**DQ2: Innovation**

JavaScript was a classic case of Netscape innovation being followed by Ecma standardization.’ Critically evaluate this statement in the wider context of the relationship between innovation and standardisation on the Web.

1. Introduction
   1. What is JavaScript?
      1. Wikipedia
         1. JavaScript (JS) is a dynamic computer programming language.[5] It is most commonly used as part of web browsers, whose implementations allow client-side scripts to interact with the user, control the browser, communicate asynchronously, and alter the document content that is displayed.[5] It is also being used in server-side programming, game development and the creation of desktop and mobile applications.
         2. JavaScript is a prototype-based scripting language with dynamic typing and has first-class functions. Its syntax was influenced by C. JavaScript copies many names and naming conventions from Java, but the two languages are otherwise unrelated and have very different semantics. The key design principles within JavaScript are taken from the Self and Scheme programming languages.[6] It is a multi-paradigm language, supporting object-oriented,[7] imperative, and functional[1][8] programming styles.
         3. The application of JavaScript in use outside of web pages—for example, in PDF documents, site-specific browsers, and desktop widgets—is also significant. Newer and faster JavaScript VMs and platforms built upon them (notably Node.js) have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript was traditionally implemented as an interpreted language but just-in-time compilation is now performed by recent (post-2012) browsers.
         4. JavaScript was formalized in the ECMAScript language standard and is primarily used as part of a web browser (client-side JavaScript). This enables programmatic access to objects within a host environment.
         5. JavaScript was originally developed by Brendan Eich, while working for Netscape Communications Corporation. While battling with Microsoft over the Web, Netscape considered their client-server offering a distributed OS, running a portable version of Sun Microsystems' Java. Because Java was a competitor of C++ and aimed at professional programmers, Netscape also wanted a lightweight interpreted language that would complement Java by appealing to nonprofessional programmers, like Microsoft's Visual Basic (see JavaScript and Java).[9]
         6. Standardization
            1. 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. In June 1997, Ecma International published the first edition of the ECMA-262 specification. A year later, in June 1998, some modifications were made to adapt it to the ISO/IEC-16262 standard, and the second edition was released. The third edition of ECMA-262 was published on December 1999.[26]
      2. (Mozilla Developer Network, 2013a)
         1. JavaScript® is the Netscape-developed object scripting language used in millions of web pages and server applications worldwide. Netscape's JavaScript is a superset of the ECMA-262 Edition 3 (ECMAScript) standard scripting language, with only mild differences from the published standard.
         2. Contrary to popular misconception, JavaScript is not "Interpretive Java". In a nutshell, JavaScript is a dynamic scripting language supporting prototype based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so.)
         3. JavaScript can function as both a procedural and an object oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects at run time, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.
      3. (Mozilla Developer Network, 2013b)
         1. JavaScript® (often shortened to JS) is a lightweight, interpreted, object-oriented language with first-class functions, most known as the scripting language for Web pages, but used in many non-browser environments as well such as node.js or Apache CouchDB. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.
         2. The JavaScript standard is ECMAScript. As of 2012, all modern browsers fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. A 6th major revision of the standard is in the works.
   2. What is Ecma?
      1. (Ecma International, n.d.)
         1. Ecma International is an industry association founded in 1961 and dedicated to the standardization of Information and Communication Technology (ICT) and Consumer Electronics (CE).
         2. For over forty years Ecma has actively contributed to world-wide standardization in information technology and telecommunications. More than 400 Ecma Standards and 100 Technical Reports of high quality have been published, more than 2/3 of which have also been adopted as International Standards and/or Technical Reports
   3. Evaluation of the statement that “JavaScript was a classic case of Netscape innovation being followed by Ecma standardization”
      1. It has been asserted in the instructions for this DQ that JavaScript resulted from innovation at Netscape being accepted as a standard by the Ecma International Group.
2. Evaluation 1
   1. (W3C, 2012)
      1. In 1996 - 1997 JavaScript was taken to ECMA to carve out a standard specification, which other browser vendors could then implement based on the work done at Netscape. The work done over this period of time eventually led to the official release of ECMA-262 Ed.1: ECMAScript is the name of the official standard, with JavaScript being the most well known of the implementations. ActionScript 3 is another well-known implementation of ECMAScript, with extensions (see below).
