Autoestudio 3: Sentencia de control IF

Cuando escribimos un algoritmo es común llegar a casos donde se requiere tomar una decisión dependiendo de un estado o de una condición. En estos casos, los lenguajes de programación tienen un conjunto de sentencias llamadas **sentencias de control**, las cuales como su nombre lo indica, controlan la ejecución de instrucciones en un programa.

En el lenguaje de programación **C**,cuando tenemos un proceso de decisión con relación a una condición lo representamos mediante la estructura de control **if**. Como su nombre lo indica, la sentencia de control if corresponde a decir: **si (condición) entonces**

Es decir, la instrucción siguiente se ejecuta sólo si la condición se cumple, lo que nos permite determinar que una sección se ejecute o no, dependiendo de una condición.

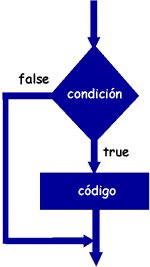
La **condición** es una expresión de comparación que al evaluarse obtiene como resultado **verdadero** (distinto de cero) o **falso** (cero), y es necesario que se encuentre entre paréntesis. La sintaxis es la siguiente:

**if** ( **condición** )

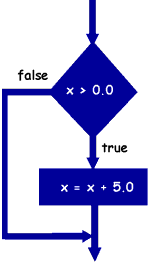
{

Instrucciones a ejecutarse si la condición es verdadera;

}

****

Cuando se ejecuta la estructura if, **primero** se evalúa la expresión de comparación (**condición**), si el resultado es **verdadero** (**true**) entonces se ejecutan las instrucciones del código del if. El diagrama de flujo en azul representa la forma en la que se ejecuta el código del estatuto de control if

****Un ejemplo de un código que hace uso del if sería:

**if** ( x > 0.0 )

{

x = x + 5.0;

}

Donde si el valor de la variable **x** es mayor que **0**, entonces su valor se incrementa en **5**. El diagrama de flujo a la derecha corresponde a este código.

Como se puede observar los operadores condicionales como el if se usan en conjunto con los operadores de comparación para especificar las condiciones que queremos verificar. Es por eso que es común encontrar los operadores de comparación < > == <= >= en la parte de la condición de la estructura de control.

# if - else

Cuando queremos expresar opciones excluyentes usamos la sentencia **else**, que correspondería a decir **si (condición) entonces**, **sino entonces.**

**if** ( **condición** )

{

Instrucciones a ejecutarse si la condición es ***verdadera***;

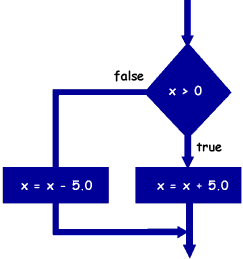
}

**else**

{

Instrucciones a ejecutarse si la condición es ***falsa***;

}

Cuando se ejecuta la estructura **if-else**, **primero** se evalúa la expresión de comparación (**condición**), si el resultado es **verdadero** (true) entonces se ejecutan solamente las instrucciones entre las llaves del if, pero si la condición es **falsa** (false) entonces se ejecutan solamente las instrucciones entre las llaves del else.

if ( x > 0.0 )

{

x = x + 5.0;

}

else

{

x = x - 5.0;

}

En este caso, si el valor de la variable **x** es Mayor que **0**, entonces su valor se incrementa en **5**, de otra manera, a su valor se le resta **5**. Es importante destacar que el uso del **else** hace que al ejecutar un código ya no ejecute ninguno de los demás que estén en la misma estructura **if**.

# if anidado

Un if o un else puede tener adentro otro if. Cuando esto ocurre se conoce como un if anidado y representa cuando queremos saber sobre varias condiciones. Su sintaxis es la misma del if.

**if** ( **condición** **1**)

{

**if** ( **condición** **2**)

{

código 1;

}

**else**

{

código 2;

}

}**else**{

código 3;

}

Esto implica que para que se pueda ejecutar el código 1 o el código 2 primero se tiene que cumplir con la condición 1. En caso de no cumplir con la condición 1 solo se ejecuta el código 3 Un ejemplo de esto sería:

if ( edad > 17)

{

if ( edad >60)

{

printf(“tercera edad”);

}

else

{

printf(“adulto”);

}

} else if (edad > 0 ) {

printf(“menor de edad”);

} else {

printf(“… dude ….eso todavía no existe :/ ”);

}

Las 3 formas de if que hemos visto if, else y else if se pueden combinar y anidar para representar las estructuras de decisión de cualquier algoritmo.

# Si deseas saber más puedes consultar:

(Joyanes Aguilar, L. y Zahonero Martínez, I. *Programación en C, C++, Java y UML*. México : McGraw Hill, 2010.) las siguientes secciones:

* Estructuras de control. (pp. 127)
* La sentencia if. (pp. 127)
* Sentencia if de dos opciones: if-else. (pp. 130)
* Sentencias if-else anidadas. (pp. 133)
* Evaluación en corto circuito de expresiones lógicas. (pp. 142)
* Puesta a punto de programas. (pp. 143)