1-INTRODUCTION TO C#

Ahmed Hafez

What is C#?

C# is a modern object-oriented programming language developed in 2000 by Anders Hejlsberg at Microsoft as a rival to Java (which it is quite similar to). It was created because Sun, (later bought by Oracle) did not want Microsoft to make changes to Java, so Microsoft chose to create their own language instead. C# has grown quickly since it was first created, with extensive support from Microsoft helping it to gain a large following; it is now one of the most popular programming languages in the world.

C# used for:

- Desktop Apps (GUI, Console)
- Mobile Apps (Android, IOS)
- Web Apps
- Cloud (Azure Cloud)
- Gaming (Wonderful games using Unity 3D and WPF)
- IoT

Getting Started-revision

To print some thing, use :

- **Console.WriteLine(“Hello World!”);**
- **Console.Write(“Hello World!”);**

To read some thing from the user, use :

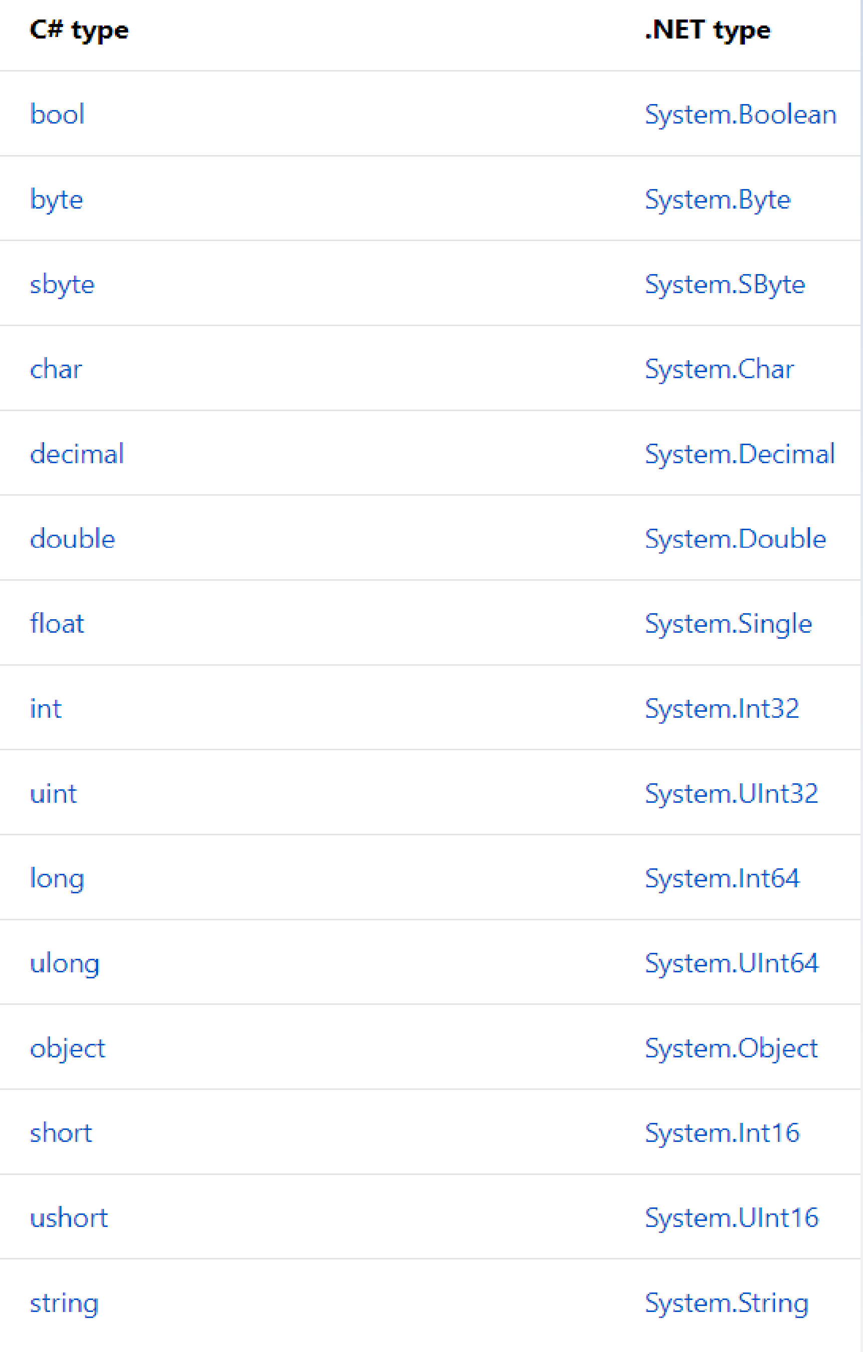
- **Console.ReadLine(); // reads input stream**
- **Console.Read();   // reads the stream as char by char**
- **Console.ReadKey(); // reads the pressed key**

