

# Métodos String en Python (con ejemplos)

## Parte #2



Luis Salcedo

24/02/2019 - Actualizado: 22/12/2020

Hola amigos de Internet. Mi nombre es Luis y les doy la bienvenida a Mi Diario Python.

En este artículo seguiremos viendo más métodos para manejar y manipular cadenas de texto.

Puedes ver la primera parte ingresando

a: <https://www.pythondiario.com/2019/02/metodos-string-en-python-con-ejemplos.html>

Sin más preámbulos, comencemos.



[Índice](#)

# Métodos String

## string.replace(string\_viejo, string\_nuevo)

El método **replace()** devuelve una copia de la cadena con la subcadena vieja remplazada por una nueva. Veamos un ejemplo para entenderlo mejor:

```
>>> cadena = "Hola Mundo"
>>> cadena.replace("Mundo", "Internet")
'Hola Internet'
```

## string.partition(char)

El método **partition()** divide la cadena en la primera aparición de **char** y devuelve una tupla que contiene la parte anterior a **char**, el mismo **char**, y la parte posterior de **char**. Suena un poco confuso, veamos un ejemplo:

```
>>> cadena = "Hola Mundo"
>>> cadena.partition("la")
('Ho', 'la', ' Internet')
```

## string.title()

El método **title()** devuelve una copia de la cadena donde las palabras comienzan con una letra mayúscula. Veamos un ejemplo:

```
>>> cadena = "mi diario python"
>>> cadena.title()
'Mi Diario Python'
```



## string.swapcase()

El método **swapcase()** devuelve una copia de la cadena con los caracteres en mayúsculas convertidos en minúsculas y viceversa.

### Ejemplo:

```
>>> cadena = "Mi Diario Python"
>>> cadena.swapcase()
'mI dIARIO pYTHON'
```

## str.startswith(prefijo)

El método **startswith()** devuelve **True** si la cadena comienza con el prefijo, de lo contrario devuelve **False**.

### Ejemplo:

```
>>> cadena = "Mi Diario Python"
>>> cadena.startswith("Mi")
True
```

## string.split(sep)

El método **split()** devuelve una lista de las palabras en la cadena, utilizando a **sep** como la cadena delimitadora.

### Ejemplo:

```
>>> cadena = "Luis,Jose,Maria,Sofia,Miguel"
>>> cadena.split(",")
['Luis', 'Jose', 'Maria', 'Sofia', 'Miguel']
```

## string.zfill(ancho)

El método **zfill()** devuelve una copia de la cadena que se rellena con **0** dígitos ASCII para hacer una cadena de *ancho* de longitud .

### Ejemplo:

```
>>> "356".zfill(6)
'000356'
```



¿Alguna duda? ¿Alguna sugerencia? No olvides dejarnos tu comentario.

Sin más nada que decir mi nombre es Luis y fue un placer compartir mis conocimientos con todos ustedes :D.

cadenas

metodos

String en python



Luis Salcedo

< 5 minutos o menos: array - Arreglos eficientes de valores numéricos

Métodos String en Python (con ejemplos) >



Ejercicios graficos de programacion Python (Parte 1)



Analizador de Texto - Programas Python

Como instalar python

```
GET /index.html HTTP/1.1
Host: www.ejemplo.com
User-Agent: Mozilla/4.5 [en]
Accept: image/gif, image/jpeg, text/html
Accept-language: en
Accept-Charset: iso-8859-1
```



El nacimiento de la World Wide Web y los Fundamentos de la web (HTTP

y HTML)

## Métodos Mágicos - Programación en Python



## Ejercicios resueltos en python (Parte 2)

Deja una respuesta

Publicar el comentario



Anónimo

febrero 24, 2019 a las 5:44 pm

Excelente muy buena la explicación.

Responder





Luis Salcedo

febrero 24, 2019 a las 5:46 pm

Muchas gracias. Saludos :D.

Responder



Nombre Apellido

febrero 24, 2019 a las 9:14 pm

Bién definido, Luis 😊

Responder

# Métodos String en Python (con ejemplos)



Luis Salcedo

21/02/2019 - Actualizado: 22/12/2020

Hola amigos de Internet. Mi nombre es Luis, y les doy la bienvenida a Mi Diario Python.

En el artículo hoy veremos los diferentes métodos, que Python nos proporciona, para manejar y manipular cadenas de texto.

Así que prepara tu mejor taza de café, y comencemos.





Índice



## Métodos String

### string.capitalize()

El método **capitalize()** devuelve una copia de la cadena con su primera letra en mayúscula.  
**Ejemplo:**







```
>>> texto.find("m")
0
>>> texto.find("i")
1
>>> texto.find("python")
10
>>> texto.find("py")
10
>>> texto.find("p y")
-1
```

## string.format()

Nos permite realizar una operación de formato de cadena. La cadena en la que se llama a este método puede contener texto literal o campos de reemplazo delimitados por llaves `{}`. Cada campo de reemplazo contiene el índice numérico de un argumento posicional o el nombre de un argumento de palabra clave. Devuelve una copia de la cadena donde cada campo de reemplazo se reemplaza con el valor de la cadena del argumento correspondiente.

