

# JavaScript

- [Javascript](#)
- [Numeric Validation JavaScript](#)
- [The advantages of Javascript](#)
- [jQuery Tutorial](#)
- [Which Javascript framework should I use?](#)
- [jQuery and JavaScript Coding: Examples and Best Practices](#)

# Javascript

Javascript, also known as ECMAScript, is used to make web browsers perform certain actions before returning to the web server. It depends on web browsers behaving the same. This was problematic for a long time, but is getting much better. Here are a few Javascript resources.

- <http://www.javascript.com/>
- [JavaScript Puzzlers: A 37 Question JavaScript Quiz](#)
- [JavaScript Tools of the Trade: JSBin](#)
- [JavaScript: The World's Most Misunderstood Programming Language – Douglas Crockford © 2001](#)
- His more recent book [JavaScript: The Good Parts – Douglas Crockford © 2008](#)
- [The Young Developer's Guide to Debugging JavaScript](#)  
By ANDRE BEHRENS – N.Y. Times
- [JavaScript Language Resources](#)
- [JavaScript Weekly](#) – A free, once-weekly e-mail round-up of JavaScript news and articles.
- [Select All\\_ JavaScript for Forms Posting to an Array](#)
- [WYSIWYG Editors written in Javascript](#)
- <http://jquery.com/>
- [Frequently Misunderstood JavaScript Concepts](#) by Michael Bolin, October 28, 2013
- [Learn to play Javascript by playing this game](#) – Code Combat
- (new) [Speaking Javascript](#) by Dr. Axel Rauschmayer – free HTML version – March 2014
- (new) [You Don't Know JS book series](#) – free online, or you can buy them – Feb. 2015

*It is simultaneously a simple, easy-to-use language that has broad appeal, and a complex and nuanced collection of language mechanics which without careful study will elude true understanding even for the most seasoned of JavaScript developers.*

[UCLA KB Articles tagged with Javascript](#)

# Numeric Validation JavaScript

## Series I : Validation (AJAX Presentation CWP)

Several people have asked me to discuss some of the details from the AJAX presentation from our last Campus Web Publishers (CWP) meeting.

In this article, which is part of the AJAX presentation series, I will share some validation functions I have used to validate the grid. In addition to validating the field, notice how it maps the arrows keys to give the user a Microsoft Excel like feel.

### Validating a numeric field

```
function NumbersOnly( Object ) {  
    if( CurrentCellClicked ) {  
        if(( event.shiftKey && event.keyCode == 9 ) || event.keyCode == 37 ) {  
            event.returnValue = true;  
            return ;  
        }  
        else if( event.keyCode == 39 || event.keyCode == 38 ) {  
            event.returnValue = true;  
            return ;  
        }  
        else {  
            event.returnValue = false;  
            return ;  
        }  
    }  
}
```

### Validating a currency field

```
function CurrencyOnly( Object ) {  
    if( CurrentCellClicked ) {  
        if(( event.shiftKey && event.keyCode == 9 ) || event.keyCode == 37 ) {  
            event.returnValue = true;  
            return ;  
        }  
        else if( event.keyCode == 39 || event.keyCode == 38 ) {  
            event.returnValue = true;  
            return ;  
        }  
        else {  
            event.returnValue = false;  
            return ;  
        }  
    }  
}
```

```

if(event.keyCode == 36 || event.keyCode == 35 || event.keyCode == 46 || event.keyCode == 8) event.preventDefault();
else if((event.shiftKey && event.keyCode == 9) || event.keyCode==37) {
    var t = parseInt(Object.nextField);
    var t = GlobalForm['c'+Object.nextField];
    if(t.disabled) {
        GlobalForm['c'+t.nextField].select();
        GlobalForm['c'+t.nextField].focus();
    }
    else {
        t.select();
        t.focus();
    }
    event.returnValue = false;
}
else {
    event.returnValue = (event.keyCode == 110 || event.keyCode == 190 || event.keyCode == 109 || event.keyCode == 107 || event.keyCode == 105 || event.keyCode == 103 || event.keyCode == 101 || event.keyCode == 99 || event.keyCode == 97 || event.keyCode == 95 || event.keyCode == 93 || event.keyCode == 91 || event.keyCode == 89 || event.keyCode == 87 || event.keyCode == 85 || event.keyCode == 83 || event.keyCode == 81 || event.keyCode == 79 || event.keyCode == 77 || event.keyCode == 75 || event.keyCode == 73 || event.keyCode == 71 || event.keyCode == 69 || event.keyCode == 67 || event.keyCode == 65 || event.keyCode == 63 || event.keyCode == 61 || event.keyCode == 59 || event.keyCode == 57 || event.keyCode == 55 || event.keyCode == 53 || event.keyCode == 51 || event.keyCode == 49 || event.keyCode == 47 || event.keyCode == 45 || event.keyCode == 43 || event.keyCode == 41 || event.keyCode == 39 || event.keyCode == 37 || event.keyCode == 35 || event.keyCode == 33 || event.keyCode == 31 || event.keyCode == 29 || event.keyCode == 27 || event.keyCode == 25 || event.keyCode == 23 || event.keyCode == 21 || event.keyCode == 19 || event.keyCode == 17 || event.keyCode == 15 || event.keyCode == 13 || event.keyCode == 11 || event.keyCode == 9 || event.keyCode == 7 || event.keyCode == 5 || event.keyCode == 3 || event.keyCode == 1 || event.keyCode == 0);
}
}

