

# JavaScript: an analysis of the language and its impact on web development.

Thomas Henson – ST20060907

Software Engineering with Web Development

## Table of Figures

Figures	Page Number
Figure 1: Example code excerpt, as shown on <a href="http://dionach.com">http://dionach.com</a>	7
Figure 3: Graph showing the number of tags (in millions) that have been used in posts on Stackoverflow. (Stackoverflow.com, 2017)	8
Figure 4: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on January 2012. (ref)	9
Figure 5: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on January 2017. (ref)	9
Figure 6: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on July 2015. (ref)	10
Figure 7: Graph depicting the amount of internet users and online purchases made over the space of 12 months in 2016. Data collected in December 2016. (Ec.europa.eu, 2016). *Survey conducted on a total of 156,152 households containing at least one person aged 16-74, and 203,798 individuals aged 16-74 across the EU.	12
Figure 8: Graph showing the top 5 most popular JavaScript frameworks, as written on DA14 site. Statistics taken directly from respective GitHub pages (GitHub, 2017).	14
Figure 9: Table showing the preferred research approach dependent on various research conditions	18
Figure 10: Table showing loading times for both web locations with JavaScript enabled in the browser	21

Figure 11: Table showing loading times for both web locations with JavaScript disabled in the browser	22
Figure 12: Screenshot showing fully loaded webpage ( <a href="http://stuff.tv">http://stuff.tv</a> ) with JavaScript enabled in the browser (Stuff.tv, 2017)	22
Figure 13: Screenshot showing fully loaded webpage ( <a href="http://stuff.tv">http://stuff.tv</a> ) with JavaScript disabled in the browser (Stuff.tv, 2017)	22
Figure 14: Screenshot showing fully loaded webpage ( <a href="http://docs.google.com">http://docs.google.com</a> ) with JavaScript enabled in the browser (Google, 2017)	23
Figure 15: Screenshot showing fully loaded webpage ( <a href="http://docs.google.com">http://docs.google.com</a> ) with JavaScript disabled in the browser (Google, 2017)	23

## Table of Contents

Table of Figures.....	1
1 - Introduction .....	7
1.1 - Aims and objectives .....	7
1.2 - How research will be undertaken .....	8
2 - Literature Review .....	9
2.1 - Abstract.....	9
2.2 - A brief history of JavaScript and the Internet.....	9
2.3 - How JavaScript has developed since it was first created .....	10
2.4 – Security issues of JavaScript in web applications.....	10
2.5 - How JavaScript has impacted web development .....	13
2.6 - JavaScript frameworks and libraries in modern use.....	18
2.6.1 - What is a framework? How does it differ from a library? .....	18
2.6.2 - Why use a framework or library? .....	19
2.6.3 – Popular JavaScript frameworks and libraries.....	19
2.7 - Will it still be relevant in the future? .....	20
3 - Methodology.....	22
3.1 - Introduction .....	22
3.2 - Research philosophy .....	22
3.3 - Research approach .....	23
3.4 - Data collection methods.....	24
3.5 - Ethical issues .....	25
3.6 - Limitations .....	25
4 - Discussion and Testing.....	27
4.1 - Brief overview of discussion topics.....	27
4.2 - Testing – Disabling JavaScript in the browser.....	27
4.2.1 - Brief overview of testing.....	27
4.2.2 - Results of testing.....	28
4.3 - Discussion .....	30
4.3.1 - The Future of JavaScript .....	30
4.3.2 - Predictions .....	32
5 - Conclusion.....	33
5.1 - Summary .....	33
5.2 - Conclusions .....	33
5.3 - Importance of this research.....	34
5.4 - Recommendations .....	34

5.5 - Reflection .....	35
References .....	36
Appendix .....	41
Ethics Application Form .....	41

## Abstract

For this dissertation project, research will be undertaken into the JavaScript language, so that an in-depth analysis can be carried out. The aims and objectives of the research is to gather a solid understanding of the origins and history of the JavaScript language, discuss the potential implications of the security issues within JavaScript and look at the impact it has had on the web development industry. In addition, popular JavaScript frameworks will be discussed, looking at why they are used and preferred by developers. Finally, predictions will be made surrounding the future of JavaScript within web development in the near future and ultimately conclude on the importance of the language.

## Acknowledgements

I would like to give special thanks to my supervisor, Dr. Eysin Chew, who has assisted me through various set-backs during this project. To my friends and family, I can't thank you enough for giving me the motivation to stay focussed and complete my dissertation successfully.

## 1 - Introduction

*Have you ever wondered how JavaScript has become one of the most popular programming language in the world? Or maybe why it is considered “The Language of the Web”?*

The name ‘JavaScript’ is one that almost everybody has heard of, even if they are not in the software development industry or just not interested in coding or technology in general. The reason why it sounds familiar is simply because the language is so popular in modern society. JavaScript is the only language that is implemented in every mainstream web browser that you can think of. Chrome, Firefox, Safari, Edge and Opera all utilise JavaScript as their primary client-side scripting language. Much of the advertisements, animations, features and general functionality of the web pages you browse on a regular basis are powered by the JavaScript language.

This dissertation looks to delve deeper into the JavaScript language, pursuing a greater understanding of the language as a whole and why it is now considered ‘the language of the web’. It will be looking at how JavaScript is used in modern web applications and the impact that it has had on web development today. It also seeks to discover why developers use the vast array of JavaScript frameworks and libraries, and why specific frameworks and libraries are more popular than others. In addition, it will look to discover the statistical aspect of the languages popularity and how the number of users has grown over a period of 15 years.

### 1.1 - Aims and objectives

The global aim of this dissertation is to be able to identify the key aspects of JavaScript’s popularity and how it became such a widely used and adopted programming language. It also seeks to analyse the various frameworks and libraries that have been developed in the JavaScript language and why they are used. Most importantly is to discover the impact that this programming language has had on not just the web development industry, but the consumer and how we interact with web pages on the internet. It will also be attempting to predict, based on various findings, the future of the JavaScript language and if it will still be a viable option in 10 years’ time.

To work towards these targets, there are a few important aspects that need to be followed closely:

- To gain an understanding of the JavaScript language and its history
- Look at the security issues surrounding JavaScript-based web applications
- Discuss various frameworks and libraries, if they should be used and why
- To talk about how JavaScript has impacted the web development industry and the way we interact with the web
- Discuss the future of the JavaScript language in 10 years’ time



## 1.2 - How research will be undertaken

This dissertation will use a case study methodology to explore the history of the JavaScript language, why it has gained such a tremendous following and how it will fair against potential rivals in the future. It will use descriptive quantitative research methods to actively seek out the information that is required, so that it can be analysed and compared to produce findings, provide topics for discussion and help in making final conclusions. This will be discussed in greater detail in the full methodology.

## 2 - Literature Review

### 2.1 - Abstract

This section of the report, the literature review, will be looking into the JavaScript language, investigating areas such as the history of the programming language and how it is used in modern applications. It will be comparing both past and present literature, which can then be used to aid with my writing on specific subjects. For the first part, the report will be looking at the JavaScript language including its history and how the language came into fruition.

### 2.2 - A brief history of JavaScript and the Internet

The scope of the internet has increased dramatically over the years. It's very first origins as "ARPANET", a realisation of Dr J.C.R Licklider which was developed as a way for several remote computers to share a single computers resources concurrently (Hauben, 2007), set the foundation for the internet as we know it today and was a big shifting point in computer history. At the time, 'ARPANET' was revolutionary, as it allowed for something that had never been accomplished before. It made thinking about the potential developments of the system very easy, with his ideas and proposals starting to become an obvious realisation to most (Hauben, 2007). As Robert Taylor, Licklider's successor at the Information Processing Techniques Office (IPTO), has said: "In pointing out the community phenomena created, in part, by the sharing of resources in one timesharing system, Lick made it easy to think about interconnecting the communities, the interconnection of interactive, on-line communities of people".

With these ideas changing the face of computer networking and the way that people interact with computers, it wasn't long before Sir Tim Berners-Lee capitalised on the opportunity of expansion. When the World Wide Web was first conceptualised and commercialised, everything was very basic. The world's first ever website, which was hosted on Berners-lee's own personal NeXT desktop computer, was a very basic HTML page that described how to access other people's documents and how to create your own server (Home.cern, n.d.). There was no external styling for these web pages or any form of additional features. It was just a basic hypertext document that anyone could gain access to.

As the World Wide Web increased in popularity during the late 1990's and early 2000's, developers started to look for new ways to improve the overall appearance and usability of their websites. For what was to be the first of many new languages, the JavaScript language was created in May of 1995, after only ten days of work (W3.org, n.d.). The language was created by a man called Brendan Eich, who was working for NetScape at the time of the development, but now works for Mozilla (W3.org, n.d.). Interestingly, during the first development stages, JavaScript was actually given the name of 'Mocha'. This was to be changed to LiveScript in September 1995, when finally, after obtaining a trademark licence from Sun, the name JavaScript was given (W3.org, n.d.). In an interview with InfoWorld editor Paul Krill, Berners-lee was asked about the naming of the language and how it came to its finalised title. He said the following: "It was all within six months from May till December (1995) that it was Mocha and then LiveScript. And then in early December, Netscape and Sun did a license agreement and it became JavaScript. And the idea was to

make it a complementary scripting language to go with Java, with the compiled language” (Krill, 2008).

The whole idea of the JavaScript language encapsulates the notion of accessible programmability for programmers with varying levels of expertise. It was designed for people to be able to add “a little bit of animation of a little bit of smarts” to their web pages and applications, without having to learn an entire new language (Krill, 2008). The greatest thing about JavaScript is that you can just learn it as you go along. You can look at snippets of code from someone else's work and apply it in your own developments, or you can read articles, forums or even books that detail how to do certain tasks specifically. All this can be done with little to no programming knowledge, which can be one of the reasons why JavaScript has become, and still is, so popular and considered to be the foundation of most modern-day web pages and applications.

When JavaScript was first conceived and developed back in 1995, the web was very basic. HTML was in development stage 3.2 (currently in 5.1 working draft) at the time, and people did not have much to work with in terms of programmability (Krill, 2008). Java was picking up at the time, but it was at a stage where developing in the Java language would require the use of high-powered, high-performance computers (Krill, 2008). This obviously meant that Java was not an accessible language for everybody and was mainly being used by professional programmers. Therefore, JavaScript was created to act as a ‘little brother’ language to Java. The beauty of JavaScript is that it was being marketed as a ‘little brother’ language to Java. Much the same as how Visual Basic was developed to compliment C++ (Krill, 2008). Evidently, it was to become something of a sensation in the world of web development, and it has changed the way we interact with the internet forever.

### 2.3 - How JavaScript has developed since it was first created

After its initial creation and release, the JavaScript language was taken to ECMA in (1996/1997) to “carve out a standard specification” (W3.org, n.d.). ECMA International is an industry association that was founded in 1961. The purpose of this organisation is to create and enforce standards for ICT and Consumer Electronics sectors. This was so that the various browser vendors, such as Internet Explorer and Opera, could implement such standards in their applications, based on the work that had already been carried out at Netscape (W3.org, n.d.).

### 2.4 – Security issues of JavaScript in web applications

Whilst the popularity and widespread use of JavaScript has positive effects on web development and the quality of web pages produced, this can also have a negative effect on the security of applications. More and more people learning and utilising JavaScript in their web pages presents greater opportunity for malicious hackers and cyber criminals to inject harmful code into a webpage, putting the consumer at risk, as well as having potentially disastrous effects on the web applications themselves.

