

# Condicional

martes, 7 de septiembre de 2021 07:20 a. m.

Nos permite elegir entre un camino u otro.

Condicional If-else:

```
If(condicional (T o F)){
    //instrucciones secuenciales que se ejecutan cuando el condicional se cumple
} else {
    //instrucciones secuenciales que se ejecutan cuando el condicional no se cumple
}
```

//Condicional if-else if- else:

```
If(condicional 1){
    //instrucciones secuenciales que se ejecutan cuando el "condicional 1" se cumple
} else if(condicional 2) {
    //instrucciones secuenciales que se ejecutan cuando la condicional 2 se cumple, pero la
    condicional 1 no se cumple
} else{
    //instrucciones que se ejecutan para cualquier otro caso
}
```

// Anidación de condicionales if

```
If(condicional 1){
    If(condicional 2){
        //Instrucciones que se ejecutan siempre y cuando se cumplan las dos condicionales
    } else{
        //Se ejecutara cuando la condicional 1 es verdadera y la condicional 2 sea falsa.
        If(condicional 3){
            //instrucciones que se ejecutan cuando la condicional 1 es verdadera; la
            condicional 2 es falsa y la condicional 3 es verdadera.
        }
    }
} else {
    //Para cualquier caso donde no se cumpla la condición 1.
}
```

**Ejemplo:**

```
/*
Programa: Permitir al usuario las siguientes operaciones:
1.- Imprimir "Hola"
2.- Imprimir "Adios"
3.- Raiz cuadrada de 2
4.- Salir del programa
```

```
*/
```

```
#include <stdio.h>
```

```
#include <math.h> //Funciones sqrt, exp, log...
```

```
void main(){
```

```
    int opcionUsuario=0;
```

```
    puts("Seleccione una opcion:");
```

```
    puts("1.- Imprimir \"Hola\"");
```

```
    puts("2.- Imprimir \"Adios\"");
```

```
    puts("3.- Raiz cuadrada de 2");
```

```
    puts("4.- Salir del programa");
```

```
    fflush(stdin);
```

```
    scanf("%d",&opcionUsuario); //Se leera un entero del teclado y se almacenará en la variable  
    opcionUsuario
```

```
    if(opcionUsuario==1){
```

```
        puts("Hola");
```

```
    } else if (opcionUsuario==2){
```

```
        puts("Adios");
```

```
    } else if (opcionUsuario==3){
```

```
        printf("El resultado de la raiz de 2 es: %1.3f",sqrt(2));
```

```
    } else{
```

```
        puts("Fin del programa");
```

```
    }
```

```
}
```

```
/*
```

```
    Programa: Apartir de un valor ingresado por el usuario, indicar si este es un }  
    número para o impar positivo o negativo.
```

```
*/
```

```
#include <stdio.h>
```

```
#include <math.h> //Funciones sqrt, exp, log...
```

```
void main(){
```

```
    unsigned int valorDeUsuario=0;
```

```
    puts("Ingrese un numero entre 0-4294967295:");
```

```
    fflush(stdin);
```

```
    scanf("%u",&valorDeUsuario); //Se leera un entero del teclado y se almacenará en la variable  
    opcionUsuario
```

```
    printf("El valor leido es: %u\n", valorDeUsuario);
```

```
    if(valorDeUsuario%2==0){
```

```
        if(valorDeUsuario>=0)
```

```
            puts("El valor ingresado es un par positivo");
```

```
        else
```

```
            puts("El valor ingresado es un par negativo");
```

```
    } else{
```

```
        if(valorDeUsuario>=0)
```

```
            puts("El valor ingresado es un impar positivo");
```

```
        else
```

```
            puts("El valor ingresado es un impar negativo");
```

```
    }
```

```
}
```

Condicional basada en casos, es decir, switch & case:

```
//Sintaxis
switch(variable(expresión)){ // variable debe almacenar un simple valor: 'a', 1, -10, '$'
    case constante1:
        //instrucciones que se deben ejecutar si se cumple la siguiente condición:
        //expresión == constante1
        break;
    case constante2:
        //instrucciones que se deben ejecutar si se cumple la siguiente condición:
        //expresión == constante2
        break;
    ...
    default:
        //instrucciones que se ejecutarán para cualquier otro caso
        break;
}
```

