

Mostrar por pantalla

10. MOSTRAR POR PANTALLA

Ya vistes que puedes usar `print()` para mostrar datos por pantalla. Hay muchas otras maneras útiles de utilizar el comando `print()`.

Lo que aprenderás

- Más usos para el comando `print()`
- Cambiar la impresión con (separadores)
- Cómo hacer una lista con las salidas de datos por pantalla.
- Cómo imprimir caracteres especiales.

Mostrar por pantalla

HOLA MUNDO!

Ya sabes imprimir con `print()` datos del tipo **Strings**,

```
print('¡Hola mundo!')
```

Imprimir el valor de variables numéricas

```
a=10  
print(a)
```

Imprimir más de un elemento usando comas

```
print('a tiene el valor' a)
```

Mostrar por pantalla

SEPARADORES Y FINAL DE LINEA

Cuando envía más de un elemento para imprimir, como por ejemplo `print(a, b, c)`, Python coloca un espacio llamado separador entre cada elemento.

Ahora vas a cambiar los separadores enviando una cadena separadora diferente a la función de impresión, usando `sep = '/'` dentro de los corchetes del comando `print()`

```
a = 1; b = 2; c = 3;  
print(a, b, c)
```

Se mostrará 1 2 3 (con espacio entre los números)

```
print(a, b, c, sep='/')
```

Se mostrará 1/2/3 (con separador / en los números)

```
print(a, b, c, sep='')
```

Se mostrará 123, sin espacios entre los números. Fíjate que en el código no hemos puesto nada entre las comillas.

Mostrar por pantalla

No te tienes que limitar a un solo carácter como separador, puedes usar cualquier carácter o cadena de caracteres, incluyendo comas y dos puntos.

```
a=1  
b=2  
c=3  
print(a,b,c, sep=' más ')
```

Tendrá como resultado 1 más 2 más 3

Normalmente, `print()` comienza una nueva línea después de que se haya impreso lo que se le ha pedido. Puedes cambiar esto poniendo `end = ' '` dentro de los paréntesis. Esto le dice a `print()` que coloque un espacio al final de la impresión en lugar de iniciar una nueva línea. Puedes usar esto en un bucle `for` para imprimir toda su salida en una misma línea. Puedes poner lo que quieras entre las comillas.

```
for n in range(1, 6):  
    print(n, end=' ')
```

Se mostrará 1 2 3 4 5 en la misma línea

TRUCOS PARA LAS STRINGS

Algunas cadenas pueden causar problemas en Python - por ejemplo, ¿qué pasa si quieres poner un apóstrofe dentro de una cadena?

Para evitar confundir a Python usando tres comillas simples, ponemos un carácter de código especial antes del apóstrofe. Este tipo de carácter se llama **carácter de escape**.

```
print('I'm very happy.')
```

Las apóstrofes en una frase confunden la interpretación de Python. Esto se debe resolver marcando que el carácter de comilla simple se debe escribir como tal y no interpretarse como final de línea.

```
print('I\'m very happy.')
```

Mostrar por pantalla

Otra combinación útil es `\n`, que inserta una nueva línea en el punto en el cual se escribe esta combinación de caracteres.

```
print ('hello\nGoodbye')
```