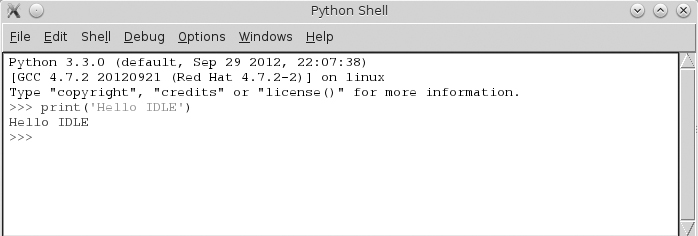
Capítulo 1: Trabajar con números

### **Operaciones matemáticas básicas**

El *shell interactivo de* Python va a ser nuestro amigo en este libro. Inicia el intérprete de comandos IDLE de Python 3 y di "hola" (ver [Figura 1-1](ch01.html#ch1fig1)) escribiendo print('Hello IDLE') y pulsando INTRO. (Para obtener instrucciones sobre cómo instalar Python e iniciar IDLE, consulta [el Apéndice A](app01.html#app01).) IDLE obedece tu orden e imprime las palabras en la pantalla. Enhorabuena: ¡acabas de escribir un programa!

Cuando vuelvas a ver el prompt >>>, IDLE estará listo para recibir más instrucciones.



*Figura 1-1: Shell IDLE de Python 3*

Python puede actuar como una calculadora glorificada, haciendo cálculos sencillos. Sólo tienes que escribir una expresión y Python la evaluará. Después de pulsar ENTER, el resultado aparece inmediatamente.

Pruébalo. Puedes sumar y restar números utilizando los operadores de suma (+) y resta (–). Por ejemplo:

>>> 1 + 2  
3  
>>> 1 + 3.5  
4.5  
>>> -1 + 2.5  
1.5  
>>> 100 – 45  
55  
>>> -1.1 + 5  
3.9

Para multiplicar, utiliza el operador multiplicación (\*):

>>> 3 \* 2  
6  
>>> 3.5 \* 1.5  
5.25

Para dividir, utiliza el operador división (/):

>>> 3 / 2  
1.5  
>>> 4 / 2  
2.0

Como puedes ver, cuando pides a Python que realice una operación de división, te devuelve también la parte fraccionaria del número. Si quieres el resultado en forma de número entero, sin los valores decimales, debes utilizar el operador división suelo (//):

>>> 3 // 2  
1

El operador de división piso divide el primer número entre el segundo y redondea el resultado al entero inmediatamente inferior. Esto resulta interesante cuando uno de los números es negativo. Por ejemplo:

>>> -3 // 2  
-2

El resultado final es el número entero inferior al resultado de la operación de división (-3/2 = -1.5, por lo que el resultado final es -2).

En cambio, si sólo quieres el resto, debes utilizar el operador módulo (%):

>>> 9 % 2  
1

Puedes calcular la potencia de los números utilizando el operador exponencial (\*\*). Los ejemplos siguientes lo ilustran:

>>> 2 \*\* 2  
4  
>>> 2 \*\* 10  
1024  
>>> 1 \*\* 10  
1

También podemos utilizar el símbolo exponencial para calcular potencias menores que 1. Por ejemplo, la raíz cuadrada de un número *n* puede expresarse como n1/2 y la raíz cúbica como n1/3:

>>> 8 \*\* (1/3)  
2.0

Como muestra este ejemplo, puedes utilizar paréntesis para combinar operaciones matemáticas en expresiones más complicadas. Python evaluará la expresión siguiendo la regla estándar *PEMDAS* para el orden de los cálculos: paréntesis, exponentes, multiplicación, división, suma y resta. Considera las dos expresiones siguientes: una sin paréntesis y otra con paréntesis:

>>> 5 + 5 \* 5  
30  
>>> (5 + 5) \* 5  
50

En el primer ejemplo, Python calcula primero la multiplicación: 5 por 5 es 25; 25 más 5 es 30. En el segundo ejemplo, se evalúa primero la expresión entre paréntesis, tal como cabría esperar: 5 más 5 es 10; 10 por 5 es 50.

Estos son los fundamentos absolutos de la manipulación de números en Python. Aprendamos ahora cómo podemos asignar nombres a los números.

[anterior](ch01_1.html)[Subtema 2 de 8: (Ver todo)](ch01.html)[siguiente](ch01_3.html)