

Decisiones y más decisiones

La tercera columna de nuestro edificio fundamental son las bifurcaciones o las *instrucciones condicionales*. Estos términos expresan la facultad dentro de un programa de ejecutar una u otra porción de código (bifurcación) según se cumpla una determinada condición.

En los viejos tiempos del Assembler la bifurcación más simple era la instrucción JUMP que permitía "saltar" a una dirección de memoria específica si el resultado de la instrucción anterior era cero. Se escribieron programas asombrosamente complejos utilizando esta simple instrucción como única bifurcación posible reivindicando la opinión de Dijkstra acerca de los requerimientos mínimos para la programación. Cuando surgieron los lenguajes de alto nivel, apareció una nueva versión de la instrucción JUMP llamada GOTO. De hecho BASIC aun provee la instrucción GOTO y podés probarla utilizando el siguiente código:

```
10 PRINT "Comenzamos en línea 10"
20 J = 5
30 IF J < 10 GOTO 50
40 Print "Esta línea nunca se imprime"
50 STOP
```

Fijate que incluso en un programa tan breve como este resulta difícil darse cuenta de qué está pasando. No hay una estructura de código, simplemente uno tiene que imaginársela mientras lee el listado. En un programa más largo esto se vuelve imposible, por lo cual la mayor parte de los lenguajes de programación modernos carecen de instrucciones JUMP o GOTO, o si las incorporan, no recomiendan en absoluto su utilización.

La instrucción IF

La instrucción condicional más obvia e intuitiva es la formada por el constructo if, then, else ("si... entonces... de lo contrario"). Sigue la lógica de nuestro lenguaje cotidiano en tanto que si (*if*) cierta condición booleana es verdadera, entonces (*then*) un bloque de instrucciones es ejecutado; de lo contrario (*else*) se ejecuta un bloque distinto de instrucciones.

En BASIC esto resultaría así:

```
PRINT "Aquí empezamos"
J = 5
IF J > 10 THEN
    PRINT "Esto nunca se imprimirá"
ELSE
    STOP
END IF
```

Afortunadamente esta estructura es más sencilla de leer y comprender que la anterior del ejemplo del GOTO. En el IF podemos utilizar cualquier tipo de condición que pueda evaluarse en términos de verdadero o falso (valores booleanos).

En Python es muy similar:

```
import sys # para permitirnos salir del programa
print "Aquí comenzamos"
j = 5
if j > 10:
    print "Esto nunca se imprime"
else:
    sys.exit()
```

Es prácticamente igual ¿no?

Es posible encadenar varias estructuras if/then/else unas dentro de otras con una técnica denominada *condiciones anidadas*:

```
# Suponemos ancho creado previamente
if ancho == 100:
    area = 0
else:
    if ancho == 200:
        largo = largo * 2
    else:
        if ancho == 500:
            ancho = ancho/10
        else:
            print "El ancho tiene un valor inesperado"
```

Nota: utilizamos == para comprobar la igualdad en cada una de las instrucciones IF, mientras que utilizamos = para asignar un valor a una variable. Confundir el uso de estos dos signos es un error común en la programación en Python. Afortunadamente, el intérprete nos advierte que hay un error de sintaxis, pero a veces hay que revisar bastante hasta hallar el problema.

Las instrucciones CASE

Este tipo de construcción es tan común que muchos lenguajes proveen una bifurcación especial para ella. En general se conocen como instrucciones CASE o SWITCH. En el caso de Tcl se utiliza así:

```
switch $ancho {
  100 { set area 0}
  200 { set largo [expr {$largo * 2}] }
  500 { set ancho [expr {$ancho / 2}] }
}
```

Ni BASIC ni Python incluyen esta construcción, pero presentan un formato más sencillo bajo la forma de IF/ELSEIF/ELSE:

```
if ancho < 100:
  area = 0
elif ancho < 200:
  largo = largo * 2
elif ancho < 500:
  ancho = ancho/10
else:
  print "El ancho es muy grande!"
```

Notá el uso de ELIF y el hecho de que el sangrado (tan importante en Python) no se modifica. Es importante destacar que las dos versiones en Python de este programa son igualmente válidas, siendo la segunda un poco más sencilla de leer si hay muchas comprobaciones.

BASIC incluye una versión un poco más extraña en la forma de ElseIf...THEN que se utiliza exactamente igual que el ELIF de Python.

Anterior Próxima Contenido

Si tenés sugerencias o dudas podés enviar un email en inglés a: alan.gauld@btinternet.com o en español a: manilio@xoommail.com