TAGS

a
abbr
acronym
address
applet
area
b
base
bdo
big
blockquote
body
br
button
caption
cite
code
col
colgroup
dd
del
dfn
div
dl
DOCTYPE
dt
em
fieldset
form
frame
frameset
h1,h2,h3,h4,h5,h6
head
hr
html
i
img
input
ins
kbd
label
legend
li
link
map
meta
noframes
noscript
object
ol
optgroup
option
p
param
pre
q
samp
script
select
small
span
strong
style
sub
sup
table
tbody
td
textarea
tfoot
th
thead
title
tr
tt
ul
var

ATTRIBUTES

accesskey
class
dir
id
lang
style
tabindex
title

EVENTS

onblur
onchange
onclick
ondblclick
onfocus
onkeydown
onkeypress
onkeyup
onload
onmousedown
onmousemove
onmouseout
onmouseover
onmouseup
onreset
onselect
onsubmit
onunload

XHTML CODING RULES AND SYNTAX

While these coding rules apply to XHTML, it is strongly recommended that you also follow these same rules when coding HTML.

All attributes, events, and tags must be written in lower case

In HTML it did not matter if you coded with a mixture of upper and lower case letters. In fact, many programmers typically coded HTML in upper case because they felt that it made the code more readable. Those days are over!

To be correct XHTML, you can only code attributes, events and tags in lower case. This is a requirement because we need our XHTML code to be compatible with XML, and XML is case sensitive (XML considers **<HR />** and **<hr />** to be different tags).

Wrong:
```
<A HREF="http://www.devguru.com" Target="_self">DevGuru</A>
```

Right:
```
<a href="http://www.devguru.com" target="_self">DevGuru</a>
```

All elements must be closed

In HTML we have many types of tags that need to be closed. The purpose of a close is to signify that the element stops at that point in the code. For example, consider using the **a** tag to create a link. The closing **** tag is placed immediately after the link text. If you did not include the closing ****, then the entire rest of the HTML document is considered to be part of the link text (as most programmers have discovered the hard way!).

In truth, HTML is very forgiving when it comes to closing tags. For example, most browsers will successfully display the following code.

```
<HTML>
Hello world!
```

In comparison, XHTML is absolutely NOT forgiving. In XHTML, all elements must be closed. (There is one exception, the **DOCTYPE** DTD element which is discussed below.) You open and close an element by using a pair of tags. The first tag opens the element. The second tag closes the element. In the closing tag, the tag name is preceded with a mandatory slash.

Here are some examples:
```
<table border="2">
<tr><td>
DevGuru is great!
</td></tr>
</table>

<ul>
<li> DevGuru is great! </li>
<li> XHTML rules! </li>
</ul>
```

Even tags like **hr** and **br** must be closed. You close this type of tag by putting a blank space after the tag name and then a slash. (The blank space is needed for maximum browser

compatibilty.) Therefore, there is only one tag (i.e., no separate closing tag). Such tags are referred to as self-closing.

For example:

<hr />

The value assigned to an attribute must be enclosed in quotes

In truth, HTML does not require that a value assigned to an attribute be enclosed in quotes. However, XHTML is very strict about this. If you forget to enclose in quotes, then XHTML will completely ignore the attribute.

Wrong:
<table border=1 cellpadding=10 cellspacing=5>

Right:
<table border="1" cellpadding="10" cellspacing="5">

No attribute may be minimized

In HTML the following code is legal. Note that the **Checked** attribute is not assigned a value. This is called attribute minimization. This attribute will cause the radio button to be checked.

<INPUT Type=radio Name=Radix Value="16" Checked>hexidecimal

Having an attribute that does not have a value assigned to it is illegal in XHTML. In other words, no attribute minimization. The correct syntax is to assign "checked" as a value to **checked**. (In addition, we must also switch to all lower case, enclose values in quotes, and close the element to be correct XHTML.)

<input type="radio" name="Radix" value="16" checked="checked" />hexidecimal

All elements must be properly nested

Nested elements are contained inside of other elements. Here is an example of an unordered list element that contains nested elements.

 XHTML
 XML
 XPATH
 XSL
 XSLT

Note how each **li** element is closed before you start the next **li** element. Further note the **ul** element is not closed until after all of the internal **li** elements have been closed.

In this next example, we bold and italicize the text **DevGuru**. We must close the **i** element before we close the **b** element.

Wrong:
DevGuru

Right:
DevGuru

XHTML documents must be well-formed

At a minimum, an XHTML document must contain **html, head, title,** and **body** elements. They must be properly nested and closed. (A **frameset** tag can be used in place of **body**.)

