El futuro digital es de todos | MinTIC

UNIVERSIDAD EL BOSQUE | MisiónTIC 2022

Ciclo 1

Semana 4

Estructuras de control: repetición o iteración

Lectura 2 – *La Sentencia while*

Convenio de asociación no. 777 de 2021 suscrito entre el Fondo Único de Tecnologías de la Información y las Comunicaciones y la Universidad el Bosque.

La Sentencia while

Esta sentencia comienza con la palabra reservada ***while***, luego viene una condición que terminará en el símbolo (:) dos puntos, y después todo el cuerpo del ***while***, es decir, todas las instrucciones que serán repetidas indentadas a la derecha. El siguiente ejemplo cuenta hacia atrás desde 5 y luego muestra la palabra '¡Despegue!':

```
n = 5
while n > 0:
    print(n)
    n = n - 1
print('¡Despegue!')
```

```
5
4
3
2
1
¡Despegue!

Process finished with exit code 0
```

El flujo de ejecución de la sentencia ***while*** es el siguiente:

1. Se evalúa la condición, obteniéndose verdadero o falso. Para el ejemplo se evalúa si n es o no positiva.
2. Si la condición es falsa, para el ejemplo n es menor o igual a 0, el cuerpo del ***while*** no se ejecuta y se continúa con la siguiente instrucción, para el caso ***print('¡Despegue!').***
3. Si la condición es verdadera, para el ejemplo n es positiva, se ejecuta el cuerpo del ***while*** y luego se retorna al paso 1, a evaluar nuevamente la condición. En el ejemplo el cuerpo del ***while*** es: ***print(n) y n = n – 1***, solo esas dos instrucciones.

Ahora se puede entender mejor por qué se le llama bucle o ciclo, al observar que el paso 3 enlaza o retorna de nuevo al paso 1. En el ejemplo, se puede decir, que el ciclo o bucle se ejecutó 5 veces, o también que tuvo 5 iteraciones. Es importante señalar que en el cuerpo del ciclo deben ocurrir algunos cambios en las variables, de tal forma que la condición en algún momento sea evaluada falsa y el bucle termine, de lo contrario se tendría un ciclo infinito que nunca terminaría. La variable sobre la cual se evalúa la condición recibe el nombre de variable de iteración, para el ejemplo, la variable n.

Suponga que se va a leer una serie de números positivos desde el teclado, y se requiere decir cuántos fueron, el ciclo debe detenerse cuando el usuario ingrese un valor menor o igual a cero. Se debe pensar en dos variables, una para el número ingresado y otra para el conteo de los números. Una posible solución sería:

```
algoritmo cuenta_enteros
var
  entero : numero, contador
inicio
  contador ← 0
  leer(numero)
  mientras numero > 0 hacer
    leer(numero)
    contador ← contador + 1
  fin_mientras
  escribir('El numero de enteros positivos es', contador)
fin
```

Figura 1: Tomado de Joyanes

La secuencia de las acciones de este algoritmo se puede reflejar en el siguiente pseudocódigo:

Paso	Pseudocódigo	Significado
1	`contador ← 0`	inicializar contador a 0
2	**`leer`**`(numero)`	leer primer número
3	**`mientras`** `numero > 0` **`hacer`**	comprobar si `numero > 0`. Si es así, continuar con el paso 4. Si no, continuar con el paso 7
4	`sumar 1 a contador`	incrementar `contador`
5	**`leer`**`(numero)`	leer siguiente `numero`
6	`regresar al paso 3`	evaluar y comprobar la expresión booleana
7	**`escribir`**`(contador)`	visualizar resultados

Puede suceder que un ciclo ***while*** no se ejecute ni una sola vez, porque lo primero que se hace en esta estructura es evaluar la expresión booleana, y si es falsa el ciclo no se ejecuta. Como se puede ver en el siguiente ejemplo la expresión booleana ***(n <= 4)*** nunca se cumplirá y no se realizará ninguna acción del cuerpo del ciclo, se seguirá con la instrucción fin. Antes del ciclo mientras se asigna el valor de 5 a la variable n y el valor de cero a la variable s, lo que se hace una sola vez.

```
inicio
  n ← 5
  s ← 0
  mientras n <= 4 hacer
    leer(x)
    s ← s + x
  fin_mientras
fin
```

Figura 2: Tomado de Joyanes