

A STUDY ON JAVASCRIPT

Javascript was rolled out in December 1995 and was originally developed by Netscape to extend the capabilities of HTML and provide an alternative to CGI (Common Gateway Interface) scripts. JavaScript quickly became the de facto standard for **client-side scripting** of web pages. As a consequence, Microsoft reverse-engineered Javascript and came up with their own version called JScript (for legal reasons). This was released with Internet Explorer 3.0, in August 1996.

JavaScript and JScript are often (mistakenly) used as though they are the same thing, **they are not JScript is not ECMA-compliant**, this is especially evident with Microsoft's non-standard event model. Before any other versions of Javascript appeared, in November, 1996 Netscape announced that it had submitted JavaScript to Ecma International for consideration as an industry standard, and subsequent work resulted in the standardized version named ECMAScript.

W3C (World Wide Web consortium) specify the standards for the web, (HTML, XML CSS etc.) while Javascript is managed by The ECMA (Ecma International is an industry association founded in 1961). Both organisations are ostensibly vendor neutral, but Javascript relies on the two organisations working in harmony; which they appear to do,

This has not stopped individual vendors from adding their own tweaks to the specifications. Care must be taken to avoid these if you are seeking cross browser compatibility

Javascript is syntactically very similar to C++ and Java and copying many names and naming conventions from Java, however the two languages are otherwise unrelated and differ conceptually in a number of respects.

Javascript is **not** a cut-down version of any other language.

- Javascript is an **interpreted** language; meaning it can run only in the presence of an interpreter, either in a Web server or a Web browser.
- You cannot write standalone applications with it, and mainly for web security reasons, it has very limited capability for reading or writing files.

Although Javascript has fewer capabilities than a full-fledged object-oriented languages like C++ and Java, it is sufficiently powerful for its intended purposes. One of its main uses is to provide **client-side scripting** (or processing) of web pages. For example,

- **Pre processing of web based forms.** Before sending a web based form to a server, the contents can be checked, to make sure the data is valid. All this processing takes place on the client machine.
- **Accessing Cascading Style sheets** and their properties. For example hiding or revealing the contents of a <DIV> by changing the display style property.
- **To add dynamic content to web pages** by interacting with a users actions (or **events**), for example with drop down menu's, or animation.

Javascript is an Object based language

Javascript is **not** an object- oriented language, because it has no **class** constructor. Rather it creates object instances, borrowing the notions of object **properties** and **methods**.

In Objected Orientated languages, such as C++ or Java, class is an abstraction. For example *student* may be a **class** while a particular student is said to be an **instance of** that class. javascript simply deals with instances, however this does not stop programmers trying to *layer in* the concepts of class structures such as **sub classes** and **inheritance**.

The properties and methods of an object are accessed using the **dot notation**. The brackets distinguish a method from a property. Understanding how Javascript accesses the HTML tags on a web page is crucial.

Javascript has a number of **built-in** objects so as well as getting to grips with the syntax of the language, learning how to use these objects effectively is also at the heart of program development.

objectName.propertyName

objectName.methodName(optional arguments)

Warning: Javascript is case-sensitive! So a variable called, myColor, is different from a variable called mycolor. This, along with the dot notation, can make debugging programs a frustrating business if you are not careful.

Javascript is event driven

Javascript is an event driven language, i.e. it responds to events generated by either the user , or the browser. Each event has an **event handler** (or **event listener**) that responds to its particular event. Event handlers are usually associated with (HTML) tags and enable you to attach functions to deal with just those events relevant to the program. E.g. a user clicking on a button.

Events are triggered irrespective of whether there is a corresponding event handler or not. If there is no event handler, and corresponding function to deal with it, then the event is ignored.

Is javascript compatible across all browsers?

That is a qualified yes, provided the browser in question follows both W3C and ECMA specifications. As noted above, Microsoft Explorer is not ECMA

compliant, so code that will run on other browsers will not work without modification.

Even if the browser is compliant, the vendor may be slow in implementing new features, so again differences arise between the browsers. This is a moving target which the site quirksmode.org monitors Popular browsers are tested against specification features, but they only answer to this problem is to run your scripts in the various browsers and see.

Source : <http://www.soslug.org/wiki/javascript>