Here is a minimal XHTML document.

```
<html>
<head>
<title>DevGuru</title>
</head>
<body>
DevGuru is great!
</body>
</html>
```

There must be a DOCTYPE declaration

The **DOCTYPE** tag is used to declare the DTD (Document Type Definition) for an XHTML document. This tag is mandatory and must appear at the top (on the first line) of all XHTML code. If the **DOCTYPE** tag is not present, then it is not XHTML code.

See the DOCTYPE write-up for details.

DTD TAG: DOCTYPE

<!DOCTYPE ... >

The **DOCTYPE** tag is used to declare the DTD (Document Type Definition) for an XHTML document.

On a broader scale, XHTML, like HTML, is a subset of the SGML (Standardized Generalized Markup Language). SGML, and its various family members, use DTDs to define the context of the language. The W3C has defined a DTD to be:

> "...a collection of declarations that, as a collection, defines the legal structure, elements, and attributes that are available for use in a document that complies to the DTD."

Specifically, the XHTML DTD precisely defines the grammar, rules, and syntax that will apply to a document that has been created using XHTML. To repeat this in a different way, to be valid XHTML, the XHTML code that creates the XHTML document must obey all of the grammar, rules, and syntax in the XHTML DTD.

This tag is mandatory and must appear at the top (on the first line) of all XHTML code. If the **DOCTYPE** DTD tag is not present, then it is not XHTML code.

The exclamation mark (!) is required. This is the only tag in an XHTML document that is not closed. Obey the case and syntax.

Currently, there are three types of DTDs that apply to XHTML: **Frameset, Strict,** and **Transitional**. You must obey the syntax.

Frameset

This is declared when you have partitioned the HTML document into two or more frames (with or without using Cascading Style Sheets).

Syntax:
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "DTD/xhtml-frameset.dtd">

Strict

This is declared when you use Cascading Style Sheets (CSS) to enhance the appearance and style of your HTML document. In general, you should only use this type of DTD if you are certain that your viewer has access to modern browsers that recognize CSS.

Syntax:
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml-strict.dtd">

Transitional

This is declared when you are simply using HTML to create the appearance of the web page, rather than using Cascading Style Sheets (CSS). This type of DTD would ensure the widest viewing audience to your XHTML document.

Syntax:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">

The basic XHTML document is composed of **DOCTYPE, html, head, title,** and **body** tags. Note that all tags in XHTML are closed except for the **DOCTYPE**.

Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/EN" "DTD/xhtml-transitional.dtd">
<html>
<head>
<title>DevGuru</title>
</head>
<body>
DevGuru is great!
<br />
<img src="images/xhtmlvalid.gif" />
</body>
</html>
```

Output:

DevGuru is great!

Welcome to the **DevGuru XHTML Quick Reference** guide. This is a handy 180 page reference source that defines and explains all of the tags, events, and associated attributes that compose XHTML (eXtensible Hyper Text Markup Language, version 1.0. This Quick Reference also includes useful, real world, working examples of code for each tag. In addition, there are write-ups, with code examples, for the 17 events and another 7 attributes that are commonly available to many tags.

To understand the future importance of XHTML, you first need to understand the current importance of HTML.

HTML is one of the most widely used computer languages in the world. The popularity of HTML is due to the fact that it is the coding technology used to publish content on the World Wide Web (also referred to as the Internet or Web). Programmers quickly discovered that HTML is a user friendly language and is very easy to learn. This ease of coding significantly aided in the proliferation of Web sites. The latest version of HTML is 4.01 which is defined by the standard published on 24 December 1999 by the World Wide Web Consortium (W3C).

In the foreseeable future, HTML will continue to serve as the basic structure for the Internet, but it is doubtful that there will be any newer versions or significant changes beyond HTML version 4.01. It is probably fair to predict that the future of HTML is XHTML and a search of the literature will reveal that XHTML is regularly referred to as the the next generation of HTML.

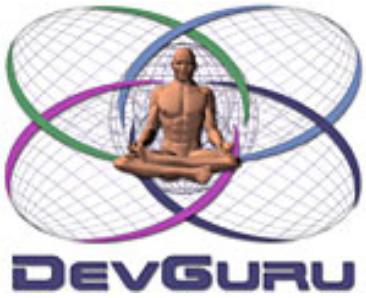
The history of XHTML is very simple; it is derived directly from HTML version 4.01 and is designed to be used with XML. Indeed, XHTML is part of a whole new suite of "X" technologies, with acronyms such as XML, XPATH, XSL, and XSLT, that are destined to have a profound effect on the Internet.

This is a new technology. On 26 January 2000, the W3C issued the recommendation for XHTML version 1.0. It is also a rapidly evolving technology. The recommendation for version 1.1, which is a module-based concept for XHTML, has already been published.

There are a few fundamental differences between HTML and XHTML that will profoundly effect how you code with XHTML. While HTML is a loose and forgiving language, XHTML will quickly remind you of a strict English teacher who demands firm adherence to the rules of grammar. Fortunately, the syntax and coding rules are very straightforward, easy to implement, and make common sense. The real purpose of these rules is to allow a seamless integration of XHTML with XML and other related "X" technologies.

- All attributes, events, and tags must be written in lower case.
- All elements must be closed.
- The value assigned to an attribute must be enclosed in quotes.
- No attribute may be minimized.
- All elements must be properly nested.
- XHTML documents must be well-formed.
- There must be a DOCTYPE declaration.

Click here for an explanation of the above rules.



Quick Reference Library

XHTML Quick Reference

TAG: a

**<a> ... **

The **<a>** tag is called the anchor tag and it is most commonly used with the **href** attribute to create a hypertext link (usually referred to as just a link). By simply clicking on the link with a mouse or using a keyboard command, you can travel from one page to another page on the same web site or a different one. By default, the new page is normally displayed in the same window or frame, unless you specify otherwise using the **target** attribute.

Note that links cannot be nested. An **<a>...** element cannot contain other **a** elements.

You can insert characters, images, line breaks (**
** or **<p />**), and text between the opening and closing **a** tags. It is recommended that you should not insert Cascading Style Sheets code or any other HTML tags between the opening and closing tags. Rather, place all such tags outside of the **a** element.

The separate closing tag is mandatory.

Attributes and Events

accesskey **class** **dir** **id** **lang** **onblur** **onclick** **ondblclick** **onfocus** **onkeydown** **onkeypress** **onkeyup** **onmousedown** **onmousemove** **onmouseout** **onmouseover** **onmouseup** **tabindex** **title**

charset
The **charset** attribute is used to specify the character encoding used on the page that is the target of the link. Character encoding defines how a sequence of bytes is to be converted into characters for display.

coords
The **coords** attribute is currently not supported by most browsers. It is used to define an area of influence around the **a** tag.

href
The **href** attribute is used to specify the target URL address for a link.

hreflang
The **hreflang** attribute can only be used when the **href** attribute is also being used. It specifies the language of the page that is the target of the link.

name
The **name** attribute is a string of characters that is used to label an element with a name. The name must be unique to that document and cannot be reused. The name can be the target of a link.

rel
The **rel** attribute is currently not fully supported by most browsers. It is a space-separated list of one or more values that specify the relationship from the source page to the target for a link. Some of the proposed values are, **appendix, bookmark, chapter, contents, copyright, glossery, help, index, next, prev, section, stylesheets,** and **subsection**.

rev
The **rev** attribute is currently not fully supported by most browsers. It is a space-separated list

of one or more values that specify the relationship from the target page to the source for a link. Perhaps the most useful value is **relation**.

shape
The **shape** attribute is currently not supported by most browsers. It is used to define an area of influence around the **a** tag.

target - not allowed in Strict XHTML
The **target** attribute specifies the name of the frame or window in which the target page should appear when a link is clicked. The four reserved names are **_blank, _parent, _self,** and **_top**.

type
The **type** attribute specifies the content type of the target page of a link. It has a MIME encoding type value.

This example demonstrates the code for creating a link. The **target** attribute is used to open the linked document in a new window.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML a Tag Example</title>
</head>
<body>
<a href="http://www.devguru.com" target="_blank"> DevGuru</a>
</body>
</html>
```

ATTRIBUTE: accesskey

accesskey="character"

The **accesskey** attribute allows you to designate a character on the keyboard that when pressed, along with the alt or meta key, will bring focus to an HTML element. By focus, we mean that the cursor will go to that element (for example, a link or an input box in a form).

The character can be any single character on the keyboard, including upper and lower case alphabets, numbers, symbols, or punctuation marks. You cannot repeat a character since each **accesskey** must uniquely identify an element.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML accesskey attribute example</title>
</head>
<body>
<form>
Simultaneously press the alt and 1 buttons and the cursor goes to the First Name input box.
<br />
Simultaneously press the alt and 7 buttons and the cursor goes to the Last Name input box.
<br />
Simultaneously press the alt and r buttons and the cursor goes to the Maiden Name input box.
<br />
<hr />
<br />
First Name: (accesskey = 1)
<br />
<input type="text" name="firstlname" size="30" accesskey="1" />
<br />
Last Name: (accesskey = 7)
</br>
<input type="text" name="lastname" size="30" accesskey="7" />
<br />
Maiden Name: (accesskey = r)
<br />
<input type="text" name="maidenname" size="30" accesskey="r" />
</form>
</body>
</html>
```

ATTRIBUTE: class

class="name"

The **class** attribute is used to assign the name of a style sheet class to a tag.

For example, all of the code samples on the **DevGuru** site are in blue colored text. This is done by enclosing the code sample inside a pair of **span** tag elements and designating a class of "CODE". In turn, the "CODE" class has been assigned a text color of blue in the style sheets file (a .css file) that is located in the Include/ directory on the **DevGuru** web site. Therefore, only one file is needed to define the appearance of the entire site.

There are four attributes common to most tags: **class, id, style,** and **title**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML class Attribute Example</title>
</head>
<body>
<span class="CODE">
The code example is blue.
</span>
</body>
</html>
```

Output:

The code example is blue.

