

## DHTML

DHTML is a combination of HTML, the Document Object Model (DOM), style sheets and scripting languages.

DHTML allows you to create web pages that are more animated and responsive to user interaction than HTML. Many users still use old browsers. Therefore, you have to create 2 versions of each site and serve them to the different browser versions. DHTML uses the structure of HTML to create interactive features. Each HTML page element is seen as an object. This is known as the DOM. It provides a standard interface to make objects do what the developers want them to do. The DOM enables any HTML tag to be seen as an object. A web page is composed of single objects that are elements with their own properties. This characteristic allows you to add script specific to the actions of that element. Web page style sheets describe the default style characteristics of a page as well as the default background color or image, hypertext link colors and content. Style sheets are used to ensure consistency across the pages of a web site. DHTML allows you to specify cascading style sheets. Dynamic fonts are also a characteristic of DHTML. They allow web page designers to include font files, containing specific styles, sizes and colors with the page. Therefore the font choice is not depended on what the browser provides.

### Quick Note:

HTML - describes how its surrounding text should appear in a browser window.  
 XML - describes what the text means. Allows the user to define their own tags as required.  
 XML is as much a data-interchange tool for communicating data among applications as a language for displaying data to the user. XSL (Extensive Style Language) - is XML style companion. It has 2 sources: DSSSL (document style semantics and specification language) and CSS (cascading style sheets).

## HTML

HTML 1.0 was created in 1984 by CERN and is now controlled by the WWW Consortium (W3C), which sets its standards. www is only one of the protocols on the web and HTML is only one of many languages used to deliver information over the internet. HTML is a markup language. Primarily use is to create web pages. It provides an international computer coding language that facilitates universal access independent of platform, network or terminal. Originally provided access to text-based environment and facilitated the exchange of research information. HTML is a markup language. This means it tells the browser how to treat the document that it is displaying. It shows where the headings and spaces are to be inserted for example. Another important function of HTML is to provide a reference system for linking documents or parts of documents. HTML enables this file or its URL to be linked to a key word or phrase in the text. Click on the word and the file is displayed.

### HTML 4.01 Features:

- Non-Linearity - it is the function of the architecture of web pages to make sense to visitors.
- Supports XHTML - that enables you to use the features of XML while creating web pages.

- Supports multiple languages - web pages can be created in multiple languages. Called internationalization. It provides language identifiers that describe the language in which the text on a web page is to be displayed.
- Supports embedded controls, such as ActiveX controls and plug-ins, are the interactive controls that are inserted in a web page from a file that exists at another location on the web.
- Supports scripting - enables you to create dynamic web pages that allow user interactivity and personalized interface to a web site.
- Supports enhanced printing options - for example you can print all web pages that are linked to a specific web page.

## DOM

[www.w3.org/DOM/](http://www.w3.org/DOM/)

The document object model is an application-programming interface (API) that was developed by the W3C. It is a set of logical rules. It defines the logical structure of a document and the way it can be accessed and manipulated. Objects, properties and methods enable you to manipulate the content, structure and style of a web page and display the changes even as they are being displayed in a browser.

Objects have properties and methods. A property is an essential characteristic for example fgcolor. A method is a programmed procedure for example: the contents of a frame can be printed by using the objects print method. The DOM is platform independent and language neutral. The document object model (DOM) is the proposed specification for how objects on a Web page are represented. Of course, Microsoft and Netscape each have their own versions of the DOM and have submitted them to the World Wide Web Consortium (W3C) to decide on a standard. A DOM defines each object on a Web page (images, text, scripts, links, etc.) and also defines what attributes are associated with these objects and how they can be manipulated. The fact that Netscape Navigator and Microsoft Internet Explorer use different DOMs is one reason why each browser's implementation of DHTML is different.

## XHTML

[www.w3.org/TR/xhtml1/](http://www.w3.org/TR/xhtml1/)

XHTML is a standard proposed by the W3C that adapts HTML into an extensible concept by using XML (extensible markup language). XML defines data that can be shared on the web. It is extensible because anyone can invent a set of purposes such as describing the appearance of a web page. To enhance web pages, HTML was redesigned by using XML to form XHTML. XHTML is portable to enable small devices to support embedded programming. XHTML brings different programming practices. It has strict code rules such as symmetrical form, use lowercase, enclose elements with quotes and end tag with a forward slash at the end of the element and before the closing angle bracket.

