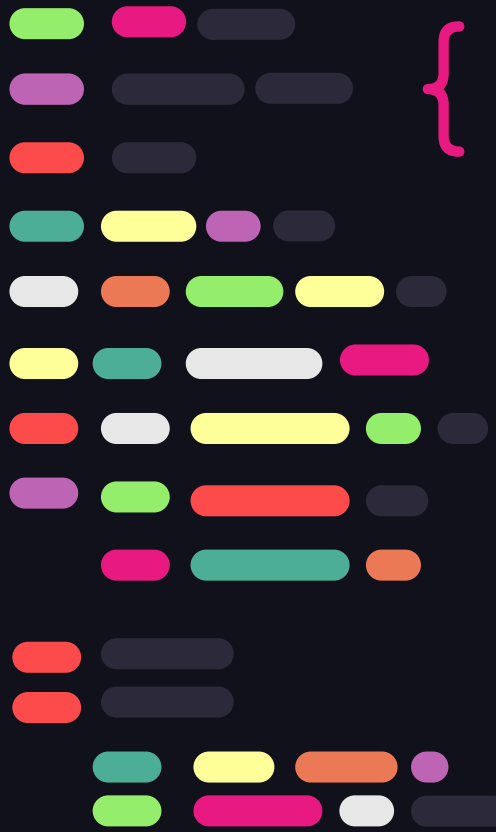
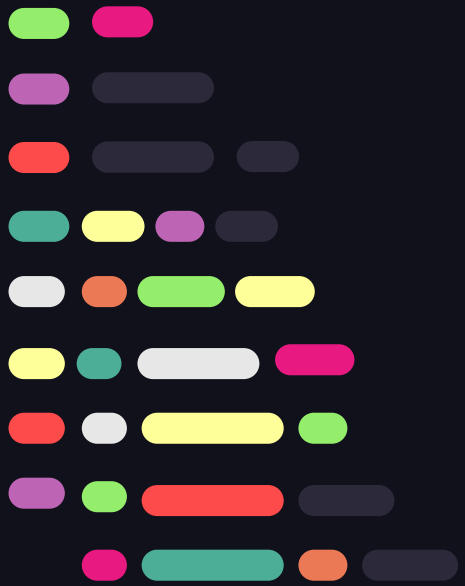


Strings

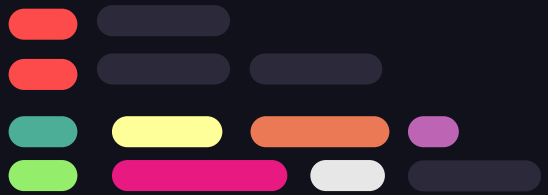
Cadenas de Caracteres

< Juan José Betancur-Muñoz >

Tabla de contenidos



01 { .. Introducción a las cadenas



} ..

Introducción a las Cadenas

En programación, una cadena es un conjunto de letras, números y símbolos que representan texto.

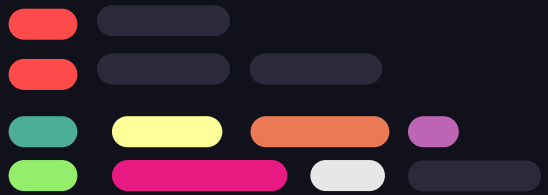
En Python, usamos comillas simples ('') o comillas dobles ("") para crear cadenas.

A cada uno de los caracteres le corresponde un número que llamamos índice.

```
message = “Hello, World!”

0 1 2 3 4 5 6 7 8 9 10 11 12
“Hello, World!”

message[0]  = H
message[7]  = W
```

02 { ..

Comillas y formatos



} ..

Comillas y formatos

No se pueden mezclar comillas simples y dobles en una misma cadena.

Se pueden usar tres comillas (''') para cadenas multilínea.

A cada uno de los caracteres le corresponde un número que llamamos índice.

```
'''Un sabio alguna vez dijo: "Quizá en el futuro alguien sepa por qué after no es antes y before después."'''
```

Comillas y formatos

Las **f**-strings permiten incrustar valores en cadenas de manera sencilla.

