

Salida de números con print()

Edison Achalma

Escuela Profesional de Economía, Universidad Nacional de San Cristóbal de Huamanga

Resumen

Este abstract será actualizado una vez que se complete el contenido final del artículo.

Tabla de contenidos

Introduction	2
0.1 Motivación	2
0.2 Intuición	2
0.3 Modelo mental	2
0.4 Desarrollo	2
0.4.1 1. Imprimir números simples	2
0.4.2 2. Operaciones matemáticas dentro de print()	2
0.4.3 3. Combinar números y texto	3
0.4.4 4. Texto vs números: la diferencia clave	3
0.4.5 5. Múltiples números en un print()	3
0.5 Ejemplos	4
0.6 Errores comunes	4
0.7 Buenas prácticas	5
0.8 Ejercicio	5
0.9 Síntesis	6
0.10 Qué sigue	6
1 Publicaciones Similares	6

Edison Achalma  <https://orcid.org/0000-0001-6996-3364>

El autor no tiene conflictos de interés que revelar. Los roles de autor se clasificaron utilizando la taxonomía de roles de colaborador (CRediT; <https://credit.niso.org/>) de la siguiente manera: Edison Achalma: conceptualización, metodología, análisis formal, investigación, recursos, curación de datos, redacción, visualización, supervisión, administración del proyecto

La correspondencia relativa a este artículo debe dirigirse a Edison Achalma, Escuela Profesional de Economía, Universidad Nacional de San Cristóbal de Huamanga, Portal Independencia N° 57, Ayacucho, AYA 5001, Perú, Email: elmer.achalma.09@unsch.edu.pe

Salida de números con print()

0.1 Motivación

Los números son el alma de la programación: cálculos, mediciones, estadísticas, todo gira en torno a ellos. Saber imprimir números correctamente te permite ver resultados de operaciones, depurar cálculos y mostrar información numérica a los usuarios. La diferencia clave con el texto es que **los números no llevan comillas** — y entender esa diferencia es fundamental.

0.2 Intuición

Piensa en `print()` como una calculadora que también habla. Si le pides que imprima 3, simplemente muestra el número 3. Si le pides `3 + 3`, primero hace la cuenta (como una calculadora) y luego muestra el resultado. Pero si le pides `"3"` (con comillas), lo trata como texto, como si fuera una palabra escrita con el símbolo “3”.

0.3 Modelo mental

Python distingue entre:

1. **Números (int, float):** valores matemáticos. **Van sin comillas.**
2. **Texto (str):** caracteres. **Van con comillas.**

Cuando usas `print()`:

- Si pasas un número, lo muestra como número.
- Si pasas una expresión matemática, **la calcula primero** y muestra el resultado.
- Si pasas texto con comillas, lo muestra literalmente.

0.4 Desarrollo

0.4.1 1. Imprimir números simples

```
print(3)      # Número entero
print(3.14)   # Número decimal (float)
print(-5)     # Número negativo
# Salida:
# 3
# 3.14
# -5
```

0.4.2 2. Operaciones matemáticas dentro de print()

Python evalúa las operaciones antes de imprimir:

```
print(5 + 3)   # Suma
print(10 - 4)  # Resta
print(6 * 7)   # Multiplicación
print(15 / 3)  # División (resultado decimal)
print(10 // 3) # División entera (trunca decimales)
print(10 % 3)  # Módulo (resto de la división)
```

```
print(2 ** 4)      # Potencia
# Salida:
# 8
# 6
# 42
# 5.0
# 3
# 1
# 16
```

0.4.3 3. Combinar números y texto

Usa **comas** para separar texto y números en el mismo `print()`:

```
print("Tengo", 25, "años")
# Salida: Tengo 25 años

print("El resultado es", 10 + 5)
# Salida: El resultado es 15

print("Precio:", 99, "€")
# Salida: Precio: 99 €
```

0.4.4 4. Texto vs números: la diferencia clave

```
print(5)          # Muestra el número 5
print("5")        # Muestra el carácter "5" (texto)
print(5 + 3)      # Calcula 5+3, muestra 8
print("5+3")      # Muestra literalmente "5+3"
# Salida:
# 5
# 5
# 8
# 5+3
```

0.4.5 5. Múltiples números en un `print()`

```
print(10, 20, 30)
# Salida: 10 20 30

print(1, 2, 3, sep=" - ")
# Salida: 1 - 2 - 3

print(5, 10, 15, sep=", ")
# Salida: 5, 10, 15
```

0.5 Ejemplos

Mínimo — Números y operaciones básicas:

```
print(100)
print(50 + 50)
print(10 * 2)
# Salida:
# 100
# 100
# 20
```

Progresivo — Cálculos más complejos:

```
print(15 + 3 * 4)      # Prioridad: multiplicación primero
print((15 + 3) * 4)    # Paréntesis cambian el orden
# Salida:
# 27   (15 + 12)
# 72   (18 * 4)
```

Real — Calculando propina en un restaurante:

```
cuenta = 45.50
propina_porcentaje = 15
propina = cuenta * propina_porcentaje / 100
total = cuenta + propina

print("Cuenta:", cuenta, "€")
print("Propina (15%):", propina, "€")
print("Total a pagar:", total, "€")
# Salida:
# Cuenta: 45.5 €
# Propina (15%): 6.825 €
# Total a pagar: 52.325 €
```

