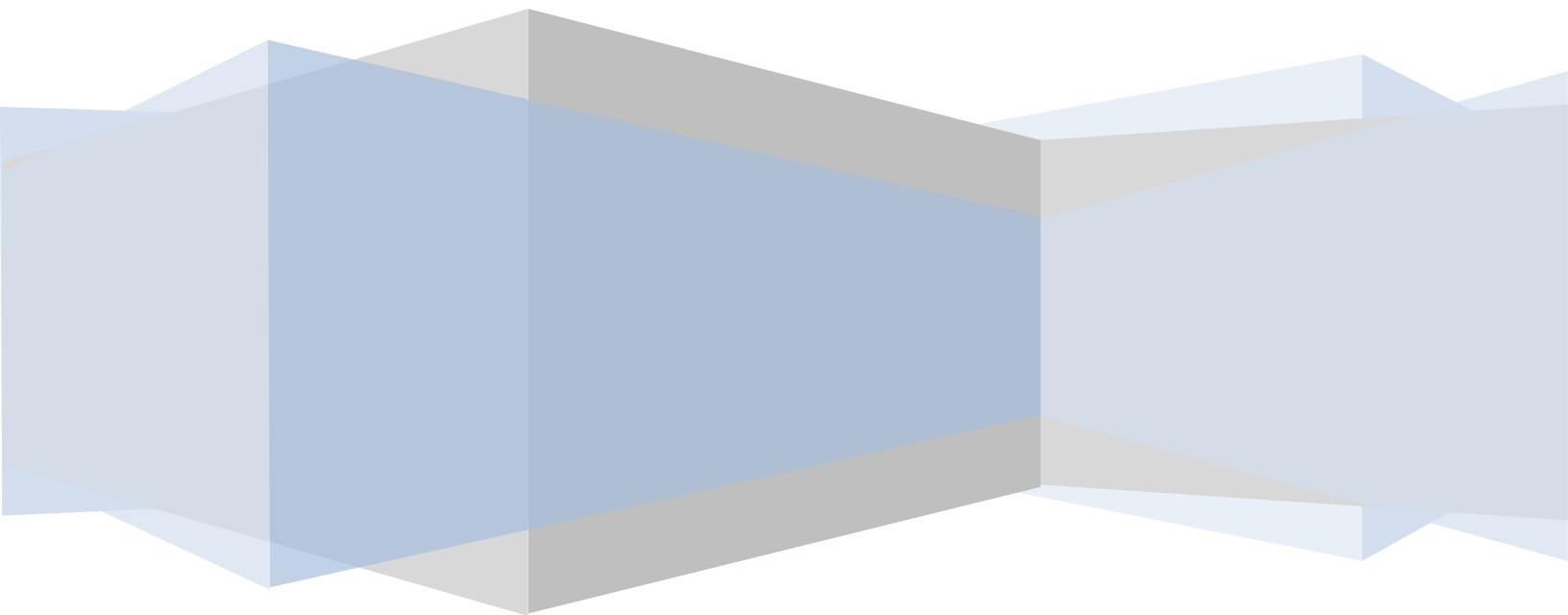


# JavaScript Essentials



JavaScript Essentials – First Edition

© 2009 Techotopia.com. This eBook is provided for personal use only. Unauthorized use, reproduction and/or distribution strictly prohibited. All rights reserved.

## Table of Contents

Chapter 1. About this Book.....	9
1.1 Intended Audience .....	10
Chapter 2. The History of JavaScript.....	11
2.1 Jim and Marc Meet in Palo Alto .....	11
2.2 Moving on from Static Web Pages .....	11
2.3 LiveScript, JavaScript, JScript, ECMAScript - What's in a Name? .....	12
Chapter 3. A Simple JavaScript Example.....	13
3.1 A First JavaScript Example.....	13
Chapter 4. Embedding JavaScript into Web pages.....	15
4.1 The <script> element .....	15
4.2 Where to place JavaScript in a Web Page.....	16
Chapter 5. Introducing JavaScript Variables.....	17
5.1 Variable Declaration and the Assignment Operator.....	17
5.2 JavaScript Variable Naming Conventions.....	18
Chapter 6. JavaScript Variable Types.....	19
6.1 JavaScript Variables.....	19
6.2 The JavaScript <i>typeof</i> Operator.....	20
Chapter 7. JavaScript Operators.....	21
7.1 Assignment Operators.....	21
7.2 Arithmetic Operators .....	22
7.2.1 Unary Operators - Incrementing and Decrementing Variable Values.....	23
7.3 Comparison Operators.....	24
7.4 String Operators .....	24
7.5 Conditional Operators.....	26
7.6 Logical Operators .....	27
7.7 Operator Precedence .....	27
Chapter 8. Comments in JavaScript.....	29