Ejemplo switch & case:

```
/*
    Programa: Permitir al usuario las siguientes operaciones:
        1.- Imprimir "Hola"
        2.- Imprimir "Adios"
        3.- Raiz cuadrada de 2
        4.- Salir del programa
*/

#include <stdio.h>
#include <math.h> //Funciones sqrt, exp, log...

void main(){
    int opcionUsuario=0;
    puts("Seleccione una opcion:");
    puts("1.- Imprimir \"Hola\"");
    puts("2.- Imprimir \"Adios\"");
    puts("3.- Raiz cuadrada de 2");
    puts("4.- Salir del programa");
    fflush(stdin);
    scanf("%d",&opcionUsuario); //Se leera un entero del teclado y se almacenará en la variable
    opcionUsuario
    switch(opcionUsuario){
        case 1:
            puts("Hola");
            break;
        case 2:
            puts("Adios");
    }
```

```

        break;
    case 3:
        printf("El resultado de la raiz de 2 es: %1.3f",sqrt(2));
        break;
    default:
        puts("Fin del programa");
        break;
}
}

```

Condicional con el operador ternario: "?" ":"

Sintaxis:

<<Condición>> ? <Expresión1> : <Expresión2>

La "Expresión1" se ejecuta cuando la condicional da como resultado "Verdadero", en otro caso se ejecutará la "Expresión2".

Sintaxis de la anidación de operador ternario:

<<Condición1>> ? <<Condición2>> ? <Expresión1> : <Expresión2> : <Expresión3>

La "Expresión1" se ejecuta cuando la condicion1 y la condicion2 son verdaderas, la "Expresión2" se ejecutará cuando la Condición1 es verdadera y la condición2 es falsa y finalmente, la "Expresión3" se ejecutará cuando la condición1 se falsa.

Se utilizará en la medida de lo posible cuando sea necesario ejecutar una instrucción cuando una expresión es verdadera o falsa.

Ejemplo:

```

#include <stdio.h>
#include <math.h> //Funciones sqrt, exp, log...

void main(){
    int opcionUsuario=0;
    puts("Ingrese un valor:");
    fflush(stdin);
    scanf("%d",&opcionUsuario); //Se leera un entero del teclado y se almacenará en la variable
    opcionUsuario
    opcionUsuario%2==0?opcionUsuario<0?puts("El valor es par negativo"):puts("El valor es par
    positivo"):puts("El numero es impar");

    /*if(opcionUsuario%2==0){
        puts("El numero es par")
    } else{
        puts("El numero es impar"
    }*/
}

```



# Estructuras cíclicas

miércoles, 8 de septiembre de 2021 08:04 a. m.

Las estructuras cíclicas nos sirven para repetir N veces una secuencia de instrucciones (pueden contener otras estructuras de control). Las estructuras cíclicas son:

- For
- While
- Do-while
- Recursividad

Estructura cíclica "for".

Sintaxis:

```
<tipo dato> indice; //declaración del contador/índice para un ciclo
for(<inicialización de contador indice>;<Condición de paro>; <incremento/decremento indice>){
    //instrucciones que se repiten mientras la condición se cumpla
}
```

**Ejemplo:**

```
int indice;
for(indice=1; indice<=10;indice++){
    //instrucciones que se ejecutaran 10 veces
    instrucción X;
}
```

...

Flujo:

```
Inicializa al índice (contador) en 1.
Evalúa la condición, si se cumple ejecuta la instrucción X
Se incrementa el contador (indice++)
Evalúa la condición, si se cumple ejecuta la instrucción X
Se incrementa el contador (indice++)
Evalúa la condición, si se cumple ejecuta la instrucción X
Se incrementa el contador (indice++)
Evalúa la condición, si se cumple ejecuta la instrucción X
Se incrementa el contador (indice++)
Evalúa la condición, si se cumple ejecuta la instrucción X
Se incrementa el contador (indice++)
...
Evalúa la condición, si esta no se cumple el ciclo termina.
```