Utilizamos **{}** como marcadores de posición para variables.

```
nombre = “Jorge”
dia = “martes”

message = f’’’Hola, {nombre}. Esperamos que
te encuentres muy bien hoy {dia}.’’’

print(message)
```

```
Hola, Jorge. Esperamos que
te encuentres muy bien hoy martes.
```

Comillas y formatos

Puedes incrustar valores en una cadena usando %s, que es como un marcador para un valor que deseas agregar más adelante.

Se escribe el mensaje completo con la combinación %s en los lugares en los que deben ir los valores.

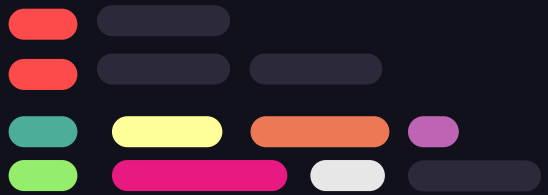
Al final puedes guardar el mensaje en una variable y usar print() para imprimirla, escribiendo nuevamente la combinación y la lista de valores en orden.

```
nombre = “Jorge”
dia = “martes”

message = ’’’Hola, %s. Esperamos que te
encuentres muy bien hoy %s.’’’

print(message %s (nombre, dia))
```

```
Hola, Jorge. Esperamos que
te encuentres muy bien hoy martes.
```

03 { ..

Métodos para strings



} ..

Métodos básicos

`len(cadena)`

Devuelve la longitud de la cadena.

`cadena.upper()`

Convierte la cadena a mayúsculas.

`cadena.lower()`

Convierte la cadena a minúsculas.

`cadena[inicio:fin]`

Devuelve una subcadena desde el índice de inicio hasta el índice de fin.

`cadena.replace(valAntes, valFinal)`

Reemplaza todas las ocurrencias de un valor con otro.

`cadena.find(subcadena)`

Regresa el índice de la primera ocurrencia de la subcadena especificada. Si no se encuentra la subcadena, devuelve -1.

{ . . Métodos básicos

cadena.upper() cadena.lower() cadena[inicio:fin]

```
len(cadena)

text = "Hello, world!"
length = len(text)  # Length of the string is 13
print(length)  # Output: 13
```

cadena.

```
text = "Hell
new_text = text.replace("world", "Python")  # Replace "world"
with "Python"
print(new_text)  # Output: "Hello, Python!"
```

```
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

{ . . Métodos básicos

len(cadena)

cadena.lower()

cadena[inicio:fin]

cadena.upper()

```
text = "Hello, world!"
upper_text = text.upper()  # Convert to uppercase
print(upper_text)  # Output: "HELLO, WORLD!"
```

```
text =
length
of the
print(
```

```
cad
```

```
text =
new_text = text.replace("world", "Python")  # Replace "world"
with "Python"
print(new_text)  # Output: "Hello, Python!"
```

```
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

Métodos básicos

len(ca

```
text = "Hel
length = le
of the stri
print(lengt
```

cadena.lower()

```
text = "Hello, World!"
lower_text = text.lower()  # Convert to lowercase
print(lower_text)  # Output: "hello, world!"
```

cadena.replace(valAntes, valFinal)

```
text = "Hello, world!"
new_text = text.replace("world", "Python")  # Replace "world"
with "Python"
print(new_text)  # Output: "Hello, Python!"
```

cadena.find(subcadena)

```
text = "Hello, world!"
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

Métodos básicos

```
cadena[inicio:fin]
```

```
text = "Hello, world!"
sliced_text = text[7:12]  # Slice from index 7 to 11
print(sliced_text)  # Output: "world"
```

cadena.replace(valAntes, valFinal)

```
text = "Hello, world!"
new_text = text.replace("world", "Python")  # Replace "world"
with "Python"
print(new_text)  # Output: "Hello, Python!"
```

cadena.find(subcadena)