C# Data Types

C# primitive data types are implemented in the .Net Framework, All of these types is converted directly to zeros and ones at runtime.

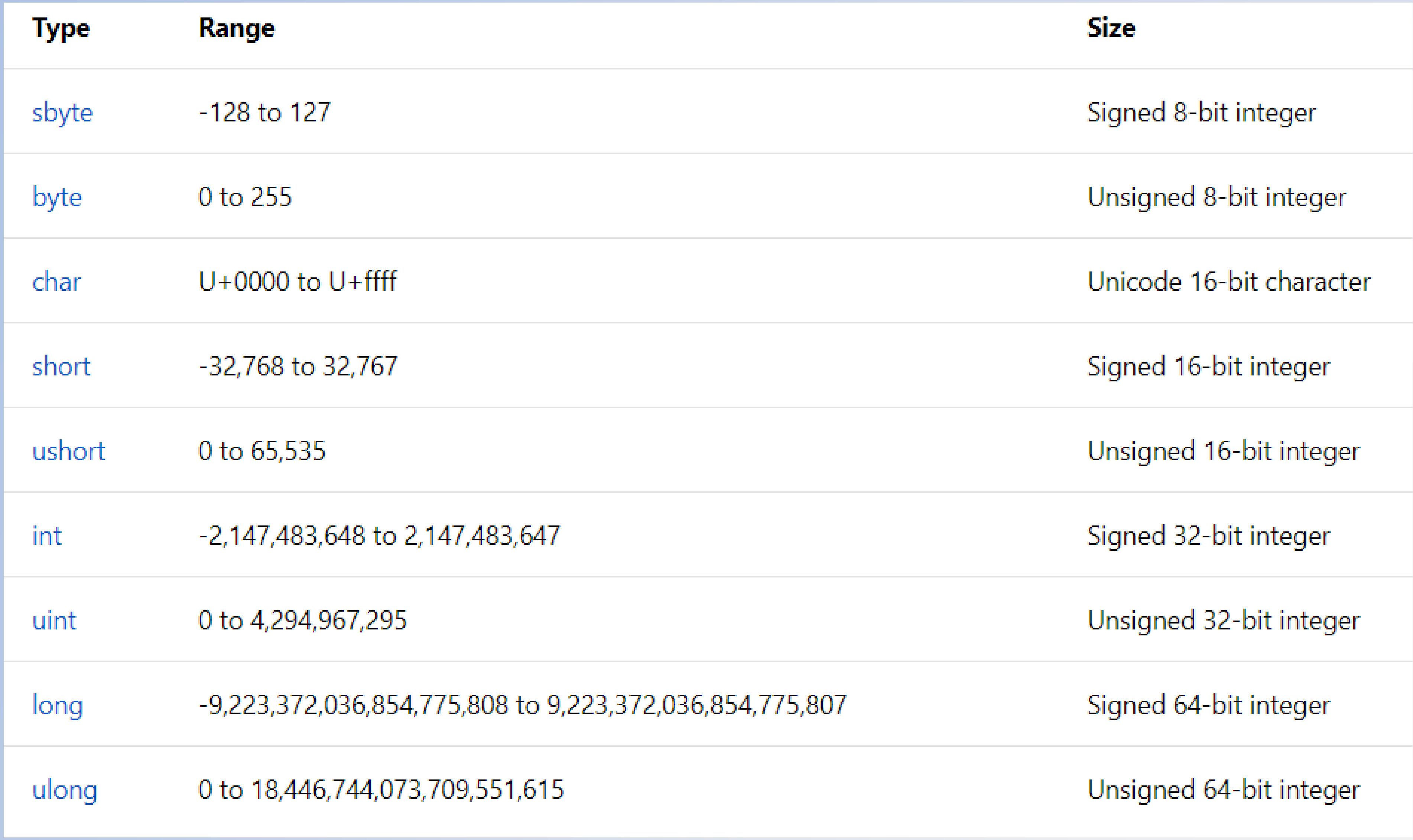
But other Built-in Types are translated to zeros and ones by CSC.exe