JavaScript has a variety of common vulnerabilities that can be exploited for criminal gains, however there are two main vulnerabilities that are frequently presenting issues to developers:

- ❖ Cross-Site Scripting (XSS)
- ❖ Cross-Site Request Forgery (CSRF)

Cross-Site Scripting, or XSS, can be defined as the manipulation of the client-side scripts of a web application, in this case the client-side scripting being JavaScript, to execute in the way that is desired by the malicious hacker (Assis, 2016). This can allow for a malevolent script to be embedded into a web page, which can then be executed each time the web application is loaded or an associated event is performed (Veracode, n.d.). It is one of, if not the 'most prevalent high risk web application vulnerability' facing web developers today, however it is often overlooked (Dionach, 2016). The nature of XSS attacks mean that in general, they require some form of interaction from the user (victim) for the script to be initiated. This, or it will wait for the user to visit a specific page on the website. This is one of the reasons why developers may not take it so seriously (Assis, 2016). However, if these vulnerabilities in the scripting of a web application are left open, they can be very dangerous to the user. Some types of malicious scripting can even change the HTML code of a web application, therefore rendering the page differently to the end user (Assis, 2016).

The most common way for a hacker to inject malicious scripts into a web application is using inputs. This can either be through comment boxes, contact forms, login panels, or any other form of user input that is available on the web page (Assis, 2016). The hacker can inject an illegal script into a genuine application through this method, which when accepted by the program can transform functionality of certain aspects of the application and even change the way it looks. An example of a piece of script that could be implanted into a JavaScript web application could look something like the following:

```
<script>
  image = new Image();
  image.src='http://[Attacker IP]:8080/?'+document.cookie;
</script>
```

Figure 1: Example code excerpt, as shown on <http://dionach.com>

The result of this example code being injected into a web application using JavaScript would allow the hacker to store the web user's session cookies in addition to other methods. As a quick description for those who don't know, a browser 'cookie' is a simple text file that stores information about your activity on the web (Bbc.co.uk, 2012). In this instance, the cookie can store the user's session state, which is essentially a snapshot of the user's current web browsing activity. The danger here is that once a hacker has retrieved these cookies from the victim's browser, they can then implement these cookies into their own browser, as to 'trick' the browser into believing that they are logged into the victim's account (Dionach, 2016).

Additionally, there is the issue of CSRF attacks on web applications. “Cross-Site Request Forgery (CSRF) is an attack that forces an end user to execute unwanted actions on a web application in which they're currently authenticated” (Owasp.org, n.d.). This type of web attack specifically targets ‘state change requests’, rather than the outright theft of data, which can range from actions such as the changing of an email address, to even transferring funds from the victims account(s) (Owasp.org, n.d.). Also, depending on if a user’s account is an administrator, the hacker could potentially gain access to the entire web application.

Consider the following example excerpt of code, showing a potential hack that a hacker could inject into a vulnerable web application using the CSRF method:

```

```

Figure 2: Example code excerpt, as shown on <http://dionach.com>

The result of this example code being injected into a web application would instantiate a request for transfer of funds from the victims account. Instead of including a real image within the img tag, the hacker has placed a link inside the source that carries out a bank transaction from the victim’s bank account website. The fact that this is possible makes the use of JavaScript in web pages questionable, especially so with web applications that don’t validate their inputs correctly to keep them secure.

Comparing the advantages of security of using JavaScript with the disadvantages, a good way to describe the experience of using the web with JavaScript turned off is this:

---

*“disabling JavaScript due of security concerns is like wearing a bubble suit every time you go outside because you’re afraid of getting hurt. It won’t actually protect you from much, but it will make your life miserable.” (Lee, 2014)*

---

It is important to note that the increase in number of users actively adopting and using JavaScript in the web browser has improved dramatically over the years. Looking at the Screen Reader Survey’s conducted by WebAim (Webaim.org, 2009), we can see that in October 2009, the percentage of total users (from 665 valid responses) that stated they browse with JavaScript forcibly disabled was 10.4% (Webaim.org, 2009). Five years later, in January 2015, the 5<sup>th</sup> Screen Reader was conducted by WebAim, stating that number of total users browsing with JavaScript disabled (from 1465 valid responses) was at 2.4% (Webaim.org, 2015). This is a sizable decrease of the number of users browsing without JavaScript, with a percentage of 8% over 5 years.

Unfortunately for JavaScript’s reputation, a common theme that seems to present itself prominently whilst conducting research into the JavaScript language is the fact that it has security issues, those of which are easily exploited.

## 2.5 - How JavaScript has impacted web development

As covered in the previous sections of this literature review, it is easy to conclude that the JavaScript language is a very versatile one. One that allows many people, of varying skill levels and programming experience, to harness the full potential and power that the World Wide Web must offer when developing their applications.

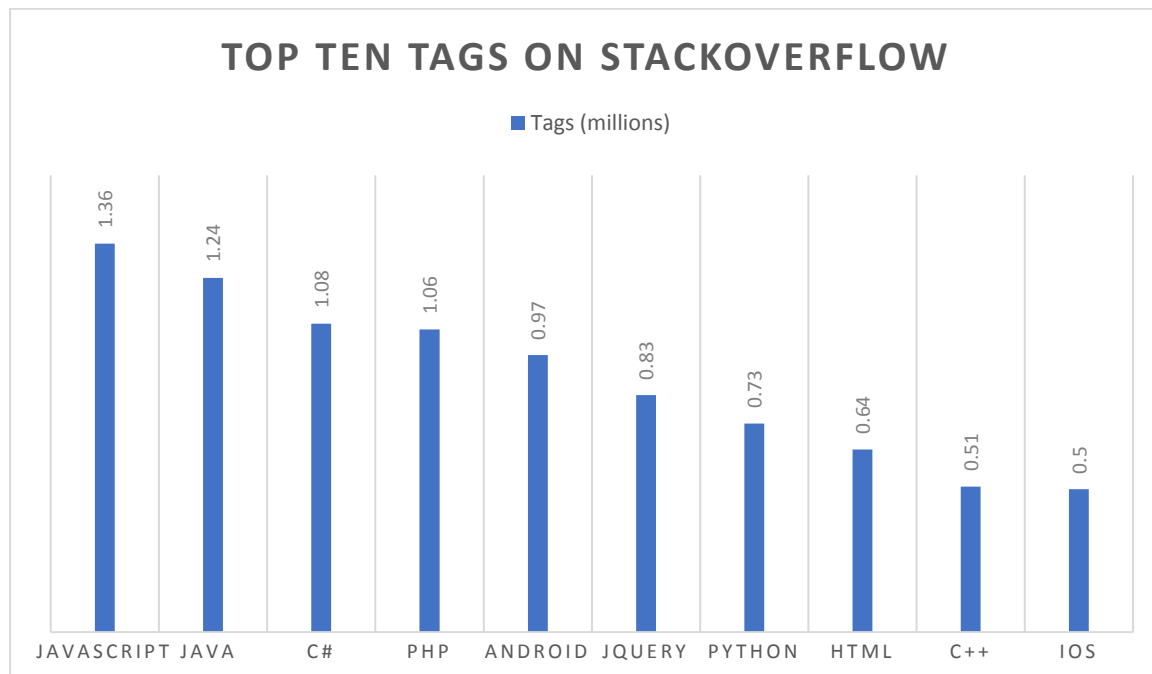


Figure 3: Graph showing the number of tags (in millions) that have been used in posts on Stackoverflow. (Stackoverflow.com, 2017)

As shown in Figure 1, JavaScript has proven to be the most talked about subject on the popular programming forum, Stackoverflow. The way that this has been calculated is by looking at the number of 'tags' that are attached to each individual post on the Stackoverflow forum. The numbers shown, in millions, are rounded to the nearest ten thousandth for simplicity and coherency. Therefore, we can conclude from this dataset that, from looking at the 'top ten' most used tags on Stackoverflow, that JavaScript is the most talked about topic on the forum. What this means for the language is enormous, as it backs up the theory that JavaScript truly is 'the language of the web' as famously quoted by Douglas Crockford, and helps to plant the language further into the top spot for the most popular web programming language (Cleverism, 2015).

As further testament to the surge in popularity of the JavaScript language, research was conducted on the increased use of the JavaScript tag on Stackoverflow. Using web application tagtrends.com, we can see how the use of the tag has increased dramatically over a period of 5 years.

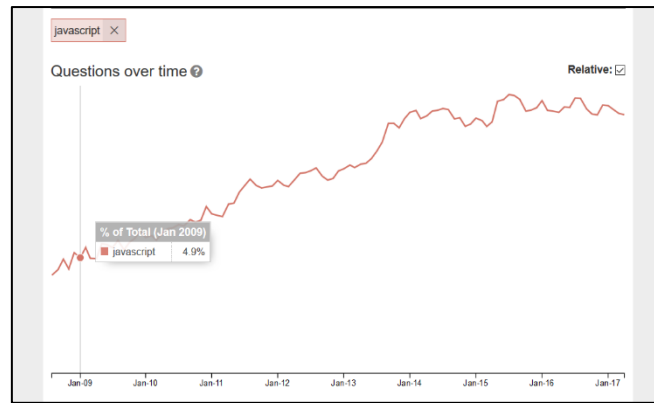


Figure 4: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on January 2012. (ref)

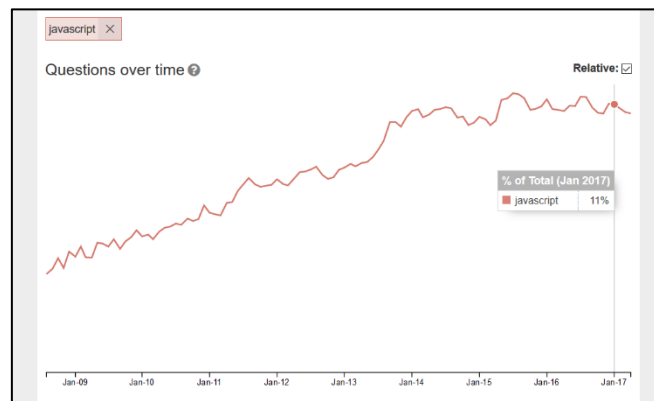


Figure 5: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on January 2017. (ref)

In figure 10, we can see that the JavaScript tag takes up 4.9% of the total amount of tags used in posts on Stackoverflow in 2012 (Sotagtrends.com, 2017). However, in figure 11, we can see that over a period of 5 years, the amount of people posting on Stackoverflow using the JavaScript tag has increased substantially, by a staggering 6.1% to make up 11% of the total tags used in 2017 (Sotagtrends.com, 2017). This goes to show just how popular JavaScript has become, and by all accounts is still becoming, as more and more people appear to be discussing the language and learning how to use it. What seems to be of slight concern, however, is that although the amount of tags being used is at a very high level it was at its highest level in July 2015, when JavaScript took up 12% of the total tags on Stackoverflow (Sotagtrends.com, 2017).

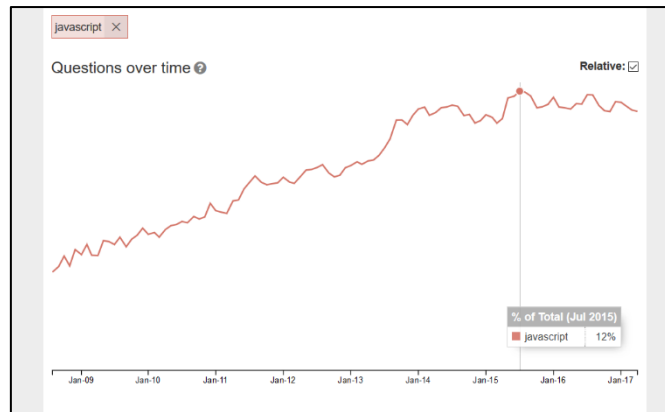


Figure 6: Line graph showing the use of the JavaScript tag on Stackoverflow, specifically focussing on July 2015. (ref)

It appears to be remaining at a somewhat flat level, however the decrease in use of the JavaScript tag is to be expected, as more and more people start to adopt various newer languages, such as Ruby and Python. This is not to say, however, that JavaScript is not being utilised or is becoming irrelevant, as the following work covers in more detail.

