Desconocido

### Un bloque es un grupo de sentencias de programación

Un *bloque* de código es un conjunto agrupado de sentencias de programación. Por ejemplo, cuando if age > 20: es True , puede que quieras hacer algo más que imprimir "¡Eres demasiado viejo!". Tal vez quieras imprimir más sentencias, como ésta:

>>> age = 25  
>>> if age > 20:  
 print('You are too old!')  
 print('Why are you here?')  
 print('Why aren\'t you mowing a lawn or sorting papers?')

Este bloque de código está formado por tres llamadas a print que se ejecutan sólo si la condición age > 20 resulta ser True . Cada línea del bloque tiene cuatro espacios al principio. Veamos de nuevo ese código, con espacios visibles:

>>> age = 25  
>>> if age > 20:  
 .... print('You are too old!')  
 .... print('Why are you here?')  
 .... print('Why aren\'t you mowing a lawn or sorting papers?')

En Python, los espacios en *blanco -como* un tabulador (que se inserta al pulsar la tecla TAB) o un espacio (que se inserta al pulsar la barra espaciadora)- tienen significado. El código que está en la misma posición (o sangrado el mismo número de espacios desde el margen izquierdo) se agrupa en un bloque. Siempre que empieces una nueva línea con más espacios que la anterior, estarás iniciando un nuevo bloque. Este nuevo bloque también forma parte del bloque anterior, como en la Figura 5-1 [.](ch05.xhtml#ch05fig01)

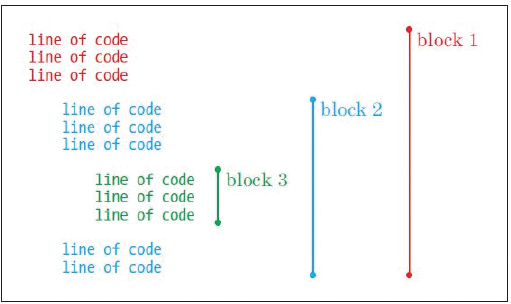
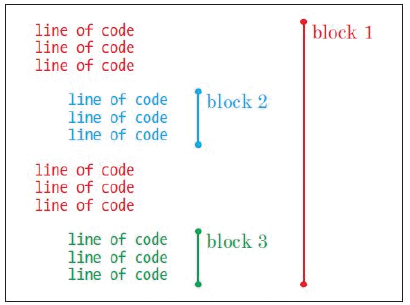


Figura 5-1*: Funcionamiento de los bloques de código*

Agrupamos las sentencias en bloques porque están relacionadas y deben ejecutarse juntas. Cuando cambias la sangría del código, generalmente estás creando nuevos bloques. La Figura [5-2](ch05.xhtml#ch05fig02) muestra cómo se crean tres bloques con sólo cambiar la sangría.



Figura*5-2: Un segundo ejemplo que muestra cómo funcionan los bloques de* código

Aquí, aunque los bloques 2 y 3 tienen la misma sangría, se consideran bloques diferentes porque hay un bloque con menos sangría (menos espacios) entre ellos.

Un bloque con cuatro espacios en una línea y seis espacios en la siguiente producirá un *error de indentación* cuando lo ejecutes. Esto se debe a que Python espera que utilices el mismo número de espacios para todas las líneas de un bloque. Aquí tienes un ejemplo:

>>> if age > 20:  
 .... print('You are too old!')  
 ...... print('Why are you here?')

He hecho visibles los espacios para que puedas ver las diferencias. Observa que la segunda línea print tiene seis espacios en lugar de cuatro. Cuando intentamos ejecutar este código, Python resalta la línea problemática con un bloque rojo y muestra un mensaje explicativo SyntaxError :

>>> age = 25  
>>> if age > 20:  
 print('You are too old!')  
 print('Why are you here?')  
SyntaxError: unexpected indent

Python no esperaba ver dos espacios de más al principio de la segunda línea print .

**NOTA**

*Utiliza un espaciado coherente para que tu código sea más fácil de leer. Si empiezas a escribir un programa y pones cuatro espacios al principio de un bloque, sigue utilizando cuatro espacios al principio de los demás bloques de tu programa. Asegúrate de sangrar cada línea de un bloque con el mismo número de espacios.*

[anterior](ch05_2.html)[Subtema 3 de 11: (Ver todo)](ch05.html)[siguiente](ch05_4.html)