

# Datatypes in C#

<https://csci-1301.github.io/about#authors>

March 15, 2022 (09:28:05 AM)

## Contents

<b>1 Value Types</b>	<b>1</b>
1.1 Numeric	1
1.1.1 Signed Integer	1
1.1.2 Unsigned Integer	1
1.1.3 Floating-point Numbers	2
1.2 Logical	2
1.3 Character	2
<b>2 Literals</b>	<b>2</b>
<b>3 Compatibility</b>	<b>2</b>
<b>4 Result Type of Operations</b>	<b>2</b>
<b>References</b>	<b>3</b>

## 1 Value Types

### 1.1 Numeric

#### 1.1.1 Signed Integer

Type	Range	Size
<code>sbyte</code>	-128 to 127	Signed 8-bit integer
<code>short</code>	-32,768 to 32,767	Signed 16-bit integer
<code>int</code>	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
<code>long</code>	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer

#### 1.1.2 Unsigned Integer

Type	Range	Size
<code>byte</code>	0 to 255	Unsigned 8-bit integer
<code>ushort</code>	0 to 65,535	Unsigned 16-bit integer
<code>uint</code>	0 to 4,294,967,295	Unsigned 32-bit integer

Type	Range	Size
<code>ulong</code>	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer

### 1.1.3 Floating-point Numbers

Type	Approximate Range	Precision
<code>float</code>	$\pm 1.5e-45$ to $\pm 3.4e38$	7 digits
<code>double</code>	$\pm 5.0e-324$ to $\pm 1.7e308$	15–16 digits
<code>decimal</code>	$(-7.9 \times 10^{28}$ to $7.9 \times 10^{28}) / (100$ to $10^{28})$	28–29 significant digits

## 1.2 Logical

Type	Possible Values	Size
<code>bool</code>	<code>true</code> , <code>false</code>	8-bit

## 1.3 Character

Type	Range	Size
<code>char</code>	U+0000 to U+ffff	Unicode 16-bit character

## 2 Literals

Name	Corresponding datatype	Examples
Integer Literal	<code>int</code>	<code>40</code> , <code>-39</code> , <code>291838</code> , <code>0</code> , ...
Float Literal	<code>float</code>	<code>3.5F</code> , <code>-43.5f</code> , <code>309430.70006F</code> , ...
Double Literal	<code>double</code>	<code>28.98</code> , <code>239.0</code> , <code>-391.089</code> , <code>0.0</code> , ...
Decimal Literal	<code>decimal</code>	<code>8.95m</code> , <code>3283.9M</code> , <code>-30m</code> , ...
Boolean Literal	<code>bool</code>	<code>true</code> , <code>false</code>
Character Literal	<code>char</code>	<code>'Y'</code> , <code>'a'</code> , <code>'0'</code> , <code>'\n'</code> , <code>'\x0058'</code> , <code>'\u0058'</code> , ...

## 3 Compatibility

	Integer Literal	Float Literal	Double Literal	Decimal Literal
<code>int</code>	✓	✗	✗	✗
<code>float</code>	✓	✓	✗	✗
<code>double</code>	✓	✓	✓	✗
<code>decimal</code>	✓	✗	✗	✓

## 4 Result Type of Operations

	<code>int</code>	<code>float</code>	<code>double</code>	<code>decimal</code>
<code>int</code>	<code>int</code>	<code>float</code>	<code>double</code>	<code>decimal</code>
<code>float</code>	<code>float</code>	<code>float</code>	<code>double</code>	illegal
<code>double</code>	<code>double</code>	<code>double</code>	<code>double</code>	illegal
<code>decimal</code>	<code>decimal</code>	illegal	illegal	<code>decimal</code>

## References

- <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/types-and-variables>
- <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/integral-types-table>
- <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/floating-point-types-table>
- <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/value-types-table>
- <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/implicit-numeric-conversions-table>
- <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/keywords/explicit-numeric-conversions-table>