System.Object and System.String are not Primitive (According to MSDN)



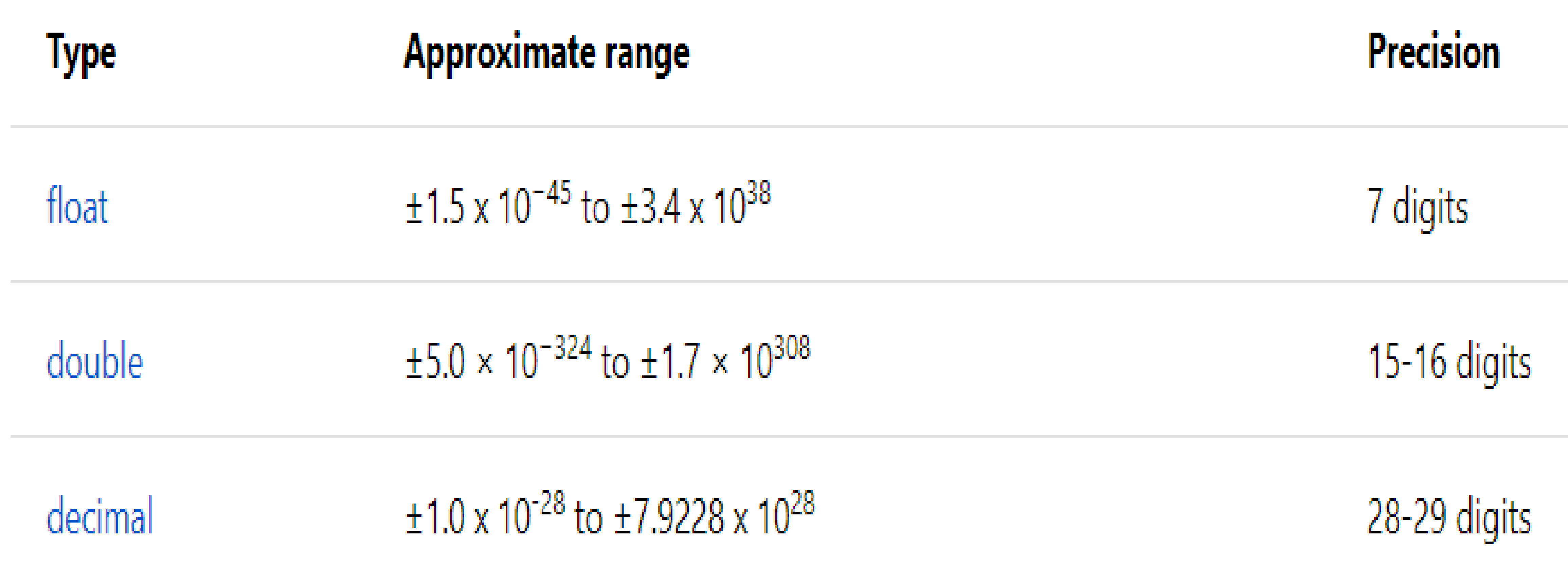
C# type	.NET type
bool	System.Boolean
byte	System.Byte
sbyte	System.SByte
char	System.Char
decimal	System.Decimal
double	System.Double
float	System.Single
int	System.Int32
uint	System.UInt32
long	System.Int64
ulong	System.UInt64
object	System.Object
short	System.Int16
ushort	System.UInt16
string	System.String

