Más secuencias y otras cosas

¿De qué nos ocuparemos?
Introducimos una nueva herramienta para ingresar programas en Python. Revisamos el uso de las variables para guardar información para su uso posterior y cómo combinar una secuencia de comandos para realizar una tarea.

Hasta ahora sabemos cómo introducir comandos simples en Python y nos hemos introducido en el mundo de los datos y en qué hacer con ellos. Veamos ahora qué ocurre cuando tipeamos comandos múltiples en Python.

La utilización de IDLE

Si instalaste la versión 1.5.2 de Python (esta información la provee el programa cuando se inicia), encontrarás una herramienta denominada IDLE que se incluye en el paquete. Este programa muestra una ventana en la cual se pueden introducir los comandos de Python. Tiene varias ventajas con respecto a la versión estándar de DOS:

- Permite cortar y pegar
- Permite la repaginación del cursor facilitando el retipeo de las líneas
- Uso de distintos colores para demarcar la sintaxis, lo que permite detectar fácilmente errores de tipeo, etc.
- Otras más que veremos luego

Si utilizás MS Windows, hay otra opción denominada *PythonWin*, la cual puede bajarse de Internet. Esta aplicación permite el acceso a las funciones de bajo nivel de Windows y es una muy buena alternativa a IDLE. Sólo funciona en Windows, y en mi opinión es levemente superior a IDLE. Sin embargo, IDLE es muy reciente y en versiones subsiguientes puede superar a PythonWin. Pase lo que pase, siempre es bueno poder elegir...

Un rápido comentario

Uno de los elementos más importantes de la programación son los comentarios, algo que la mayor parte de los principiantes a primera vista considera una pérdida de tiempo. Los comentarios son simplemente líneas de texto que describen lo que está pasando. No tienen ningún efecto en la ejecución del programa, son puramente decorativos. Sin embargo, tienen una función muy importante para el programador: explican qué es lo que ocurre, y más importante aún, *por qué*. Y esto es fundamental si el programador que analiza el código no es el autor del programa, o si ha pasado mucho tiempo desde que el programa fue escrito. Una vez que uno comienza a escribir programas más complejos, la presencia de los comentarios se hace prácticamente fundamental. A partir de este momento, utilizaré comentarios para explicar los fragmentos que veamos. Gradualmente el texto explicativo disminuirá y pasará a incluirse en los respectivos comentarios.

Cada lenguaje tiene su forma particular de indicar los comentarios. En BASIC utilizamos REM al principio del comentario. Todo lo que aparece después del comando REM es ignorado:

```
REM print "Esto nunca se imprime."
print "Esto sí que se imprime."
```

Si alguna vez escribiste archivos .bat en MSDOS, el comando REM te resultará familiar, ya que se utiliza en ambos lenguajes.

La mayor parte de las versiones de BASIC permiten el uso de ' en lugar de REM, lo cual es más sencillo de tipear pero más difícil de distinguir. La elección queda a tu cargo.

Tanto Python como Tcl usan el símbolo # como demarcador de comentarios. Todo lo que sigue al signo # es ignorado:

```
v = 12      # asignar el valor 12 a v
x = v*v     # x es v al cuadrado
```

Dicho sea de paso, este estilo de comentario es un tanto *defectuoso*. Los comentarios no deben limitarse a explicar qué sucede con el código (algo que podemos ver por nosotros mismos) sino que debe explicar *por qué* hace lo que hace.

```
v = 3600    # 3600 es la cantidad de segundos en una hora
s = t*3600  # t guarda el tiempo transcurrido en horas, así que lo convertimos a segundos
```

Estos comentarios son mucho más útiles.

Uso de Sequencias mediante variables

Escribí lo siguiente en IDLE o en la línea de comandos de Python (DOS) o en la ventana de Python (UNIX):

```
>>> v = 7
>>> w = 18
>>> x = v + w    # usamos las variables en un cálculo
>>> print x
```

Aquí hemos creado una serie de variables (v, w, x) y las estamos manipulando. Este procedimiento se asemeja al uso de la tecla M en una calculadora de bolsillo para guardar un resultado que será utilizado más tarde.

Podemos mejorar esto utilizando una cadena de formato para mostrar el resultado:

```
>>> print "La suma de %d y %d es: %d" % (v,w,x)
```

Una ventaja de las cadenas de formato es que podemos almacenarlas también en una variable:

```
>>> s = "La suma de %d y %d es: %d"
>>> print s % (v,w,x)   # es útil si se imprime el mismo formato con diferentes valores
```

Recordá que podíamos tipear cadenas largas encerrándolas entre comillas triples. Ahora usaremos este procedimiento para construir una tabla de multiplicación:

```
>>> s = """
1 x 12 = %d
2 x 12 = %d
3 x 12 = %d
"""     # CUIDADO, no se pueden poner comentarios dentro de
>>>     # una cadena, ya que estos se incorporarán a la misma.
>>> print s % (12, 2*12, 3*12)
```

Si extendemos este sistema, podríamos imprimir las doce tablas de multiplicar, desde la del uno hasta la del doce. ¿Pero hay una mejor forma de hacerlo?

Para recordar
• IDLE es una herramienta multiplataforma para escribir programas en Python. • Los comentarios facilitan la comprensión del programa pero no tienen ningún efecto sobre el funcionamiento del mismo. • Las variables pueden guardar resultados intermedios para un uso posterior.

Anterior Próxima Contenido

Si deseás contactarte con el autor podés hacerlo en inglés a alan.gauld@btinternet.com o en español a manilio@xoommail.com