

Tipos de datos básicos en C++

Tipos de datos básicos en C++

El **tipo de dato** determina la naturaleza del valor que puede tomar una variable. Un tipo de dato define un **dominio** de valores y las **operaciones** que se pueden realizar con éstos valores.

- **int** (Números enteros)
- **float** (Números reales)
- **double** (Números reales más grandes que float)
- **bool** (Valores lógicos true o false)
- **char** (Caracteres y cualquier cantidad de 8 bits)
- **void** (Nada. Sirve para indicar que una función no devuelve valores)

Modificadores

Tipo	Bytes	Bits	Mínimo valor	Máximo valor
bool	1	8	False	True
char	1	8	-128	127
short	2	16	-32768	32767
int	4	32	-2147483648	2147483647
long	4	32	-2147483648	2147483647
long long	8	64	-9223372036854775808	9223372036854775807
unsigned char	1	8	0	255
unsigned short	2	16	0	65535
unsigned	4	32	0	4294967295
unsigned long	4	32	0	4294967295
unsigned long long	8	64	0	18446744073709551615
float	4	32	1.17549435e-38	3.40282347e+38
double	8	64	2.2250738585072014e-308	1.7976931348623157e+308
long double	12	96	3.36210314311209350626e-4932	1.18973149535723176502e+4932

Tipo int

Tamaño en bytes: 4 bytes (32 bits)

Dominio: valores entre -2,147,483,648 y 2,147,483,647

Operaciones:

- + Suma
- Resta
- * Producto
- / División entera
- % Resto de la división entera
- , + Signo negativo, positivo
- ++ Incrementación
- Decrementación

// File: Números enteros01.cpp

#include<iostream>

using namespace std;

int main()

{

int x,y;

x=2147483647; //Valor máximo de números enteros con 32 bits

x++;

cout<<"x="<<x<<endl;

x=5; y=10;

x=y++; cout<<"x="<<x; cout<<"y="<<y<<endl; //x toma valor de y (10), y aumenta 1 (11)

x=5; y=10;

x=++y; cout<<"x="<<x; cout<<"y="<<y; //x toma valor de y incrementada (11), y aumenta 1 (11)

return 0;

}

//File: Números enteros02.cpp

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int x,y;
```

```
    x=100;
```

```
    y=23;
```

```
    cout<<"Suma es "<<x+y<<endl;
```

```
    cout<<"Diferencia es "<<x-y<<endl;
```

```
    cout<<"Producto es "<<x*y<<endl;
```

```
    cout<<"División entera es "<<x/y<<endl;
```

```
    cout<<"Residuo es "<<x%y<<endl;
```

```
    cout<<"Incremento de x "<<++x<<endl;
```

```
    cout<<"Decremento de y "<<--y<<endl;
```

```
    return 0;
```

```
}
```

Tipo float

Tamaño en bytes: 4 bytes

Dominio: son todos los números reales que contienen una coma decimal comprendidos entre los valores absolutos: $3.4E-38$ y $3.4E+38$ (normalmente dependiendo de la representación interna)

Operaciones:

- + Suma
- Resta
- * Producto
- / División en coma flotante
- , + Signo negativo, positivo
- ++ Incrementación
- Decrementación

// File: Números reales float01.cpp

```
#include<iostream>

using namespace std;

int main()
{
    const float PI = 3.141592;
    float radio, longitud;
    cout<<"Radio de la circunferencia=";
    cin>>radio;
    longitud=2*PI*radio;
    cout<<"La longitud es:"<<longitud<<endl;

    return 0;

}
```

// File: Números reales float02.cpp

```
#include<iostream>

using namespace std;

int main()
{
    const float v_luz = 3E+5;
    float anios, kilometros;
    cout<<"Años luz=";
    cin>>anios;
    kilometros=v_luz*anios*365*24*3600;
    cout<<"La distancia recorrida es:"<<kilometros<<"Km"<<endl;

    return 0;

}
```

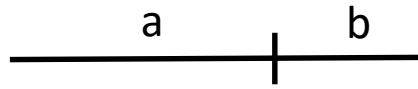

Tipo **double**

Tamaño en bytes: 8 bytes

Dominio: son todos los números reales que contienen una coma decimal comprendidos entre los valores absolutos: $1,7E-308$ y $1,7E+308$ (normalmente dependiendo de la representación interna)

Operaciones:

- + Suma
- Resta
- * Producto
- / División en coma flotante
- , + Signo negativo, positivo
- ++ Incrementación
- Decrementación



$$\frac{a+b}{a} = \frac{a}{b}$$

Condición de proporción áurea

$$\varphi = \frac{a}{b}$$

$$1 + \frac{1}{\varphi} = \varphi$$

$$\varphi^2 - \varphi - 1 = 0 \rightarrow \varphi = \frac{1 + \sqrt{5}}{2}$$