## XML

[xml.coverpages.org/xml.html](http://xml.coverpages.org/xml.html), [www.xml.org](http://www.xml.org)

XML is a flexible way to create common formats for data that client- and server-side developers use to enable them to control data. It enables developers to share the format and the data through intranets and on the web. XML is a W3C recommendation. It describes the content of a web page by what the data means. XML enables developers to use diverse data types. Converting data to XML can reduce the complexity of exchanging data and create data that can be read by different types of applications. It

provides easy-to-parse syntax to represent data by isolating the content component from the descriptive rendering instructions. This capability separates content and presentation. XML tags are not predefined. XML allows you to define your own tags and your own document structure, to suit your needs.

### **XML Features:**

- Written in plain text and is readable. You can edit or view an XML document with a simple text editor.
- Cross-platform software and hardware independent tool that is used for transmitting information.
- Used primarily for Web pages. XML allows the author to extend and customize basic HTML formatting by creating proprietary tags and text behaviors. XML is meant to emphasize intelligent and logical formatting within technical documents in order to streamline searching and categorizing, and to ensure total cross-browser compatibility.

### **WML**

[www.oasis-open.org/cover/wap-wml.html](http://www.oasis-open.org/cover/wap-wml.html)

WML is loosely based on HDML (handheld device markup language). It allows you to present text portions of a web page on cell phones and PDAs (personal digital assistants) by wireless access. WML is an open source language that is the part of the WAP (wireless application protocol). It is also an XML that offers similar functionality to HTML4.0 but is used for small screen devices.

### **WML Features:**

- Small display
- Limited input capacity
- Narrowband network connection
- Limited memory
- Limited CPU power
- It provides WML users with games, email services and instant messaging.

WML supports the Meta element that supports names, schemes and content attributes. WML does not support the link element because there is no mechanism for linking to external metadata. WML files have strict parameters in order to maintain small file sizes. WML uses WAP, which has limited memory. It supports scripting and event handling for navigation or script invocation. These elements enable users to interact with the page and also to enable automated page jumps. WML syntax is used to manage text and graphics. It controls data input, hyperlinks and navigation. Decks are logical root element that contains the cards entities. Cards represent parts of tasks while decks represent the complete task. Decks minimize network time and reduce download because you download a deck at a time and you do not need network connection to navigate between the cards. WML is strictly defined language and it is domain specific. WML elements must be written in lowercase, attributes must be quoted and elements must have closing tags. Maximum size of a deck is 1492 bytes.

### **RSS**

What is RSS?

Intorduction to RSS

RSS is a method of describing news or other Web content that is available for "feeding" (distribution or syndication) from an online publisher to Web users. RSS is an application of the Extensible Markup Language (XML) that adheres to the World Wide Web Consortium's Resource Description Framework (RDF). Originally developed by Netscape for its browser's Netcenter channels, the RSS specification is now available for anyone to use. All RSS files must conform to the XML 1.0 specification, as published on the World Wide Web Consortium (W3C) website. A Web site that wants to "publish" some of its content, such as news headlines or stories, creates a description of the content and specifically where the content is on its site in the form of an RSS document. The publishing site then registers its RSS document with one of several existing directories of RSS publishers. A user with a Web browser or a special program that can read RSS-distributed content can read periodically-provided distributions. Some current directories of RSS files include Meerkat, GropSoup, NewsIsFree, UserLand, and XML Tree; these sites are sometimes known as content aggregators. RSS browsers include Headline Viewer and Novobot. News is only one form of content that can be distributed with an RSS feed. Other possibilities include discussion forum excerpts, software announcements, and any form of content retrievable with a URL.

At the top level, a RSS document is a <rss> element, with a mandatory attribute called version that specifies the version of RSS that the document conforms to. If it conforms to this specification, the latest version attribute is 2.0. Subordinate to the <rss> element is a single <channel> element, which contains information about the channel (metadata) and its contents (items).

### **Intermediate Javascript tutorial**

If you are reading these pages it is assumed that you have already read the [beginning JavaScript](#) pages or have written and used some JavaScript in Web pages before.

