
Métodos de formato

Convertir a mayúscula la primera letra

Método: capitalize()

Retorna: una copia de la cadena con la primera letra en mayúsculas

```
>>> cadena = "bienvenido a mi aplicación"
>>> print cadena.capitalize()
Bienvenido a mi aplicación
```

Convertir una cadena a minúsculas

Método: lower()

Retorna: una copia de la cadena en minúsculas

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

Convertir una cadena a mayúsculas

Método: upper()

Retorna: una copia de la cadena en mayúsculas

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

Convertir mayúsculas a minúsculas y viceversa

Método: swapcase()

Retorna: una copia de la cadena convertidas las mayúsculas en minúsculas y viceversa

```
>>> cadena = "Hola Mundo"
>>> print cadena.swapcase()
hOLA mUNDO
```

Convertir una cadena en Formato Título

Método: title()

Retorna: una copia de la cadena convertida

```
>>> cadena = "hola mundo"
>>> print cadena.title()
Hola Mundo
```

Centrar un texto

Método: center(longitud[, "caracter de relleno"])

Retorna: una copia de la cadena centrada

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.center(50, "=")
=====Bienvenido a mi aplicación=====
>>> print cadena.center(50, " ")
        Bienvenido a mi aplicación
```

Alinear texto a la izquierda

Método: ljust(longitud[, "caracter de relleno"])

Retorna: una copia de la cadena alineada a la izquierda

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.ljust(50, "=")
Bienvenido a mi aplicación=====
```

Alinear texto a la derecha

Método: rjust(longitud[, "caracter de relleno"])

Retorna: una copia de la cadena alineada a la derecha

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.rjust(50, "=")
=====Bienvenido a mi aplicación
>>> print cadena.rjust(50, " ")
        Bienvenido a mi aplicación
```

Rellenar un texto anteponiendo ceros

Método: zfill(longitud)

Retorna: una copia de la cadena rellena con ceros a la izquierda hasta alcanzar la longitud final indicada

```
>>> numero_factura = 1575
>>> print str(numero_factura).zfill(12)
000000001575
```

Métodos de Búsqueda

Contar cantidad de apariciones de una subcadena

Método: `count("subcadena", [posicion_inicio, posicion_fin])`

Retorna: un entero representando la cantidad de apariciones de *subcadena* dentro de *cadena*

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.count("a")
3
```

Buscar una subcadena dentro de una cadena

Método: `find("subcadena", [posicion_inicio, posicion_fin])`

Retorna: un entero representando la posición donde inicia la subcadena dentro de *cadena*. Si no la encuentra, retorna -1

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.find("mi")
13
>>> print cadena.find("mi", 0, 10)
-1
```

Métodos de Validación

Saber si una cadena comienza con una subcadena determinada

Método: `startswith("subcadena"[, posicion_inicio, posicion_fin])`

Retorna: True o False

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.startswith("Bienvenido")
True
>>> print cadena.startswith("aplicación")
False
>>> print cadena.startswith("aplicación", 16)
True
```

Saber si una cadena finaliza con una subcadena determinada

Método: `endswith("subcadena"[, posicion_inicio, posicion_fin])`

Retorna: True o False

```
>>> cadena = "bienvenido a mi aplicación".capitalize()
>>> print cadena.endswith("aplicación")
True
>>> print cadena.endswith("Bienvenido")
False
>>> print cadena.endswith("Bienvenido", 0, 10)
True
```

Saber si una cadena es alfanumérica

Método: `isalnum()`

Retorna: True o False

```
>>> cadena = "pepegrillo 75"
>>> print cadena.isalnum()
False
>>> cadena = "pepegrillo"
>>> print cadena.isalnum()
True
>>> cadena = "pepegrillo75"
>>> print cadena.isalnum()
True
```

Saber si una cadena es alfabética

Método: `isalpha()`

Retorna: True o False

```
>>> cadena = "pepegrillo 75"
>>> print cadena.isalpha()
False
>>> cadena = "pepegrillo"
>>> print cadena.isalpha()
True
>>> cadena = "pepegrillo75"
>>> print cadena.isalpha()
False
```

Saber si una cadena es numérica

Método: `isdigit()`

Retorna: True o False

```
>>> cadena = "pepegrillo 75"
>>> print cadena.isdigit()
False
>>> cadena = "7584"
>>> print cadena.isdigit()
True
>>> cadena = "75 84"
>>> print cadena.isdigit()
False
>>> cadena = "75.84"
>>> print cadena.isdigit()
False
```

Saber si una cadena contiene solo minúsculas

Método: `islower()`

Retorna: True o False

```
>>> cadena = "pepe grillo"
>>> print cadena.islower()
True
>>> cadena = "Pepe Grillo"
>>> print cadena.islower()
False
>>> cadena = "Pepegrillo"
>>> print cadena.islower()
False
>>> cadena = "pepegrillo75"
>>> print cadena.islower()
False
```

True

Saber si una cadena contiene solo mayúsculas

Método: `isupper()`

Retorna: True o False

```
>>> cadena = "PEPE GRILLO"
>>> print cadena.isupper()
True
>>> cadena = "Pepe Grillo"
>>> print cadena.isupper()
False
>>> cadena = "Pepegrillo"
>>> print cadena.isupper()
False
>>> cadena = "PEPEGRILLO"
>>> print cadena.isupper()
True
```

Saber si una cadena contiene solo espacios en blanco

Método: `isspace()`

Retorna: True o False

```
>>> cadena = "pepe grillo"
>>> print cadena.isspace()
False
>>> cadena = "      "
>>> print cadena.isspace()
True
```

Saber si una cadena tiene Formato De Título

Método: `istitle()`

Retorna: True o False

```
>>> cadena = "Pepe Grillo"
>>> print cadena.istitle()
True
>>> cadena = "Pepe grillo"
>>> print cadena.istitle()
False
```