





# PYTHON

## **Cadenas de Caracteres Funciones y Métodos**

**Lic. Gustavo Bianchi**

Junio 2023

# Contenido

---

- Explicación de uso de **algunas funciones** que podemos utilizar con cadenas de caracteres.
- Explicación de uso de **algunos métodos** que les podemos aplicar a las cadenas de caracteres.

# Cadenas: Algunas funciones aplicables

```
>>> palabra = "Algoritmo"  
>>> len(palabra)  
9
```

*La función **len()**, nos devuelve la cantidad de caracteres que hay en la cadena*

```
>>> valor = "2023"  
>>> int(valor)  
2023
```

*La función **int()**, nos devuelve en caso que pueda convertir la cadena a número entero, el número entero; sino dará error.*

# Cadenas: Algunas funciones aplicables

```
>>> min("Zapato")  
Z
```

La función **min()**, nos **devuelve el menor de los caracteres**, según su valor en la tabla ASCII, por eso en este caso, la “Z” mayúscula, está antes en la tabla que la “a” minúscula

```
>>> max("Zapato")  
t
```

La función **max()**, de forma análoga que la **min()**, nos **devuelve el mayor de los caracteres**, según su valor en la tabla ASCII, por eso en este caso, nos devuelve la “t” minúscula, que ocupa la mayor posición en la tabla ASCII, entre los caracteres que componen la cadena.

# Cadenas: Algunos métodos aplicables

```
>>> palabra = "Algoritmo"  
>>> palabra.count("o")  
2
```

*Método **str.count()***

*Nos devuelve la cantidad de letras "o" que hay en la cadena*

```
>>> palabra = "encadenado"  
>>> palabra.count("en")  
2
```

*Método **str.count()***

*Nos devuelve la cantidad de subcadenas "en" que hay en la cadena*

```
>>> palabra.count("en", 0, 3)  
1
```

*Método **str.count()** alternativa*

*Nos devuelve la cantidad de subcadenas "en" que hay en la cadena entre las posiciones 0 y 3*

# Cadenas: Algunos métodos aplicables

```
>>> palabra = "Algoritmo"
```

```
>>> palabra.index("o")
```

```
3
```

*Método **str.index()***

*Nos devuelve la primer posición de la letra "o" en la cadena. Si la letra no se encuentra, dará error.*

```
>>> palabra = "Algoritmo"
```

```
>>> palabra.index("i", 4, 7)
```

```
5
```

*Método **str.index()** alternativa*

*Se puede indicar entre cuáles posiciones buscar el caracter.*

```
>>> palabra.find("ori")
```

```
3
```

*Método **str.find()***

*Nos devuelve la posición en la que comienza la subcadena "ori", ó -1 si no está incluida.*

Note que la diferencia entre usar el método index o el método find, será que el primero en caso de no encontrar lo que se busca devuelve error; mientras que find, devuelve -1.

# Cadenas: Algunos métodos aplicables

```
>>> palabra = "Algoritmo"  
>>> palabra.upper()  
'ALGORITMO'
```

*Método str.upper()*  
*Nos devuelve la cadena completa convertida a mayúsculas*

```
>>> palabra.lower()  
'algoritmo'
```

*Método str.lower()*  
*Nos devuelve la cadena completa convertida a minúsculas*

```
>>> palabra.isupper()  
False
```

*Método str.isupper()*  
*Nos devuelve True si todos los caracteres en la cadena están en mayúscula, sino False*

```
>>> palabra.islower()  
False
```

*Método str.islower()*  
*Nos devuelve True si todos los caracteres en la cadena están en minúscula, sino False*



# Cadenas: Algunos métodos aplicables

```
>>> palabra = "Algoritmo"  
>>> palabra.isalpha()  
True
```

*Método **str.isalpha()***  
*Nos devuelve True si la cadena sólo está formada por letras, sino False*

```
>>> palabra.isnumeric()  
False
```

*Método **str.isnumeric()***  
*Nos devuelve True si la cadena sólo está formada por números, sino False*

```
>>> palabra.isalnum()  
True
```

*Método **str.isalnum()***  
*Nos devuelve True si la cadena sólo está formada por letras y/o números, sino False*

**[www.ingenieria.uba.ar](http://www.ingenieria.uba.ar)**

**f**    /ingenieriauba

 /FIUBAoficial