8.1	Single Line Comments .....	29
8.2	Multi-line Comments .....	30
8.3	Summary .....	31
Chapter 9.	JavaScript Flow Control and Looping.....	32
9.1	JavaScript Conditional Statements.....	32
9.1.1	The JavaScript <i>if</i> Statement .....	32
9.1.2	The JavaScript <i>if ... else</i> Statements.....	33
9.2	JavaScript Looping Statements .....	34
9.2.1	JavaScript <i>for</i> loops .....	34
9.3	JavaScript <i>while</i> loops .....	36
9.3.1	JavaScript <i>do ... while</i> loops .....	37
9.4	JavaScript <i>switch</i> Statements.....	38
9.5	Breaking a Loop.....	40
9.5.1	label Statements .....	41
9.6	Skipping Statements in Current Loop Iteration .....	42
Chapter 10.	Understanding JavaScript Functions.....	43
10.1	Declaring and Calling a JavaScript Function.....	43
10.2	Passing Arguments to a Function.....	44
10.3	Returning a Value from a Function .....	45
10.4	Where to Place Function Declarations.....	47
Chapter 11.	JavaScript Object Basics .....	48
11.1	An Overview of JavaScript Objects.....	48
11.2	Creating a Custom JavaScript Object .....	48
11.3	Creating and Using Object Instances .....	49
11.4	Extending Objects.....	51
11.5	Summary .....	51
Chapter 12.	JavaScript String Object .....	52
12.1	Creating a JavaScript String Object .....	52

12.2	JavaScript String Object Methods .....	53
12.3	JavaScript String Object Properties .....	54
12.4	JavaScript String Object Examples .....	55
12.4.1	Getting the length of a String Object .....	55
12.4.2	Setting String Object text Effects .....	55
12.4.3	Performing Conversions on String Objects .....	56
12.4.4	Manipulating Strings with the String Object .....	56
12.5	Summary .....	57
Chapter 13.	JavaScript Date Object .....	59
13.1	Understanding System Time .....	59
13.2	Creating a JavaScript Date Object .....	59
13.3	Setting the Time and Date of the JavaScript Date Object .....	60
13.4	Reading the Date and Time from a JavaScript Date Object .....	60
13.4.1	UTC Time .....	61
13.4.2	Finding the Time Zone Offset and Getting Local Time .....	61
Chapter 14.	JavaScript Math Object .....	63
14.1	JavaScript Math Object Methods .....	63
14.2	JavaScript Math Object Properties .....	63
14.3	Using JavaScript Math Object and Properties .....	64
14.4	Rounding and Truncating Numbers with the the Math Object .....	64
Chapter 15.	JavaScript Window Object .....	65
15.1	Referencing the JavaScript window Object .....	65
15.2	JavaScript window Object Properties .....	65
15.3	Opening Browser Windows using JavaScript .....	66
15.4	Closing Browser Windows using JavaScript .....	67
15.5	Moving and Resizing Windows .....	68
15.6	Changing Window Focus .....	69
15.7	Displaying Message Box Dialogs .....	70

Chapter 16. JavaScript Document Object.....	72
16.1 What is the JavaScript Document Object?.....	72
16.2 JavaScript Document Object Methods and Properties.....	72
16.2.1 Document Object Methods .....	73
16.2.2 Document Object Properties .....	73
16.3 Using the JavaScript Document Object.....	73
16.3.1 Writing text to a document .....	74
16.3.2 Writing Text to a document in a different Window .....	74
16.3.3 Changing the Document Title .....	75
16.3.4 Changing the Document Colors .....	76
16.3.5 Getting a List of Objects in a Document .....	77
Chapter 17. JavaScript Location Object.....	79
17.1 Loading a New URL into the Current Window .....	79
17.2 Reading the URL of the Current Window.....	79
17.3 Extracting Parts of the URL.....	79
17.4 JavaScript Location Object Methods.....	79
Chapter 18. JavaScript Arrays.....	81
18.1 What is a JavaScript Array .....	81
18.2 How to Create a JavaScript Array.....	81
18.3 Initializing the Elements of an Array .....	82
18.4 Accessing the Elements of a JavaScript Array .....	83
18.5 JavaScript Array Object Methods and Properties .....	84
18.6 JavaScript Array Sorting .....	84
Chapter 19. JavaScript Timeouts .....	86
19.1 Setting up a JavaScript Timeout.....	86
19.2 Cancelling a JavaScript Timeout.....	86
19.3 Setting up JavaScript Timeout to Repeat.....	87
19.4 Summary .....	87

