Comparison on JS vs C#

	JavaScript	C#
Designed by	Brendan Eich	Microsoft
Type	Object-oriented language	Type-safe object-oriented language
Static typing	Dynamic	Static
Platform	Cross-platform compatible	Limitation on platform
Lambda Support	No Lambda support	Supports Lambda
Generic Support	No	Great generic support

	JavaScript	C#
LINQ (Language integrated query)	Doesn't support	Supports LINQ
Garbage collection	Don't have garbage collection	Automatic garbage collection
Platform	Less cross-platform support	Amazing cross-platform support
Consistency	Horrible	Very Consistent
Flexibility	Less flexible	Very flexible
Frameworks	Support various frameworks	Supports.Net framework
Frontend/Backend support	Supports both	Supports frontend
Versatile	Very versatile	Less versatile

	JavaScript	C#
Server-side scripting	Supports server-side scripting	No support for server-side scripting
Syntax	Complex syntax	Concise syntax
IDE	Notepad, Notepad ++, Sublime, Visual Studio Code	Visual Studio
Operator overloading	It doesn’t support operator overloading	Supports Operator overloading
Readability	Less Readable	Great readability
OOPS concept	Supports the OOPS concept	OOPS simplified with great syntax
Complexity	Less complex in term of understanding	More complex

	JavaScript	C#
Performance	Less	High
Resource	Easy to find JavaScript developer	Hard to find a resource
Closure script	Supports closure script	It doesn’t support closure script
Over threading	C# gives the programming explicit control over threading.	JavaScript hides much of this with its call-and-response function structure.
Integer data type	Doesn’t support	Supports integer data type
Array data type	Doesn’t support	Supports Array data type
Verbosity/Simplicity	C# is Static Typed. It adds type adding capabilities	JavaScript is Dynamic Typed. It does not require variable

	to a variable. This highly reduces coding errors much before it is executed and we can quickly debug applications written in C#.	type assignments, i.e., a variable can hold String and again the same variable can hold Integer without any modification to the code.
Error Detection	With the static typing approach, we have the ability to detect errors while coding. Unlike JavaScript, where the code needs to be compiled.	We cannot detect errors in JavaScript while coding. Errors are detected only while executing JavaScript code.
Compilation	Codes written in C# needs to be compiled. The compiled code results in executable files that can be run. This adds an additional	We need not compile JavaScript codes. Our browsers run them.

	step to executing C# codes.	
Code Maintenance	In C# We have the ability to refactor codes and make small changes easily. Hence, it becomes easier to maintain large applications	Writing code in JavaScript for large applications is cumbersome to maintain.
Productivity	Using Microsoft Visual Studio, a developer gets the integrated development environment (IDE) support for C#. With this they get the added advantages of code auto-completion, error detection while coding	Unlike C#, developers have to write JavaScript codes without any IDE IDEsupport.

	as well as dynamic typing, resulting in enhanced productivity and worry-free coding.	
Syntax	We write codes using the concepts of types and interfaces in C#. It is similar to OOPs syntax.	We can simply write JavaScript codes inside a script tag in HTML. We write JavaScript codes in OBSL syntax (Object Based Script language).
Client Side/Server Side Language	Commonly used on Server Side. For running C# On the client-side, we use the Silverlight plugin on the Microsoft Internet Explorer browser.	Commonly used on Client Side. We have a few frameworks based on JavaScript that allows us to use JavaScript on the server side.