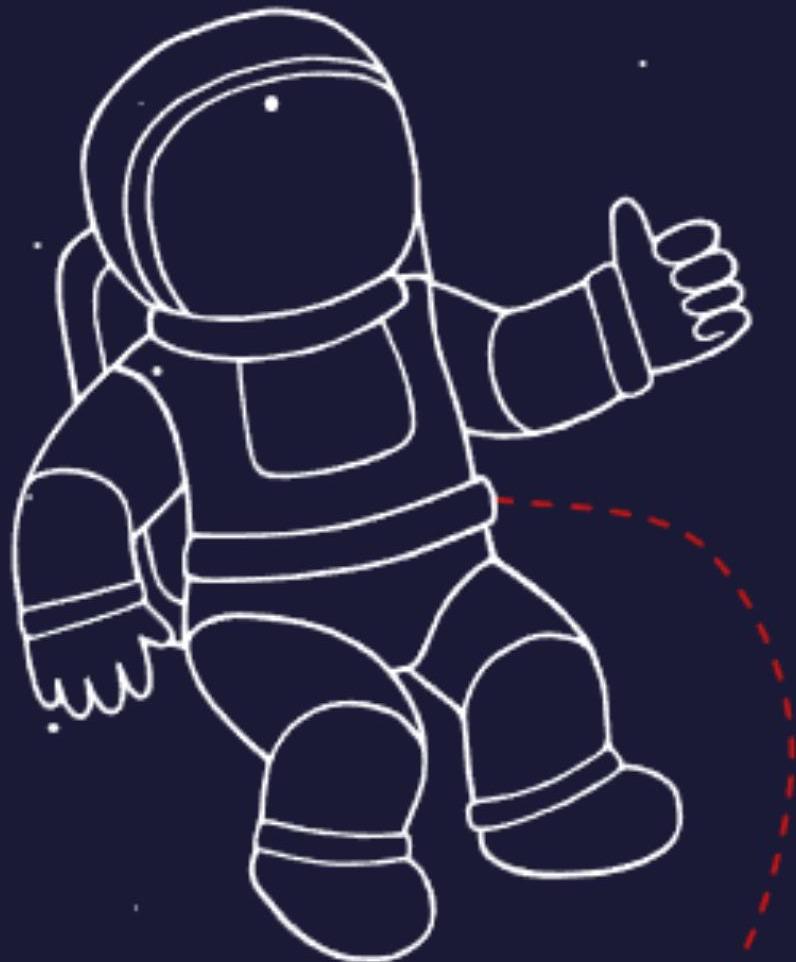




# Programa académico **CAMPUS**

**Trainer**  
**Ing. Carlos H. Rueda C.**





# Métodos de cadenas I



# CADERAS PYTHON

Podemos acceder a una caracter específico del string mediante su índice, pero no modificarlo.

y	o		s	o	y		t	u		p	a	d	r	e
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

```
1 s = "yo soy tu padre"
2 print(s[7])    # >>> t
3 print(s[-8])   # >>> t
4 s[7] = "s"      # >>> Error!
```

# len()

Función len() permite conocer el largo de un string (número de caracteres). Así podemos recorrer un string de largo cualquiera.

```
s = input("Ingrese string: ")
j = 0
while(j < len(s)):
    print(s[j])
    j += 1
```

# Recorrer un string

Para recorrer un string también podemos utilizar el comando for (que es una alternativa a usar whiles).

```
s = input("Ingrese string: ")  
for i in s:  
    print(i)
```

```
s = input("Ingrese string: ")  
j = 0  
while(j < len(s)):  
    print(s[j])  
    j += 1
```

# Operaciones con string

Finalmente existen 4 operadores básicos para trabajar con strings.

- $a + b \rightarrow$  Concatena  $a$  y  $b$ .
- $n * a \rightarrow$  Concatena  $n$  veces  $a$ .
- $a \text{ in } b \rightarrow$  Es True ssi  $a$  es parte de  $b$ .
- $a \text{ not in } b \rightarrow$  Es True ssi  $a$  no es parte de  $b$ .

