**Ciclo while**

En lenguaje C, para escribir una instrucción while (repetitiva mientras) se utiliza la sintaxis:

While (<expresión\_lógica>)

{

<bloque\_de\_instrucciones>

}

Cuando el <bloque\_de\_instrucciones> sólo contiene una instrucción, los caracteres abrir llave ({) y cerrar llave (}) son opcionales.

Por otra parte, al igual que en las instrucciones alternativas doble y simple, a la <expresión\_lógica> de una instrucción repetitiva while, también se le llama condición.

Para que se ejecute el <bloque\_de\_instrucciones>, la condición tiene que ser verdadera. Por el contrario, si la condición es falsa, el <bloque\_de\_instrucciones> no se ejecuta.

Por tanto, cuando el flujo de un programa llega a un bucle while, existen dos posibilidades:

Si la condición se evalúa a falsa, el bloque de instrucciones no se ejecuta, y el bucle while finaliza sin realizar ninguna iteración.

Si la condición se evalúa a verdadera, el bloque de instrucciones sí que se ejecuta y, después, se vuelve a evaluar la condición, para decidir, de nuevo, si el bloque de instrucciones se vuelve a ejecutar o no. Y así sucesivamente, hasta que, la condición sea falsa.

El <bloque\_de\_instrucciones> de un bucle while puede ejecutarse cero o más veces (iteraciones). Si el <bloque\_de\_instrucciones> se ejecuta al menos una vez, seguirá ejecutándose repetidamente, mientras que, la condición sea verdadera. Pero, hay que tener cuidado de que el bucle no sea infinito.

Cuando la condición de un bucle while se evalúa siempre a verdadera, se dice que se ha producido un bucle infinito, ya que, el programa nunca termina. Un bucle infinito es un error lógico.

Es importante hacer hincapié en el hecho de que, en un bucle while, primero se evalúa la condición y, en el caso de que ésta sea verdadera, entonces se ejecuta el bloque de instrucciones. Veremos que, en el bucle do while, el procedimiento es al revés. En él, primero se ejecuta el bloque de instrucciones y, después, se evalúa la condición.

Para que un bucle while no sea infinito, en el bloque de instrucciones debe ocurrir algo para que la condición deje de ser verdadera. En la mayoría de los casos, la condición se hace falsa al cambiar el valor de una variable.

En resumen, una instrucción repetitiva while permite ejecutar, repetidamente, (cero o más veces) un bloque de instrucciones, mientras que, una determinada condición sea verdadera.

**Ciclo do while**

En lenguaje C, para escribir una instrucción do while (repetitiva hacer mientras) se utiliza la sintaxis:

Do

{

<bloque\_de\_instrucciones>

} While (<expresión\_lógica>);

Cuando el <bloque\_de\_instrucciones> sólo contiene una instrucción, los caracteres abrir llave ({) y cerrar llave (}) son opcionales.

Además, como se puede apreciar, la instrucción repetitiva do while, también hace uso de una condición.

En un bucle do while, primero se ejecuta el bloque de instrucciones y, después, se evalúa la condición. En el caso de que ésta sea verdadera, se vuelve a ejecutar el bloque de instrucciones. Y así sucesivamente, hasta que, la condición sea falsa.

Por consiguiente, cuando el flujo de un programa llega a un bucle do while, existen dos posibilidades:

Se ejecuta el bloque de instrucciones y, después, si la condición se evalúa a falsa, el bloque de instrucciones no se vuelve a ejecutar, de manera que, el bucle do while finaliza, habiendo realizado una sola iteración.

Se ejecuta el bloque de instrucciones y, a continuación, si la condición se evalúa a verdadera, el bloque de instrucciones se vuelve a ejecutar. Y así sucesivamente, hasta que la condición sea falsa.

El <bloque\_de\_instrucciones> de un bucle do while puede ejecutarse una o más veces (iteraciones). También hay que prevenir que el bucle no sea infinito.

En resumen, una instrucción repetitiva do while permite ejecutar repetidamente (una o más veces) un bloque de instrucciones, mientras que, una determinada condición sea verdadera.

**Ciclo for**

El ciclo For es una de las instrucciones más sencillas de aprender, y consiste en utilizar mayormente “rangos” en los cuales se define el número de iteraciones que se pueden hacer en este ciclo.

Declaración y sintaxis

La sintaxis es la siguiente:

1

For (inicio; mientras; incremento)

2

{

3

//CODIGO A EJECUTAR

4

}

Donde el inicio es la declaración de una variable que funciona como un “contador” mientras ejecutamos el ciclo. Continuamente el “mientras” especifica los valores o el rango de valores que puede tomar el contador de “inicio” y el “incremento” especifico cuanto se va a incrementar el contador con cada iteración, lo que indicaría que eventualmente el contador saldría de su posible rango de valores y terminaría el ciclo.

Ejemplo de serie de números pares con For

El siguiente ejemplo muestra un sencillo programa que imprime una serie de números pares utilizando el ciclo For, especificando que x comienza con un valor de 2, y que tendrá incrementos de 2 en 2 mientras sea menor a 20, el resultado será la impresión de valores del 2 al 18(solo números pares).

View source

01

#include <stdio.h>

02

int main(int argc, const char \* argv[])

03

{

04

//CICLOS FOR EN C

05

int y;

06

for(int x=2;x<20;x+=2)

07

{

08

printf("El contador X vale: %d\n",x);

09

}

10

return 0;

11

}