Métodos principales de cadenas

Cuando creamos una cadena de caracteres estamos creando un objeto de la clase `str`, que tiene definido un conjunto de métodos:

```
 cadena.capitalize    cadena.isalnum       cadena.join          cadena.rsplit
cadena.casefold      cadena.isalpha       cadena.ljust         cadena.rstrip
cadena.center        cadena.isdecimal     cadena.lower         cadena.split
cadena.count         cadena.isdigit       cadena.lstrip        cadena.splitlines
cadena.encode        cadena.isidentifier  cadena.maketrans     cadena.startswith
cadena.endswith      cadena.islower       cadena.partition     cadena.strip
cadena.expandtabs    cadena.isnumeric     cadena.replace       cadena.swapcase
cadena.find          cadena.isprintable   cadena.rfind         cadena.title
cadena.format        cadena.isspace       cadena.rindex        cadena.translate
cadena.format_map    cadena.istitle       cadena.rjust         cadena.upper
cadena.index         cadena.isupper       cadena.rpartition    cadena.zfill
```

Métodos de formato

```
 >>> cad = "hola, como estás?"
>>> print(cad.capitalize())
Hola, como estás?

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

>>> cad = "hola mundo"
>>> print(cad.upper())
HOLA MUNDO

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

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

>>> print(cad.center(50))
                    hola mundo
>>> print(cad.center(50,"="))
====================hola mundo====================

>>> print(cad.ljust(50,"="))
hola mundo========================================
>>> print(cad.rjust(50,"="))
========================================hola mundo

>>> num = 123
>>> print(str(num).zfill(12))
000000000123
```

Métodos de búsqueda

```
>>> cad = "bienvenido a mi aplicación"
>>> cad.count("a")
3
>>> cad.count("a",16)
2
>>> cad.count("a",10,16)
1

>>> cad.find("mi")
13
>>> cad.find("hola")
-1

>>> cad.rfind("a")
21
```

El método `index()` y `rindex()` son similares a los anteriores pero provocan una excepción `ValueError` cuando no encuentra la subcadena.

Métodos de validación

```
>>> cad.startswith("b")
True
>>> cad.startswith("m")
False
>>> cad.startswith("m",13)
True

>>> cad.endswith("ción")
True
>>> cad.endswith("ción",0,10)
False
>>> cad.endswith("nido",0,10)
True
```

Otras funciones de validación: `isalnum()`, `isalpha()`, `isdigit()`, `islower()`, `isupper()`, `isspace()`, `istitle()`,...

Métodos de sustitución

`format`

En la unidad **"Entrada y salida estándar"** ya estuvimos introduciendo el concepto de formateo de la cadenas. Estuvimos viendo que hay dos métodos y vimos algunos ejemplos del *nuevo estilo* con la función predefinida `format()`.

El uso del estilo nuevo es actualmente el recomendado (puedes obtener más información y ejemplos en algunos de estos enlaces: enlace1 y enlace2) y obtiene toda su potencialidad usando el método `format()` de las cadenas. Veamos algunos ejemplos:

```
>>> '{} {}'.format("a", "b")
'a b'
>>> '{1} {0}'.format("a", "b")
'b a'
>>> 'Coordenadas: {latitude}, {longitude}'.format(latitude='37.24N', longitude='-115.81W')
'Coordenadas: 37.24N, -115.81W'
>>> '{0:b} {1:x} {2:.2f}'.format(123, 223,12.2345)
'1111011 df 12.23'
>>> '{:^10}'.format('test')
'   test   '
```

Otros métodos de sustitución

```
>>> buscar = "nombre apellido"
>>> reemplazar_por = "Juan Pérez"
>>> print ("Estimado Sr. nombre apellido:".replace(buscar, reemplazar_por))
Estimado Sr. Juan Pérez:

>>> cadena = "   www.eugeniabahit.com   "
>>> print(cadena.strip())
www.eugeniabahit.com
>>> cadena="00000000123000000000"
>>> print(cadena.strip("0"))
123
```

De forma similar `lstrip(["caracter"])` y `rstrip(["caracter"])`.

Métodos de unión y división

```
>>> formato_numero_factura = ("Nº 0000-0", "-0000 (ID: ", ")"
>>> print("275".join(formato_numero_factura))
Nº 0000-0275-0000 (ID: 275)

>>> hora = "12:23"
>>> print(hora.rpartition(":"))
('12', ':', '23')
>>> print(hora.partition(":"))
('12', ':', '23')
>>> hora = "12:23:12"
>>> print(hora.partition(":"))
('12', ':', '23:12')
>>> print(hora.split(":"))
['12', '23', '12']
>>> print(hora.rpartition(":"))
('12:23', ':', '12')
>>> print(hora.rsplit(":",1))
['12:23', '12']


>>> texto = "Linea 1\nLinea 2\nLinea 3"
>>> print(texto.splitlines())
['Linea 1', 'Linea 2', 'Linea 3']
```