ATTRIBUTE: dir

dir="direction"

The **dir** attribute is used to declare the direction that the text should run, either left to right (default) or right to left. This allows the internationalization of **HTML** since several important foreign languages read right to left, such as Chinese and Hebrew.

Also, see the **lang** core attribute that defines the language being used. However, note that using **lang** by itself does not specify language direction.

If used as an attribute of the **HTML** tag, it will apply to the entire document. When used with other tags, it only applies to the text under the influence of that tag. For example, when used with the **table** tag, the first column of the table will start on the right side and each additional column will be farther to the left.

This attribute should not be used to simply reverse a string. That is best done by using the VBScript StrReverse( ) function.

You may only use the following values:

dir="ltr"
The **ltr** value, which directs the text to flow from left to right, is the default.

dir="rtl"
The **rtl** value directs the text to flow from right to left.

Note that .html texts that contain foreign languages that use special characters will have to be saved as a Unicode file, rather than as an ANSI file, in order for the characters to be properly displayed. Please be aware that some browsers may not be capable of correctly displaying a Unicode file.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="rtl" xml:lang="he" lang="he">
<head>
<title>DevGuru XHTML dir attribute Example</title>
</head>
<body>
... place text of right to left language here ...
</body>
</html>
```

ATTRIBUTE: id

id="name"

The **id** attribute is used to assign a identifier value to a tag. Each **id** must be unique within the document and each element can only have one id.

In XHTML, the **id** attribute has essentially replaced the use of the **name** attribute. The value of the **id** must start with an alphabetic letter or an underscore. The rest of the value can contain any alpha/numeric chacracter.

Because it is unique, the value of the **id** can be the target of a URL or used to define a style rule.

There are four attributes common to most tags: **class, id, style,** and **title**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML id attribute Example</title>
</head>
<body>
<a href="#ckbx2">Go to Check Box Two</a>
<p />
<input type="checkbox" id="ckbx1" /> Check Box One
<p />
<input type="checkbox" id="ckbx2" /> Check Box Two
<p id="_mytext">
How now purple cow?
</p>
</body>
</html>
```

Output:
The use of **id** does not effect the output in any manner.

ATTRIBUTE: lang

lang="languagecode"

The **lang** attribute is used to define the base language to be used for displaying text and characters on a Web site. This allows an internationalization of **HTML** for a very large number of languages.

The languages are designated by a two letter code, such as "en" for English or "el" for Greek. One or more hypenated values can be tacked on to the initial two letter code to specify regional or ethentic variations, such as "en-us" for U.S. English.

Note that .html texts that contain foreign languages that use special characters will have to be saved as a Unicode file, rather than as an ANSI file, in order for the characters to be properly displayed. Please be aware that some browsers may not be capable of correctly displaying a Unicode file.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML lang Attribute Example</title>
</head>
<body>
<span lang="el">
... place text of foreign language here ...
</span>
</body>
</html>
```

EVENT: onblur

onblur="action"

The **onblurs** event occurs when the element that is in focus, loses that focus. You use the term, in blur, to signify that an element does not have focus. Since only one element can be in focus at one time, all other elements are in blur.

By focus, we mean that the cursor is at that element. For example, if the element was an input text box in a form, the cursor would appear inside the element and you could type and enter data into the box.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit**- runs a script when the form is submitted.

In this example, alert messages advise which element is in focus and which has lost focus.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onfocus and onblur Example</title>
</head>
<body>
<form>
Click back and forth between input text windows One and Two.
<br />
Alert messages will state who is in focus and who has lost focus (become blur).
<br />

One:
<input onfocus="JavaScript:alert('One is in focus')"
onblur="JavaScript:alert('One has lost focus')" />
<br />

Two:
<input onfocus="JavaScript:alert('Two is in focus')"
onblur="JavaScript:alert('Two has lost focus')" />

</form>
</body>
</html>
```

EVENT: onclick

onclick="action"

The **onclick** event is associated with the single click of a mouse on a selected HTML tag (element). Whenever you single click the element, the script code of the **onclick** is executed. This script can also call functions or subroutines which contain code that you want to run when the single click occurs.

The similar **ondblclick** event is associated with the double click of a mouse.

