

# JScript .NET

**JScript .NET** is a .NET programming language developed by Microsoft.

The primary differences between JScript and JScript .NET can be summarized as follows:

Firstly, JScript is a scripting language, and as such programs (or more suggestively, scripts) can be executed without the need to compile the code first. This is not the case with the JScript .NET command-line compiler, since this next-generation version relies on the .NET Common Language Runtime (CLR) for execution, which requires that the code be compiled to Common Intermediate Language (CIL), formerly called Microsoft Intermediate Language (MSIL), code before it can be run. Nevertheless, JScript .NET still provides full support for interpreting code at runtime (e.g., via the Function constructor or the eval function) and indeed the interpreter can be exposed by custom applications hosting the JScript .NET engine via the VSA interfaces.

Secondly, JScript has a strong foundation in Microsoft's ActiveX/COM technologies, and relies primarily on ActiveX components to provide much of its functionality (including database access via ADO, file handling, etc.), whereas JScript .NET uses the .NET Framework to provide equivalent functionality. For backwards-compatibility (or for where no .NET equivalent library exists), JScript .NET still provides full access to ActiveX objects via .NET / COM Interop using both the ActiveXObject constructor and the standard methods of the .NET Type class.

Although the .NET Framework and .NET languages such as C# and Visual Basic .NET have seen widespread adoption, JScript .NET has never received much attention, by the media or by developers. It is not supported in Microsoft's premier development tool, Visual Studio .NET. However, ASP.NET supports JScript .NET.

## JScript .NET

<b><u>Paradigm</u></b>	<u>multi-paradigm</u>
<b><u>Developer</u></b>	<u>Microsoft Corporation</u>
<b><u>Typing discipline</u></b>	<u>duck</u> , <u>weak</u> , <u>dynamic</u>
<b><u>Platform</u></b>	<u>.NET Framework</u>
<b>Influenced by</b>	
<u>JScript</u> , <u>ECMAScript</u>	

## JScript .NET

<b><u>Filename extension</u></b>	<u>.js</u>
----------------------------------	------------

## Contents

### Language differences

Differences with C#

Differences with C++

Differences with Java

Differences with older versions of JScript

### See also

### Notes

### References

## Language differences

The following are prime examples of language differences between JScript .NET and other .NET languages, including comparisons.

## Differences with C#

- JScript .NET does not require a main() function that the operating system must call directly when executing a JScript .NET application, as such, JScript .NET program flow can be based on global code.
- JScript .NET, because of its very loose type checking system can be very easy to learn, since the convention of explicit type declaration is not required at all.
- JScript .NET does not require explicit references to the .NET Framework Base Class Library, as certain functions found in earlier versions of JScript are present in JScript .NET (e.g. functions for finding the tangent of an angle for a right triangle).
- JScript .NET is closely linked to C syntax, and is thus very easy to learn for C#, Java or C++ developers.
- While JScript .NET can be used to create Windows Forms applications, JScript .NET will have some trouble, as delegates can only be consumed in JScript .NET and not created. Thus, custom events are hard to emulate in JScript .NET.

## Differences with C++

- JScript .NET does not require a main() function.
- JScript .NET does not require explicit type declaration on variables. (In C++, the use of templates and generics can be compared to this, loosely emulated with template specialization, etc.)
- JScript .NET also does not require explicit type casts on variable use in the program. Code used to retrieve a string of characters, but only used for integer numbers can be cast implicitly; the vice versa can be done without error at compile time, but there is a chance of loss of precision or data.

e.g.:

```
import System;

Console.WriteLine("Hello, what's your name?");
Console.WriteLine("Type your name: ");

var _name = Console.ReadLine();

Console.WriteLine("Hello, " + _name);
```

## Differences with Java

- JScript .NET syntax and lexical conventions are similar to Java in that both are derived from C. JScript was originally Microsoft's implementation of ECMAScript, which is more commonly known as JavaScript, though it is unrelated to Java. Thus, users of Java and other C-derived languages will find JScript easier to learn.
- JScript .NET allows developers to use untyped variables, and can sometimes infer their type from their usage to optimize the compiled code. On the other hand, Java requires all variables to be typed.
- JScript .NET can add properties and methods to objects in run-time, while Java objects always conform to their declared interface.

- JScript .NET supports global variables, something Java does not.

## Differences with older versions of JScript

- JScript .NET allows developers to declare variables and functions with type information (e.g., **var x : String;**), while type information for JScript's variables and functions can not be declared (e.g., **var x;**).
- JScript .NET scripts are not interpreted, but executed independently. When executed, a JScript .NET application will invoke the CLR. The CLR will execute the CIL instructions without using an interpreter.
- JScript .NET can be run without the presence of a browser or another scripting engine as the compiler can generate standalone executables and assemblies. However these still require .NET Framework to be installed in order to run.
- JScript .NET provides access to the .NET Framework BCL (Base Class Library), providing much more functionality.
- JScript .NET is only available as a scripting language for ASP.NET, the technology used to generate web pages; thus, JScript .NET takes a similar role to PHP and other server-side scripting languages. Internet Explorer, however, is still using only the older JScript engine, so JScript.NET cannot be used to script web pages (or HTAs or HTC's). In this regard, JScript is much more versatile than JScript .NET.

## See also

---

- JavaScript OSA – A system-level scripting language for the Apple Macintosh
- JScript
- ActionScript

## Notes

---

## References

---

- Getting Started With JScript .NET (<http://msdn.microsoft.com/en-us/library/3bf5fs13.aspx>)
- Introducing JScript .NET (<http://msdn.microsoft.com/en-us/library/ms974588.aspx>)
- JScript.NET tutorial and form tutorial ([https://web.archive.org/web/20091027083329/http://geocities.com/Jeff\\_Louie/jscript\\_net\\_windows\\_form.htm](https://web.archive.org/web/20091027083329/http://geocities.com/Jeff_Louie/jscript_net_windows_form.htm))

---

Retrieved from "[https://en.wikipedia.org/w/index.php?title=JScript\\_.NET&oldid=960913008](https://en.wikipedia.org/w/index.php?title=JScript_.NET&oldid=960913008)"

---

This page was last edited on 5 June 2020, at 15:58 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.