Ejemplo anidación de ciclos for:

```
int indice, indice2;
for(indice=1; indice<=10;indice++){
    //instrucciones que se ejecutaran 10 veces
    for(indice2=0; indice2<=2; indice2++){
        //instrucciones se ejecutaran 3*10 veces
        instrucción X;
    }
}
```

```

        //instrucciones que se ejecutaran 10 veces
        instrucción Y;
    }

    int indice, indice2, indice3;
    for(indice=1; indice<=10; indice++){
        //instrucciones que se ejecutaran 10 veces
        for(indice2=0; indice2<=2; indice2++){
            //instrucciones se ejecutaran 3*10 veces
            for(indice2=0; indice2<=1; indice2++){
                //instrucciones se ejecutaran 10*3*2 veces
                instrucción z

                //operador ternario
                //switch & case
                //if-else
                //if-else if-else

            }
        }

        //instrucciones que se ejecutaran 10 veces
        instrucción Y;
    }

```

Sintaxis de while es la siguiente:

```

while( <condición>){
    //instrucciones que se ejecutaran mientras la condición se cumple
}

```

Sintaxis de anidación del while

```

while(<condición 1>){
    //instrucciones que se ejecutan únicamente si la condición 1 se cumple.
    while(<condición 2>){
        //instrucciones que se ejecutan mientras la condición 1 y la condición 2 se cumplen
    }
}

```

```

while(<condición> && <condición 2>){
    //instrucciones que se ejecutan mientras la condición y la condición 2 se cumplen

    //operador ternario
    //switch & case
    //if-else
    //if-else if-else
    //for
    //while
    //do-while
}

```

```
}
```

Sintaxis do-while:

```
do{
    //instrucciones que se ejecutan al menos una vez y más veces si la condición se cumple.
    //operador ternario
    //switch & case
    //if-else
    //if-else if-else
    //for
    //while
    //do-while
}while(<condición>)
```

El do-while permite que por lo menos una vez se ejecuten las instrucciones dentro de esta estructura:

Ejemplo:

```
/*
    Programa:
        1.- Ejecutar un contador de 0 a 1000
        2.- Ejecutar un contador de 1000 a 0, de 2 en 2
        3.- Evaluar numeros pares positivos e imprimir el doble del valor
        4.- Obtener raiz cuadrada de todo valor impar entre 100 y 200
        5.- Salir del programa

    Nota El menú de usuario se debe desplegar cada vez que se termine de ejecutar una
    operación y hasta que el
    usuario decida terminar el programa.
*/

#include<stdio.h>
#include<math.h>
#include<stdlib.h>

void main(){
    int contador=0; //Se inicializa en 0 para no tener basura en la variable
    int opcionUsuario=0;
    int valorUsuarioEvaluacion=0;
    double raizCuadrada=0.0;
    do{
        //Por lo menos una vez se muestra el menú de usuario y se
        realiza la operación
        puts("Seleccione una opción:");
        puts("1.- Ejecutar un contador de 0 a 1000");
        puts("2.- Ejecutar un contador de 1000 a 0, de 2 en 2");
        puts("3.- Evaluar numeros pares positivos e imprimir el doble del valor");
        puts("4.- Obtener raiz cuadrada de todo valor impar entre 100 y 200");
        puts("5.- Salir del programa");
        fflush(stdin);
        scanf("%d",&opcionUsuario);
        switch(opcionUsuario){
```



```

case 1:
    contador=0;
    while(contador<=1000){
        printf("El estdo actual del contador es: %d\n",contador);
        contador++;
    }
    break;
case 2:
    for(contador=1000; contador>=0;contador-=2) //no se requieren llaves
    porque solo tiene 1 instrucción que repetir
        printf("El estdo actual del contador es: %d\n",contador);
    break;
case 3:
    puts("Por favor dame un numero");
    fflush(stdin);
    scanf("%d",&valorUsuarioEvaluacion);
    if(valorUsuarioEvaluacion%2==0 && valorUsuarioEvaluacion>=0){
        printf("El nuevo valor es: %d\n",2*valorUsuarioEvaluacion);
    } else{
        puts("El valor dado no es par positivo");
    }
    break;
case 4:
    for(contador=100; contador<=200; contador++){
        if(contador%2!=0)
            printf("La raiz de %d es: %1.3f\n",contador, sqrt(contador));
    }
    break;

default:
    puts("Nada que hacer");
    exit(0); //Se cierra el programa avisando al S.O. que no ocurrio ningun error.
    break;
}
}while(1); // Condición infinita que requiere de que internamente exista una llamada a la
función "exit()"
//while(opcionUsuario!=5) // condición simple que valida la opción del usuario y se repite
mientras que el usuario no seleccione 5
}

```