      2. The standards process continued in cycles, with releases of ECMAScript 2 in 1998 and ECMAScript 3 in 1999, which is the baseline for modern day JavaScript. The "JS2" or "original ES4" work led by Waldemar Horwat (then of Netscape, now at Google) started in 2000 and at first, Microsoft seemed to participate and even implemented some of the proposals in their JScript.net language.
3. Evaluation 2
   1. Eich, B 2008
      1. I'd done JavaScript and taken it through standardization at Ecma, which hadn't really done much software before then.
      2. A lot of people contributed. Sun contributed. Microsoft certainly contributed. And [Microsoft was] very interested at that point in gaining market share on Netscape. They were the minority browser, so they were saying, "What's this Netscape churn that's going on from release to release?" We want standards. And they helped make the standards. And they helped make them be real-world standards. Some of the personalities at Microsoft were very bright, and they would have liked to change some things. But as we got into it, they realized that any change meant that if they made the change in their implementation first, as the minority browser vendor, they might not work on a certain page. And then Netscape might renege or be late, and they'd lose market share.
      3. But it's a funny thing that's happened to the Web: The Web is a mess. It's formally unsound. It's a bunch of de facto and du jour standards; browsers kind of agree, and content authors worry about the top few browsers when they write something and roll it out. You still see stuff rolled out on some of the big sites that doesn't work in Safari right away. But ifs getting better. So there's this sort of intersection between the browsers and what works—like the Venn diagram and the intersection semantics. And it moves over time. And it can be moved if you put some energy into it. Web developers do want to do new things. They do want to make their life easier, and they do look at certain browsers first.
4. Evaluation 3 (optional)
   1. Zelnick, N. 1998
      1. Nonconformance or incomplete implementation of standards makes building sophisticated Web pages that work across browsers a difficult and expensive job. Because Web developers have to spend extra time and effort working around quirks in different implementations, they almost always have to privilege general functionality over technological enhancements.
      2. Netscape and Microsoft both agreed to support W3C standards when it became clear that their differing approaches were going to get them nowhere. Developers couldn’t use the nifty new features without sacrificing a significant percentage of the potential audience for their pages.
      3. It seemed for a short time that the tragedy of the Web being broken into separate, parallel worlds would be averted and that the much heralded “browser war” would settle into détente. But now that the 5.0 browsers are approaching release, it’s becoming clearer that neither company is going to live up to its pledge.
      4. Finally, if you view the Web as a new medium that changes the way people deal with information, then the browser standards debacle is nothing less than a giant step backward. By building pages for particular browsers, developers are either building documents with limited lifespans or ensuring that the hacks and workarounds used today will have to continue for the foreseeable future.
      5. Innovation in browsers is great. Developers would love to use all the nifty technologies that Microsoft and Netscape introduce. But because they spend all of their time working around what should be there but isn’t, they can’t.
      6. A browser war that is fought on the basis of core standards can do nothing but destroy the Web along the way.
5. Conclusion
   1. W3Schools, n.d.
      1. To make internet a better place, for both developers and visitors, it is important that both browsers and Web developers follow the Web standards.
      2. When developers follow the Web standards, the development is simplified, since it is easier for a developer to understand another's coding.
      3. Using Web standards will ensure that all browsers will display your Web site properly, without time-consuming rewrites.
      4. Web pages that conforms to the standard are easier for search engines to access and index, easier to convert to other formats, and easier to access with program code (like JavaScript and the DOM).

References:

Chapman, S. (n.d.) ‘A Brief History of Javascript’, *About.com* [Online]. Available from: <http://javascript.about.com/od/reference/a/history.htm> (Accessed: 31 May 2014)

Ecma International (n.d.) *What is Ecma International?* [Online]. Available from: <http://www.ecma-international.org/memento/index.html> (Accessed: 31 May 2014)

Eich, B 2008, 'The new browser war. (Cover story)', Eweek, 25, 18, pp. 47-53, Academic Search Complete, EBSCOhost, viewed 1 June 2014.

Mozilla Developer Network (2013a) *About JavaScript* [Online]. Available from: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/About_JavaScript> (Accessed: 31 May 2014)

Mozilla Developer Network (2013b) *JavaScript* [Online]. Available from: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (Accessed: 31 May 2014)

W3C (2012) A Short History of JavaScript [Online]. Available from: <https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript> (Accessed: 31 May 2014)

W3Schools (n.d.) *Web Standards* [Online]. Available from: <http://www.w3schools.com/web/web_standards.asp> (Accessed: 31 May 2014)

Zelnick, N. 1998, "Nifty technology and nonconformance: the web in crisis", Computer, vol. 31, no. 10, pp. 115-116, 119.