In this example, every time you single click on the **button** element, one line of JavaScript code is executed which causes an alert box to be displayed with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onclick Tag Example</title>
</head>
<body>
<input type="button" onclick="JavaScript: alert( 'onclick event' )" value="Click On This Button" />
</body>
</html>
```

EVENT: ondblclick

ondblclick="action"

The **ondblclick** event is associated with the double click of a mouse on a selected HTML element. Whenever you double click the element, the script code of the **ondblclick** is executed. This script can also call functions or subroutines which contain code that you want to run when the double click occurs.

The similar **onclick** event is associated with the single click of a mouse.

In this example, every time you double click the **button** element, one line of JavaScript code is executed which causes an alert box to be displayed with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML ondblclick Tag Example</title>
</head>
<body>
<input type="button" ondblclick="JavaScript: alert( 'ondblclick event' )" value="Double Click Me!" />
</body>
</html>
```

EVENT: onfocus

onfocus="action"

The **onfocus** event occurs when the element come into focus. By focus, we mean that the cursor is at that element. For example, if the element was an input text box in a form, the cursor would appear inside the element and you could type and enter data into the box.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit**- runs a script when the form is submitted.

In this example, alert messages advise which element is in focus and which has lost focus.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onfocus and onblur Example</title>
</head>
<body>
<form>
Click back and forth between input text windows One and Two.
<br />
Alert messages will state who is in focus and who has lost focus (become blur).
<br />

One:
<input onfocus="JavaScript:alert('One is in focus')"
onblur="JavaScript:alert('One has lost focus')" />
<br />

Two:
<input onfocus="JavaScript:alert('Two is in focus')"
onblur="JavaScript:alert('Two has lost focus')" />

</form>
</body>
</html>
```

EVENT: onkeydown

onkeydown="action"

The **onkeydown** event is associated with pressing down a key on the keyboard while the mouse is inside a selected HTML tag (element). In other words, the key down event occurred while the HTML element was in focus. Whenever you press down the key, the script code of the **onkeydown** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs. For example, if you only want numbers to be entered, you could capture the value of the key and take appropriate action if a non-numeric key was pressed.

The similar **onkeyup** event is associated with the release of a key after it was pressed down, and the **onkeypress** event is associated with the pressing of a key.

In this example, place the mouse inside the **textarea** element and type in a word. Every time you press a key down, one line of JavaScript code is executed which causes an alert box to be displayed with a message. Note how the alert comes up before the character is displayed inside the **textarea**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onkeydown Tag Example</title>
</head>
<body>
<textarea onkeydown="javascript: alert('A key was pressed down')" >
</textarea>
</body>
</html>
```

EVENT: onkeypress

onkeypress="action"

The **onkeypress** event is associated with press of a key on the keyboard while the mouse is inside a selected HTML element. In other words, the key press event occurred while the HTML element was in focus. Whenever you press the key, the script code of the **onkeypress** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs. For example, if you only want numbers to be entered, you could capture the value of the key and take appropriate action if a non-numeric key was pressed.

The similar **onkeydown** event is associated with the pressing down of a key, and the **onkeyup** event is associated with the release up of a key.

In this example, place the mouse inside the **textarea** element and type in a word. Every time you press a key, one line of JavaScript code is executed which causes an alert box to be displayed with a message. Note that the character does not appear in the **textarea** until after you have clicked okay in the alert.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onkeypress Tag Example</title>
</head>
<body>
<textarea onkeypress="javascript: alert('A key was pressed')">
</textarea>
</body>
</html>
```

EVENT: onkeyup

onkeyup="action"

The **onkeyup** event is associated with releasing up a key on the keyboard while the mouse is inside a selected HTML tag (element). In other words, the key up event occurred while the HTML element was in focus. Whenever you release the key, the script code of the **onkeyup** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs. For example, if you only want numbers to be entered, you could capture the value of the key and take appropriate action if a non-numeric key was pressed.

The similar **onkeydown** event is associated with the pressing down of a key, and the **onkeypress** event is associated with the pressing of a key.

In this example, place the mouse inside the **textarea** element and type in a word. Every time you release a key, one line of JavaScript code is executed which causes an alert box to be displayed with a message. Note how the alert comes up after the character is displayed inside the **textarea**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onkeyup Tag Example</title>
</head>
<body>
<textarea onkeyup="javascript: alert('A key was released up')">
</textarea>
</body>
</html>
```

EVENT: onmousedown

onmousedown="action"

The **onmousedown** event is associated with clicking down the mouse button while the mouse is inside a selected HTML tag (element). In other words, the mouse down event occurred while the HTML element was in focus. Every time you click down the mouse, while inside the element, the script code of the **onmousedown** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs.

The five mouse events are: **onmousedown, onmousemove, onmouseout, onmouseover,** and **onmouseup**.

