1. The variable is a name given to a data value.
2. A variable holds the value of specific data type e.g string, int, float etc.
3. A variable can be declared and initialized in separate statements and also in the single statement.
4. The value of a variable can be changed at any time throught out the program as long as it is accessible.
5. Multiple variables can be defined seperated by comma (,) in a single or multiple line till semicolon(;).
6. A value must be assigned to a variable before using it otherwise it will give compile time error.

Value typey

* Signed Integral
  + sbyte: 8 bits, range from -128 to 127
  + short: 16 bits, range from -32,768 to 32,767
  + int : 32 bits, range from -2,147,483,648 to 2,147,483,647
  + long : 64 bits, range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
* Unsigned integral
  + byte : 8 bits, range from 0 to 255
  + ushort : 16 bits, range from 0 to 65,535
  + uint : 32 bits, range from 0 to 4,294,967,295
  + ulong : 64 bits, range from 0 to 18,446,744,073,709,551,615
* Floating point
  + float : 32 bits, range from 1.5 × 10-45 to 3.4 × 1038, 7-digit precision
  + double : 64 bits, range from 5.0 × 10-324 to 1.7 × 10308, 15-digit precision
* Decimal
  + decimal : 128 bits, range is at least -7.9 × 10-28 to 7.9 × 1028, with at least 28-digit precision
  + Boolean: bool
* Enum types
  + User-defined types of the form enum E {...}
* Struct types
  + User-defined types of the form struct S {...}
* Nullable value types
  + Extensions of all other value types with a null value

Reference types

* Class types
  + Ultimate base class of all other types: object
  + Unicode strings: string
  + User-defined types of the form class C {...}
* Interface types
  + User-defined types of the form interface I {...}
* Array types
  + Single- and multi-dimensional, for example, int[] and int[,]
* Delegate types
  + User-defined types of the form delegate int D(...)

**\* Escape Sequence \***

1- \n new line

2- \\ printed as \

3- \t new tab

4- \" printed "

Format String : C display number as currency using local currency symbol

D Decimal

E Exponential

F Float

G Display the number as either fixed point or integer

N Display number with embedded commas

X Display number by using hexadecimal

Ex: Console.WriteLine("Currency formatting {0:C} {1:C4}",88.8, 888.8);

Console.WriteLine("Decimal formatting {0:D5}",88);

Console.WriteLine("Exponential formatting {0:E5}",888.8);

Console.WriteLine("Float formatting {0:F3}",888.8888);

Console.WriteLine("General formatting {0:G6}",88.888888);

Console.WriteLine("Embeded formatting {0:N7}",88888.8);

Console.WriteLine("HexaDecimal formatting {0:X4}",88);

This prints :

Currency formatting $88.80 $888.8000

Decimal floating 00088

Exponential formatting 8.88800E+002

Float formatting 888.889

General formatting 88.8889

Embeded formatting 88,888.8000000

HexaDecimal formatting 0058