C# Predefined Types

Type	Description	Example
`object`	The ultimate base type of all other types	`object o = null;`
`string`	String type; a string is a sequence of Unicode characters	`string s = "hello";`
`sbyte`	8-bit signed integral type	`sbyte val = 12;`
`short`	16-bit signed integral type	`short val = 12;`
`int`	32-bit signed integral type	`int val = 12;`
`long`	64-bit signed integral type	`long val1 = 12;` `long val2 = 34L;`
`byte`	8-bit unsigned integral type	`byte val1 = 12;`
`ushort`	16-bit unsigned integral type	`ushort val1 = 12;`
`uint`	32-bit unsigned integral type	`uint val1 = 12;` `uint val2 = 34U;`

`ulong`	64-bit unsigned integral type	`ulong val1 = 12;`
		`ulong val2 = 34U;`
		`ulong val3 = 56L;`
		`ulong val4 = 78UL;`

`float`	Single-precision floating point type	`float val = 1.23F;`
`double`	Double-precision floating point type	`double val1 = 1.23;` `double val2 = 4.56D;`
`bool`	Boolean type; a bool value is either true or false	`bool val1 = true;` `bool val2 = false;`
`char`	Character type; a char value is a Unicode character	`char val = 'h';`
`decimal`	Precise decimal type with 28 significant digits	`decimal val = 1.23M;`

Each of the predefined types is shorthand for a system-provided type. For example, the keyword for `int` refers to the struct `System.Int32`. As a matter of style, the use of the keyword is favored over the use of the complete system type name.

The types `string` and `object` are reference types; the others are value types.

SOURCE: ECMA Standard ECMA-334, December 2001