In this example, place the mouse inside on the **button** element, click down, and hold it. Every time you click down, one line of JavaScript code is executed which causes an alert box to be displayed with a message. Finally, release the mouse and click okay in the alert.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onmousedown Tag Example</title>
</head>
<body>
<input type="button" onmousedown="JavaScript: alert( 'onmousedown event' )" value="Click Down With Mouse!" />
</body>
</html>
```

EVENT: onmousemove

onmousemove="action"

The **onmousemove** event is associated with moving the mouse around while inside a selected HTML tag (element). In other words, the mouse move event occurred while the HTML element was in focus. Every time you move the mouse around the element, the script code of the **onmousemove** is executed. This script can also call functions or subroutines that contain code that you want to run when this event occurs.

The five mouse events are: **onmousedown, onmousemove, onmouseout, onmouseover,** and **onmouseup**.

In this example, simply move the mouse inside on the **button** element.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onmousemove Tag Example</title>
</head>
<body>
<input type="button" onmousemove="JavaScript: alert( 'onmousemove event' )" value="Move The Mouse Inside This Button!" />
</body>
</html>
```

EVENT: onmouseout

onmouseout="action"

The **onmouseout** event is associated with moving the mouse out of a selected HTML tag (element). In other words, the mouse out event occurred while the HTML element was in focus. Every time you move the mouse out of the element, the script code of the **onmouseout** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs.

The five mouse events are: **onmousedown, onmousemove, onmouseout, onmouseover,** and **onmouseup**.

In this example, simply place the mouse inside the **button** element and then move it out. As soon as you move out, an alert appears with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onmouseout Tag Example</title>
</head>
<body>
<input type="button" onmouseout="JavaScript: alert( 'onmouseout event' )" value="Move The Mouse Out Of This Button!" />
</body>
</html>
```

EVENT: onmouseover

onmouseover="action"

The **onmouseover** event is associated with moving the mouse over a selected HTML tag (element). In other words, the mouse over event occurred while the HTML element was in focus. Every time you move the mouse over the element, the script code of the **onmouseover** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs.

The five mouse events are: **onmousedown, onmousemove, onmouseout, onmouseover,** and **onmouseup**.

In this example, simply move the mouse over on the **button** element. As soon as you move over, an alert is displayed with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onmouseover Tag Example</title>
</head>
<body>
<input type="button" onmouseover="JavaScript: alert( 'onmouseover event' )" value="Move The Mouse Over This Button!" />
</body>
</html>
```

CORE ATTRIBUTE: onmouseup

onmouseup="action"

The **onmouseup** event is associated with the releasing up portion of a click of the mouse while inside a selected HTML tag (element). In other words, the mouse up event occurred while the HTML element was in focus. Every time you release the mouse up, while in the element, the script code of the **onmouseup** is executed. This script can also call functions or subroutines which contain code that you want to run when this event occurs.

The five mouse events are: **onmousedown, onmousemove, onmouseout, onmouseover,** and **onmouseup**.

In this example, simply place the mouse inside the **button** element, click down, then release up. As soon as you release up, an alert appears with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onmouseup Tag Example</title>
</head>
<body>
<input type="button" onmouseup="JavaScript: alert( 'onmouseup event' )" value="Click And Release The Mouse While Inside This Button!" />
</body>
</html>
```

ATTRIBUTE: tabindex

tabindex="number"

The **tabindex** attribute specifies an integer that defines the rank in the tabbing order for the specified element when you use the keyboard to navigate (tab through) a page.

The lower the integer, the higher the ranking in the tabbing order. A higher rank is tabbed to before a lower rank. The integer cannot be negative or zero.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML tabindex Attribute Example</title>
</head>
<body>
<form>
Place the cursor in the top input window.
<br />
Then type the <b>tab</b> button on the keyboard.
<br />
Note that after the "First Name" we jump over "Last Name" to "Maiden Name".
<br />
After "Maiden Name" we return up to "Last Name".
<br />
<hr />
<br />
First Name: (tabindex = 1)
<br />
<input type="text" name="firstlname" size="30" tabindex="1" />
<br />
Last Name: (tabindex = 3)
</br>
<input type="text" name="lastname" size="30" tabindex="3" />
<br />
Maiden Name: (tabindex = 2)
<br />
<input type="text" name="maidenname" size="30" tabindex="2" />
</form>
</body>
</html>
```

ATTRIBUTE: title

title="string"

The **title** attribute is used to assign a name to a tag. This name can be any string of characters or words. Further, you can use the HTML character entities in the name. For example, you could use

 which will cause a line break (just like the **br** tag) in the name.

If you wish, you can reuse the same title values repeatedly inside an HTML document, or you could have each value be unique. In contrast, the **id** attribute requires that each id value must be unique.

Do not confuse this attribute with the **title** tag which is used to assign a title to a web document.