```
//Proporción áurea
#include <iostream>
#include <math.h>
#include <iomanip>

using namespace std;

int main()
{
    double f;
    f=(1+sqrt(5))/2;
    cout<<setprecision(50)<<f<<endl;
    cout<<setprecision(50)<<1/f<<endl;
    return 0;
}
```

Tipo **bool**

Tamaño en bytes: 1 byte

Dominio: dos únicos valores: { true (1), false(0) }

Operadores lógicos:

&&	Y lógica (and)
 	O lógica (or)
!	Negación lógica (not)

Operadores relacionales:

<	Menor que
<=	Menor o igual que
>	Mayor que
>=	Mayor o igual que
==	Igual a
!=	Distinto
cond ? valor1 : valor2	Asignación condicional (ternario)

```
// File: Tiposlogicos01.cpp
```

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int edad;
```

```
    bool mayor;
```

```
    string mensaje;
```

```
    const int mayoria_edad = 18;
```

```
    cout << "Dame tu edad=";
```

```
    cin >> edad;
```

```
    mayor = (edad>=mayoria_edad);
```

```
    mensaje = mayor ? "Mayor de edad" : "Menor de edad";
```

```
    cout << mensaje;
```

```
    return 0;
```

```
}
```

Tipo **char**

Tamaño en bytes: 1 byte

Dominio: dígitos, letras mayúsculas, letras minúsculas y signos de puntuación.

```
#include<iostream>
using namespace std;
int main ()

{
    //Creamos dos variables
    char a;
    int i;

    //Guardamos el código en la variable entera.
    cout <<"Dime el código ascii"<<endl;
    cin >>i;

    //Igualamos nuestras variables
    a=i;

    //Mostramos "char"
    cout <<"El caracter es: "<<a<<endl;

    return 0;
}
```

```
#include <iostream>
using namespace std;
int main ()

{
    //Creamos nuestras variables
    char a;
    int i;

    //Pedimos el carácter
    cout <<"Dime el carácter"<<endl;
    cin >>a;

    //Igualamos ahora el "int" al "char"
    i=a;

    //Mostramos "int"
    cout <<"Su codigo ascii es: "<<i<<endl;

    return 0;
}
```

¡OJO!

Operaciones, que matemáticamente son equivalentes, pueden ser computacionalmente diferentes.

```
//errores precisión.cpp
```

```
#include <iostream>
```

```
#include <stdlib.h>
```

```
using namespace std ;
```

```
int main()
```

```
{
```

```
    bool ok;
```

```
    ok=(4*(5/8))== (4*5/8);
```

```
    cout << boolalpha; //activa la salida true o false en lugar de 1 o 0
```

```
    cout <<"La expresión lógica (4*(5/8))== (4*5/8) es ";
```

```
    cout << ok << endl ;
```

```
    system("pause");
```

```
}
```

```
//Tiposdatos.cpp
#include<iostream>
#include <limits.h>
#include <float.h>

using namespace std;

int main()
{
    cout<<"Menor entero "<<INT_MIN<<" Mayor entero: "<<INT_MAX<<endl;
    cout<<"Menor float "<<FLT_MIN<<" Mayor flotante: "<<FLT_MAX<<endl;
    cout<<"Menor doble "<<DBL_MIN<<" Mayor doble: "<<DBL_MAX<<endl;
    cout<<"Menor char "<<CHAR_MIN<<" Mayor char: "<<CHAR_MAX<<endl;

    return 0;

}
```

Tipos Simples Enumerados

Las constantes enumeradas permiten crear listas de elementos afines.

```
enum <nombre_de_constante> { <lista_de_valores> };
```

```
enum Colores {Rojo, Verde, Azul, Amarillo};
```

```
enum Botones {Salir, Jugar};
```

```
...
```

```
Colores favorito = Rojo;
```

```
...
```

El compilador asigna un número a cada elemento del conjunto (comenzando con 0).


```
// File: Enumerados.cpp
```

```
#include<iostream>
```

```
#include<stdlib.h>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    enum Dias { Lunes, Martes, Miercoles, Jueves, Viernes };
```

```
    enum Meses {Enero, Febrero, Marzo, Abril, Mayo, Junio, Julio, Agosto};
```

```
    Dias libre = Viernes;
```

```
    Meses vacaciones= Agosto;
```

```
    cout <<"Dia libre= "<< libre<<endl;
```

```
    cout <<"Mes Vacaciones= "<< vacaciones << endl;
```

```
    system ("pause");
```

```
    return 0;
```

```
}
```