C# Data Types - Integral types table



Type	Range	Size
sbyte	-128 to 127	Signed 8-bit integer
byte	0 to 255	Unsigned 8-bit integer
char	U+0000 to U+ffff	Unicode 16-bit character
short	-32,768 to 32,767	Signed 16-bit integer
ushort	0 to 65,535	Unsigned 16-bit integer
int	-2,147,483,648 to 2,147,483,647	Signed 32-bit integer
uint	0 to 4,294,967,295	Unsigned 32-bit integer
long	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	Signed 64-bit integer
ulong	0 to 18,446,744,073,709,551,615	Unsigned 64-bit integer

C# Data Types - Floating-point types table

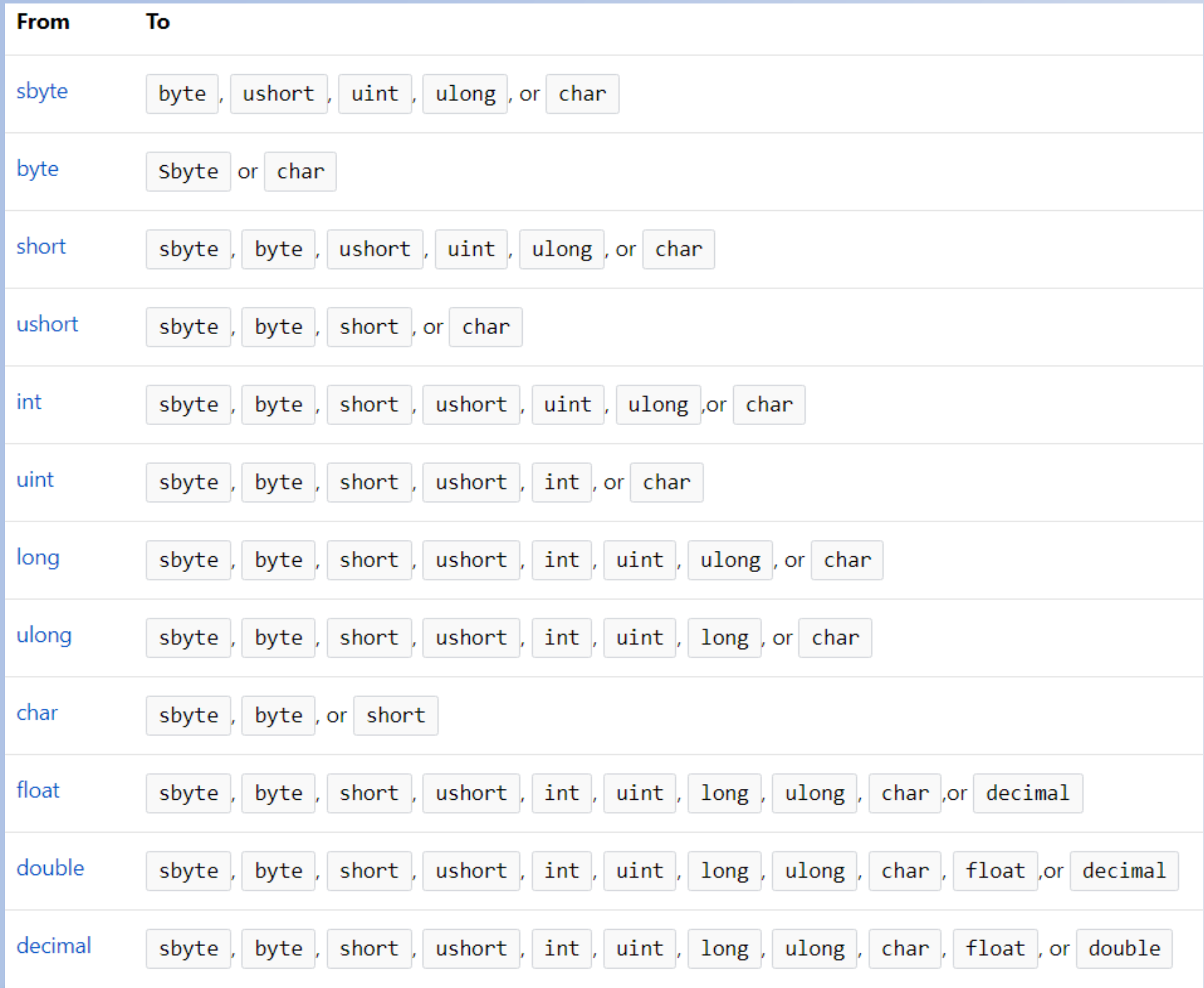


Type	Approximate range	Precision
float	±1.5 x 10^{-45} to ±3.4 x 10^{38}	7 digits
double	±5.0 × 10^{-324} to ±1.7 × 10^{308}	15-16 digits
decimal	±1.0 x 10^{-28} to ±7.9228 x 10^{28}	28-29 digits