To further understand how much of an impact that JavaScript has had on the internet and web development in general, we need to look at how the web would look if JavaScript was 'disabled'. Little known to the majority is that you can turn JavaScript off in your web browser, meaning that any JavaScript contained on the websites you visit will be blocked, rendering the code useless. Whilst this may seem like an easy way to enhance protection and remove annoyances when browsing the web, the security advantages are minimal and most, if not all websites will not load correctly. Disabling JavaScript in your browser may not only affect the way that the website looks and feels, but it could also hinder its performance and disable some essential features, for example, being able to log into your accounts or posting comments on social media or forums. Whilst some sites, such as Gmail, provide an alternative basic HTML option for those with the JavaScript feature disabled, some websites may not load at all. Users do have the ability to manually select specific websites to be exempt from the deactivation, but this takes 'precious time' (Hoffman, 2013).

In addition, the notion that web developers should strive to enhance their web pages for use without JavaScript have been criticised. Certain web pages are designed to work more like a software application than a traditional website, with much of these 'web applications' being heavily based on the JavaScript language. Apps such as Google Docs, which rely on JavaScript to maintain its excellent user interface, would be inoperable without the JavaScript code that powers it and makes it such a great application. There have been arguments to prove that turning off JavaScript completely is far from the best option for improving security and blocking ads whilst browsing the web. As quoted from an article on HowToGeek, a popular online tech magazine founded in 2006, writer Chris Hoffman said:

---

*"Disabling JavaScript takes websites back to a time when they were simple documents without any other features. While some people may long to return to that time, that's not the web we live on today."* (Hoffman, 2013)

---



This further reiterates the fact that JavaScript has etched itself into the very fabric of the internet as we know and love it today. As mentioned previously, the web was once a very basic, 8-bit expanse that wasn't very easy on the eye. Basic HTML documents with flashy (in the literal sense of the word) animations. With advancements in technology came new programming languages, and JavaScript took the reins and became the godfather of the web browser.

Whilst it may seem like JavaScript is in its glory days, there are strong arguments to suggest that all is not as it seems, with some actively encouraging the deactivation of the language in the browser. For a language that has such an impact on the way we browse the internet, this may seem like a farfetched attempt. However, what would happen if one day support for the JavaScript language in the browser was phased out?

In a recent discovery, it was found that 'the majority' of ecommerce software had not be configured to work for those that had disabled JavaScript in the browser. This, in effect, is turning away an average of 1 in 20 customers (Boag, 2010). Considering the statistics for web users in Britain alone, 95% of households in Britain alone used the internet in 2016\*, with 83% percent of those making online purchases in the same year, as shown in Figure 3 below:

	Proportion of individuals who:	
	Used internet within the last 12 months	Purchased online within the last 12 months
<b>EU-28</b>	<b>84</b>	<b>55</b>
Belgium	87	57
Bulgaria	62	17
Czech Republic	83	47
Denmark	97	82
Germany	91	74
Estonia	88	56
Ireland	83	59
Greece	70	31
Spain	81	44
France	88	66
Croatia	74	33
Italy	71	29
Cyprus	76	29
Latvia	81	44
Lithuania	75	33
Luxembourg	98	78
Hungary	81	39
Malta	78	47
Netherlands	94	74
Austria	85	58
Poland	75	42
Portugal	71	31
Romania	66	12
Slovenia	76	40
Slovakia	83	56
Finland	94	67
Sweden	95	76
United Kingdom	95	83
Norway	98	78
FYR of Macedonia	75	15
Turkey	59	17

Figure 7: Graph depicting the amount of internet users and online purchases made over the space of 12 months in 2016. Data collected in December 2016. (Ec.europa.eu, 2016). \*Survey conducted on a total of 156,152 households containing at least one person aged 16-74, and 203,798 individuals aged 16-74 across the EU.

Tests will be carried out to experience the effects of disabling JavaScript in the browser from point of view. A shortlist of two different web applications, Google Docs and Stuff.tv, has been created, each of which will be visited both with JavaScript enabled and disabled respectively. The results of these tests will serve to prove just how much of an importance that the JavaScript language has become for the consumer when browsing the web. The results will be included as screenshots, each with a small discussion on each of the findings.

## 2.6 - JavaScript frameworks and libraries in modern use

### 2.6.1 - What is a framework? How does it differ from a library?

A software framework, or in this case, a JavaScript framework, can be defined as a 'skeleton', for which the developer can define the applications own individual, unique features to fill out the skeletal structure (Kumar, 2015). Therefore, a framework can be defined as a structure for which a developer can follow and use to their advantage during the creation of a web project (Kumar, 2015). The way in which a framework is designed to work is through a paradox, known as "Inversion of Control".

#### *Inversion of Control*

Inversion of Control, otherwise known as the Hollywood Principle ("Don't call use, we'll call you"), is a phenomenon that has been cited as a 'defining characteristic of a framework' (Fowler, 2005). It allows the developer to focus on the key elements and functionality of their code, whilst the framework is then left to take care of calling the methods and functions that have been created. In an online article written by Martin Fowler, a ThoughtWorks employee and avid software development enthusiast and writer, he quoted the following:

---

*"One important characteristic of a framework is that the methods defined by the user to tailor the framework will often be called from within the framework itself, rather than from the user's application code. The framework often plays the role of the main program in coordinating and sequencing application activity. This inversion of control gives frameworks the power to serve as extensible skeletons. The methods supplied by the user tailor the generic algorithms defined in the framework for a particular application."* (Johnson and Foote, 1988).

---

It seems that many people believe that frameworks and libraries in software development are almost identical, or at least very similar. However, this is not the case, as they are in-fact completely different by design and in their purported uses. As described above, a framework is defined as the 'structure' of an application, one within which the developer can tailor the features and characteristics of their code, allowing the framework to 'take control' by calling methods and functions in accordance with how the code has been written.

A library, however, differs from the description of a traditional software framework, in the sense that it does not follow the 'Hollywood Principle'. The reason for this is because a library is simply a collection of pre-defined classes, each of which have been designed to complete specific tasks and carry out certain functions (Kumar, 2015). It allows the developer to utilise various aspects of code that have been written and perfected by other developers and use this to their advantage in their own works. Whereas a framework is something that one would use as the foundation of a project, a library can be seen as more

of a 'retrofit', allowing developers to add various classes and functions to their code without dramatically changing the structure or handing over control (Kumar, 2015).

### 2.6.2 - Why use a framework or library?

#### *Why you should use frameworks and libraries*

One of the biggest advantages of using a framework, from a developer's perspective, is the fact that one can fixate on perfecting the features of a project without having to 're-invent the wheel' each time. A framework can encapsulate the common aspects of every web page across the internet and make it readily available for anyone to utilise in their work. This must be one, if not *the* main reason, why certain JavaScript frameworks and libraries have gained such popularity.

Another idealistic feature of popular JavaScript frameworks, such as Angular.js and Node.js, is simply the fact they make a developer's life much, much easier. It allows one to work within a pre-fabricated environment, with the only tasks left to do being adding functionality and design to suit. Frameworks are also a great way to ensure cross-browser compatibility, as most are designed to work in all browsers supporting the JavaScript language.

### 2.6.3 – Popular JavaScript frameworks and libraries

When it comes to choosing a framework to use when coding in JavaScript, the options are certainly plentiful. There are many to choose from, but it begs the question of which one should you choose and which will suit your application the best. There seems to be a few that stand out from all the rest, in the sense they are widely adopted and favoured by developers. In an article posted on DA14's website, a bespoke software solutions company, the 5 most popular JavaScript frameworks in 2017 were highlighted and discussed in brief detail. This was used as the basis of the following research (Da-14.com, n.d.).

Consider the following graph, which shows statistics in relation to each framework respectively using data from GitHub:

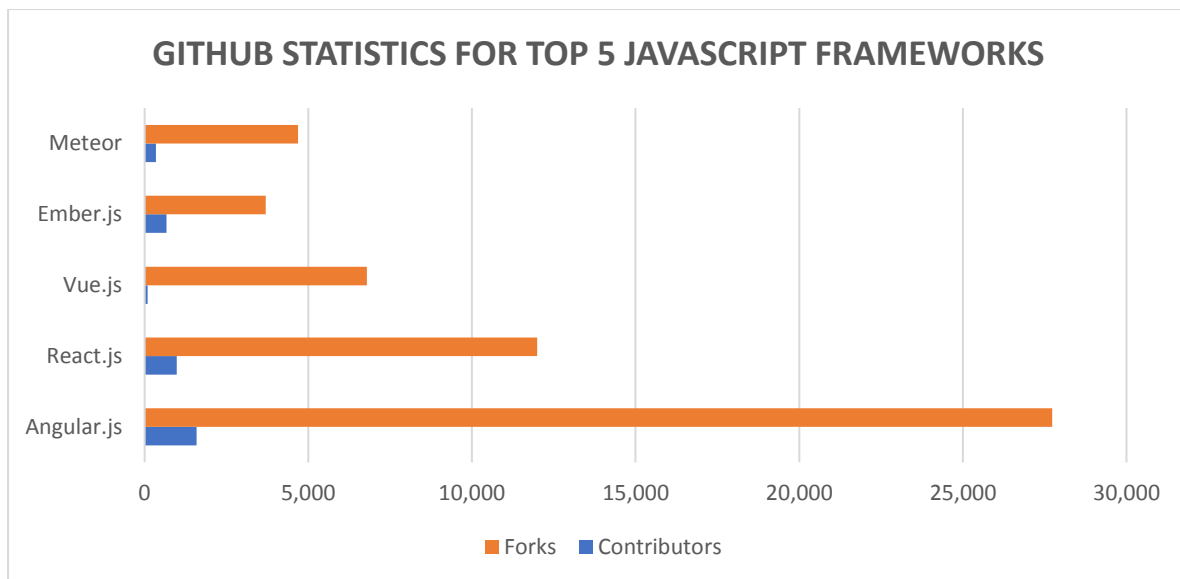


Figure 8: Graph showing the top 5 most popular JavaScript frameworks, as written on DA14 site. Statistics taken directly from respective GitHub pages (GitHub, 2017).

Let's analyse the data being displayed in Figure 2. There are two key elements on display; forks and contributors. A 'fork' on GitHub is when a user copies the repository, so that they can make changes to the source code freely, without it affecting the official releases. This is a huge benefit of organisations adopting the 'open-source' format with their projects, as it allows developers to experiment with and adapt the source code to suit their specific requirements.

'Contributors', on the other hand, are those that contribute to a particular project by adding elements to the source code, which have then been accepted by the developers.

Contributions can be anything from a simple error fix to larger functional changes. Thusly, this information can be used to predict which frameworks are being utilised the most by software developers, as we can see the various amounts of forks and contributions for each individual framework.

## 2.7 - Will it still be relevant in the future?

Whilst JavaScript may currently hold the title for the most popular programming language (Stackoverflow.com, 2017), it does not necessarily mean that it will stay this way. In a famous article, written by JavaScript author and a huge contributor to the development of the language, Douglas Crockford said the following:

---

*"Programming languages are like cats. It is easier to get a new cat than to get an old cat fixed." (Crockford, 2008)*

---

From this statement alone, the future of the JavaScript language and its frequented use in almost all websites and web applications is put into question. It is difficult to enforce this statement using the JavaScript language itself, as rather than replace an existing language, it looked to compete with the popular languages at the time, including C and Java especially.

However, unfortunate as it may be, JavaScript seems to have quite a lot of competition in the modern market.

