

# COMPILADO DE SINTAXIS EN C++

## 1. Tipos de datos y operadores

- **Identificadores o variables. –**

- Formados por letras, números y '\_'.
- Las mayúsculas y las minúsculas se distinguen.
- Las **palabras reservadas** van en **minúsculas**.

- **Tipos de datos.-**

TIPO DE DATO	DESCRIPCION	ANCHO DE BIT	RANGO EN PC
<b>char</b>	Carácter (un solo carácter)	8	-128 a 127
<b>int</b>	Número entero	16	-32768 a 32767
<b>float</b>	Números con punto decimal	32	3.4E-38 a 3.4E+38
<b>double</b>	Números con punto decimal de mayor rango	64	1.7E-308 a 1.7E+308
<b>void</b>	Sin tipo(uso) especial	0	Sin valores

- **Declaraciones:** Todas las variables deben declararse antes de usarse. Una declaración especifica un tipo de dato seguido de una lista no vacía de variables de ese tipo (separadas por comas).

Ejemplos:

```
float f1, f2;  
char d, e;  
int numero;  
int x, y, z;  
int i = 1; /* Inicializada al definirla */
```

- **Constantes:**

**#define** <nombre> <valor>

El #define es una directiva para el compilador que indica que se reemplazará <nombre> por <valor>.

Ejemplos:

```
#define MAYOR 50
```

```
#define pi 3.1416
```

```
#define verdad 1
```

- Operadores

### Aritméticos

Operador	Función	Uso
=	asignación	<code>int i = 7;</code>
*	multiplicación	<code>double r = 3.5 * i;</code>
/	división	<code>double t = r / 1.2;</code>
%	modulo (resto)	<code>i = 21 % 6; // i = 3</code>
+	suma	<code>double rt = r + t;</code>
-	resta	<code>r = rt - t;</code>
++, --	incremento	<code>int i = 0;</code> <code>int j = i++; //j = 0</code> <code>int k = --i; //k = 0</code>
+=, -=, *=, /=	opera y asigna	<code>r += 2.6; //r = r + 2.6</code> <code>r *= 2.6; //r = r * 2.6</code>

### Relacionales y Lógicos

Operador	Función	Uso
<	menor que	<code>i &lt; 5</code>
<=	menor o igual que	<code>r &lt;= 5</code>
>	mayor que	<code>i &gt; 5</code>
>=	mayor o igual que	<code>i &gt;= 5</code>
==	igualdad	<code>i == 5</code>
!=	desigualdad	<code>i != 5</code>
!	NOT lógico	<code>!(i==5); //(i != 5)</code>
&&	Y lógico	<code>i &lt; 5 &amp;&amp; j &gt; 4</code>
	O lógico	<code>i &lt; 5    j &gt; 4</code>

## 2. Entrada y salida de datos.-

- **Entrada.-**

El comando **cin>>** se usa para leer cualquier tipo de dato predefinido desde el teclado, y convertirlo, de forma automática, al formato interno adecuado. La función se define como:

**cin>> nombre de variable;**

**Ejemplos:**

**Definidas las variables como:**

char genero;

int edad;

float peso;

cin>> nombre;

cin>>genero;

cin>>edad>>peso;

- **Salida. -**

El comando **cout<<** permite impresión por pantalla un mensaje o el valor de una variable:

**Ejemplos:**

cout<<"Introduzca su nombre (20 letras máximo): " ;

cout<<"El nombre que ha escrito es:"<< nombre;

- 

## 3. Estructuras de control.-

### 3.1 Condicionales

- **Sentencia Simple if** (si-entonces)

```
if (condición){  
    Instrucciones;  
}
```

- **Sentencia Compuesta if – else** (si-entonces-sino)

```
if (condición)
{
    Instrucciones, si la condición es verdadera;
}
else
{
    Instrucciones, si la condición es falsa;
}
```

- **Sentencia Condicional Múltiple switch** (Según hacer)

```
switch (variable)
{
    case constante1:
        Instrucciones;
        break;
    case constante2:
        Instrucciones;
        break;
    .....
    .....
    default :
        Instrucciones;
        break;
}
```

### 3.2 Ciclos.-

- **Sentencia for** (para -hacer)

```
for (inicialización; condición; incremento)
{
    Instrucciones del bucle;
}
```

**Ejemplo:**

```
for (i=1; i<=10; i++)
{
    Instrucciones del bucle;
}
```

## Un Programa ejemplo utilizando for

**EJEMPLO:** Realizar un programa para calcular el factorial de un numero

```
*main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3  //Funcion factorial
4  int factorial(int n){
5      int i,f=1;
6      for(i=n;i>=1;i--){
7          f=f*i;
8      }
9      return f;
10 }
11 //Programa Principal
12 int main()
13 {
14     int fact,nume;
15     printf("Ingrese un numero entero");
16     scanf("%d",&nume);
17     fact=factorial(nume);
18     printf("El factorial del numero %d es %d",nume,fact);
19 }
20
```

- **Sentencia while** (Mientras-hacer)