Chapter 20. Building Forms with JavaScript .....	88
20.1 JavaScript and The Form Object.....	88
20.2 Accessing Objects in a Form.....	88
20.3 JavaScript and the Text Object.....	89
20.4 JavaScript and The TextArea Object.....	91
20.5 JavaScript and The Button Object .....	92
20.6 JavaScript Check Boxes.....	93
20.7 JavaScript and Radio Buttons.....	94
20.8 JavaScript and Drop-down / Select Object.....	95
20.8.1 Obtaining the Current Selection from the JavaScript Select Object.....	97
20.9 JavaScript and the Password Object .....	98
Chapter 21. Understanding JavaScript Cookies.....	99
21.1 What is a JavaScript Cookie? .....	99
21.2 The Structure of a Cookie.....	99
21.2.1 Cookie Name / Value Pair .....	99
21.2.2 Cookie Expiration Setting.....	100
21.2.3 Cookie <i>path</i> Setting.....	100
21.2.4 Cookie <i>domain</i> Setting.....	100
21.3 Creating a Cookie .....	100
21.4 Reading a Cookie .....	101
21.5 Removing a Cookie by Setting the Expiration Date .....	103
Chapter 22. Understanding Cascading Style Sheets (CSS).....	104
22.1 What are Cascading Style Sheets for?.....	104
22.2 Embedding CSS Styles into a Web Page .....	104
22.3 Importing an External Style Sheet into a Web Page .....	104
22.4 Creating CSS Rules.....	105
22.5 Setting Styles for Individual Elements.....	105
22.6 Setting Styles using the <i>id</i> Attribute of an HTML Element.....	106

22.7	Using Classes with CSS Rules.....	106
22.8	Summary .....	106
Chapter 23.	JavaScript and CSS - Dynamic Styles and Layers.....	107
23.1	Making Basic Dynamic Style Changes .....	107
23.2	Hiding and Showing Content Using JavaScript.....	108
23.3	Using Overlaying Layers .....	110
23.4	Making Layers that Move.....	113



## Chapter 1. About this Book

---

Before JavaScript was created, a typical web browsing experience consisted largely of displaying static pages of information. Any type of user interaction with a web page usually involved the browser sending information back to the web server where some type of server based script or other program would validate the user's data input. The server would then send back either a static error page if the user had entered invalid data or, assuming successful validation, return a static page to the web browser containing the results of the transaction. In addition, the lack of a mechanism to implement dynamic web page content and behavior dictated that an entirely new web page be loaded in order to display new information.

It quickly became clear that a way of performing "client-side" or "web browser-side" processing was needed. The original intent for JavaScript, therefore, was to address this need by facilitating such tasks as validating user input before it is transmitted to the web server, and creating dynamic web page content and layouts independently of the web server (for example, displaying different text depending on a user's actions within the page).

JavaScript met and, indeed, exceeded these requirements by providing an extremely flexible and intuitive scripting language. The problem today is that, as with any useful technology, JavaScript has been used in ways never anticipated by its original authors. As a result, the potential uses of JavaScript have grown far beyond the needs of the average person looking to add dynamic behavior to web pages. The result of this is that JavaScript book authors also feel the need to cover every possible capability of JavaScript, often to the extent that the average JavaScript reference book now runs to over 1,000 pages.

The purpose of this book, therefore, is to provide the essentials of JavaScript so that anyone with a web site can use this powerful scripting language quickly and effectively without having to learn things that, whilst within the capabilities of JavaScript, aren't actually needed by the average web designer. That said this book does not take any shortcuts. It provides detailed and in depth knowledge of all aspects of JavaScript from language basics through to advanced topics.

This book is designed to provide a detailed and comprehensive, yet easy to follow, guide to developing powerful, dynamic web pages using JavaScript. It could be said that the object of **JavaScript Essentials** is to provide you with everything you need to use JavaScript, and nothing that you don't.

## 1.1 Intended Audience

This book assumes that the reader has some experience in designing web pages using either HTML or XHTML. Experience using other programming languages is a benefit, but not a requirement when reading this book. Every effort has been made to carefully explain the concepts of JavaScript to make this book accessible to those with little or no programming experience. The early chapters of JavaScript Essentials cover the basics of the JavaScript scripting language with plenty of examples illustrating the topics covered and providing "cut and paste" solutions to common problems. Only once the basics are covered does the book go on to cover the more advanced topics.

It is hoped that after completing this book, the typical reader will be confidently adding dynamic, JavaScript powered content to their web sites.

## Chapter 2. The History of JavaScript

---

To fully understand the history of JavaScript it is first necessary to go back to a time before Internet Explorer and Firefox dominated the web browser market, and before most of us even knew what a web browser was.