In an interview with Brendan Eich, the ‘founding father’ of the JavaScript language, he was asked if he had heard of ARAX (Asynchronous Ruby and XML) and APAX (Asynchronous Python and XML), which allows Ruby and Python to be used in the browser instead of JavaScript. Eich has this to say:

---

*“Yes. I think that’s great, and I think we will support other programming languages.” (Krill, 2008)*

---

This statement speaks of collaborations between competing languages and is a promising prospect from the eyes of the JavaScript development team. It feels as though Eich is confident in his language from this declaration, as he doesn’t seem to be phased by the possibility of another language ‘taking over’ as an alternative browser scripting language. Furthermore, Eich finished the interview question by stating: “What will be interesting will be to see how it’s standardized because until it’s really in all browsers, developers can’t count on it.”. This again proves that JavaScript still has a very strong hold on the web development industry, as crucially it is still the only server-side language that has been adopted by most, if not all web browsers. For another language to take over, big changes would need to be made to browsers, both in the development and release stages. Developers would need to take a different approach when developing a new website or application so that it could incorporate these new languages. In addition, they would also need to edit their existing projects to meet the new requirements of all the browsers. Whilst it’s possible for JavaScript to be replaced as the foundations of our web browsers and the websites we frequent, it wouldn’t be an easy transition. Something that Brendan Eich makes crystal clear in his interview with InfoWorld editor, Paul Krill (Krill, 2008).

The implications of this type of activity, in pairing Python and Ruby with XML to replace JavaScript, could be detrimental to the success of the language in the future. The fact that the substitution of the JavaScript language is even being considered at this stage is a huge blow in terms of the expansion of JavaScript on the web. Not only does it have to worry about other languages taking the spotlight, there also some serious security and privacy issues associated with JavaScript.

Taking into consideration the alarming rate of expansion within the JavaScript language, such as the various libraries and frameworks that are continuously being updated and released, it is easy to get caught up in it all and forget about the essentials of security on the client-side. The problem with using frameworks or libraries when writing code is this; you only need to know the framework. Consequently, this means that you don’t have to know the ins and outs of JavaScript to successfully utilise these frameworks or libraries in your works. By means of this, malicious hackers and cyber criminals may be able to exploit the ‘gaps’ that have been left open in your code through lack of knowledge on the subject.

Accordingly, as JavaScript is an interpreted language, instead of the code being compiled to machine-readable format before its distribution, it is compiled a runtime, meaning that it is being compiled as the application is running. Theoretically, this can allow essentially anyone that downloads software that is JavaScript based to gain access to the source code, which obviously in some scenarios is less than ideal.

### 3 - Methodology

#### 3.1 - Introduction

Research can be defined, in the non-technical sense, as the search for knowledge (Kothari, 2004). Thus, research can be depicted as an original contribution, one of which is added to an existing pool of knowledge, making for its advancement and further expansion.

The research methodology, on the other hand, is looked upon as a way to 'systematically solve the research problem' (Kothari, 2004). A methodology can be understood as the study of how research is conducted scientifically, specifically looking at the various steps adopted by the researcher in pursuing knowledge around a particular problem, along with the logic behind these steps and methods.

This methodology seeks to analyse the research methods that have been adopted throughout this dissertation, looking at the research philosophy and approach that has been adopted for the project. It will also be discussing the type of data that has been collected for the project, along with the ethical issues associated with data collection and the limitations that were faced during the process of data collection.

#### 3.2 - Research philosophy

The research philosophy that has been adopted for this research project is Positivism. The Positivism philosophy follows the principle that only factual knowledge gained through observation, including measurement, is trust worthy. In other words, it aims only to seek out confirmed knowledge and statistics for the purpose of research. The research that has been undertaken for this project has been limited under the specified philosophy, however it suits the requirements of the project as whole, in terms of the target resolution. Data collection and interpretation through the means of an objective research approach, being qualitative, are expected when utilising Positivism.

The researcher is also considered to be completely independent from the study, which in context means that in addition, there is minimal interaction with research participants during the period of research. Thusly meaning that as a positivist project, the findings will be mostly based on fact, with little opinionated scripting. As only secondary data is being collected for this dissertation and relative findings, the Positivism philosophy was best suited to the task at hand, as the sole purpose and main question of this work is to analyse the history and impact of the JavaScript language, which is mostly factual.

It is considered to be good practice to adopt a deductive approach when undertaking a research project under the positivist paradigm. This is simply because Positivism is based around facts and statistics, which suits the deductive approach, as it is geared towards the hypothesising and testing of a particular subject area. The research approach that has been

used for this research will be discussed in greater detail in the following section of the methodology.

The main reason that this philosophy was chosen for this project is because it is focussed primarily on collecting secondary data. An in-depth analysis has been conducted of the JavaScript language, its impact on the World Wide Web and the various JavaScript frameworks being utilised by today's web developers. A common philosophy that students tend to adopt when conducting a piece of work such as the dissertation is Interpretivism. Interpretivism is focussed with human participants and their interaction with the study, therefore Interpretivism and Positivism are almost opposites. Additionally, as stated previously, the positivist movement aims to distance its self from any participants, such that the findings produced towards the end of the research are not affected by bias or the perspective of the researcher (ref). Results should be based upon reason and logic, consequently preserving the importance of remaining neutral when researching.

Whilst the Positivist movement and its traits in relation to research have been beneficial to the structure of this research project, there are various downfalls of this philosophy. In relation to Positivism being purely factual and emotionally-detached, the research findings produced because of abiding by the positivist principles are only descriptive. This, in turn, can mean that the results of research conducted lack insight into the more in-depth issues surrounding the research topic. In hindsight, this could be considered a reason why philosophies such as Interpretivism are preferred.

### 3.3 - Research approach

The research approach that has been adopted for this research project is the deductive method. A deductive research approach is focussed on developing a hypothesis, or in some cases hypotheses (plural), that are based upon existing theories, e.g. previous findings in relation to the research topic. Thereafter, a research strategy is to be designed with the purpose of testing the hypothesis.

#### *What is a hypothesis?*

A hypothesis can be defined as an explanation, or 'supposition', that is provisionally accepted in order to predict certain events or phenomena (BusinessDictionary.com, n.d.). Additionally, a hypothesis seeks to provide guidance for further investigation into the subject area. It may be proven wrong or it may be proven correct, and it must be able to produce arguments (BusinessDictionary.com, n.d.).

Contradictory to the deductive approach is the inductive approach. Briefly, an inductive research approach is focused with early observations during research, with theories not being produced until research is coming to an end (Research Methodology, n.d.). It is clear to see how this approach differs from deductive, as with deductive a theory is presented at the start of the research. The basis of which is then used for testing against and constructing arguments to further develop the research.

Consider the following table, using data provided from (Research Methodology, n.d.), that depicts the main differences between these two popular research approaches. It is clear to see why the deductive approach has been chosen for this research project. As only



secondary data is being collected, less time is required to complete the study. Additionally, the 'abundance of sources' that are available in relation to the title of the dissertation edges more towards deductive, as no primary data needs to be collected to successfully work towards answering the topic question.

	Deductive preferred	Inductive preferred
Wealth of literature	Abundance of sources	Scarcity of sources
Time available	Short time available to complete the study	There is no shortage of time to complete the study
Risk	To avoid risk	Risk is accepted, no theory may emerge at all

Figure 9: Table showing the preferred research approach dependent on various research conditions

The main reason that the deductive approach has been chosen for this research topic is simply because there an underlying theory presenting itself throughout this project. The theory is that the JavaScript language could be irrelevant in the near future. This project seeks to prove, one way or the other, if this theory has any foundations through the analysis of past literature and statistical data.

### 3.4 - Data collection methods

For this project, only secondary data is being collected for the purpose discovering the origins and impact of the JavaScript language. Secondary data collection, or analysis, can be defined as the use of existing literature to find the answer a question that differs from the original source (ref). For existing works to be analysed and utilised, various research methods need to be considered. There are three main types of data collection methods, which apply to both primary and secondary research:

- ❖ Qualitative methods
- ❖ Quantitative methods
- ❖ Mixed methods

For the research being undertaken in this project, mixed data collection methods have been used, utilising a combination of both qualitative and quantitative methods. Primarily the research will be carried out using qualitative research methods, as most research is text-based, focussing on existing literature and the factual and opinionated (unbiased) information contained within them. However, some instances require for quantitative methods, as there are statistical findings in this report, including graphs and tables depicting various scenarios.

Conducting only secondary research has many benefits, especially given the title of the project and the expected outcome of the research. Secondary research is preferred simply for its convenience, as little time is required to carry out secondary data collection in comparison to primary data collection (questionnaires, interviews, etc.). Although, despite

these benefits, there are also some limitations when conducting secondary research, which will be discussed in greater detail in the following sections.

### 3.5 - Ethical issues

There is an abundance of potential ethical issues that one can run into whilst conducting secondary data collection, those of which need to be thought about during the research process. The moral implications of using somebody else's work, either without permission or without accrediting their findings, are plentiful.

For this task, all the literature referenced has been readily available in the public domain. This has either been through the universities library facilities, both physical and online, or has been various other sources on the internet, including journal articles, conference proceedings and newspaper articles. This can be seen as one of the major implications of secondary data collection, as it is crucial that the authors of any literature used to enhance the quality of your research is properly referenced and credit is given to the owner.

---

*"If the data is freely available on the Internet, books or other public forum, permission for further use and analysis is implied. However, the ownership of the original data must be acknowledged. If the research is part of another research project and the data is not freely available, except to the original research team, explicit, written permission for the use of the data must be obtained from the research team and included in the application for ethical clearance." (Tripathy, 2013)*

---

Ethics are a big part of any human being, as they help shape us into the people we are and help to model the society that we live in. Traits like being able to distinguish between right and wrong, being polite to other human beings and just generally being a good person are the things that have been taught to us through school, at home or wherever it may have been. We all have an ethical 'standard', if you will, that allows us to live in relative harmony.

It is important to transfer these ethical 'skills' into research, as they aid in producing work to a good standard and help 'promote the aims of research' (Resnik, 2015), such as genuine knowledge, truthful and error-free work. A good example of this is with the strict constraints that students are faced with when conducting research, such as copyright protection, falsifying or misrepresenting information. They help to support the notion of producing truthful, ethical pieces of work.

### 3.6 - Limitations

The challenges that one faces when carrying out not just secondary research, but primary research and research can be difficult and thusly need to be considered in detail. This project is based upon secondary data collection, therefore there are a few key limitations that present themselves as obstacles within the scope of the project. These are some of the main limitations that are faced when conducting secondary research for this project:

- ❖ Accuracy of information – reputable sources
- ❖ Non-timely – information could be ‘out-of-date’
- ❖ Incomplete research – free preview, expensive full text
- ❖ Relevance – not specific to researchers needs

The accuracy of information can ultimately affect the quality of your findings; therefore, a lot of time should be spent finding current literature from a reliable source. Reliable sources can include, but are not limited to, thesis and dissertations, journal articles, newspaper articles and literature from online sources. Consistently finding information from a reputable source can be challenging at best, as there may be a shortage of information available surrounding the subject area that is being researched.

Then, there is the issue of information being ‘out-of-date’, as it may have been written far in the past and could therefore be deemed as irrelevant today. This is something that needs to be greatly considered when using any past literature in your research, especially in an industry that is constantly changing. This was found to be a large obstacle when conducting research, especially so when considering specific areas such as the development of the JavaScript language, as there were minimal modern articles supporting this theory.

