The advantages of Javascript

Introduction

Javascript is a browser-interpreted language that was created to access all elements of HTML and the browser. The processing is done entirely by the client-side browser which makes it very useful tool to handle processing which would have otherwise been checked server-side, thereby reducing overhead. Javascript is also used to increase user interaction, animate objects, create drop down navigation, grab data from databases, and more! If you are working with webpages in any way, javascript will help. In this article, I attempt to point out the advantages in learning this powerful language.

Making Magic with DHTML

You have probably encountered javascript through your years of web surfing without knowing it. Currently, the most popular implementation of javascript is the rollover. You know, when your cursor "rolls over" an image and it changes. Another popular use of javascript is drop down navigation menus. Javascript is used to create "magic" effects on webpages in conjunction with CSS (Cascading Style Sheets). The combination of HTML, CSS, and javascript is called DHTML for Dynamic HyperText Markup Language. The Dynamic in DHTML refers to how elements in the webpage appear to move or change, usually by user interaction. Not surprisingly, the brains behind these effects is javascript.

HTML was designed for web authors to quickly and easily create web pages. With only knowledge of HTML, designers will not have access to certain areas of the web browser. Examples include the status bar, browser information, browser window size, cookies, and browser history. Javascript is needed in those cases in order to access those elements and do things with them.

CSS can give properties to HTML elements that can't be created through tags. CSS is also used in conjunction with Javascript in order to identify and categorize certain HTML elements. By systemitizing these elements, javascript can then access them much more easily.

If you are designing a website for public access, implementing some DHTML helps your audience and navigate interact with your pages.

Dynamic Data with AJAX

The "Dynamic" in DHTML is a bit of a misnomer in the sense that the content does not change. In some effects, the content is simply hidden from the audience and revealed later with javascript. True dynamic content would require some sort of access to external data such as a text file or a database. Unfortunately, since javascript was designed as a client-side tool, it is not the ideal way to fetch data for you. However, with a bit of clever programming, developers have come up with a solution that integrates javascript, XML, and PHP together. This technology is called AJAX –

Asynchronous Javascript And XML.

The basic idea of AJAX is to use the event handlers of javascript to fetch external data from an external PHP/ASP page. This page, in turn, fetches the data from the database and *generates* an XML file with the content. The javascript parses the *xml* file and embeds the data straight into the HTML. While the backend programming may seem a bit complicated, the advantages of AJAX benefit both server and client sides.

Traditionally, dynamic data can only be generated by a page refresh. Not only does this interrupt the flow of your web applications, the server must also resend the *entire* webpage, resulting in unnecessary overhead. One of the major benefits of AJAX is that your web applications *do not need to be refreshed*. To the client, a simple button click generates dynamic data straight on the screen *seemlessly*. To the server, only the data that is requested by the client is sent. Furthermore, since the raw data is being parsed and processed through javascript, AJAX saves the server precious processing cycles. The result: dynamic, seamless webpages that are optimized for speed.

Javascript Form Validation

As I've hinted in the introduction, javascript can be used to validate HTML form fields. Let's imagine a familiar scenario in which a server is at maximum and webpages are being served slowly. Imagine a innocuous client who forgets to type in the "at" on the email field of a form. He clicks submit and waits … and waits and … waits only to receive a cold "invalid input" message. Then he clicks the back button on the browser, which asks if he wants to retrieve the webpage again. The user has no choice but to say yes, and he waits…and waits.. and waits only to go back to the original screen and try again. Not only does all this waiting frustrate the client, the server was also bothered by having to do the check on the email field.

Javascript can act as the intermediary between the submitted form and server. In our imaginary example above, javascript steps in and immediately tells the client that he forgot the "at". Only when the form is valid does the page go to bother the already-busy server.

Javascript is a completely object-oriented language. While the "script" in javascript may imply on-the-fly procedural programming, it has capabilities to create objects, methods, and even inheritance! This, along with dynamic data fetching through AJAX, allows you to truly create complex and powerful applications.

Conclusion

To learn more about javascript and all the aforementioned technologies, I highly recommend http://www.w3schools.com/ both as a tutorial and a reference. Be sure to read up on the DOM model to take advantage of the power of this language.

In the center of all of these web technologies is javascript. The language itself is simple in that you don't need to worry about data types, compilers, or forgetting that semicolon! Javascript is also powerful in how integral it is to so many existing technologies. It saves server frustration, client frustration, and programmer frustration. The next time you create a web page, think about how

might it be improved with javascript.