Introduction to C#

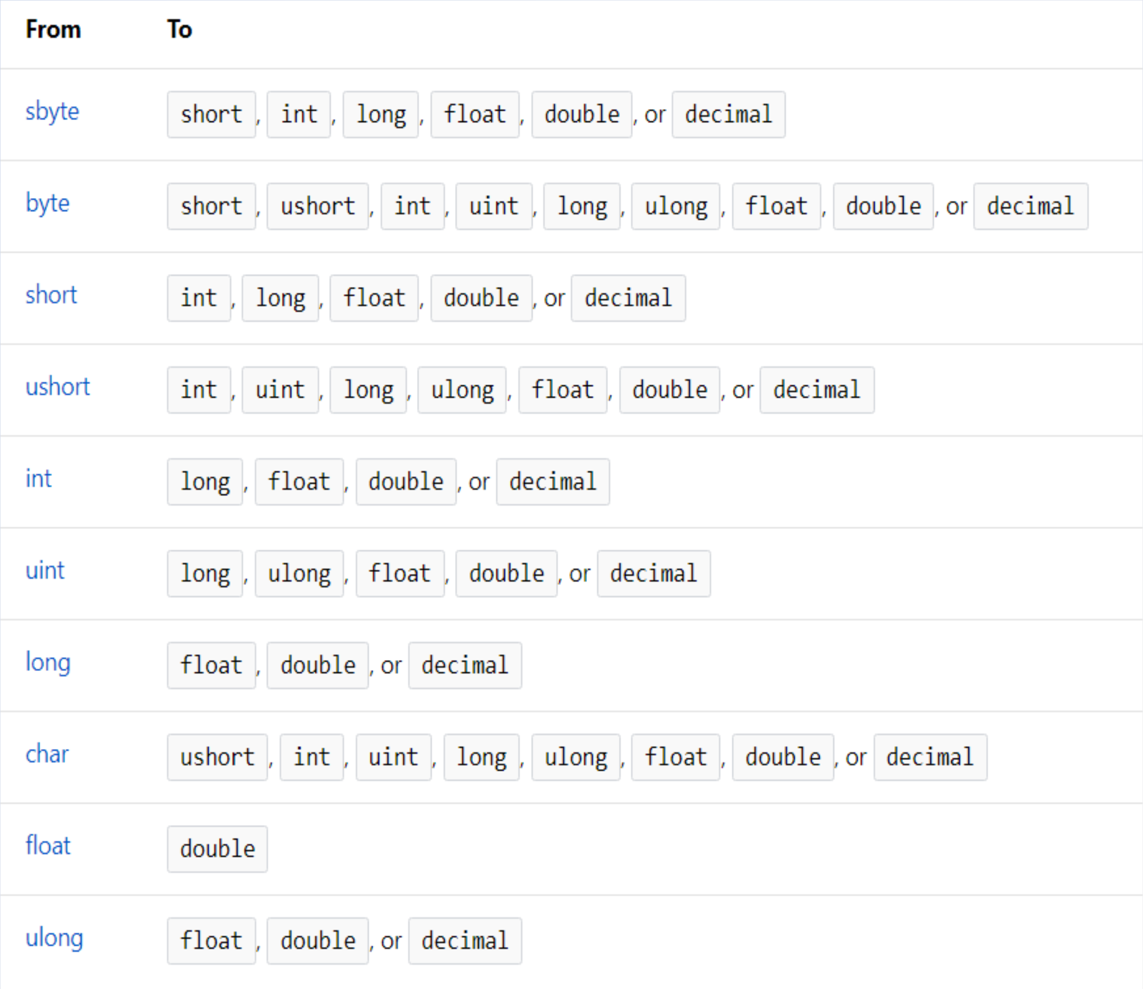
Numeric Conversions

Explicit Conversions



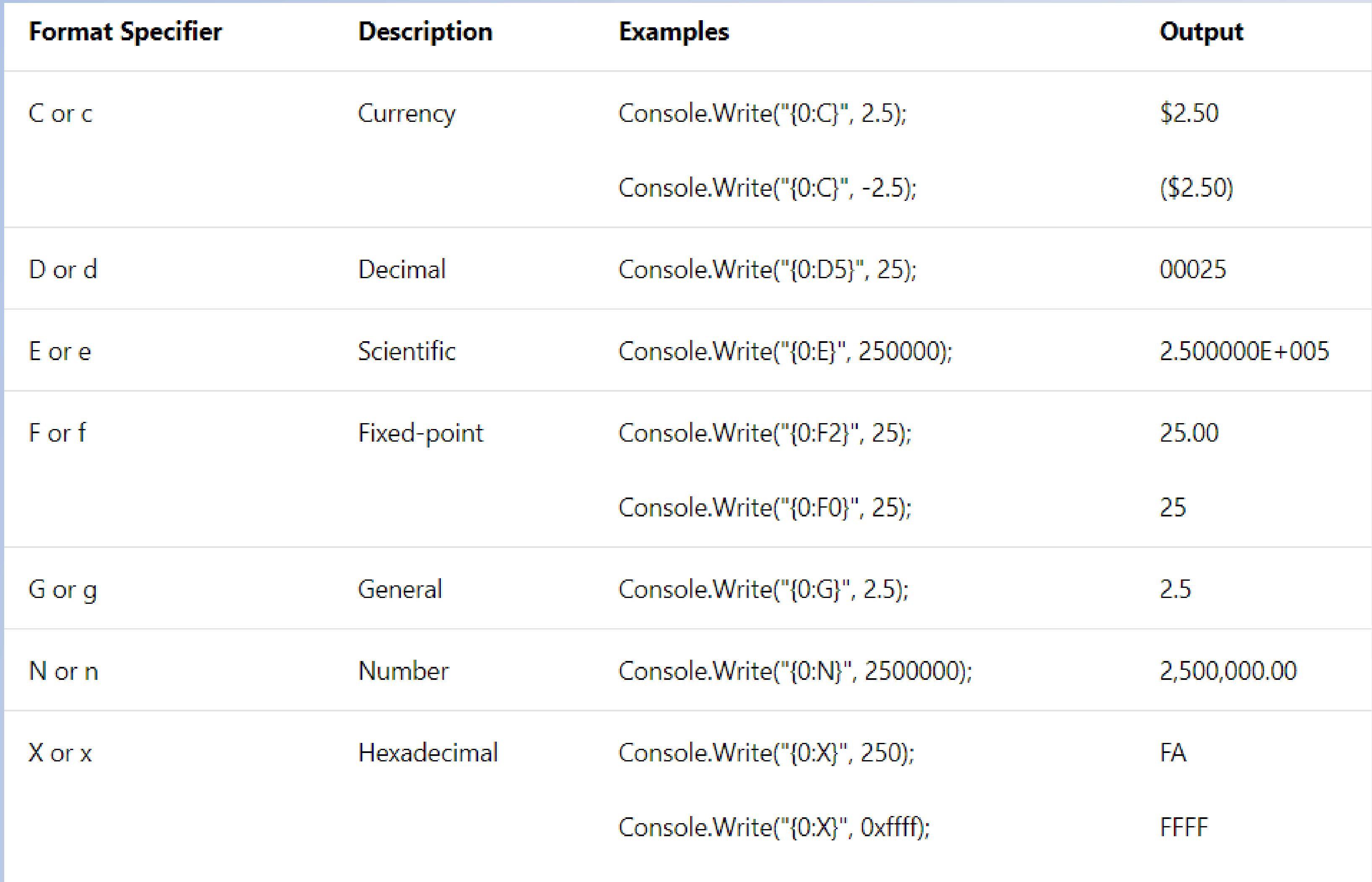
From	To
sbyte	byte, ushort, uint, ulong, or char
byte	Sbyte or char
short	sbyte, byte, ushort, uint, ulong, or char
ushort	sbyte, byte, short, or char
int	sbyte, byte, short, ushort, uint, ulong,or char
uint	sbyte, byte, short, ushort, int, or char
long	sbyte, byte, short, ushort, int, uint, ulong, or char
ulong	sbyte, byte, short, ushort, int, uint, long, or char
char	sbyte, byte, or short
float	sbyte, byte, short, ushort, int, uint, long, ulong, char,or decimal
double	sbyte, byte, short, ushort, int, uint, long, ulong, char, float,or decimal
decimal	sbyte, byte, short, ushort, int, uint, long, ulong, char, float, or double