The **title** attribute is used by ToolTips on Internet Explorer to create a small window that automatically displays the text in the **title** when you run the mouse over a hyperlink or a mapped hyperlink image. This can be done with both the **a** and **area** tags. By default, this window is only displayed for about ten seconds. (This concept can also be applied to Netscape by using JavaScript functions. Please read the Knowledge Base article for details.)

There are four attributes common to most tags: **class, id, style,** and **title**.

Mouse over the link displayed in the output.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML title Attribute Example</title>
</head>
<body>
Please refer to the DevGuru Knowledge Base article:
<br />
<A href="/features/knowledge_base/a100216.html"
title="ToolTips are &#10;&#13;really neat!">
A100216: Adding ToolTips to your HTML pages.
</a>
</body>
</html>
```

Output:

Please refer to the DevGuru Knowledge Base article:
A100216: Adding ToolTips to your HTML pages.

As another example, consider the situation where you have a large number of tables or tables inside of tables. By placing a **title** in both the opening and closing **table** tags, it becomes much easier to identify and keep track of the start and stop of individual tables inside your code.

Code:

```
<table title="PhoneNumbers">
... table entries ...
```

</table title="PhoneNumbers">

TAG: abbr

<abbr> ... </abbr>

The **abbr** tag is used to signify that the enclosed text is an abbreviation of a longer word or phrase.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML abbr Tag Example</title>
</head>
<body>
<abbr>lb</abbr> is an abbreviation for pound
</body>
</html>
```

Output:

lb is an abbreviation for pound

ATTRIBUTE: style

style="string"

The **style** attribute is used to assign style sheet properties to the specific element containing the attribute.

Do not confuse this attribute with the **style** tag which is used to create document-level style sheet rules. It is referred to as document-level because the style rules will apply to the entire HTML page. In contrast, the **style** attribute behaves similarly to an inline style rule and only effects a small portion the HTML document.

There are four attributes common to most tags: **class, id, style,** and **title**.

In this example, the text enclosed by the **span** tags will be brown, size 24px and in Times font.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML style Attribute Example</title>
</head>
<body>
<span style="color: brown; font-size: 24px; font-family: times;">
When in the course of human events...
</span>
</body>
</html>
```

Output:

When in the course of human events...

TAG: acronym

<acronym> ... </acronym>

The **acronym** tag is used to signify that the enclosed text is an acronym which is a special type of abbreviation formed by using the first letter in each word of a phrase.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML acronym Tag Example</title>
</head>
<body>
<acronym>MIME</acronym> means Multipurpose Internet Mail Extension
</body>
</html>
```

Output:

MIME means Multipurpose Internet Mail Extension

TAG: address

<address> ... </address>

The **address** tag is used to signify that the enclosed text is an address. Most browsers display the address in a special manner, such as in italics. However, you can use style sheets to alter the appearance as you see fit. The text between the **address** tags can contain any HTML tag.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML address Tag Example</title>
</head>
<body>
<address>
DevGuru<br />
5106 Kingston Pike<br />
Knoxville, TN 37919<br />
</address>
</body>
</html>
```

Output:

DevGuru
5106 Kingston Pike
Knoxville, TN 37919

TAG: applet

<applet> ... </applet>

The **applet** tag is used to download and execute a Java applet. The applet is treated like an image and is rendered into the flow of the text without a line break. You can define the size of the region (the applet window) in which the applet will appear. By using the attributes for **applet**, style sheets code, and other HTML tags, you can create a presentation and appearance for the applet in the text flow to suit your needs.

Java applets can be very slow to load and may not display in many browsers.

Many Java applets require that document-specific parameters be provided to control the execution. You can use the **param** tag, inserted between the opening and closing **applet** tags, to define these parameters.

The **applet** tag was officially deprecated effective with HTML version 4.0. It was replaced with the **object** tag. However, this tag in such extensive use on the Internet that it has been resurrected.

The closing tag is mandatory.

Core Attributes

class **id** **style** **title**

Attributes

align
The **align** attribute is used to position the applet window with respect to any surrounding text, images, or other applets.

alt
The **alt** attribute provides a text message that will be displayed (in place of the applet) on browsers that cannot display an applet.

archive
The **archive** attribute is a comma-separated list of URLs of Java classes that must be pre-loaded into the local user cache before the applet can be displayed.

code
The **code** attribute is mandatory and specifies the relative URL of the Java class (a .class type file) that must be executed to display the applet. Specifically, the URL is relative to the applet base URL (otherwise the applet will not load).

codebase
The **codebase** attribute is used to specify the URL of the directory where the Java class is stored. If the directory is different from where the HTML code is stored, you must use this attribute.

height
The **height** attribute is used to set the vertical extent of the applet in pixels.

hspace

The **hspace** attribute is used to add additional blank space in pixels to each side of the applet.

mayscript
The **mayscript** attribute is only recognized by Netscape. It permits the applet to access JavaScript code.

name
The **name** attribute is used to assign a unique string of characters as the name of the applet.

object
The **object** attribute is not supported by most browsers. It is the name of a resource where a serialized version of the applet is stored. Do not confuse this attribute with the **object** tag.

vspace
The **vspace** attribute is used to add additional blank space in pixels both above and below the applet.

width
The **width** attribute is used to set the horizontal extent of the applet in pixels.

This example displays a Java applet clock.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML applet Tag Example</title>
</head>
<body>
<applet codebase="clock/classes/" code="JavaClock.class" width="150" height="150">
<param name="timezone" value="eastern" />
</applet>
</body>
</html>
```