Whilst finding information that is too old to utilise is a pressing issue, there is also the setback of incomplete information. This could be if you were to find an article that meets the requirements of your project, but when you continue reading, you discover that it is only a ‘preview’ and you must pay expensive amounts to have access to the full texts. Whilst there is a vast expanse of literature available in the universities library, free of charge, cost-effectiveness continues to cause problems when conducting research into specific areas. The importance or usefulness of the research will need to be evaluated to decide whether the extra expenditure is worth it or not.

Finally, there is the issue of finding information that is relevant. Much too often, researchers will stray from their set targets and objectives by using research that is irrelevant to the question they are answering. Therefore, it is important that a research stays on track on only makes use of literature that is specifically relative to their topic area.

Limitations are faced by any researcher undertaking a project, but it is not the problems at hand that is important, it is how they are dealt with. To overcome these matters, it is important to ensure that academic tools are utilised when searching for literature in relation to your research. For example, UWIC has a search engine called MetSearch, which not only searches for academic literature contained within the university library, but also any other literature that they have access to outside of the university, most of which can be accessed online for free. It has also been of great help utilising tools such as Google Scholar and the SCOPUS database, as all have options that allow you to filter the search results to meet your requirements, e.g. year of publication, type of literature and even the authors of the literature.

As for searching for articles online using a traditional search engine, it is quickly apparent that there is a vast array of articles and blog posts in relation to the search query. Whilst it is easy to click on the first available link, read and absorb the included information, it is important to check if the article has referred to any existing literature that could be considered reliable. The likelihood is, if no references have been made then the literature could be heavily biased or opinionated.

## 4 - Discussion and Testing

### 4.1 - Brief overview of discussion topics

In this section of the dissertation, a discussion of the literature review and previous research will be conducted, with an in-depth analysis of the previous workings and the results that have been yielded as a result. As mentioned in the literature review, tests will be undertaken to establish, from the researcher's perspective, the effects of disabling JavaScript in the browser and the first-hand implications of this. It will then go on to discuss why disabling JavaScript in the browser is a good or bad idea, whilst also drawing a conclusion on whether it should be switched off whilst browsing the web. Additionally, this section will be seeking to reveal the true impact that JavaScript has had on the modern web browser and how it has changed the way in which both developers create applications and how the consumers interact with the internet.

Finally, it will look to discuss and predict the future of the JavaScript language as a standard across all web browsers. Topics will include the relevance of the language currently and how this could change within a 10-year period, whilst also looking at potential competition for the JavaScript language, from languages such as Python and Ruby.

### 4.2 - Testing – Disabling JavaScript in the browser

#### 4.2.1 - Brief overview of testing

For this section of the report, tests are going to be carried out to analyse the effects of disabling JavaScript in the web browser. The purpose of these tests is to gain further insight into the importance of JavaScript when browsing the internet and how the web would look without it.

To gauge the effects in greater detail, the use of shortlist of popular web pages and applications will be visited, both with JavaScript enabled and disabled respectively. Screenshots of the loaded web pages will be taken and presented in the report, each with a brief description of the scenario surrounding the screenshot, the expected results and the actual results. Page loading times will also be noted, which will then be presented in a table format for comparison. In addition, certain features of each website or web application will be tested to see how they perform, or if they perform, with JavaScript disabled.

The browser that will be used for testing is the Google Chrome browser. The reason that this browser has been chosen to carry out these tests is because it is one of the fastest web browsers available, which will aid in providing the most accurate page loading times. In addition, most modern websites and applications are designed to work in Chrome, Firefox

and Safari as a preferred standard, with browsers like Opera and the new Microsoft Edge (essentially Internet Explorer with a new name and a haircut) are secondary.

So that the most accurate result can be yielded from these tests, it has been decided that a mixture of web applications and web pages will be tested, with at least one web application being dependent on JavaScript. The list of chosen web pages and applications is as follows:

- ❖ Stuff.tv – a popular tech and gadget magazine, which is available in both print and online formats
- ❖ Google Docs – a web application for editing and storing documents on the web
- ❖ Facebook – the world’s largest social media platform, with over 1.87 billion users worldwide (Smart Insights, 2017)
- ❖ Mozilla – the closest thing to JavaScript’s own webpage, providing JavaScript information and docs

The reason that these sites have been chosen is because they are web locations that the researcher does not visit regularly, if at all, therefore removing any bias or unjust descriptions of their performance under testing conditions. Furthermore, these are sites that are very popular with the general population of web users, which is especially true with the likes of Google Docs and Facebook.

#### 4.2.2 - Results of testing

##### *Tables – loading times*

	Stuff.tv	Google Docs	Facebook	Mozilla
<b>Loading time (seconds)</b>	▪ 3.31	▪ 1.68	▪ 0.55	▪ 1.74
	▪ 3.87	▪ 1.71	▪ 0.56	▪ 1.63
	▪ 3.15	▪ 1.72	▪ 0.71	▪ 1.68

Figure 10: Table showing loading times for all four web locations with JavaScript enabled in the browser

	Stuff.tv	Google Docs	Facebook	Mozilla
<b>Loading time (seconds)</b>	▪ 1.57	▪ 0.76	▪ 0.96	▪ 1.69
	▪ 0.56	▪ -	▪ 0.51	▪ 1.40
	▪ 0.44	▪ -	▪ 0.50	▪ 1.49

Figure 11: Table showing loading times for all four web locations with JavaScript disabled in the browser

##### *Screenshots*

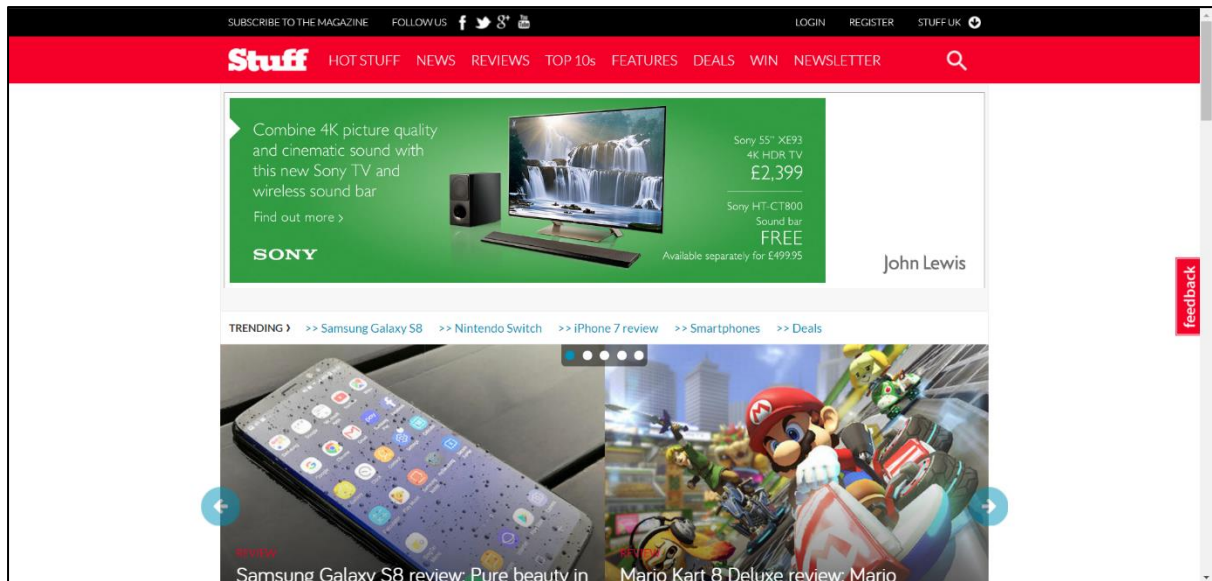


Figure 12: Screenshot showing fully loaded webpage (<http://stuff.tv>) with JavaScript enabled in the browser (Stuff.tv, 2017)

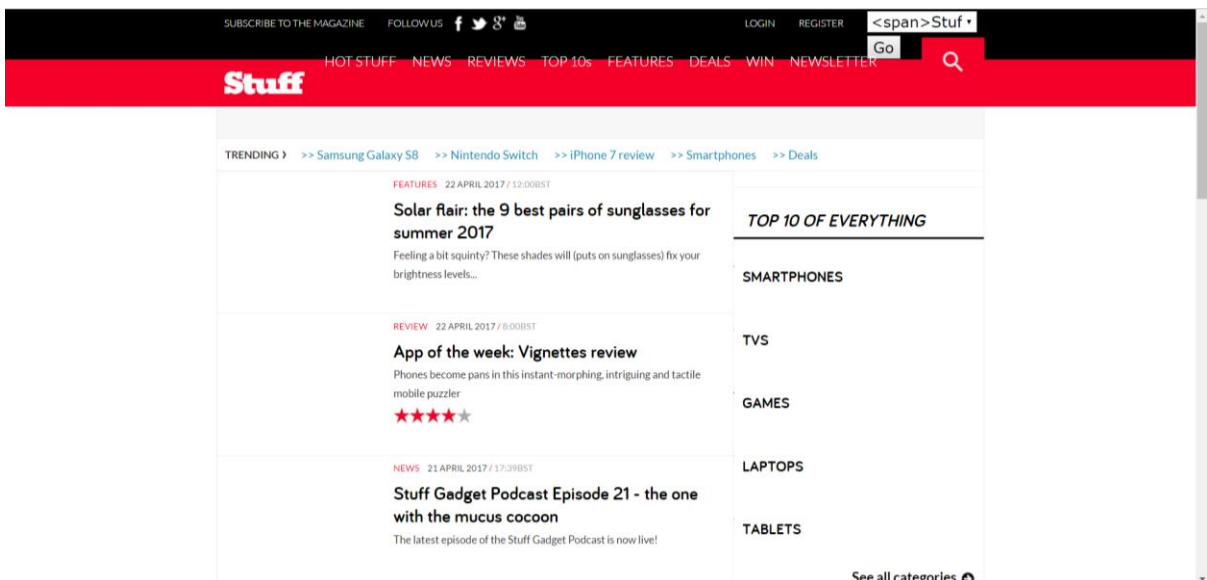


Figure 13: Screenshot showing fully loaded webpage (<http://stuff.tv>) with JavaScript disabled in the browser (Stuff.tv, 2017)

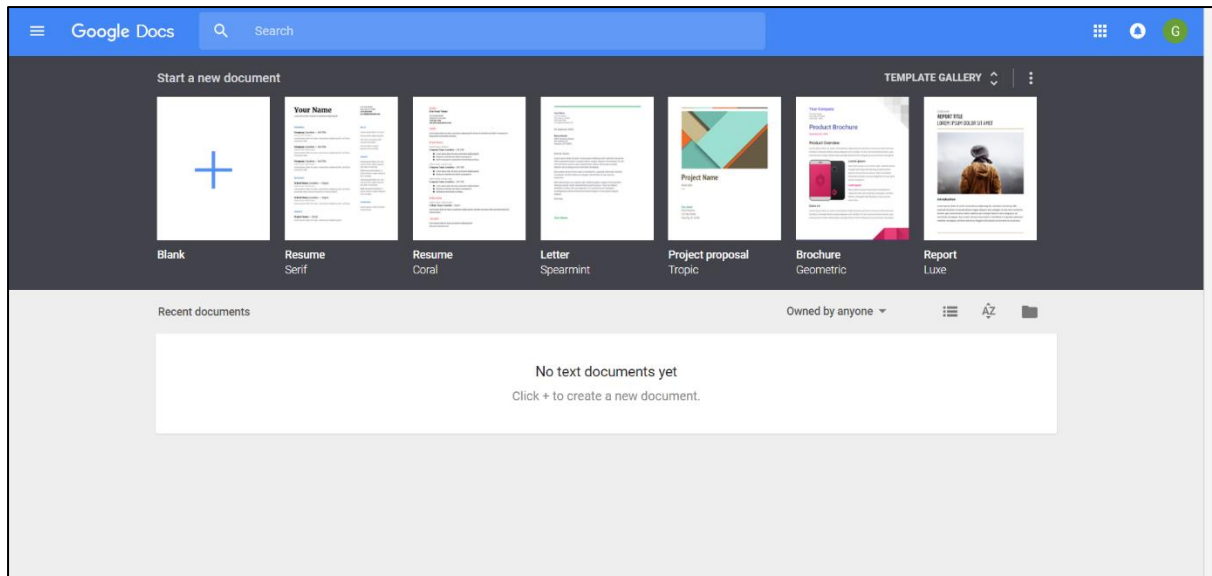


Figure 14: Screenshot showing fully loaded webpage (<http://docs.google.com>) with JavaScript enabled in the browser (Google, 2017)

