

Data Types, Variables & Literals

C#

C# Basics



Data Types

Types of data that a valid C# variable can hold



Variables

Dolor Sit Amet
Consectetuer Elit
Nunc Viverra



Literals

Dolor Sit Amet
Consectetuer Elit
Nunc Viverra

Data Types

- ❖ Specify the type of data that a valid C# variable can hold.
- ❖ C# is a strongly typed programming language
- ❖ Each type of data is predefined as part of the programming language. E.g.
 - ❖ Integer
 - ❖ Character
 - ❖ float ...
- ❖ All constants or variables defined for a given program must be described with one of the data types.

Three categories

- ❖ Value Data Types
 - ❖ directly store the variable value in memory
 - ❖ Includes both signed and unsigned
 - ❖ System.ValueType is the implied parent/base class for all value types
- ❖ Reference Data Types
 - ❖ contain a memory address of variable value
 - ❖ System.Object is the implied parent/base class for all reference types
- ❖ Pointer Data Type
 - ❖ contain a memory address of the variable value
 - ❖ Associated Operators ampersand (&) and asterisk (*)
 - ❖ In C# can only be used with the unsafe keyword

1. Value Data Types

❖ Signed & Unsigned Integral Types

Alias	Type Name	Type	Size(bits)	Range	Default Value
sbyte	System.Sbyte	signed integer	8	-128 to 127	0
short	System.Int16	signed integer	16	-32768 to 32767	0
Int	System.Int32	signed integer	32	-2^{31} to $2^{31}-1$	0
byte	System.byte	unsigned integer	8	0 to 255	0
uint	System.UInt32	unsigned integer	32	0 to $2^{32}-1$	0
long	System.Int64	signed integer	64	-2^{63} to $2^{63}-1$	0L
ushort	System.UInt16	unsigned integer	16	0 to 65535	0
ulong	System.UInt64	unsigned integer	64	0 to $2^{63}-1$	0

1. Value Data Types

❖ Floating Point Types

Alias	Type name	Size(bits)	Range (aprox)	Default Value
float	System.Single	32	$\pm 1.5 \times 10^{-45}$ to $\pm 3.4 \times 10^{38}$	0.0F
double	System.Double	64	$\pm 5.0 \times 10^{-324}$ to $\pm 1.7 \times 10^{308}$	0.0D

❖ Decimal Type

Alias	Type name	Size(bits)	Range (aprox)	Default value
decimal	System.Decimal	128	$\pm 1.0 \times 10^{-28}$ to $\pm 7.9228 \times 10^{28}$	0.0M

1. Value Data Types

- ❖ Character Types

Alias	Type name	Size In(Bits)	Range	Default value
char	System.Char	16	U +0000 to U +ffff	'\0'

- ❖ Boolean Types

Alias	Type name	Values
bool	System.Boolean	True / False

2. Reference Data Types

- ❖ Contain a memory address of variable value
- ❖ Built-in reference types
 - ❖ String
 - ❖ represents a sequence of Unicode characters and its type name is System.String. So, string and String are equivalent.
 - ❖ Object
 - ❖ All types, predefined and user-defined, reference types and value types, inherit directly or indirectly from Object
 - ❖ is the base class for all the data types in C#

More on Object class

- ❖ Since System.Object is the base class for all types (directly or indirectly)
 - ❖ All variables can be converted to System.Object or vice versa
 - ❖ Value Types inherit from System.ValueType which in turn inherits from System.Object
- ❖ Boxing
 - ❖ When variable of a value types is converted to System.Object
- ❖ Unboxing
 - ❖ When variable of type object is converted to a value type
- ❖ class, interface, delegate, array are all reference data types

Stack & Heap

Stack

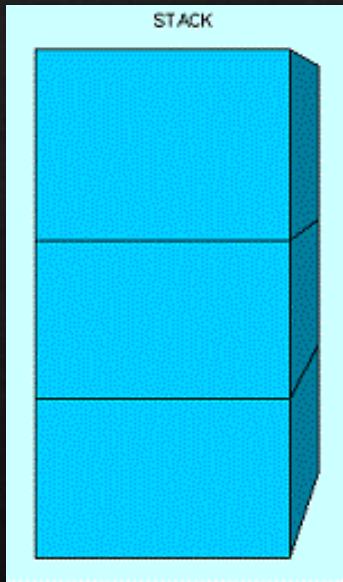
- ❖ It is an array of memory.
- ❖ It is a LIFO (Last In First Out) data structure.
- ❖ Data can be added to and deleted only from the top of it.
- ❖ Used to store entire value types as well as reference variables