Implicit Conversions



From	To
sbyte	short, int, long, float, double, or decimal
byte	short, ushort, int, uint, long, ulong, float, double, or decimal
short	int, long, float, double, or decimal
ushort	int, uint, long, ulong, float, double, or decimal
int	long, float, double, or decimal
uint	long, ulong, float, double, or decimal
long	float, double, or decimal
char	ushort, int, uint, long, ulong, float, double, or decimal
float	double
ulong	float, double, or decimal

Formatting Numeric Results



Format Specifier	Description	Examples	Output

C or c	Currency	Console.Write("{0:C}", 2.5);
		Console.Write("{0:C}", -2.5);
			$2.50
			($2.50)

D or d	Decimal	Console.Write("{0:D5}", 25);	00025
E or e	Scientific	Console.Write("{0:E}", 250000);	2.500000E+005

F or f	Fixed-point	Console.Write("{0:F2}", 25);
		Console.Write("{0:F0}", 25);
			25.00
			25

G or g	General	Console.Write("{0:G}", 2.5);	2.5
N or n	Number	Console.Write("{0:N}", 2500000);	2,500,000.00

X or x	Hexadecimal	Console.Write("{0:X}", 250);
		Console.Write("{0:X}", 0xffff);
			FA
			FFFF

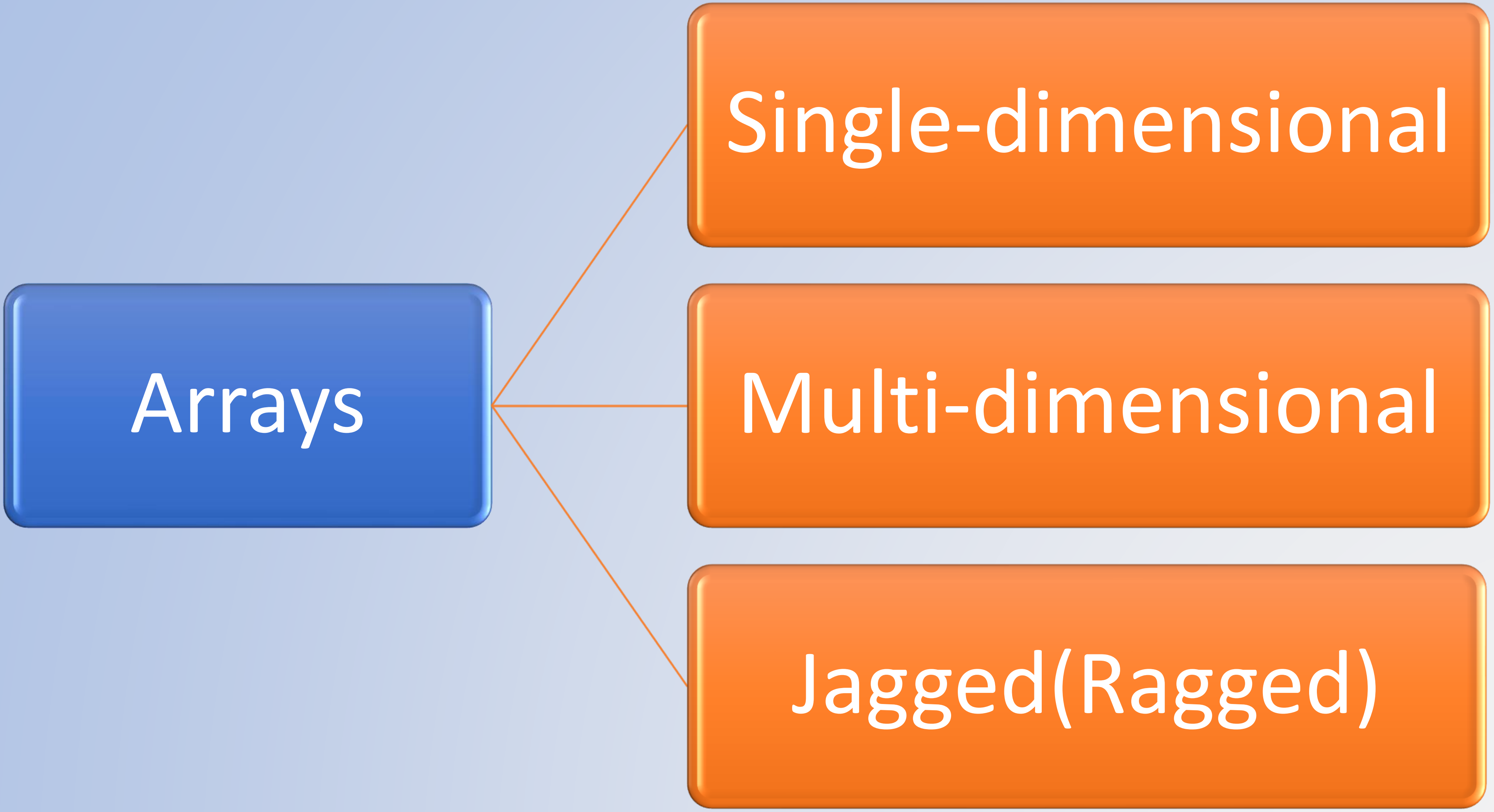
C# Statements

Selection	if-else switch

Iteration	do-while
	for
	foreach-in
	while

Jump	break
	continue
	goto
	return

Arrays and Array Type



Arrays

Single-dimensional

Multi-dimensional

Jagged(Ragged)

THANK YOU