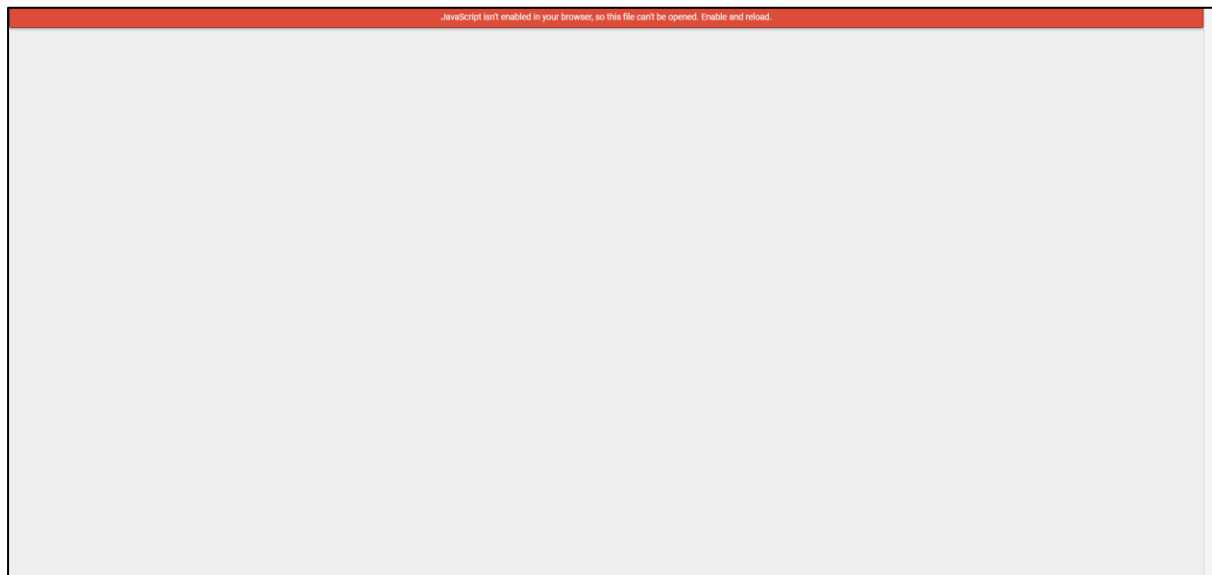


Figure 15: Screenshot showing fully loaded webpage (<http://docs.google.com>) with JavaScript disabled in the browser (Google, 2017)

## 4.3 - Discussion

### 4.3.1 - The Future of JavaScript

As proven in testing and discussion of research findings and literature, it is not hard for one to see that JavaScript is a huge part of the way in which we browse the web. The tests that were carried out in the browser, showing a selection of websites both with JavaScript enabled and disabled respectively, have proven that a web without JavaScript can be very dysfunctional. So much so in fact, that certain web applications (in this case, Google Docs), do not work at all without JavaScript being enabled in the browser. This is because JavaScript serves as the foundation of millions of websites and applications across the internet.

Whilst we know that disabling JavaScript in the browser does carry certain benefits, such as increased security and less advertisements, it doesn't appear to be worth disabling completely yet. In figure 5, we can see the popular tech bog Stuff.tv, fully loaded in the web browser with JavaScript enabled. Everything looks as it should, images have loaded advertisements have loaded. However, in figure 6, we see the exact same webpage, although this time JavaScript has been disabled in the browser. There are many key elements to consider here:

- ❖ No images have loaded on the webpage
- ❖ Slideshow, present in figure 5, is no longer present
- ❖ Advertisement, also present in figure 5, is no longer present

The removal of JavaScript has, in this instance, removed an advertisement from the webpage, further supporting the theory of using the deletion of JavaScript as an ad-blocking tool. Additionally, disabling JavaScript has dramatically lowered the functionality of the web page in general. The main slideshow, which contains the most recent and most popular articles, has not been rendered. This in turn influences the browsing experience, as the user is not able to see the same content as one that is browsing with JavaScript enabled. No thumbnail images have loaded for the individual articles, again hindering the experience. Further evidence of this can be seen in figures 9 and 10 (appendix), where it is apparent that no images at all are being rendered with JavaScript disabled.

In some more extreme circumstances, the testing continues to evaluate the performance of Google Docs, a popular online document editing application. As expected, the application performs normally with JavaScript enabled, with no drawbacks or errors (Figure 14). When JavaScript is disabled in the browser, the application does not load at all and presents the user with an error message, stating:

---

*"JavaScript isn't enabled in your browser, so this file can't be opened.  
Enable and reload." (Figure 15)*

---

As the Google Docs web application is based heavily on JavaScript to support the user interface, Google simply does not have an alternative. JavaScript is absolutely essential for this particular web application to work, again supporting the theory that JavaScript's position as a web development tool is not in jeopardy. The language does have competition – as covered in literature review section, however it seems that JavaScript being replaced as a standard for all web browsers is simply not going to happen at this stage in its life. Too many developers, along with the webpages and applications that they are producing, rely dominantly on JavaScript as the structural foundation of their work. Without it, things don't work correctly, or in some cases just don't work at all.

Statistics would also argue that the future of the JavaScript language within web development is secured for the foreseeable future. Reflecting on the literature review, the amount of people that are utilising the JavaScript language, both past and present, is



investigated. Stackoverflow, a question and answer forum for software engineers from many different industries, is an excellent way to gauge the number of developers that are actively engaging with a programming language. This is made possible by being able to view the number of 'tags' that are being attached to posts on the forum, allowing us to gain some insight into the most popular and most topical languages at present. This information is then archived by websites like tagtrends.com, which uses a line graph to show the increase and/or decrease of the use of any one tag.

From the results in the archive on tagtrends.com, we can see that in 2012, the percentage of JavaScript tags attached to posts on Stackoverflow was at 4.9% of the total. In January 2017, the percentage has risen dramatically to 11% of the total tags. However, whilst the numbers are still promising, the usage of the tag was at its highest in July 2015, as shown in figure (figure), therefore showing a slight decline in the number of users discussing the topic.

Finally, one of the main aspects to consider when discussing the impact and future relevance of the JavaScript language are its many security issues, most of which are easily exploitable. As covered in the lit review, we can see the varying types of exploits that can be used against web applications that utilise JavaScript, such as Cross-Site Scripting and Cross-Site Request Forgery. These should present themselves as important and challenging issues for web developers, who should strive to ensure any JavaScript functionality in their applications is appropriately validated and thus, secure.

To support the future development and improvement of the JavaScript language, developers will need to successfully take hold of any opportunity to secure their scripting. This can be done by using input validation in their applications to protect from XSS attacks, or blocking the use of GET requests, allowing only POST requests. Developers should also seek to use escape characters when writing their code, as this can help to scramble malicious scripts that are input into the application.

#### 4.3.2 - Predictions

If one were even attempt to answer the question of JavaScript's future on the World Wide Web, the answer would have to be that there seems to be no sign of it departing from the web browser anytime soon. Based upon the information that has been unearthed through the research within this dissertation, along with the testing of web locations with JavaScript disabled, it is hard to see how the JavaScript language is set to be replaced anytime soon.

As quoted by Douglas Crockford, as shown in the literature review, a programming language is just like cat. It is easier to get a new one than to try and fix the old one. In the software engineering industry, with new languages, frameworks and libraries sprouting up constantly, this is a very accurate statement. For a developer to stay relevant within the software industry, they should always strive to keep up-to-date with the latest advancements in their language of choice, as well as being open to change in the sense of learning new languages and frameworks when required. That being said, it is important to note how much of a drastic effect has been measured from simply disabling JavaScript in the web browser. If change is imminent, and there is another standardised, widely adopted scripting language

lurking in the shadows, the amount of effort required by the developers of not just web browsers, but of the websites and applications that millions of people frequent on a daily basis would be ginormous. It would mean they would have to essentially redesign their projects from the ground up, to accommodate new standards and work with new languages.

Another important subject to discuss in more detail here is security of websites and applications that have been developed using JavaScript, as covered in section 2.2 of the literature review. It was discovered whilst conducting research into the security features and potential risks of the JavaScript language that it is easy for malicious hackers and cyber criminals to inject harmful excerpts of code into web applications.

## 5 - Conclusion

### 5.1 - Summary

This dissertation research project has sought to discover the origins of the JavaScript language and its impact on the web development industry. The main aims and objectives of this project were to discover how and why JavaScript was initially developed, how it has progressed and changed overtime and the full impact of the language on the World Wide Web as we know it today. Additionally, the security issues associated with the JavaScript language also needed to be considered, and the findings that show the outlook for the history of the JavaScript language as the most popular client-side scripting language.

In the conclusion section of the dissertation, the final drawings on the information gathered during the research stages will be discussed, as well as the proposed importance of the research project within the web development industry. In addition, recommendations for future research into this subject area will be given, along with final notes, discussing the personal experience of the researcher.

### 5.2 - Conclusions

Looking back at all the research findings discovered as a direct result of this dissertation, coupled with the results from browsing the web with JavaScript disabled, there are several key objectives that have been realised.

Firstly, is the issues surrounding the security of the JavaScript language when utilised in web applications. As hackers possess the ability to 'inject' vulnerable web applications with malicious excerpts of code, therefore changing the way the application behaves, JavaScript remains a risky language to include in your source code. Attacks can be prevented if following the appropriate cleaning and validation procedures, however some sites will always be vulnerable. These issues will need to be addressed by the JavaScript development community, as threats such as this seem to be turning people away from using and wanting to learn the JavaScript language. This is evident as there has been an increase in the number of web users that are browsing with JavaScript disabled. Concurrently, within the community of users on Stackoverflow, the most desired language to learn is currently Python, which serves as large competition for the JavaScript language.

It is also fair to conclude that the research has confirmed the future of the JavaScript language in the not-so-distant future. Clearly, JavaScript has etched itself onto our web browsing experience. The web would not look the same without JavaScript, both metaphorically and literally, as proven in the testing phase. The notion that JavaScript use in the browser should be discontinued has evidently been proven unscrupulous, as over 97% of all web users are actively browsing with JavaScript enabled. It is difficult to call the decision to not maximise a web application for use without JavaScript as unjust, simply because the percentage of users browsing without JavaScript is minimal and has been declining for many years. The numbers set to decrease even further in the future, as the popularisation of frameworks such as React continues to rage.

To conclude on what the research has proven it is clear to see that the JavaScript language is not going to be replaced as of yet. JavaScript is structurally integral to the 'persona' of the modern internet, and the browsing experience without it is unpleasant at best, acting almost as a step back in time. JavaScript is, and will continue to be for the foreseeable future, the language of the web.

### 5.3 - Importance of this research

This research could be considered to the development of the JavaScript language for many reasons. Firstly, the questioning of the security flaws within the JavaScript language should serve as a stark reminder to the development community. These are current and pressing issues in terms of security for the language, as some standardised safety measures must be deployed in order to obtain language-wide protection for all developers. Whilst there are current methods that allow for some protection against the insertion of malicious code, it feels as though there should be some ingrained feature in JavaScript, allow for the prevention of malevolent attacks on source code.

