

Introducción a C++

Tipos de datos y Operadores

Tomás Peiretti

Tipos de datos fundamentales

Una variable es una porción de memoria que almacena un valor. Pero, ¿Qué **tipos de valores** se pueden almacenar?



Dentro de los tipos de datos fundamentales de c++ existen:

- Caracteres
- Números enteros
- Números de punto flotante
- Booleanos (valores de verdad)

Tipos de datos fundamentales

Grupo	Nombres de tipo	Precisión/Tamaño	Rango de valores
Caracteres	char	1 byte = 8 bits	0 a 255
	char16_t	2 bytes = 16 bits	0 a 65,535
	char32_t	4 bytes = 32 bits	0 a 4,294,697,295
	wchar_t	Puede almacenar cualquier caracter soportado	
Números enteros con signo	short	2 bytes = 16 bits	-32,768 a 32,767
	int	4 bytes = 32 bits	-2,147,483,684 a 2,147,483,647
	long	4 bytes = 32 bits	-2,147,483,684 a 2,147,483,647
	long long	8 bytes = 64 bits	-9,223,372,036,854,775,808 a 9,223,372,036,854,775,807
Números enteros sin signo	unsigned short	Mismo tamaño que los enteros con signo	0 a 65,535
	unsigned		0 a 4,294,967,295
	unsigned long		0 a 4,294,967,295
	unsigned long long		0 a 18,446,744,073,709,551,615
Números de punto flotante	float	4 bytes = 32 bits	Ver docs
	double	8 bytes = 64 bits	
	long double	8 bytes o mayor	
Boolean	bool	True o false	true, false
Void	void	No almacena valor	

<https://en.cppreference.com/w/cpp/language/types>

Operadores

Grupo	Operador	Ejemplo
Aritméticos	+	a + b
	-	a - 5
	*	x * y
	/	x / y
	%	a % 10
Asignación	=	a = 1
	+=	a += 1
	-=	a -= 1
	*=	a *= 2
	/=	a /= 10

Grupo	Operador	Ejemplo
Relacionales	==	x == b
	!=	x != 'a'
	<	x < z
	>	x > 100
	<=	x <= y
	>=	x >= 0
Lógicos	!	! (x >= 0)
	&&	x > 0 && x <= 100
		x < -10 x > 10
Otros	++	x++ o ++x
	--	x-- o --x
	sizeof	sizeof(x)

<https://cplusplus.com/doc/tutorial/operators>

Operador de "casteo" de tipo

Como sabemos, el compilador de C++ lleva a cabo conversiones implícitas. Por lo que, en algunos casos, necesitamos transformar un tipo de dato para poder manipularlo.

El operador de casteo permite convertir un valor que es de un determinado tipo de dato a otro tipo de dato. Hay dos maneras de utilizarlo:

```
1 int main() {
2     float x = 101.25;
3     // casteando x a int para poder usar %
4     cout << (int) x % 10 << endl;
5     // otra manera de castear x a int
6     cout << int(x) % 10 << endl;
7     // ejemplo: castear el dividendo para que el resultado de la
      division sea float y no int
8     cout << "1/2 = " << 1/2 << endl; // 1/2 = 0
9     cout << "1/2 = " << float(1)/2 << endl; // 1/2 = 0.5
10    return 0;
11 }
```

Ejercicios

- Beecrowd 1001
- Beecrowd 1002
- Beecrowd 1003
- Beecrowd 1004
- Beecrowd 1005
- Beecrowd 1006
- Beecrowd 1007
- Beecrowd 1008
- Beecrowd 1009
- Beecrowd 1010
- Beecrowd 1011
- Beecrowd 1012
- Beecrowd 1013
- Beecrowd 1014
- Beecrowd 1015
- Beecrowd 1016
- Beecrowd 1017
- Beecrowd 1018
- Beecrowd 1019
- Beecrowd 1020
- Beecrowd 1021