Data Structures and Algorithms

In programming languages like C# and JavaScript, variables can be categorized as primitive or non-primitive (also known as reference types). Here’s an explanation of each in both languages:

**C#**

**Primitive Types**

Primitive types in C# are basic data types provided by the language, and they hold simple values. These include:

1. **Numeric Types:**
   * byte, sbyte
   * short, ushort
   * int, uint
   * long, ulong
   * float
   * double
   * decimal
2. **Other Types:**
   * char
   * bool
3. **Special Types:**
   * string (though it's technically a reference type, it behaves like a primitive type in many contexts)
   * object (the base type for all other types; however, it's generally considered non-primitive)

**Non-Primitive (Reference) Types**

Non-primitive types in C# are more complex and include:

* **Classes** (e.g., class MyClass { })
* **Interfaces** (e.g., interface IMyInterface { })
* **Arrays** (e.g., int[] myArray)
* **Delegates** (e.g., delegate void MyDelegate())
* **Strings** (internally handled as reference types)
* **User-Defined Types** (e.g., custom classes and structs)

**JavaScript**

**Primitive Types**

Primitive types in JavaScript are immutable and include:

1. **Numbers:**
   * number (includes both integer and floating-point numbers)
2. **String:**
   * string
3. **Boolean:**
   * boolean
4. **Special Types:**
   * undefined
   * null
   * Symbol (a unique and immutable data type introduced in ECMAScript 6)
   * BigInt (for arbitrarily large integers, introduced in ECMAScript 2020)

**Non-Primitive (Reference) Types**

Non-primitive types in JavaScript are objects and can store collections of data and more complex entities:

* **Objects:**
  + Object
  + Array
  + Function
  + Date
  + RegExp
* **Special Objects:**
  + Map
  + Set
  + WeakMap
  + WeakSet

## **What is the difference between System.String and string in C#?**

Basically, there is no difference between string and String in C#. "string" is just an alias of System.String and both are compiled in the same manner. String stands for System.String, and it is a .NET Framework type. "string" is an alias in the C# language for System.String. Both of them are compiled to System.String in IL (Intermediate Language), so there is no difference.

string s="Hello ";

String st="World";

C#

Copy

So, the question is if both strings do the same job, then why do we need to invent another one for the confused programmer? The answer is very simple. "string" is just a keyword which provides limited functionality and is mainly used in creating a variable. Whereas, System.String is a class which gives you a rich set of functions and properties to manipulate the string.

## **Known differences between string and System.String in  C#?**

1. string is a keyword and widely used for declaring a variable. Whereas System.String is a class and used for accessing string static methods like String.Format(), String.Compare() etc.
2. You can use string directly in your program and you don’t need to add Using System namespace at the beginning of your program whereas to using System.String you need to import Using System namespace.
3. As a long map to System.Int64, int maps to System.Int32 and short map to System.Int16; string map to System.String.

**Code Example: String vs string in C#**

using System;

namespace ConsoleApplication {

public class Program {

public static void Main(string[] args) {

string s = "Hello ";

string t = "World";

string st = String.Concat(s, t);

Console.WriteLine(st);

}

}