# Operaciones con string

```
a = "hola"; b = "chao"  
print(a+b)                      # >>> holachao  
print(3*a)                      # >>> holaholahola  
print("ol" in a)                 # >>> True  
print("ol" not in b)              # >>> True
```

# MÉTODOS DE CADENAS

El tipo de dato `str` es una clase incorporada cuyas instancias incluyen variados métodos para analizar, transformar, separar y unir el contenido de las cadenas de caracteres.

En esta presentación detallamos los que consideramos más relevantes para tener siempre a mano.

# upper()

Devuelve la cadena con todos sus caracteres a mayúscula:

```
"Hola Mundo".upper()
```

```
'HOLA MUNDO'
```

# **lower()**

Devuelve la cadena con todos sus caracteres a minúscula:

```
"Hola Mundo".lower()
```

```
'hola mundo'
```

# capitalize()

Devuelve la cadena con su primer carácter en mayúscula:

```
"hola mundo".capitalize()
```

```
'Hola mundo'
```

# **title()**

Devuelve la cadena con el primer carácter de cada palabra en mayúscula:

```
"hola mundo".title()
```

```
'Hola Mundo'
```

# count()

Devuelve una cuenta de las veces que aparece una subcadena en la cadena:

```
"Hola mundo".count('mundo')
```

```
1
```

# find()

Devuelve el índice en el que aparece la subcadena (-1 si no aparece):

```
"Hola mundo".find('mundo')
```

```
5
```

# find()

Devuelve el índice en el que aparece la subcadena (-1 si no aparece):

```
"Hola mundo".find('mundoz')
```

```
-1
```

# rfind()

Devuelve el índice en el que aparece la subcadena, empezando por el final:

```
"Hola mundo mundo mundo".rfind('mundo')
```

# **isdigit()**

Devuelve True si la cadena es todo números (False en caso contrario):

```
c = "100"  
c.isdigit()
```

True

# isalnum()

Devuelve True si la cadena es todo números o caracteres alfabéticos:

```
c = "ABC10034po"  
c.isalnum()
```

True

# isalpha()

Devuelve True si la cadena es todo caracteres alfabéticos:

```
c = "ABC10034po"  
c.isalpha()
```

False

# isalpha()

Devuelve True si la cadena es todo caracteres alfabéticos:

```
"Holamundo".isalpha()
```

```
True
```

# islower()

Devuelve True si la cadena es todo minúsculas:

```
"Hola mundo".islower()
```

```
False
```

# isupper()

Devuelve True si la cadena es todo mayúsculas:

```
"Hola mundo".isupper()
```

```
False
```

# **istitle()**

Devuelve True si la primera letra de cada palabra es mayúscula:

```
"Hola Mundo".istitle()
```

```
True
```

# isspace()

Devuelve True si la cadena es todo espacios:

```
" " .isspace()
```

```
False
```

# startswith()

Devuelve True si la cadena empieza con una subcadena:

```
"Hola mundo".startswith("Mola")
```

```
False
```

# **endswith()**

Devuelve True si la cadena acaba con una subcadena:

```
"Hola mundo".endswith('mundo')
```

```
True
```

# split()

Separa la cadena en subcadenas a partir de sus espacios y devuelve una lista:

```
"Hola mundo mundo".split()[0]
```

```
'Hola'
```

# split()

Podemos indicar el carácter a partir del que se separa:

```
"Hola, mundo, mundo, otra, palabra".split(' , ')
```

```
[ 'Hola' , 'mundo' , 'mundo' , 'otra' , 'palabra' ]
```

# join()

Une todos los caracteres de una cadena utilizando un carácter de unión:

```
", ".join("Hola mundo")
```

```
'H,o,l,a,,m,u,n,d,o'
```

# join()

Une todos los caracteres de una cadena utilizando un carácter de unión:

```
" ".join("Hola")
```

```
'H o l a'
```

# strip()

Borra todos los espacios por delante y detrás de una cadena y la devuelve:

```
" Hola mundo ".strip()
```

```
'Hola mundo'
```

# strip()

Podemos indicar el carácter a borrar:

```
"-----Hola mundo---".strip(' -')
```

```
'Hola mundo'
```

# replace()

Reemplaza una subcadena de una cadena por otra y la devuelve:

```
"Hola mundo".replace('o', 'θ')
```

```
'Holθa mundθ'
```