Subsequently, this research project will make good reading for anyone within the industry that is interested to learn about the short comings of the language, as well as an analysis of its impact on web development and predictions regarding its future.

### 5.4 - Recommendations

There is a vast array of area that still require research, which have been discovered because of this research. As a suggestion, there should be more research conducted into the security of JavaScript and the potential implications of the security flaws present in the current releases. The importance of this cannot be stressed enough, as it is in the best interest of both the developer and the end user to keep browser activity secure, both for privacy and personal security reasons.

In addition, it is apparent that there is minimal data and research available regarding the impact of the JavaScript language on the web development industry. For the most part, most of the literature review at section (section) is purely assumptive, using information from various topic areas to form some form of understanding of JavaScript's impact.

## 5.5 - Reflection

This dissertation has been a very challenging project and has served to push the boundaries of academic ability. Not only has it been tedious and time consuming by nature, pushing both mentally and physically, but it has also tested research, analytical and writing skills. It has been thoroughly interesting to learn and discover aspects of the JavaScript language not previously known, as it is a language that is of personal interest and one that is used on a regular basis.

It has been an enlightening experience, providing a sense of confidence about the work that has been carried out, both past and present. It has proven that working under a time limit against set targets is possible. This is an invaluable skill for a graduate student looking to enter into full time employment, as it will help with focussing and managing time effectively within a new job role.

If anything were to be done differently within the scope of this dissertation, it would have to be to spend more time researching the JavaScript language and learning about it prior to starting the project. Having a deeper understanding of JavaScript before carrying out research would have helped in the understanding of various articles and code excerpts.

## References

1. Acunetix. (n.d.). *Defending against CSRF Attacks, XSRF or Sea-Surf*. [online] Available at: <https://www.acunetix.com/websitesecurity/csrf-attacks/> [Accessed 23 Apr. 2017].
2. Acunetix. (n.d.). *What is Cross-site Scripting and How Can You Fix it?*. [online] Available at: <https://www.acunetix.com/websitesecurity/cross-site-scripting/> [Accessed 23 Apr. 2017].
3. Allen, J. (2015). *The Importance of Client-Side JavaScript Security — SitePoint*. [online] SitePoint. Available at: <https://www.sitepoint.com/importance-client-side-javascript-security/> [Accessed 19 Apr. 2017].
4. Assis, R. (2016). *What is XSS? - Cross Site Scripting Vulnerability*. [online] Sucuri Blog. Available at: <https://blog.sucuri.net/2016/04/what-is-an-xss-vulnerability.html> [Accessed 23 Apr. 2017].
5. Bbc.co.uk. (2012). *BBC - WebWise - What are cookies?*. [online] Available at: <http://www.bbc.co.uk/webwise/guides/about-cookies> [Accessed 23 Apr. 2017].
6. Berners-lee, T. (1996). *The World Wide Web: Past, Present and Future*. [online] W3.org. Available at: <https://www.w3.org/People/Berners-Lee/1996/ppf.html> [Accessed 7 Apr. 2017].
7. Boag, P. (2010). *Ecommerce solutions fail their customers*. [online] Boagworld - User Experience Advice. Available at: <https://boagworld.com/accessibility/ecommerce-failure> [Accessed 21 Apr. 2017].
8. Boulton, J. (2014). *100 ideas that changed the web*. 1st ed. London: Laurence King Publishing LTD.
9. Burgess, M. (2016). *JavaScript Frameworks Are Great – Matt Burgess – Medium*. [online] Medium. Available at: <https://medium.com/@mattburgess/javascript-frameworks-are-great-2df4a3f0b24d> [Accessed 23 Apr. 2017].
10. Burgess, M. (2017). *All JavaScript frameworks are terrible – Matt Burgess – Medium*. [online] Medium. Available at: <https://medium.com/@mattburgess/all-javascript-frameworks-are-terrible-e68d8865183e> [Accessed 23 Apr. 2017].
11. Cassim, H. (2013). *Using Frameworks to Build Websites and Web Applications*. [online] OSTRaining. Available at: <https://www.ostraining.com/blog/webdesign/frameworks/> [Accessed 23 Apr. 2017].
12. Cleverism. (2015). *Top Programming Languages used in Web Development*. [online] Available at: <https://www.cleverism.com/programming-languages-web-development/> [Accessed 23 Apr. 2017].
13. Connolly, D. (2000). *A Little History of the World Wide Web*. [online] W3.org. Available at: <https://www.w3.org/History.html> [Accessed 23 Apr. 2017].
14. Crockford, D. (2001). *JavaScript: The World's Most Misunderstood Programming Language*. [online] Crockford.com. Available at: <http://www.crockford.com/javascript/javascript.html> [Accessed 23 Apr. 2017].

15. Crockford, D. (2008). *JavaScript: The World's Most Misunderstood Programming Language Has Become the World's Most Popular Programming Language*. [online] Javascript.crockford.com. Available at: <http://javascript.crockford.com/popular.html> [Accessed 23 Apr. 2017].
16. Da-14.com. (n.d.). *5 Best JavaScript Frameworks in 2017 | DA-14*. [online] Available at: <https://da-14.com/blog/5-best-javascript-frameworks-2017> [Accessed 17 Apr. 2017].
17. Dionach. (2016). *The Real Impact of Cross-Site Scripting*. [online] Available at: <https://www.dionach.com/blog/the-real-impact-of-cross-site-scripting> [Accessed 23 Apr. 2017].
18. Doernhoefer, M. (2006). Surfing the net for software engineering notes. *ACM SIGSOFT Software Engineering Notes*, [online] 31(4), p.8. Available at: [http://delivery.acm.org/10.1145/1150000/1142972/p16-doernhoefer.pdf?ip=193.62.97.110&id=1142972&acc=ACTIVE%20SERVICE&key=BF07A2EE685417C5%2E8720C5823304BAAB%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=741160152&CFTOKEN=26093351&\\_\\_acm\\_\\_=1490013400\\_1d7a1d220a19214a30ae551cb857f6a7](http://delivery.acm.org/10.1145/1150000/1142972/p16-doernhoefer.pdf?ip=193.62.97.110&id=1142972&acc=ACTIVE%20SERVICE&key=BF07A2EE685417C5%2E8720C5823304BAAB%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=741160152&CFTOKEN=26093351&__acm__=1490013400_1d7a1d220a19214a30ae551cb857f6a7).
19. Ec.europa.eu. (2016). *E-commerce statistics for individuals - Statistics Explained*. [online] Available at: [http://ec.europa.eu/eurostat/statistics-explained/index.php/E-commerce\\_statistics\\_for\\_individuals#Data\\_sources\\_and\\_availability](http://ec.europa.eu/eurostat/statistics-explained/index.php/E-commerce_statistics_for_individuals#Data_sources_and_availability) [Accessed 21 Apr. 2017].
20. Ec.europa.eu. (2016). *E-commerce statistics for individuals - Statistics Explained*. [online] Available at: [http://ec.europa.eu/eurostat/statistics-explained/index.php/E-commerce\\_statistics\\_for\\_individuals#Data\\_sources\\_and\\_availability](http://ec.europa.eu/eurostat/statistics-explained/index.php/E-commerce_statistics_for_individuals#Data_sources_and_availability) [Accessed 23 Apr. 2017].
21. Finley, K. (2015). *I Turned Off JavaScript for a Whole Week and It Was Glorious*. [online] Wired.com. Available at: <https://www.wired.com/2015/11/i-turned-off-javascript-for-a-whole-week-and-it-was-glorious/> [Accessed 14 Apr. 2017].
22. Fowler, M. (2004). *bliki: Lambda*. [online] martinfowler.com. Available at: <https://martinfowler.com/bliki/Lambda.html> [Accessed 17 Apr. 2017].
23. Fowler, M. (2005). *bliki: InversionOfControl*. [online] martinfowler.com. Available at: <https://martinfowler.com/bliki/InversionOfControl.html> [Accessed 17 Apr. 2017].
24. Glynn, F. (n.d.). *JavaScript Security*. [online] Veracode. Available at: <https://www.veracode.com/security/javascript-security> [Accessed 23 Apr. 2017].
25. Greif, S. (2016). *The State Of JavaScript: Front-End Frameworks – Sacha Greif – Medium*. [online] Medium. Available at: <https://medium.com/@sachagreif/the-state-of-javascript-front-end-frameworks-1a2d8a61510> [Accessed 23 Apr. 2017].
26. Hafiz, M., Hasan, S., King, Z. and Wirfs-Brock, A. (2016). Growing a language: An empirical study on how (and why) developers use some recently-introduced and/or recently-evolving JavaScript features. *Journal of Systems and Software*, [online] 121, pp.191-208. Available at: <http://ac.els-cdn.com.ezproxy.cardiffmet.ac.uk/S0164121216300309/1-s2.0->

S0164121216300309-main.pdf?\_tid=51110d58-0d6f-11e7-90f6-00000aacb35f&acdnat=1490015921\_e5ed4d65404644ab14c1995ff1f4bad7.

27. Hauben, M. (2007). *History of ARPANET*. 1st ed. [ebook] Available at: <http://www.dei.isep.ipp.pt/docs/arpa.html> [Accessed 4 Apr. 2017].
28. Hedenborg, M., Hagelbäck, J. and Ladan, Z. (2015). *Comparing performance between plain JavaScript and popular JavaScript frameworks*. Uppsala University.
29. hello JS. (2017). *2017 Stack Overflow Survey — JavaScript data – hello JS*. [online] Available at: <https://blog.hellojs.org/2017-stack-overflow-survey-javascript-data-e68275477a8c> [Accessed 23 Apr. 2017].
30. Help.github.com. (n.d.). *Fork A Repo - User Documentation*. [online] Available at: <https://help.github.com/articles/fork-a-repo/> [Accessed 20 Apr. 2017].
31. Hoffman, J. (2013). *Should You Disable JavaScript?*. [online] Howtogeek.com. Available at: <https://www.howtogeek.com/138865/htg-explains-should-you-disable-javascript/> [Accessed 20 Apr. 2017].
32. Home.cern. (n.d.). *The birth of the web | CERN*. [online] Available at: <https://home.cern/topics/birth-web> [Accessed 7 Apr. 2017].
33. iadt.edu. (2013). *The Pros and Cons of JavaScript: Is it Still Necessary? - IADT*. [online] Available at: <http://www.iadt.edu/student-life/iadt-buzz/december-2013/the-pros-and-cons-of-javascript-is-it-still-necessary> [Accessed 23 Apr. 2017].
34. Ibm.com. (2017). *Which Are The Best Programming Languages to Learn in 2017? - Tech for Life Blog - Tech for Life*. [online] Available at: [https://www.ibm.com/developerworks/community/blogs/d13a8e32-0870-49cb-9b0d-ba0e34fa6561/entry/Which\\_Are\\_The\\_Best\\_Programming\\_Languages\\_to\\_Learn\\_in\\_2017?lang=en](https://www.ibm.com/developerworks/community/blogs/d13a8e32-0870-49cb-9b0d-ba0e34fa6561/entry/Which_Are_The_Best_Programming_Languages_to_Learn_in_2017?lang=en) [Accessed 17 Apr. 2017].
35. International, E. (n.d.). *What is Ecma International*. [online] Ecma-international.org. Available at: <http://www.ecma-international.org/flat/memento/index.html> [Accessed 13 Apr. 2017].
36. Johnson, R. and Foote, B. (1988). *Designing Reusable Classes*. [online] Laputan.org. Available at: <http://www.laputan.org/drc/drc.html> [Accessed 17 Apr. 2017].
37. Johnston, M. (2014). *Secondary Data Analysis: A Method of which the Time Has Come*. PhD. University of Alabama.
38. Keefer, C. (2015). *The What and Why of Javascript Frameworks - Art & Logic: Custom Software Development*. [online] Art & Logic: Custom Software Development. Available at: <https://artandlogic.com/2015/05/the-what-and-why-of-javascript-frameworks/> [Accessed 17 Apr. 2017].
39. Knowthis.com. (n.d.). *Secondary Research - Disadvantages*. [online] Available at: <http://www.knowthis.com/data-collection-low-cost-secondary-research/secondary-research-disadvantages> [Accessed 23 Apr. 2017].