```
text = "Hello, world!"
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

{ . . Métodos básicos

cadena.replace(valAntes, valFinal)

```
text = "Hi, world!"
newText = text.replace("world", "Py")  # Replace "world" with "Py"
print(newText)  # Output: "Hi, Py!"
```

cadena[inicio:fin]

cadena.find(subcadena)

```
text = "Hello, world!"
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

{ . . Métodos básicos

len(cadena) cadena.upper() cadena.lower() cadena[inicio:fin]

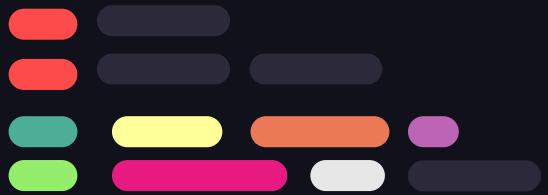
cadena.find(subcadena)

```
text = "Hello, world!"
index = text.find("world")  # Find the index of "world"
print(index)  # Output: 7
```

```
text = "Hello, world!"
new_text = text.replace("world", "Python")  # Replace "world"
with "Python"
print(new_text)  # Output: "Hello, Python!"
```

04 { ..

Ejemplo de Aplicación



} ..

Ejemplos de Aplicación

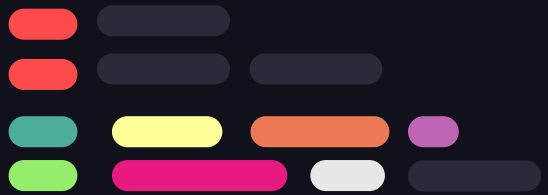
Ejemplo Métodos String

Cree un programa que reciba una frase en string y regrese esa frase sometida a cada uno de los métodos vistos.

```
textoString = "Hello, World!"

length = len(textoString)
upper_text = textoString.upper()
lower_text = textoString.lower()
sliced_text = textoString[0:5]  Slice desde index 0 a 4
replaced_text = textoString.replace("Hello", "Hi")

print(f"Length: {length}")
print(f"Uppercase: {upper_text}")
print(f"Lowercase: {lower_text}")
print(f"Sliced: {sliced_text}")
print(f"Replaced: {replaced_text}")
```

05 { . .

Retos de Práctica



} . .

Retos de Práctica

1. Cree un programa que reciba una frase que incluya la palabra Python y regrese un mensaje que diga cuántos caracteres posee, imprima el texto en mayúsculas y en minúsculas, imprima los primeros 5 caracteres de la frase, e imprima la frase con la palabra Python cambiada por la palabra lenguaje.

Ingreso: **“Me encanta Python!”**
Resultado: Su frase tiene 18 caracteres. Aquí está en mayúsculas: **ME ENCANTA PYTHON!** y aquí en minúsculas: **me encanta python!**. Estos son los primeros 5 caracteres de su frase: **Me en**. Esta es su frase sin la palabra Python: **Me encanta JavaScript!**.

2. Un programa que reciba el nombre del usuario en minúscula y entregue el mismo nombre, pero con mayúscula inicial.

Ingreso: **juan.**
Resultado: ¡Hola, **Juan**!

3. Un programa que tome cualquier palabra y la entregue con un diseño como el de Metallica.

Ingreso: **camino**
Resultado: **CaminO**

Retos de Práctica

4. Un formateador de citas bibliográficas con la norma APA 6 para libros. El programa debe solicitar: nombre y apellido del autor, año de publicación, título del libro, ciudad, país de publicación y editorial. Debe entregar un String con el siguiente formato:
Apellido autor, inicial nombre en mayus. (año). Título. Ciudad, País: Editorial.

Ingreso: **juan, pérez, 1978, La casa de medina, madrid, gredos.**
Resultado: **Pérez, J. (1978). La Casa de Medina. Madrid: Gredos.**

5. Un programa que reciba una palabra que debe censurarse y un mensaje escrito por el usuario que contenga la palabra censurable. Debe regresar el mismo mensaje, pero debe cambiar la palabra que no puede utilizase por otra expresión.

Ingreso: **“This is shit!”**
Resultado: **This is %$&$!**