

Integer Data type

The integer data type is a set of whole numbers. Every integer value does not have the decimal value. We use the keyword "**int**" to represent integer data type in c. We use the keyword `int` to declare the variables and to specify the return type of a function. The integer data type is used with different type modifiers like `short`, `long`, `signed` and `unsigned`. The following table provides complete details about the integer data type.

Type	Size (Bytes)	Range	Specifier
int (signed short int)	2	-32768 to +32767	%d
short int (signed short int)	2	-32768 to +32767	%d
long int (signed long int)	4	-2,147,483,648 to +2,147,483,647	%d
unsigned int (unsigned short int)	2	0 to 65535	%u
unsigned long int	4	0 to 4,294,967,295	%u

Floating Point data types

Floating-point data types are a set of numbers with the decimal value. Every floating-point value must contain the decimal value. The floating-point data type has two variants...

- `float`
- `double`

We use the keyword "**float**" to represent floating-point data type and "**double**" to represent double data type in c. Both `float` and `double` are similar but they differ in the number of decimal places. The `float` value contains 6 decimal places whereas `double` value contains 14 decimal places. The following table provides complete details about floating-point data types.

Type	Size (Bytes)	Range	Specifier
float	4	1.2E - 38 to 3.4E + 38	%f
double	8	2.3E-308 to 1.7E+308	%ld
long double	10	3.4E-4932 to 1.1E+4932	%ld

Character data type

The character data type allows its variable to store only a single character. The size of the character is 1 byte. It is the most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.

Type	Size (Bytes)	Range	Specifier
char (signed char)	1	-128 to +127	%c
unsigned char	1	0 to 255	%c

Void data type

Void is an empty data type that has no value. We use void data type in functions when we don't want to return any value to the calling function.