**while**(condición)

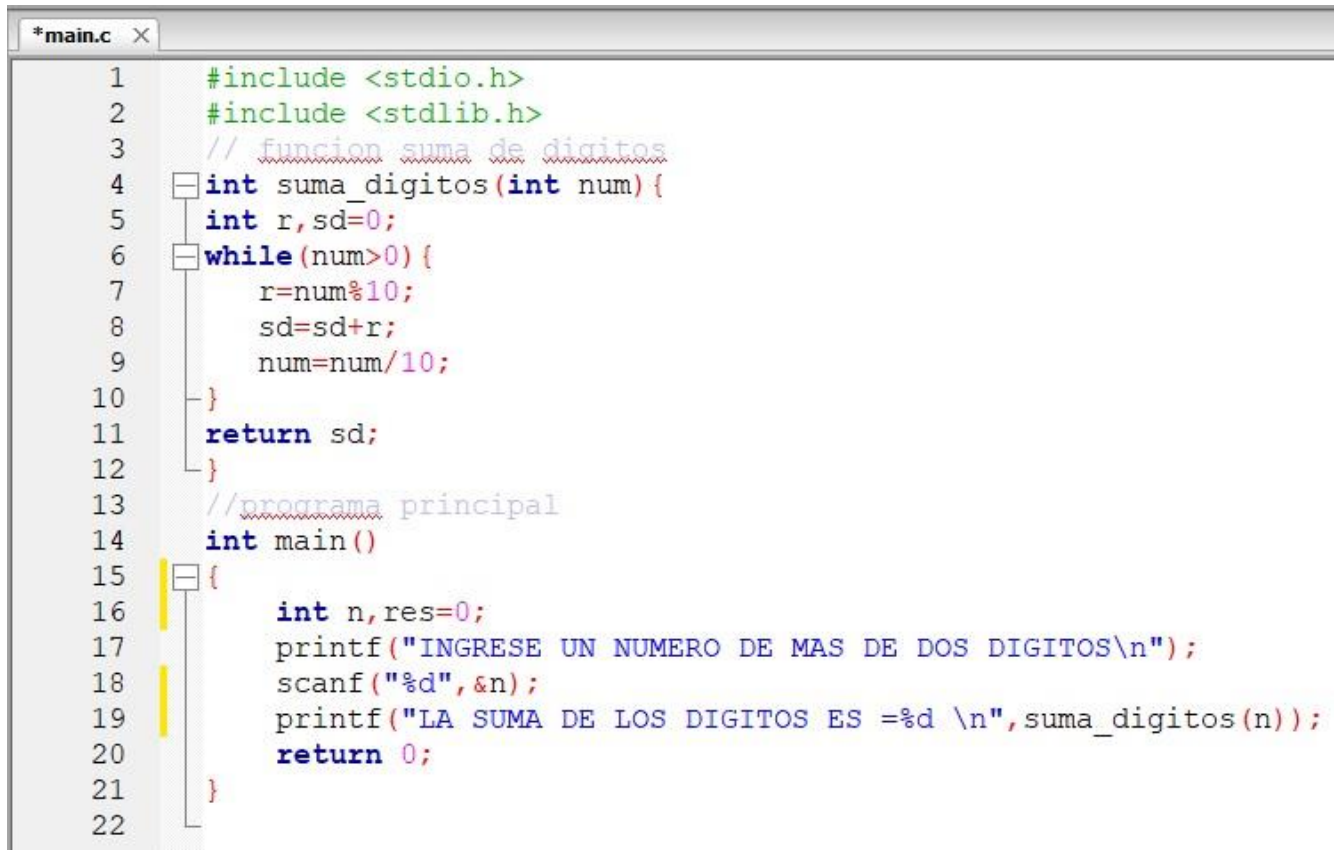
{

Instrucciones que se repetirán, si la condición es verdadera;

}

## Un Programa ejemplo utilizando **while**

**EJEMPLO:** Realizar un programa para realizar la suma de los dígitos de un numero



```
*main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3  // funcion suma de digitos
4  int suma_digitos(int num) {
5      int r, sd=0;
6      while(num>0) {
7          r=num%10;
8          sd=sd+r;
9          num=num/10;
10     }
11     return sd;
12 }
13 //programa principal
14 int main()
15 {
16     int n, res=0;
17     printf("INGRESE UN NUMERO DE MAS DE DOS DIGITOS\n");
18     scanf("%d", &n);
19     printf("LA SUMA DE LOS DIGITOS ES =%d \n", suma_digitos(n));
20     return 0;
21 }
22
```

- **Sentencia **do while**** (Mientras-hacer)

**do**

**{**

Instrucciones que se repetirán, si la condición es verdadera;

} while(condición);

### Un Programa ejemplo utilizando do-while

**EJEMPLO:** Realizar un programa para calcular los descuentos en una tienda si las compras son mayoa a 1000 se reaizara un descuento de 15% caso contrario solo del 5%.Repetir este proceso hasta que el usuario lo requiera

```
main.c x
1  #include <stdio.h>
2  #include <stdlib.h>
3  //FUNCION DESCUENTOS
4  float descuentos(float c) {
5      float desc=0;
6      if(c>1000)
7          desc=c*0.15;
8      else
9          desc=c*0.05;
10     return desc;
11 }
12 //PROGRAMA PRINCIPAL
13 int main()
14 {
15     float compra,d;
16     int resp=1;
17     do{
18         printf("Ingrese monto de compra ");
19         scanf("%f",&compra);
20         d=descuentos(compra);
21         printf("El descuento de la compra %.2f es = %.2f \n",compra,d);
22         printf("DESEA CONTINUAR? 1-->si 0-->no ");
23         scanf("%d",&resp);
24     }while(resp!=0);
25 }
```