### 2.1 Jim and Marc Meet in Palo Alto

In January 1994 Jim Clark, the founder of Silicon Graphics, and Marc Andreessen, a graduate of the University of Illinois, met at Cafe Verona in Palo Alto, California to talk about starting a company. Whilst at the University of Illinois Andreessen had worked on Mosaic - one of the earliest web browsers - and the two soon decided to bring the internet and web browsing to the masses. Many sleepless nights and millions of lines of code later the Netscape browser took the internet by storm, and the company's subsequent stratospheric Initial Public Offering (IPO) triggered what would later be referred to as the dot-com boom.

Sadly neither Netscape nor Cafe Verona ultimately stood the test of time. Under attack from Microsoft, Netscape was purchased by AOL where it failed to flourish. Eventually the underlying Netscape browser technology was spun off into the open source Mozilla organization which later spawned the hugely successful Firefox browser. Cafe Verona also fell victim to the dot com implosion and went out of business a few years ago along with many Silicon Valley restaurants that had relied on hungry and wealthy technology workers for their business.

### 2.2 Moving on from Static Web Pages

Firefox is not the only legacy of Netscape Corporation however. It was at Netscape that the need for an alternative to the static nature of web pages was first identified.

It is important to keep in mind that before JavaScript most web page functionality was provided through the process of a web browser sending a request to a web server, and the web server returning a page of HTML in response to that request. Any change to the information in the web browser window typically required the transmission of another entire page from the web server. In essence the web browser, though typically running on a desktop or laptop computer capable of performing highly complex tasks, was relegated to the task of simply displaying pages of information sent from a server.

Recognizing the need for browser or *client side* intelligence Netscape began work on a scripting language that would enable web designers to embed code into the HTML of web pages to provide features such as client side validation of user input, dynamic display of data and event driven functionality (for example performing local processing when a push button is pressed or data entered into a text field).

### 2.3 LiveScript, JavaScript, JScript, ECMAScript - What's in a Name?

The scripting language developed at Netscape was originally called LiveScript. Perhaps in an effort to bask in the reflected glory of Sun Microsystem's Java programming language, LiveScript was later renamed JavaScript, though in reality it bears no resemblance to Sun's Java.

A disagreement between Netscape and Microsoft over licensing ultimately led Microsoft to reverse engineer JavaScript and create their own version called JScript. Inevitably differences between the two variants of the language quickly appeared and linger to this day.

In order to bring some predictability to JavaScript, it was eventually standardized by the European Computer Manufacturers Association (ECMA). This standardized version of the language is, officially at least, referred to ECMAScript. In practice most people will always call it JavaScript, regardless of which implementation of the language they developing with, or browser they are targeting.

## Chapter 3. A Simple JavaScript Example

---

No programming book would be complete without an initial example application just to get the reader started, and this book is no exception. In addition to a quick confidence boost for the novice, this serves as a good opportunity to check that the environment being used for the rest of this book is fully functional.

### 3.1 A First JavaScript Example

Ever since the first book on programming for the C language was published it has become traditional that the first example consists of writing code to display the words "Hello World". With this tradition in mind we will begin with such as example.

Using your favorite editor, create an HTML file with the following content:

```
<html>
<head>
  <title>My First JavaScript Example</title>
</head>

<body>
<script type="text/javascript" language="JavaScript">
  document.writeln( "Hello World" );
</script>
</body>
```

Load the HTML file into a JavaScript enabled browser and a web page will be displayed showing the text "Hello World". All that this script does is define the standard HTML tags that you would expect to see in an HTML page, and also define embedded JavaScript code that writes text out to the current document (*document* is the JavaScript object for a web page).

Previously we discussed the event driven nature of JavaScript. In the next example we will extend the "Hello World" example to include event handling, in this case designing our script to display an alert dialog when a button is pressed.

Edit the previous HTML file and modify the script code so that it resembles the following:

```
<html>
```

```
<head>
  <title>My First JavaScript Example</title>
</head>
<body>

<script type="text/javascript" language="JavaScript">
  document.writeln( "Hello World" );
</script>

<form>
  <input type=button value="Press me" onClick="alert('Goodbye cruel
world')">
</form>

</body>
</html>
```

When loaded into a web browser the page will appear as before, but this time a push button is visible in the page which, when pressed, will display an alert dialog containing the words "Goodbye cruel world". In order to make this happen we have used the HTML `<form>` and `<input>` tags to make the button visible and combined this with the `onClick` event handler. We've specified that when the `onClick` event of the button is triggered (i.e. when the user clicks on it) then the JavaScript to display the alert dialog should be executed.