

8. CADENAS Y ENTRADAS

Los ordenadores no sólo tratan con números, sino también con letras, palabras y símbolos.

Los programadores llaman a estas cadenas de caracteres: "strings".

Lo que aprenderás

- Qué es una String o Cadena.
- Cómo introducir Cadenas desde el teclado.
- Cómo seleccionar partes de una Cadena.

QUE ES UNA STRING

Le puedes decir a Python que algo es una **string**, o en castellano cadena de caracteres, cuando colocamos dentro de una variable comillas simples. En Python, las cadenas son el tipo de datos **str**.

Puedes darle a una variable una cadena de caracteres como un valor. En este ejemplo, el valor de la variable llamada “**nombre**” tiene el valor **Manolo**.

Algunos ejemplos de variables tipo String:

```
'hola'  
'Manolo'  
'¡El 7 es el numero magico!'  
'G7/mk%k$3'  
'manolo@correo.com'
```

```
nombre='Manolo'  
print(nombre)
```

JUNTANDO STRINGS

Puedes agregar cadenas de caracteres unidas entre si, siempre que la **String** donde los vayas a almacenar sea lo suficiente grande como para incluir la suma de dos cadenas de caracteres.

```
nombre='Manolo'  
saludo='Buenos días ' + nombre  
print(saludo)
```

¿Te has fijado que al final de 'Buenos días' hay un espacio? ¿Sabes por qué?

OBTENER STRINGS DESDE EL TECLADO

Python tiene un comando para poder entrar datos `input()`, que lee lo que se escribe en el teclado y lo guarda como una cadena o string.

Este comando muestra la pregunta en la pantalla (el mensaje) y luego espera a que se escriba una respuesta. Una vez que se pulsa la tecla `ENTER`, la cadena se almacena en la variable indicada (comida).

```
nombre='Manolo'  
comida=input('¿Qué quieres comer? ')  
print(nombre, 'quiere comer', comida)
```

BITS DE LAS STRINGS

Python numera cada carácter en una cadena, empezando desde cero.

Puedes seleccionar algunos caracteres de la cadena de manera individual.

También podemos seleccionar intervalos.

```
correr='¡Corre ya!'
      0123456789
```

```
correr='¡Corre ya!'
print(correr[0], correr[5], correr[9])
```

Resultado: ¡ e !

```
print(correr[4:8])
```

Resultado: re y

Fíjate que el último valor del rango no aparece, siempre muestra el valor anterior.