1. [JavaScript comments](#)
2. [JavaScript and external stylesheets](#)
3. [Browser identification with JavaScript](#)
4. [JavaScript arrays](#)
5. [The JavaScript DATE object](#)
6. [The JavaScript Math object](#)
7. [The JavaScript String object](#)
8. [More on JavaScript variables](#)
9. [JavaScript SWITCH](#)
10. [JavaScript and timing](#)
11. [JavaScript FOR loops](#)
12. [JavaScript WHILE loops](#)
13. [JavaScript cookie handling](#)
14. [External JavaScript](#)
15. [JavaScript Event Listeners](#)
16. [Validating form contents with JavaScript](#)
17. [JavaScript files and directories](#)

### **JavaScript and external stylesheets**

You should realise that it is possible to use [multiple external stylesheets on a Web page](#) and therefore that you might need to do more than just [access a single external stylesheet with JavaScript](#).

Unfortunately there is no obvious way to do that other than looping through the stylesheets and looking for the correct one.

There are also browser issues to contend with (and not just in IE as Chrome and Safari also fail to follow the rules on alternate stylesheets).

### **Accessing more than one stylesheet using JavaScript**

If you know the order of the sheets you could just use the index number to access them:

```
document.styleSheets[0]
```

The styleSheets object contains all stylesheets LINKed to from the current page. It can be thought of as an array holding the content of all of the stylesheets used on the page. This includes both external stylesheets and embedded stylesheets.

To write good code though you need to allow for someone changing the HTML or CSS by adding stylesheets or changing their order. You need to look through the stylesheets to find the one you want:

```
for (var i=0; i<document.styleSheets.length; i++) {
  if (document.styleSheets[i].title=="styles2") {
    alert('Success in finding the stylesheet at position: '+i);
    break;
  }
}
```

The code loops through the stylesheets looking for the one with the title "styles2" and then stops. only the first one being used on page load. Create three stylesheets and a Web page following [this page](#). Then try the above code (you can just put it in SCRIPT tags at the end of the BODY or put it in a function with an event trigger).

### **Accessing the rules within the stylesheet**

Once you have identified the correct stylesheet using the above code you can look through the rules to find the one you want. Here i is the variable containing the number of the stylesheet so you need to leave the above loop in for this to work (put this code just after it):

```
for (var j=0;j<document.styleSheets[i].cssRules.length;j++) {
  if(document.styleSheets[i].cssRules[j].selectorText=='body'){
    document.styleSheets[i].cssRules[j].style.color="#00FF00";
    alert("The body selector was found and was in position '+j+' out of
'+document.styleSheets[i].cssRules.length+' rules")
  }
}
```

The FOR loop will run through all of the rules in the stylesheet (cssRules.length is the number of rules found). If there is a matching rule there will be an alert and then the loop will be stopped. You should probably also allow for not finding the CSS rule at all.

### **Displaying the rules within all stylesheets**

If you struggled to understand that it might help you to see the stylesheets and CSS rules for a page. For this to work well you really need a page which uses more than one stylesheet (with titles) and each having more than one rule! Use a standards-compatible browser (Opera or Firefox). Paste this script after the </BODY> tag but before the </HTML> tag.

```
for (var i=0; i<document.styleSheets.length; i++) {
  var stylesheetTitle=document.createElement('h1');
  stylesheetTitle.textContent=document.styleSheets[i].title;
  document.body.appendChild(stylesheetTitle);
  for (var j=0; j<document.styleSheets[i].cssRules.length; j++) {
    var styleRule=document.createElement('p');
    styleRule.textContent=document.styleSheets[i].cssRules[j].cssText;
    document.body.appendChild(styleRule);
  }
}
```

By nesting the second FOR loop the styles are listed under the heading of the title of each sheet.

### **Application**

To amend a stylesheet style the easiest way is probably to use in-line styles and override any external styles. If you are trying to do something more sophisticated and need to change the external styles you can adapt the above code to:

1. find the appropriate stylesheet
2. find the position of the rule which sets the existing style (which you want changed)
3. use those two numbers (i and j in the above examples) to set the new value

### **Selecting alternate stylesheets**

It is also possible to change the stylesheet being used rather than changing individual properties although browser incompatibility continues to be a problem. The first step is to set up more than one stylesheet as you did above following the instructions on the [page about CSS and multiple stylesheets](#).

