

05-string

May 25, 2020

1 String

Las cadenas en python están rodeados por comillas simples o comillas dobles.

'Hola' es lo mismo que "Hola"

Se puede mostrar la cadena directamente con la función *print()*

```
print("Hola Mundo")
```

1.1 Ejemplo

```
[1]: # cadena de texto
a = "Hola Data Analytics"
print(a)
```

Hola Data Analytics

```
[2]: # cadena de texto multilinea
a = """Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
"""
print(a)
```

```

Lorem ipsum dolor sit amet,
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua.
```

1.2 Concatenar cadena de texto

En python existe 3 forma de concatenar.

1.3 Ejemplo

```
[3]: # concatenar con el símbolo +
nombre = "Yerson"
edad = 26
concatenacion = "Tu nombre es: " + nombre + " y tienes " + str(edad) + " año(s)"
concatenacion
```

```
[3]: 'Tu nombre es: Yerson y tienes 26 año(s)'
```

```
[4]: # concatenar con el método format
concatenacion = "Tu nombre es: {0} y tienes: {1} año(s)".format(nombre, edad)
concatenacion
```

```
[4]: 'Tu nombre es: Yerson y tienes: 26 año(s)'
```

```
[5]: # concatenar con la cadena F
concatenacion = f"Tu nombre es: {nombre} y tienes {edad} año(s)"
concatenacion
```

```
[5]: 'Tu nombre es: Yerson y tienes 26 año(s)'
```

1.4 Métodos en string

Python tiene el comando `dir` que nos ayuda a saber qué método tenemos disponible para x tipo de variables. Por ejemplo, queremos saber los métodos que tiene string usamos `dir("hola")`

```
[6]: # mostrar los métodos de un string
dir("Hola")
```

```
[6]: ['__add__',
      '__class__',
      '__contains__',
      '__delattr__',
      '__dir__',
      '__doc__',
      '__eq__',
      '__format__',
      '__ge__',
      '__getattr__',
      '__getitem__',
      '__getnewargs__',
      '__gt__',
      '__hash__',
      '__init__',
      '__init_subclass__',
      '__iter__',
      '__le__',
```

```
'__len__',
'__lt__',
'__mod__',
'__mul__',
'__ne__',
'__new__',
'__reduce__',
'__reduce_ex__',
'__repr__',
'__rmod__',
'__rmul__',
'__setattr__',
'__sizeof__',
'__str__',
'__subclasshook__',
'capitalize',
'casefold',
'center',
'count',
'encode',
'endswith',
'expandtabs',
'find',
'format',
'format_map',
'index',
'isalnum',
'isalpha',
'isascii',
'isdecimal',
'isdigit',
'isidentifier',
'islower',
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'replace',
'rfind',
'rindex',
```

```
'rjust',
'partition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'zfill']
```

¡Podemos observar que existen muchos! Vamos a explicar algunos métodos, “los más usuados”

- **upper()** Convierte el texto a mayúscula
- **lower()** Convierte el texto a minúscula
- **strip()** Elimina cualquier espacio en blanco desde el principio o el final
- **split()** Divide la cadena en subcadenas si encuentra instancias del separador
- **replace()** Reemplaza una cadena con otra cadena.

1.5 Ejemplo

```
[7]: texto = " Tigo Analytics "
# convertir el texto todo en minúscula
print(f"Cadena en minúscula: {texto.lower()}")
# convertir el texto todo en mayúscula
print(f"Cadena en minúscula: {texto.upper()}")
# eliminar espacio en blanco al principio y/o al final
texto = texto.strip()
print(f"Cadena en minúscula: {texto.strip()}")
# reemplazar la palabra Tigo por Data
texto = texto.replace('Tigo', 'Data')
print(f"Cadena en minúscula: {texto}")
```

```
Cadena en minúscula: tigo analytics
Cadena en minúscula: TIGO ANALYTICS
Cadena en minúscula: Tigo Analytics
Cadena en minúscula: Data Analytics
```

```
[8]: # separar texto por espacios
texto_separado = texto.split(" ")
texto_separado
```

```
[8]: ['Data', 'Analytics']
```

Aquí podemos ver todas las descripciones de los métodos en un string