```

## Validating a Date Field

```

function DateOnly(Object) {
    if(CurrentCellClicked) {
        if((event.shiftKey && event.keyCode == 9) || event.keyCode==37) {
            event.returnValue=true;
            return ;
        }
        else if(event.keyCode9 || event.keyCode39) {
            event.returnValue=true;
            return ;
        }
    }
    if(event.keyCode == 36 || event.keyCode == 35 || event.keyCode == 46 || event.keyCode == 8) event.preventDefault();
    else if((event.shiftKey && event.keyCode == 9) || event.keyCode==37) {
        var t = parseInt(Object.nextField);
        var t = GlobalForm['c'+Object.nextField];
        if(t.disabled) {
            GlobalForm['c'+t.nextField].select();
            GlobalForm['c'+t.nextField].focus();
        }
    }
}

```

```

else {
    t.select();
    t.focus();
}
event.returnValue = false;
}
else {
    event.returnValue = (event.keyCode == 109 || event.keyCode == 189 || event.keyCode == 191 || e
}
}

```

### Of course, date validation also requires:

```

function IsDate(DateObject) {
    var month, year, day;
    if(DateObject.value.length < 6) {
        var datePat = /^(\\d{1,2})(\\/| -)(\\d{1,2})$/;
        var matchArray = DateObject.value.match(datePat); // is the format ok?
        if (matchArray == null)
            return false;

        month = matchArray1; // parse date into variables
        day = matchArray2;
        year = (new Date()).getFullYear();
    }
    else {
        var datePat = /^(\\d{1,2})(\\/| -)(\\d{1,2})(\\/| -)(\\d{2,4})$/;
        var matchArray = DateObject.value.match(datePat); // is the format ok?
        if (matchArray == null)
            return false;

        month = matchArray1; // parse date into variables
        day = matchArray2;
        year = matchArray3;
    }

    if (month < 1 || month > 12) { // check month range
        throw (new Exception("Month must be between 1 and 12."));
    }
}

```

```
if (day < 1 || day > 31) {
    throw (new Exception("Day must be between 1 and 31."));
}

if ((month4 || month6 || month9 || month11) && day==31) {
    throw (new Exception("Month month doesn't have 31 days"));
}

if (month == 2) { // check for february 29th
    var isleap = (year % 4 == 0 && (year % 100 != 0 || year % 400 == 0));
    if (day > 29 || (day==29 && !isleap)) {
        throw (new Exception("February " + year + " doesn't have " + day + " days!"));
    }
}

if(year.toString().length>2) {
    year = year = year.toString().substring(2);
}

DateObject.value = (parseInt(month)) + '/' + (parseInt(day)) + '/' + year;
return true; // date is valid
}
```

# 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 an 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.

# jQuery Tutorial

Learn the basic building blocks of jQuery in a two hour combination of videos, and interactive console challenges. – <http://try.jquery.com/>

*Please list any other useful jQuery Tutorials you know of.*

# Which Javascript framework should I use?

So you're tasked with creating a new web site or enhancing an old one and you need to provide some nice UI components or a RIA, the question is, what to use? We have faced this question at the Digital Libraries, and found the best approach for us is to define the requirements and audience of the application.

Usually, when asking around about js frameworks, the default answer is often jquery and in most cases jquery is a very good choice, but it also depends on what type of application you are building.

If you are interested in mimicking a desktop application feel, Ext-js might be a library to take a look at, which is a javascript library of almost Windows-like control widgets. An example of this might if your site has an admin portion, and needs very fine grained control such as trees, file and directory browsers, editable data grids, etc...

A nice comparison of several js frameworks is available here:

<http://fictionalrealm.com/coding/2009/04/19/javascript-ui-framework-comparison-or-why-i-choose-extjs-jquery/>

If you are looking to build an app-like web project for mobile and tablet devices, the Sencha Touch library looks to fit that niche.

[Sencha Touch](#)

# jQuery and JavaScript Coding: Examples and Best Practices

When used correctly, jQuery can help you make your website more interactive, interesting and exciting. This article will share some best practices and examples for using the popular Javascript framework to create unobtrusive, accessible DOM scripting effects. The article will explore what constitutes best practices with regard to Javascript and, furthermore, why jQuery is a good choice of a framework to implement best practices.

<http://www.smashingmagazine.com/2008/09/16/jquery-examples-and-best-practices/>