

LA SENTENCIA while

Un bucle (ciclo) es cualquier construcción de programa que repite una sentencia o secuencia de sentencias un número de veces. La sentencia (o grupo de sentencias) que se repiten en un bloque se denomina cuerpo del bucle y cada repetición del cuerpo del bucle se llama iteración del bucle.

Las dos principales cuestiones de diseño en la construcción del bucle son: ¿Cuál es el cuerpo del bucle? ¿Cuántas veces se iterará el cuerpo del bucle?

Un bucle while tiene una condición del bucle (una expresión lógica) que controla la secuencia de repetición. La ejecución de la sentencia o sentencias expresadas se repite mientras la condición del bucle permanece verdadera y termina cuando se hace falsa. La condición del bucle se evalúa antes de que se ejecute el cuerpo del bucle y, por consiguiente, si esta condición es inicialmente falsa, el cuerpo del bucle no se ejecutará. En otras palabras, el cuerpo de un bucle while se ejecutará cero o más veces.

La forma general de la instrucción while es:

while (condición-bucle)

```
{  
sentencia 1;  
sentencia 2 ;  
.....  
sentencia n ;  
}
```

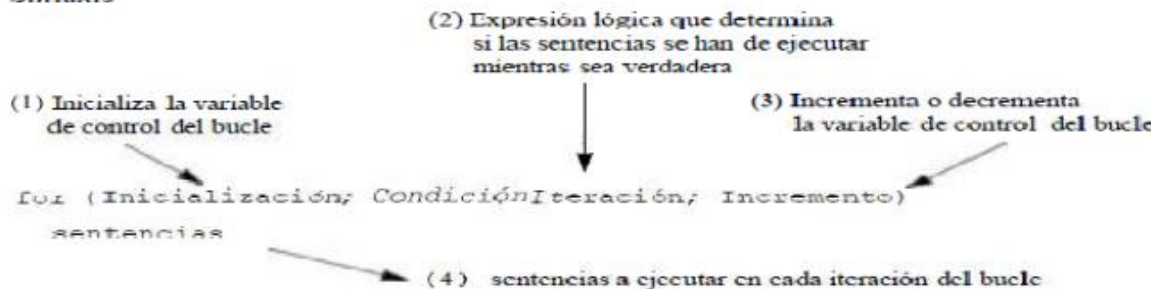
El comportamiento o funcionamiento de una sentencia
(bucle) while es:

- 1. Se evalúa la condición del bucle.
- 2. Si condición del bucle es verdadera (distinto de cero):
 - a. La sentencia especificada, denominada el cuerpo del bucle, se
 - b. Vuelve el control al paso 1.
- 3. En caso contrario: El control se transfiere a la sentencia siguiente al bucle o sentencia while.

EL BUCLE for

El bucle for es el más adecuado para implementar bucles controlados por contador que son bucles en los que un conjunto de sentencias se ejecuta una vez por cada valor de un rango especificado. La sentencia for (bucle for) es un método para ejecutar un bloque de sentencias un número fijo de veces.

Sintaxis



El bucle for se diferencia del bucle while en que las operaciones de control del bucle se sitúan en un solo sitio: la cabecera de la sentencia.

El bucle for contiene las cuatro partes siguientes:

- Parte de inicialización, que inicializa las variables de control del bucle. Se pueden utilizar variables de control del bucle simples o múltiples.
- Parte de condición, que contiene una expresión lógica que hace que el bucle realice las iteraciones de las sentencias, mientras que la expresión sea verdadera.
- Parte de incremento, que incrementa o decrementa la variable o variables de control del bucle.
- Sentencias, acciones o sentencias que se ejecutarán por cada iteración del bucle.

La sentencia for es equivalente al siguiente código while:

Inicialización;

while(condición iteración) {

Sentencia(s); incremento; }

Existen dos formas de implementar la sentencia for que se utilizan normalmente para implementar bucles de conteo: formato ascendente, en el que la variable de control se incrementa y formato descendente, en el que la variable de control se decrementa.

Formato ascendente:

```
for(var_control=valor_inicial; var_control<=valor_límite; expresión_incremento)  
sentencia(s);
```

Formato descendente:

```
for(var_control=valor_inicial; var_control>=valor_límite;  
expresión_decremento) sentencia(s);
```

LA INSTRUCCIÓN DO-WHILE

Cuando se construye un bucle utilizando la instrucción while, la comprobación para la continuación del bucle se realiza al comienzo de cada pasada. Sin embargo, a veces es deseable disponer de un bucle en el que se realice la comprobación al final de cada pasada. Esto se puede hacer mediante la instrucción do...while.

La forma general de la instrucción do-while es:

```
do  
instrucción(es);  
while(expresión)
```

La instrucción se ejecutará repetidamente, mientras que el valor de expresión sea verdadero (es decir, distinto de cero).

Observe que instrucción siempre se ejecutará al menos una vez, ya que la comprobación de la condición de repetición no se realiza hasta el final de la primera pasada por el bucle. La instrucción puede ser simple o compuesta, aunque en la mayoría de las veces será compuesta. Debe incluir algún elemento que altere el valor de expresión para que el bucle tenga un final.