Heap

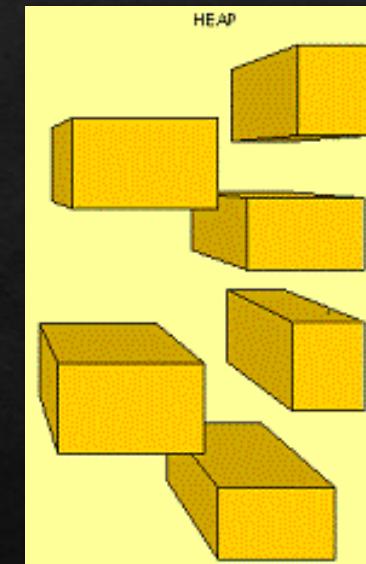
- ❖ It is an area of memory where chunks are allocated to store certain kinds of data objects.
- ❖ In it data can be stored and removed in any order.

Stack & Heap : How memory is managed

Stack

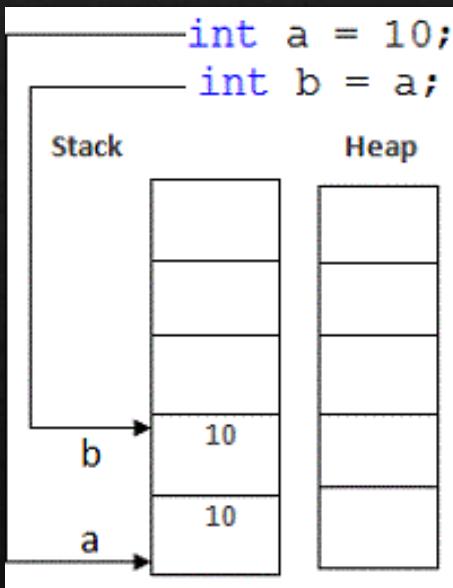


Heap

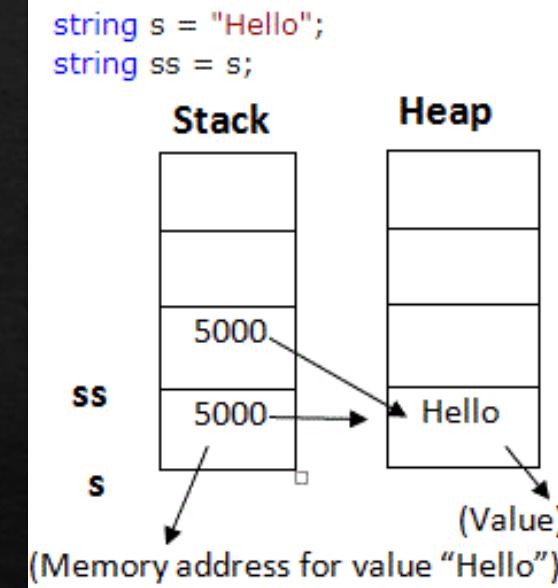


Stack & Heap: Practical Scenario

Stack



Heap



Stack & Heap: What is stored?

Stack

- ❖ "Things" declared with the following list of type declarations are Value Types
- ❖ bool, byte, char, decimal, double, enum, float, int, long, sbyte, short, struct, uint, ulong, ushort

Heap

- ❖ "Things" declared with following list of type declarations are Reference Types
- ❖ class, interface, delegate, object, string

Stack & Heap: Some more differences

Stack

- ❖ Memory allocation is Static
- ❖ It is stored Directly
- ❖ Local variables get wiped off once they lose the scope

Heap

- ❖ Memory allocation is Dynamic
- ❖ It is stored indirectly
- ❖ Only cleaned by Garbage Collector

3. Pointer Data Types

- ❖ Pointer Data Types will contain a memory address of the variable value
- ❖ To get the pointer details we have two operators
 - ❖ ampersand (&): It is Known as Address Operator. It is used to determine the address of a variable.
 - ❖ asterisk (*): It also known as Indirection Operator. It is used to access the value of an address.
- ❖ Syntax :
 - ❖ type* identifier;
- ❖ Example :
 - ❖ int* addr1;