



# Fundamentos de Programación con Python

## Unidad V Cadena de caracteres

**Docente: T.S.U Gerardo Alí Ferraro Schelijasch**  
**[gerferr83@soltecferr.com](mailto:gerferr83@soltecferr.com)**  
**<https://soltecferr.com>**

# Unidad V – Cadena de caracteres

---

## Resumen

- Cadena de caracteres
- Indicación
- Slicing
- Métodos
  - `capitalize()`
  - `lower()`
  - `upper()`
  - `strip()`
  - `replace()`
  - `swapcase()`
  - `count()`
  - `isalnum()`
  - `isalpha()`
  - `zfill()`
  - `split()`

# Unidad V – Cadena de caracteres

---

## Cadena de caracteres

Una cadena de caracteres (o string) es una secuencia de caracteres encerrada entre comillas simples `'...'` o comillas dobles `"..."`. Las cadenas son inmutables, lo que significa que no pueden ser modificadas después de ser creadas.

Ejemplo

```
cadena1 = "Hola, Mundo!"
```

```
cadena2 = 'Python es divertido'
```

# Unidad V – Cadena de caracteres

---

## Indexación

La indexación en Python permite acceder a caracteres individuales en una cadena usando índices. Los índices comienzan en 0 para el primer carácter y van incrementándose por cada carácter siguiente. También se puede usar indexación negativa para acceder a caracteres desde el final de la cadena. Ejemplo:

```
cadena = "Python"
```

```
# Acceso con indexación positiva
```

```
print(cadena[0]) # P
```

```
print(cadena[3]) # h
```

```
# Acceso con indexación negativa
```

```
print(cadena[-1]) # n
```

```
print(cadena[-4]) # t
```

# Unidad V – Cadena de caracteres

---

## Slicing:

El slicing permite obtener una subcadena (o segmento) de una cadena original especificando un rango de índices. La sintaxis para realizar slicing es:

`cadena[inicio:fin:paso]`

- inicio: El índice donde comienza el slicing (incluido).
- fin: El índice donde termina el slicing (excluido).
- paso: El intervalo entre índices. Es opcional y su valor por defecto es 1.

# Unidad V – Cadena de caracteres

---

## Slicing:

```
cadena = "Python es divertido"
```

```
# Subcadena desde el índice 0 hasta el 6 (sin incluir el 6)  
print(cadena[0:6]) # Python
```

```
# Subcadena desde el índice 7 hasta el final  
print(cadena[7:]) # es divertido
```

```
# Subcadena desde el inicio hasta el índice 6 (sin incluir el 6)  
print(cadena[:6]) # Python
```

```
# Subcadena con un paso de 2  
print(cadena[::2]) # Pto sdieio
```

```
# Subcadena en orden inverso
```

```
print(cadena[::-1]) # oditrevid se nohtyP
```



# Unidad V – Cadena de caracteres

---

## Slicing:

- Si omites el inicio, Python comienza desde el inicio de la cadena.
- Si omites el fin, Python continúa hasta el final de la cadena.
- Usar un paso negativo invierte el orden de los caracteres en la subcadena.

## Métodos:

Para poder manipular una cadena de caracteres es necesario usar los métodos que nos da el lenguaje Python los mas usados son:

# Unidad V – Cadena de caracteres

---

## Métodos:

- `capitalize()`:

Convierte el primer carácter de la cadena en mayúscula y los demás en minúscula.

```
cadena = "python es divertido"  
print(cadena.capitalize()) # Python es divertido
```

- `Lower()`:

Convierte todos los caracteres de la cadena a minúsculas.

```
cadena = "PYTHON ES DIVERTIDO"  
print(cadena.lower()) # python es divertido
```



# Unidad V – Cadena de caracteres

---

## Métodos:

- `upper()`:

Convierte todos los caracteres de la cadena a mayúsculas.

```
cadena = "python es divertido"  
print(cadena.upper()) # PYTHON ES DIVERTIDO
```

- `strip()`:

Elimina los espacios en blanco al inicio y al final de la cadena. También puede eliminar caracteres específicos si se le pasa un argumento.

# Unidad V – Cadena de caracteres

---

## Métodos:

- `strip()`

```
cadena = " Python es divertido "  
print(cadena.strip()) # "Python es divertido"
```

```
# Eliminando caracteres específicos  
cadena = "...Python..."  
print(cadena.strip('.') ) # "Python"
```

- `replace()`:

Reemplaza todas las ocurrencias de una subcadena por otra.

# Unidad V – Cadena de caracteres

---

## Métodos:

- `replace()`:

```
cadena = "Python es divertido"  
print(cadena.replace("divertido", "genial")) # Python es genial
```

- `swapcase()`:

Intercambia las mayúsculas por minúsculas y viceversa en toda la cadena.

```
cadena = "Python Es Divertido"  
print(cadena.swapcase()) # pYTHON eS dIVERTIDO
```

# Unidad V – Cadena de caracteres

---

## Métodos:

- `count()`:

Cuenta el número de veces que una subcadena aparece en la cadena.

```
cadena = "Python es divertido. Python es potente."  
print(cadena.count("Python")) # 2
```

- `isalnum()`:

Verifica si todos los caracteres en la cadena son alfanuméricos (letras o números). Devuelve True si es así, de lo contrario False.

# Unidad V – Cadena de caracteres

---

## Métodos:

- `isalnum()`:

```
cadena = "Python123"  
print(cadena.isalnum()) # True
```

```
cadena = "Python 123"  
print(cadena.isalnum()) # False (debido al espacio)
```

- `isalpha()`:

Verifica si todos los caracteres en la cadena son letras. Devuelve True si es así, de lo contrario False.

# Unidad V – Cadena de caracteres

---

## Métodos:

- `isalpha()`:

```
cadena = "Python"  
print(cadena.isalpha()) # True
```

```
cadena = "Python123"  
print(cadena.isalpha()) # False (debido a los números)
```

- `zfill()`:

Rellena la cadena con ceros a la izquierda hasta alcanzar la longitud especificada.



# Unidad V – Cadena de caracteres

---

## Métodos:

- `zfill()`:

```
cadena = "42"  
print(cadena.zfill(5)) # 00042
```

- `split()`:

Divide la cadena en una lista de subcadenas, utilizando un delimitador. Si no se especifica un delimitador, se usa el espacio por defecto.

# Unidad V – Cadena de caracteres

---

## Métodos:

- `split()`:

```
cadena = "Python es divertido"  
print(cadena.split()) # ['Python', 'es', 'divertido']
```

```
# Usando un delimitador específico  
cadena = "Python-es-divertido"  
print(cadena.split('-')) # ['Python', 'es', 'divertido']
```

## Unidad V – Cadena de caracteres

---

### Términos de la licencia.

- This work is licensed under the creative commons Attribution-shareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA,
- Este trabajo se otorga bajo los términos de la licencia Creative Commons Attribution-shareAlike License. Para obtener una copia de esta licencia visita <http://creativecommons.org/licenses/by-sa/4.0> o envía una carta a la dirección Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.