In C#:

* Data Type

when declaring a variable in C#, you must specify the data type with the variable name, and as such, you will tell the compiler what type of data the variable will hold.

Same as C++.

Preliminary data types of the C# language

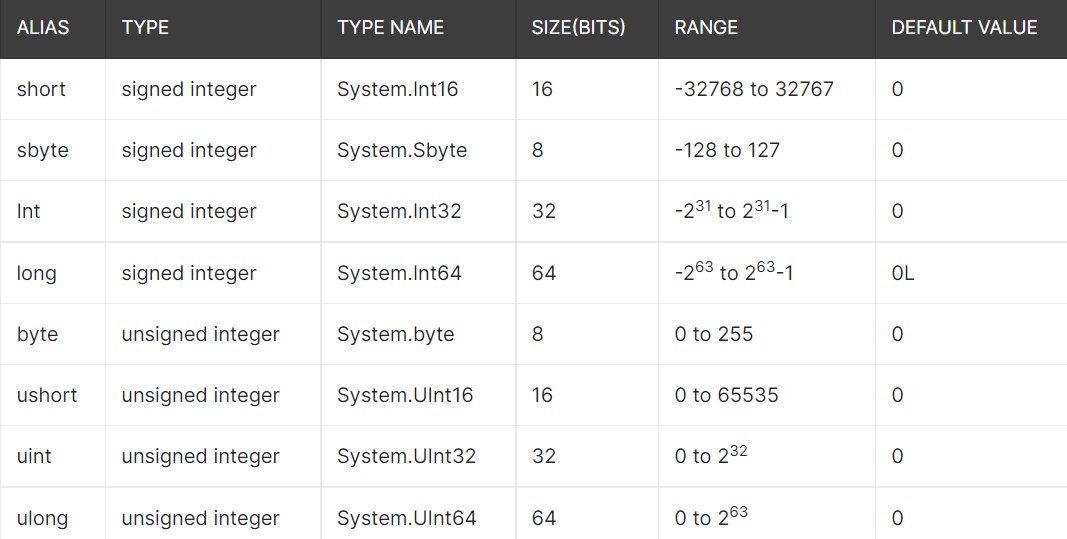
* int
* foat
* char

integral Datatypes

* C# allows you to define eight different types of integral variables.

They provide support for 8-bit, 16-bit, 32-bit, and 64-bit values with signed & unsigned modifiers.

These eight types are mentioned below with their keyword, range, class name, and default value.



**Floating point type**

Floating-point data types in C# are of two types. These are:

1. 32-bit single (7-digit) precision floating point type declared using the keyword float. For initializing any variable with float, you have to mention a 'f' or 'F' after the value. For example: float g = 62.4f; If you do not use the suffix, then the compiler treats the value as double.
2. 64-bit (14-15 digit) precision floating point type declared using the keyword double. For initializing any variable with double, you have to mention a 'd' or 'D' after the value. For example, double ks =8.403122d;

**Decimal Types**

Decimal Type is another type of variable used for extensive calculations; 128-bit is used to calculate huge economic data. It uses' or 'M' as the suffix; otherwise, the value will be treated as double.

**Character Types**

This is used for representing a 16-bit Unicode character used for storing a single character.

**Reference Type**

A reference type is another form of the variable that does not hold the actual value or data; instead, it references any memory location assigned with a variable. This refers to a memory location

The built-in reference types provided by C# are:

1. string
2. object
3. dynamic

All these will be covered in the later chapter of this tutorial.

**Pointer Type**

These types of variables are used for storing the address of any memory location, which is of another type. Pointers are considered a separate data type because they do not hold the actual value or data; instead, they are meant to store the actual memory location. The concept of pointers came in C# from C and C++.

The syntax of declaring pointers in C# is:

Syntax:

type\* identifier;

Example:

char\* str\_name;

int\* user\_id;

* Function
  + Call by Value
  + Call by Reference
  + Out Parameter
* Array
  + Array of functions
  + Jagged array
  + Params
  + Array class
* OOPs
  + Constructor
  + Destructor
  + Static clas
  + Structs
  + Enums
* Exception handling
* Delegates
* Collections
  + List
  + Hashset
  + Sorted set
  + Stack
  + Queue
  + Linked list
  + Dictionary
  + Sorted Dictionary

Sorted List