```

Ejemplo 6: Mostrar sólo cierta cantidad de decimales

```
In [12]: print('Pi = {:.2f}'.format(3.14159))

Pi = 3.14
```