0.6 Errores comunes

Error	Síntoma	Cómo detectarlo
Número con comillas	<code>print("5")</code> muestra texto “5”, no número	Si necesitas hacer operaciones, no uses comillas
Texto sin comillas	<code>print(Hola)</code> → <code>NameError: name 'Hola' is not defined</code>	El texto siempre va entre comillas
Dividir por cero	<code>print(5 / 0)</code> → <code>ZeroDivisionError: division by zero</code>	Asegura que el denominador no sea 0
Operaciones en texto	<code>print("5+3")</code> muestra “5+3”, no 8	Las comillas hacen que todo sea literal

Error	Síntoma	Cómo detectarlo
Confundir separador	<code>print("Hola" 25)</code> → SyntaxError	Usa comas para separar: <code>print("Hola", 25)</code>

0.7 Buenas prácticas

- **Números sin comillas:** siempre que quieras un valor numérico o una operación.
- **Usa paréntesis** para clarificar el orden de las operaciones.
- **Nombra variables** en lugar de números “mágicos”:

```
# Poco legible
print(45.50 * 0.15)

# Legible
cuenta = 45.50
propina = cuenta * 0.15
print(propina)
```

- **Separa texto y números con comas** para que Python los muestre correctamente.
- **Usa f-strings** para formateo avanzado (lo veremos más adelante).

0.8 Ejercicio

Básico — Calculadora de presupuesto
Escribe un programa que:

1. Tenga un presupuesto de **500** euros
2. Un gasto en comida de **125.50** euros
3. Un gasto en transporte de **45.75** euros
4. Calcule el **total gastado** y el **saldo restante**
5. Imprima los resultados con mensajes descriptivos

Requisitos: los números deben ser calculados, no escritos directamente en los `print()`.

Solución (no mires hasta haberlo intentado):

```
presupuesto = 500
gasto_comida = 125.50
gasto_transporte = 45.75

total_gastado = gasto_comida + gasto_transporte
saldo = presupuesto - total_gastado

print("Presupuesto inicial:", presupuesto, "€")
```

```
print("Gasto en comida:", gasto_comida, "€")
print("Gasto en transporte:", gasto_transporte, "€")
print("Total gastado:", total_gastado, "€")
print("Saldo restante:", saldo, "€")
```

Salida esperada:

Presupuesto inicial: 500 €
Gasto en comida: 125.5 €
Gasto en transporte: 45.75 €
Total gastado: 171.25 €
Saldo restante: 328.75 €

0.9 Síntesis

- **Números sin comillas:** `print(5)` muestra el número 5.
- **Texto con comillas:** `print("5")` muestra el texto "5".
- **Operaciones matemáticas** se calculan antes de imprimir.
- **Mezcla texto y números** con comas: `print("Edad:", 30)`.
- **Jerarquía de operadores:** respeta la matemática estándar (paréntesis primero).
- **Puedes usar `sep` y `end`** para personalizar el formato de salida.

0.10 Qué sigue

- **Relacionado:** variables (guardar números en variables), tipos de datos (`int`, `float`, `str`)
- **Siguiente tema:** Variables y tipos de datos — aprenderás a almacenar y manipular números y texto de forma estructurada.

1 Publicaciones Similares

Si te interesó este artículo, te recomendamos que explores otros blogs y recursos relacionados que pueden ampliar tus conocimientos. Aquí te dejo algunas sugerencias:

1. [!\[\]\(c580b67c7cd5c9e9e19f04ff6d5093e0_img.jpg\) Instalacion De Anaconda](#)
2. [!\[\]\(81f7c93ea32d9f7160f5d63859611838_img.jpg\) Configurar Entorno Virtual Python Anaconda](#)
3. [!\[\]\(ba314389b6ecfa02b6b9fcc7005e4045_img.jpg\) 01 Introducion A La Programacion Con Python](#)
4. [!\[\]\(b97fa2bc0f8f47a470570a614b3c9117_img.jpg\) 02 Variables Expresiones Y Statements Con Python](#)
5. [!\[\]\(a322c324b965d10daf2ba87b88bdc914_img.jpg\) 03 Objetos De Python](#)
6. [!\[\]\(41d65a787fcf1850fd2e8e87ad550f3e_img.jpg\) 04 Ejecucion Condicional Con Python](#)
7. [!\[\]\(0433665236990cd5d97323c35934349b_img.jpg\) 05 Iteraciones Con Python](#)
8. [!\[\]\(1a71d62dc9b5bf8e942632f113c9e5d5_img.jpg\) 06 Funciones Con Python](#)
9. [!\[\]\(1420310daac853834744395dadfc45d5_img.jpg\) 07 Dataframes Con Python](#)
10. [!\[\]\(edc97d0e076fb4e779eb68be1b49dcb8_img.jpg\) 08 Prediccion Y Metrica De Performance Con Python](#)
11. [!\[\]\(c251dbe21b21640472a575d3efc646cc_img.jpg\) 09 Metodos De Machine Learning Para Clasificacion Con Python](#)
12. [!\[\]\(19e3fe5821872c01e30a336da3d029ce_img.jpg\) 10 Metodos De Machine Learning Para Regresion Con Python](#)
13. [!\[\]\(ddd5de565b260afb4c7b5709a1287b53_img.jpg\) Visualizacion De Datos Con Python](#)

Esperamos que encuentres estas publicaciones igualmente interesantes y útiles. ¡Disfruta de la lectura!