**Ejemplo:**

```
>>> "La suma de 1 + 2 es {0}".format(1+2)
'La suma de 1 + 2 es 3'
```

## string.index(sub)

El método **index** es muy similar al método **find**. Con la diferencia de que cuando no se encuentra la subcadena, **index** lanza un **ValueError**.

```
>>> texto = "mi diario python"
>>> texto.index("mi")
0
>>> texto.index("PYTHON")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ValueError: substring not found
>>>
```

## string.isalpha()

Devuelve verdadero si todos los caracteres de la cadena son alfanuméricos y hay al menos un carácter, de lo contrario es falso.

**Ejemplo:**



```
>>> texto = "mi diario python"
>>> texto.isalpha()
False
>>> "midiariopython".isalpha()
True
```

Si te preguntas ¿por que “mi diario python” a lanzado False?. Es porque los espacios no son un carácter alfanumérico.

## string.isdigit()

El método **isdigit()** devuelve **True** si todos los caracteres de la cadena son dígitos.

**Ejemplo:**

```
>>> texto = "mi diario python"
>>> digitos = "12345"
>>> texto.isdigit()
False
>>> digitos.isdigit()
True
```

## string.isspace()

El método **isspace()** devuelve **True** si solo hay caracteres de espacio en blanco.

**Ejemplo:**

```
>>> " ".isspace()
True
>>> " a".isspace()
False
```

## string.lower()

El método **lower** devuelve una copia de la cadena con todos sus caracteres en minúsculas.

**Ejemplo:**



```
>>> "Hola Mundo".lower()
'hola mundo'
>>> "PYTHON".lower()
'python'
```

## string.upper()

El método **upper()** devuelve una copia de la cadena con todos sus caracteres en mayúsculas.

**Ejemplo:**

```
>>> texto = "mi diario python"
>>> texto.upper()
'MI DIARIO PYTHON'
```

## string.lstrip(chars)

El método **lstrip** devuelve una copia de la cadena con los caracteres iniciales eliminados. El argumento *chars* es una cadena que especifica el conjunto de caracteres que se eliminarán.

**Ejemplo:**

```
>>> web = "www.pythondiario.com"
>>> web.lstrip("w.")
'pythondiario.com'
>>>
```

Estos son algunos de los métodos de **string**. Puedes verlos todos en <https://docs.python.org/3.6/library/stdtypes.html#string-methods>.

¿Quieres una segunda parte **Método String**? ¿Alguna duda? Déjanos tu comentario.

Sin más nada que decir. Mi nombre es Luis y fue un placer compartir mis conocimientos con todos ustedes :D.

[metodos](#)[string](#)[String en python](#)



Luis Salcedo

◀ Métodos String en Python (con ejemplos) Parte #2

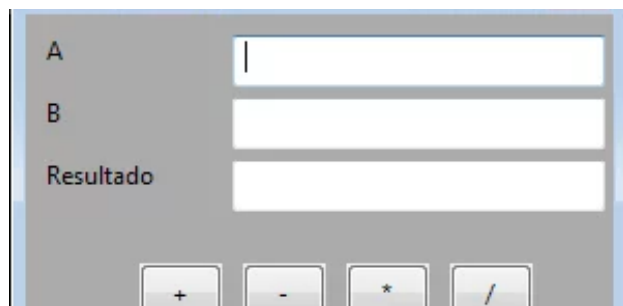
5 minutos o menos: Generar archivos y directorios temporales con Python. ▶



Crear un ejecutable .exe en python con cx\_Freeze



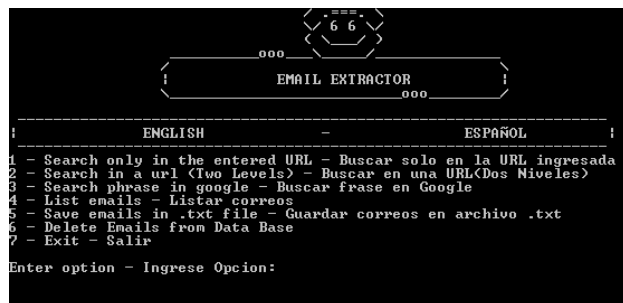
Resolviendo ecuaciones de segundo grado con Python



Mini-Curso de wxPython 4.  
Agregando controles



Como instalar Django



Introducción a las Redes Neuronales  
- Parte #1: Topología de la Redes Neuronales



## Extraer correos electrónicos de páginas web con python

### Deja una respuesta

---



Androidrh

febrero 22, 2019 a las 7:17 pm

Excelente, ojala segunda parte



Jorge Ayona Inglis

febrero 23, 2019 a las 10:06 am

Excelente!! segunda parte por favor...



Nombre Apellido

febrero 23, 2019 a las 3:01 pm



Pues sí, siempre es mejor una segunda parte que una documentación oficial un poco confusa y sin ejemplos claros.

Responder



zunga

mayo 5, 2020 a las 11:18 am

si es verdad las documentaciones son muy confusas

Responder

Instalar Python y Django

Consejos para debuggear con print()



pdf a jpg: Cómo convertir un archivo usando python





GitHub Copilot, ejemplos en Python: ¿Realmente nos ayuda a programar?



The install worked successfully! Congratulations!

You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

Cómo instalar y configurar Django en Ubuntu 20.04

¿Necesitas un Software a medida?

Newsletter

Email

Suscribirse

Mi Diario Python > 2019 > febrero > Métodos String en Python (con ejemplos) Parte #2

Email

Suscribirse

SIGUENOS