# replace()

Podemos indicar un límite de veces a reemplazar:

```
"Hola mundo mundo mundo mundo mundo mundo".replace(' mundo', '', 4)
```

```
'Hola mundo'
```

# Ejercicio 1

Los teléfonos de una empresa tienen el siguiente formato prefijo-número-extension donde el prefijo es el código del país **+34**, y la extensión tiene dos dígitos (por ejemplo **+34-913724710-56**). Escribir un programa que pregunte por un número de teléfono con este formato y muestre por pantalla el número de teléfono sin el prefijo y la extensión.

# Ejercicio 2

Hacer un programa que reciba una palabra e indicar si esta es un **palíndrome**.

Una palabra **palíndroma** es aquella que se lee igual adelante que atrás..

# Ejercicio 3

Escribir un programa que pregunte al usuario la fecha de su nacimiento en formato dd/mm/aaaa y muestra por pantalla, el día, el mes y el año. Adaptar el programa anterior para que también funcione cuando el día o el mes se introduzcan con un solo carácter.

# Ejercicio 4

Descargar y resolver el desafío planteado en el documento PDF.

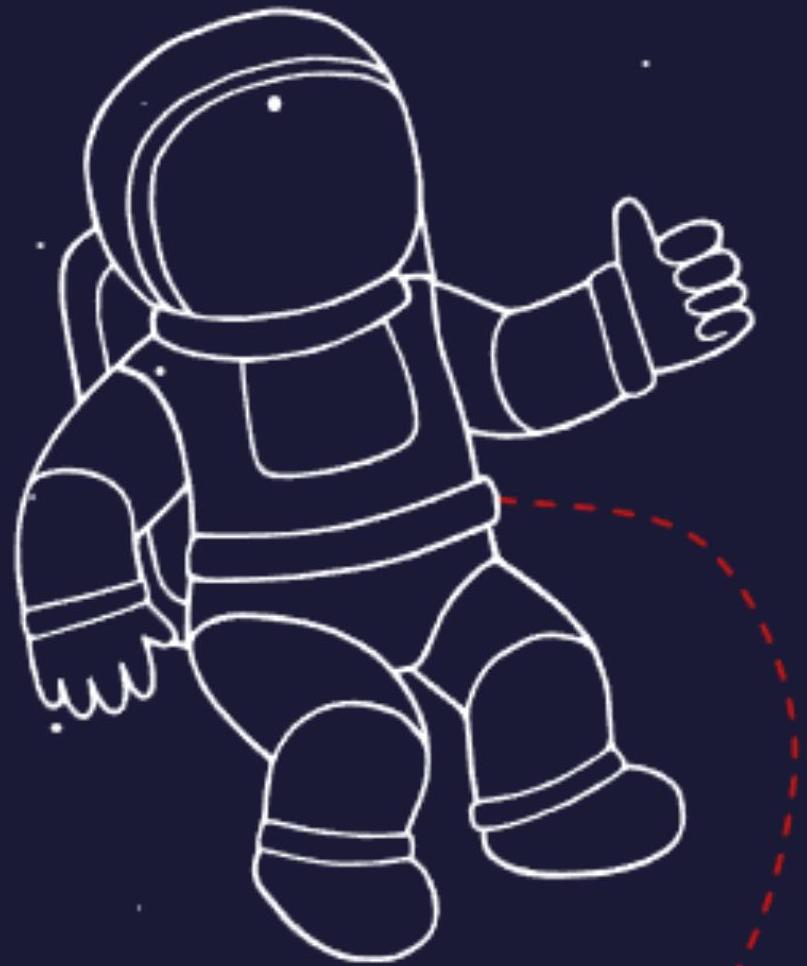
Debe crear un programa que resuelva el desafío, no una función por que el tema no se ha visto.

[Para ver documento hacer clic aquí.](#)

# MÉTODOS CADENAS

## Referencias

- [Operaciones comunes de cadena de caracteres \(sitio oficial python\)](#)
- [30 Métodos de las cadenas](#)



# Programa académico **CAMPUS**

**Trainer**  
**Ing. Carlos H. Rueda C.**

