

# Módulo 1

1

0101

01 0 1 00 011 0101

# LECCIÓN 2

## Cadenas

1 1 01 0

Las cadenas en Python o strings son un tipo inmutable que permite almacenar secuencias de caracteres. Para crear una, es necesario incluir el texto entre comillas dobles `""` o simples `"`. Puede obtener más ayuda con el comando `help(str)`.

```
s = "Esto es una cadena"
print(s)          #Esto es una cadena
print(type(s))   #<class 'str'>

s = 'Esto es otra cadena'
print(s)         #Esto es otra cadena
print(type(s))  #<class 'str'>
```

La longitud de las cadenas está limitada a la memoria del ordenador donde se ejecute el programa, incluso se puede definir una cadena vacía.

```
s = ''
s = ""
```

```
[ ] s = """
    O definir una cadena que se quiera escribir
    en diferentes líneas
    """

print(s)
```

```
O definir una cadena que se quiera escribir
en diferentes líneas
```

## Ejercitación Comillas

Cómo puedo escribir e imprimir el texto `I'm the best software developer`, utilizando comillas simples y dobles?

Encontró algún inconveniente?

Para solucionar situaciones como la anterior, existen dos estrategias, una donde utilizamos las comillas dobles para poder incluir apóstrofes dentro de la cadena y otra en donde se escapa directamente el caracter.

Para escapar un caracter y hacerlo literal, es decir que se imprima tal cual lo que se está escribiendo, se utiliza el backslash o barra invertida \.

Cómo entonces podría escribir e imprimir, utilizando comillas simples y dobles el siguiente texto

```
"I'm the best software developer", she said
```

```
[ ] print('"I\'m the best software developer", she said ')
```

```
"I'm the best software developer", she said  
8
```

## Ejercitación Mensaje Cifrado

Cada caracter que se escribe tiene su correspondiente código ASCII y Unicode, se te reta q escribir un mensaje cifrado utilizando dicha codificación, recuerde que se puede activar la impresión de la equivalencia de caracteres utilizando la barra invertida \

Qué pasará si agrega un r al inicio de la cadena con el mensaje secreto que acaba de escribir? a dicha r se le conoce como raw strings

```
print(r"\110\110")
```

## Formateo de cadenas

Qué pasa si dentro de una cadena se requiere imprimir un dato? un nombre, un valor o un flotante. Si el dato es al final, es sencillo se puede concatenar con el operador +

```
print("Su edad es: " + "25")
```

o separando por comas ,

```
[ ] print("Su edad es: " + "25")

print("su edad es:", 25, "y nacio en", 1999)
```

```
Su edad es: 25
su edad es: 25 y nacio en 1999
```

Qué tal si los datos que se quieren agregar a la cadena no van al final sino en la mitad o en varias partes, para esta funcionalidad existe el formateo de cadenas y funciona tradicionalmente así:

```
%s #para cadenas
%d #para números
%f #para flotantes
```

Una forma un poco más moderna de aplicar el mismo formateo es utilizando el método `.format()`, que admite asignarle un nombre de variable a cada uno de los datos que se quieren imprimir.

```
[ ] print("Los números son %d y %d." % (5, 10)) #Los números son 5 y 10.

s = "Los números son {} y {}".format(5, 10)
print(s) #Los números son 5 y 10.

s = "Los números son {a} y {b}".format(a=5, b=10)
print(s) #Los números son 5 y 10.
```

Los números son 5 y 10.  
Los números son 5 y 10  
Los números son 5 y 10

Finalmente, desde Python 3.6 existe un método adicional llamado los f-strings, que admiten algunas operaciones y llamado a funciones que retornen algo.

```
[ ] a = 5; b = 10
s = f"Los números son {a} y {b}"
print(s) #Los números son 5 y 10

a = 5; b = 10
s = f"a + b = {a+b}"
print(s) #a + b = 15

def funcion():
    return 20
s = f"El resultado de la función es {funcion()}"
print(s) #El resultado de la función es 20
```

Los números son 5 y 10  
a + b = 15  
El resultado de la función es 20

## Métodos de las cadenas

Las cadenas tienen diferentes métodos propios que se pueden utilizar para modificar la cadena, a continuación se listan algunos de ellos

```
.capitalize()  
.upper()  
.lower()  
.title()  
.count()  
.find('search')  
.split()  
.join()  
.strip()  
.replace()  
.zfill()
```

Y la lista completa puede ser encontrada en Métodos de cadenas en Python

Ejercitación qué hacen los métodos?

Su tarea es descubrir cómo se utilizan los métodos listados arriba y qué devuelven?