Once you have a page with multiple stylesheets you can use JavaScript to disable the preferred stylesheet and enable one of the alternates. Create a function in the HEAD section of the Web page or in an external JavaScript source file:

```
function changeStyleSheet() {
  alert("The number of stylesheets on this page is: "+document.styleSheets.length);
}
```

```
document.styleSheets[1].disabled=false;
document.styleSheets[0].disabled=true;
}
```

At the bottom of the existing page add an event trigger such as:

```
<button onmouseup="changeStyleSheet()">Press me</button>
```

Load the page in a browser and results may vary. Browsers which follow standards will only use the first stylesheet when the page first loads (so the page will be black on white). However, they will report all three stylesheets in the ALERT. Then they switch to the second stylesheet and disable the first one (resulting in yellow on blue).

Other browsers which do not follow standards may work partly or not at all. For example, Safari and Chrome do not recognise the presence of the alternate stylesheets unless you remove the TITLE attributes (and if you do that the script will fail in the other browsers). To allow for both you would need to check for compatibility and offer both approaches. That involves spotting which browser is which and using or not using TITLES though and that is notoriously difficult to keep up-to-date.

You would be able to come up with a JavaScript-based system which allows the user to express a preference (a drop down menu perhaps) and then record that in a cookie. However, a better solution is to use server-side scripting to record stylesheet choice in a cookie and then offer only one stylesheet to the browser to avoid any chance of confusing the poor thing!

### **Browser identification with JavaScript**

Traditionally this has been called browser sniffing and has been used so that you can decide which code to use with the browser. The way you learned in the beginner's pages is much more reliable for handling [JavaScript incompatibilities across browsers](#) but you might still want to find out what browsers are being used.

```
alert(navigator.userAgent);
```

The results might surprise you sometimes which is why browser sniffing is of limited use. Also try appVersion, appName, appCodeName, appVersion, oscpu, cookieEnabled, browserLanguage, onLine, userLanguage, javaEnabled and platform.

### **Other useful information**

You can also find out other useful things using JavaScript such as the resolution the user has on their system:

```
alert('resolution is '+screen.width+ ' by '+screen.height);
```

## JavaScript arrays

You should already know [what an array is](#).

This sets up a JavaScript array:

```
var choices=new Array('chicken','beef','pork','vegetarian');
```

You can leave the values out and add them later.

```
var choices=new Array();
choices.push('chicken');
choices.push("beef");
```

PUSH adds a new element at the end of the array. SHIFT does the opposite and removes the first element in the array. POP gets the last element out and SPLICE removes:

```
choices.splice(2,1);
```

or replaces:

```
choices.splice(2,1,'pork');
```

a specified element. More traditionally you can do all this by assigning values:

```
var choices=new Array();
choices[0]='chicken';
choices[1]='beef';
```

To access the data in the array you identify which entry you want with a number in square brackets:

```
alert(choices[1]);
```

That would show beef in the dialog. Array numbering starts at zero so 1 is the second entry.

If you want to create an array with a set size but not put anything in it yet:

```
choices = new Array(4);
```

## Finding things in arrays

The built in method `indexOf()` finds the array element which contains what you want:

```
fred=new Array('this', 'that', 'the other');
var location=fred.indexOf('that');
alert('At element '+location+' there is found the word '+fred["location"]);
```

### **Looping through arrays**

By replacing the number in the square brackets with a variable (an incremented counter) or a test/condition you can more easily get at each entry in turn using loops. You can use FOR to go through all elements in the array or WHILE loops to go through until one matches the condition. These will be covered soon.

An array is an object and so has built in methods. The array object in JavaScript has a method which will do the same thing for every element in the array:

```
fred.forEach(somefunction());
```

This will send each element of the array to your function. It actually sends three items of data (the element value, it's index and the whole array contents) to the function. Try using ALERT in the function to display the contents of a simple array to see what happens:

```
function somefunction(value, index, wholearray){
    alert('The value at location '+index+' within the whole array '+wholearray+' is '+value);
}
fred=new Array(5,2,8,7);
fred.forEach(somefunction);
```

### **Sorting array contents**

Another method of the array object allows the sorting of it's contents:

```
somearray.sort();
```

This will sort alphabetically even if the values are numbers (so 11 comes before 2).