40. Kothari, C. (2004). *Research Methodology: Methods and Techniques*. 3rd ed. New Delhi: New Age Publishers, pp.10-20.
41. Krill, P. (2008). *JavaScript creator ponders past, future*. [online] InfoWorld. Available at: <http://www.infoworld.com/article/2653798/application-development/javascript-creator-ponders-past--future.html> [Accessed 12 Apr. 2017].
42. Kumar, R. (2011). *Research Methodology: A Step-by-Step Guide for Beginners*. 3rd ed. London: SAGE Publications.
43. Kumar, S. (2015). *Difference Between Library and Framework*. [online] C-sharpcorner.com. Available at: <http://www.c-sharpcorner.com/UploadFile/a85b23/framework-vs-library/> [Accessed 23 Apr. 2017].
44. Lee, J. (2014). *3 Ways JavaScript Can Breach Your Privacy & Security*. [online] MakeUseOf. Available at: <http://www.makeuseof.com/tag/3-ways-javascript-can-used-breach-privacy-security/> [Accessed 23 Apr. 2017].
45. Leiner, B., Cerf, V., Clark, D., Kahn, R., Kleinrock, L., Lynch, D., Postel, J. and Roberts, D. (n.d.). *Brief History of the Internet - Internet Timeline | Internet Society*. [online] InternetSociety.org. Available at: <https://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet#concepts> [Accessed 23 Apr. 2017].
46. Medium. (2016). *The State Of JavaScript: Front-End Frameworks – Sacha Greif – Medium*. [online] Available at: <https://medium.com/@sachagreif/the-state-of-javascript-front-end-frameworks-1a2d8a61510> [Accessed 14 Apr. 2017].
47. Owasp.org. (n.d.). *Cross-Site Request Forgery (CSRF) - OWASP*. [online] Available at: [https://www.owasp.org/index.php/Cross-Site\\_Request\\_Forgery\\_\(CSRF\)](https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)) [Accessed 23 Apr. 2017].
48. Paudyal, U., Borovic, C., Eriksson, O. and Jansson, A. (2011). *Scalable Web Application using Node.JS and CouchDB*. Undergraduate. Uppsala University.
49. Research Methodology. (n.d.). *Deductive Approach (Deductive Reasoning) - Research Methodology*. [online] Available at: <http://research-methodology.net/research-methodology/research-approach/deductive-approach-2/> [Accessed 24 Apr. 2017].
50. Research Methodology. (n.d.). *Research Philosophy - Research Methodology*. [online] Available at: <http://research-methodology.net/research-philosophy/> [Accessed 23 Apr. 2017].
51. Resnik, D. (2015). *What is Ethics in Research & Why is it Important?*. [online] National Institute of Environmental Health Services. Available at: <https://www.niehs.nih.gov/research/resources/bioethics/whatis/> [Accessed 21 Apr. 2017].
52. Richards, G., Lebresne, S., Burg, B. and Vitek, J. (2010). An analysis of the dynamic behavior of JavaScript programs. In: *ACM SIGPLAN Conference*.



53. Severance, C. (2012). JavaScript: Designing a Language in 10 Days. *Computer*, [online] 45(2), pp.7-8. Available at:  
<https://www.computer.org/csdl/mags/co/2012/02/mco2012020007.html>.
54. Silver, T. (2010). *WordPress 3.0 JQuery: Enhance Your WordPress Website with the Captivating Effects of JQuery*. 1st ed. Packt Publishing.
55. Smart Insights. (2017). *Global Social Media Statistics Summary 2017*. [online] Available at: <http://www.smartinsights.com/social-media-marketing/social-media-strategy/new-global-social-media-research/> [Accessed 22 Apr. 2017].
56. Sotagtrends.com. (2017). *Tag Trends*. [online] Available at:  
[http://sotagtrends.com/?tags=\[javascript\]](http://sotagtrends.com/?tags=[javascript]) [Accessed 14 Apr. 2017].
57. Stackoverflow.com. (2017). *Tags*. [online] Available at:  
<https://stackoverflow.com/tags?tab=popular> [Accessed 14 Apr. 2017].
58. Stark, J. (2010). *Building iPhone Apps with HTML, CSS, and JavaScript*. 1st ed. Beijing: Farnham : O'Reilly.
59. Telerik (2016). *The Future of JavaScript - 2016 and Beyond*. [online] Telerik, pp.11-12. Available at: <http://d585tldpucybw.cloudfront.net/docs/default-source/whitepapers/the-future-of-javascript-2016-and-beyond.pdf?sfvrsn=0> [Accessed 23 Apr. 2017].
60. Tripathy, J. (2013). *Secondary Data Analysis: Ethical Issues and Challenges*. [online] PubMed Central (PMC). Available at:  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4441947/> [Accessed 23 Apr. 2017].
61. Veracode. (n.d.). *Cross-Site Scripting (XSS) Tutorial: Learn About XSS Vulnerabilities, Injections and How to Prevent Attacks*. [online] Available at:  
<https://www.veracode.com/security/xss> [Accessed 23 Apr. 2017].
62. W3.org. (n.d.). *A Short History of JavaScript - Web Education Community Group*. [online] Available at:  
[https://www.w3.org/community/webed/wiki/A\\_Short\\_History\\_of\\_JavaScript](https://www.w3.org/community/webed/wiki/A_Short_History_of_JavaScript) [Accessed 20 Mar. 2017].
63. Ward, R. and Smith, M. (1998). JavaScript as a first programming language for multimedia students. *ACM SIGCSE Bulletin*, 30(3), pp.249-253.
64. Webaim.org. (2009). *WebAIM: Screen Reader User Survey #2 Results*. [online] Available at: <http://webaim.org/projects/screenreadersurvey2/> [Accessed 23 Apr. 2017].
65. Webaim.org. (2015). *WebAIM: Screen Reader User Survey #5 Results*. [online] Available at: <http://webaim.org/projects/screenreadersurvey5/> [Accessed 23 Apr. 2017].
66. Williams, R. (2015). *Web browsers: a brief history*. [online] Telegraph.co.uk. Available at: <http://www.telegraph.co.uk/technology/microsoft/11577364/Web-browsers-a-brief-history.html> [Accessed 23 Apr. 2017].

67. Wyke-Smith, C. (2010). *Scriptin' with JavaScript and Ajax*. 1st ed. Berkeley, CA: New Riders.

## Appendix

### Ethics Application Form

When undertaking a research or enterprise project, Cardiff Met staff and students are obliged to complete this form in order that the ethics implications of that project may be considered.

**If the project requires ethics approval from an external agency (e.g., NHS),** you will not need to seek additional ethics approval from Cardiff Met. You should however complete Part One of this form and attach a copy of your ethics letter(s) of approval in order that your School has a record of the project.

The document ***Ethics application guidance notes*** will help you complete this form. It is available from the [Cardiff Met website](#). The School or Unit in which you are based may also have produced some guidance documents, please consult your supervisor or School Ethics Coordinator.

Once you have completed the form, sign the declaration and forward to the appropriate person(s) in your School or Unit.

#### PLEASE NOTE:

**Participant recruitment or data collection MUST NOT commence until ethics approval has been obtained.**

#### PART ONE

Name of applicant:	Thomas James Henson
Supervisor (if student project):	Dr. Esyin Chew
School / Unit:	Cardiff School of Management
Student number (if applicable):	ST20060907
Programme enrolled on (if applicable):	BSc (Hons) Software Engineering
Project Title:	JavaScript: an analysis of the programming language and its impact on web development.
Expected start date of data collection:	18/03/2017
Approximate duration of data collection:	1 month
Funding Body (if applicable):	N/A
Other researcher(s) working on the project:	None
Will the study involve NHS patients or staff?	No
Will the study involve human samples and/or human cell lines?	No

Does your project fall entirely within one of the following categories:	
Paper based, involving only documents in the public domain	No
Laboratory based, not involving human participants or human samples	No

Practice based not involving human participants (eg curatorial, practice audit)	No
Compulsory projects in professional practice (eg Initial Teacher Education)	No
A project for which external approval has been obtained (e.g., NHS)	No
If you have answered YES to any of these questions, expand on your answer in the non-technical summary. No further information regarding your project is required. If you have answered NO to all of these questions, you must complete Part 2 of this form	

In no more than 150 words, give a non-technical summary of the project
An in-depth analysis of the JavaScript language and the impact that it has had on web development. I will be looking at the origins of JavaScript, including when the language was created and why it was developed. In addition, I will conduct research into how JavaScript is used in current applications and why it is considered one of the most popular programming languages, along with its many frameworks and libraries. To finalise, I will be discussing how the language has evolved over a ten-year period and if it will still be relevant in the near future.

<b>DECLARATION:</b> I confirm that this project conforms with the Cardiff Met Research Governance Framework	
I confirm that I will abide by the Cardiff Met requirements regarding confidentiality and anonymity when conducting this project.	
<b>STUDENTS:</b> I confirm that I will not disclose any information about this project without the prior approval of my supervisor.	
Signature of the applicant: Thomas J. Henson	Date: 16/03/2017
<b>FOR STUDENT PROJECTS ONLY</b>	
Name of supervisor: Dr Eysin Chew	Date: 17 March 2017
Signature of supervisor: <i>Eysin Chew</i>	

<b>Research Ethics Committee use only</b>	
Decision reached:	Project approved <input checked="" type="checkbox"/> Project approved in principle <input type="checkbox"/> Decision deferred <input type="checkbox"/> Project not approved <input type="checkbox"/> Project rejected <input type="checkbox"/>
Project reference number: <b>2016D04792</b>	

Name: Dr Hilary Berger	Date: 18/03/2017
Signature: Hilary Berger	
Details of any conditions upon which approval is dependant: None	
<b>A-RESEARCH DESIGN</b>	
<del>A1 Will you be using an approved protocol in your project?</del>	<del>No</del>
<del>A2 If yes, please state the name and code of the approved protocol to be used<sup>‡</sup></del>	
<del>A3 Describe the research design to be used in your project</del>	
<del>A4 Will the project involve deceptive or covert research?</del>	
<del>A5 If yes, give a rationale for the use of deceptive or covert research</del>	
<del>A6 Will the project have security sensitive implications?</del>	
<del>A7 If yes, please explain what they are and the measures that are proposed to address them</del>	

<b>B-PREVIOUS EXPERIENCE</b>
<del>B1 What previous experience of research involving human participants relevant to this project do you have?</del>
<del>B2 Student project only</del> <del>What previous experience of research involving human participants relevant to this project does your supervisor have?</del>

<b>C-POTENTIAL RISKS</b>
<del>C1 What potential risks do you foresee?</del>
<del>C2 How will you deal with the potential risks?</del>

~~When submitting your application you **MUST** attach a copy of the following:~~

- ~~• All information sheets~~
- ~~• Consent/assent form(s)~~

~~An exemplar information sheet and participant consent form are available from the Research section of the Cardiff Met website.~~

<sup>‡</sup> An Approved Protocol is one which has been approved by Cardiff Met to be used under supervision of designated members of staff; a list of approved protocols can be found on the Cardiff Met website here