### **Multi-dimensional arrays**

To create the main (top or parent) array do it as before:

```
playersarray = new Array(2);
```

Then create the array contained inside each element in the main array:

```
playersarray[0]= new Array(1);
playersarray[1]= new Array(2);
```

If you need to put lots of sub-arrays in a loop will help.

To put data into the sub-arrays:

```
playersarray[0][0]='Fred';
playersarray[0][1]='1/1/11';
playersarray[0][2]='20965';
playersarray[0][0]='Freda';
playersarray[0][1]='10/10/11';
playersarray[0][2]='25467';
```

To get the data out:

```
var player=playersarray[0][0];
var date=playersarray[0][1];
```

## The JavaScript DATE object

Dates in JavaScript can be confusing at first but are fairly easy eventually.

### Getting the current date

To create a new date:

```
var currentDate=new Date();
```

Create an alert to show currentDate and you will see the format. It looks hard to manipulate but it is not:

```
var currentFullDate=new Date();
var currentDate=currentFullDate.getDate();
```

The method getDate extracts just the numeric date (day of the month) from the date. You can extract any other part in the same way by changing the bit after "get" (e.g. getDay, getYear, getMinutes and even getTimezoneOffset).

### Other dates

You can also manually fix the date:

```
var myDate=new Date('January 16, 1988');
```

JavaScript will assume it is midnight and that the time zone is GMT. You can include the whole date including seconds if you want to.

### Comparing dates

You can use dates as they are in IFs or WHILEs:

```
while (myDate<currentDate){
  document.getElementById('messagediv').innerHTML='processing...';
  currentDate=Date();
}
document.getElementById('messagediv').innerHTML='It's time!';
```

The loop will continue until the current time is more than the target time.

### Date arithmetic

You do date arithmetic by extracting the bit (date, month, year) you want, doing the addition or subtraction and JavaScript will then put the changed bit back in:

```
var myDate=new Date('January 16, 1988');
myDate.setDate(myDate.getDate()+1);
```

Add an alert box and it should give January 17. The method getDate gets just the date part and then 1 is added. That value is then used to setDate. Change setDate to setYear and getDate to getYear and try again.

If the date had been January 31 and you added one JavaScript would handle that properly by making it February 1st. You might not expect that at first but it works well.

You can also subtract one date from another to find elapsed time the result (the gap) is in milliseconds so divide by 1000 for seconds, 60000 for minutes etc..

### **The JavaScript Math object**

Like any other object Math has methods and properties. The properties are fairly obscure but do include  $\pi$ . Math methods are used to manipulate numbers:

```
squareRoot=Math.sqrt(4);
```

That will put the square root of 4 into the variable squareRoot.

Other methods can round numbers (`Math.round`), remove the numbers after the decimal point (`Math.floor`) or create random numbers (`Math.random`). If you are using an editor with good auto-complete or hint functions (or a search engine) you will find there are many more.

### The JavaScript String object

The most useful property for a string object is the length:

```
nameLength=surname.length;
```

There are a few methods which will be fairly useful for someone using JavaScript on their Web pages to make them interactive:

```
letter=String.fromCharCode(keyPressed);
```

That one converts an ASCII code (which might come from an event) into a Unicode character (e.g. B). Some other useful methods include:

- `someWords.match('go')` checks if the variable contains "go"
- `someWords.replace('go', 'goo')` replaces any instance of "go" in the variable with "goo"
- `position=someWords.search('go')` looks for "go" and returns the position of it in the variable
- `position=someWords.indexOf('go', 0)` also looks for "go" and returns the position of it in the variable but can take a starting position for the search
- `someWords.split(',')` would split a comma-separated string (often used to export database data) into an array
- `smallBit=someWords.substr(3,10)` extracts from the third character to the thirteenth from the variable
- `someWords=someWords.toLowerCase()` makes all letters lower case (`toUpperCase` does the opposite)