TAG: area

<area> ... </area>

The **area** tag is used (along with the **<map>** tag) to create a client-side image map which is divided into two or more mouse-sensitive regions. When you click on a region in the map, an attribute contained in the **area** tag can direct an action to occur. For example, a hyperlink can be created in each region using the **href** attribute. When you click onto a region, the hyperlink will send you to a target page just like any link. Also, since the map is mouse-sensitive, the various onmouse cursor events can be very useful.

The **area** tag must be contained inside the **map** element. The separate closing tag is mandatory.

Attributes and Events

accesskey **class** **dir** **id** **lang** **onblur** **onclick** **ondblclick** **onfocus** **onkeydown** **onkeypress** **onkeyup** **onmousedown** **onmousemove** **onmouseout** **onmouseover** **onmouseup** **style** **tabindex** **title**

Attributes

coords
The **coords** attribute defines the (x,y) coordinates of each shape in the image map. The coordinates are the x and y offset in pixels from the upper left corner (0,0) of the image. The number of coordinates you need to provide depends of the shape specified by the **shape** attribute.

href
The **href** attribute is the URL address for the link. Each **area** tag in the image map can have a link.

nohref
The **nohref** attribute specifies that the region defined by the **area** tag cannot be a link.

shape
The **shape** attribute defines the geometric shape of the image map. It can be a circle (**circle** or **circ**), polygon (**polygon** or **poly**), or a rectangle (**rectangle** or **rect**)

target - not allowed in strict XHTML
The **target** attribute specifies the name of the frame or window in which the target page should appear when a link is clicked. The four reserved names are **_blank, _parent, _self,** and **_top**.

The **area** and **map** tags are used on the **DevGuru** site to create image mapped links for the indexes to the various Quick References. For example, this is the code used for the ASP Quick Reference. These links are active and will take you to the ASP indexes.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML area Tag Example</title>
```

```
</head>
<body>
<img border="0" src="images/asp_method_event_property.gif" usemap="#methodproperty" />
<br />
<map name="methodproperty">
<area shape="rect" coords="192,0,248,26" href="/technologies/asp/quickref/properties.html" />
<area shape="rect" coords="133,0,191,26" href="/technologies/asp/quickref/collections.html" />
<area shape="rect" coords="90,0,134,22" href="/technologies/asp/quickref/events.html" />
<area shape="rect" coords="43,0,89,22" href="/technologies/asp/quickref/methods.html" />
</map>
</body>
</html>
```

TAG: b

** ... **

The **b** tag causes the designated text to be displayed in a bold weight. If bold is not available for a particular font, the browser may impose some other physical change to the text, such as underline.

This can also be done using the style sheets **font-weight** property.

The separate closing tag is mandatory.

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Attributes

None

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML b Tag Example</title>
</head>
<body>
<b>DevGuru</b> products are great!
</body>
</html>
```

Output:

DevGuru products are great!

TAG: base

<base />

The **base** tag is used to define the base URL for the HTML document. This allows the use of relative URL addresses elsewhere in the document. Interestingly, this base URL does not have to be an absolute URL, but rather, it can be a relative URL.

Also, when a browser arrives from another site to your site via a link, you can use the **target** attribute of this tag to ensure that your site does not appear inside a frame or window of the previous site. In other words, your site will always appear as the top window.

The **base** tag must be placed between the opening and closing **head** tags.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

href
The **href** attribute is mandatory. It is used to define the base URL.

target - not allowed in Strict XHTML
The **target** attribute specifies the default name of the frame or window in which all of the target pages should appear when a link is clicked on your site. However, this can be overridden by a **target** attribute that is inside an **a** tag. The four reserved names are **_blank, _parent, _self,** and **_top**.

In this example, the link in the **body** resolves to:
http://www.devguru.com/Technologies/HTML/quikref/html_index.html

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML base Tag Example</title>
<base href="http://www.devguru.com/">
</head>
<body>
<a href="/Technologies/HTML/quikref/html_index.html">HTML Index</a>
</body>
</html>
```

This code directs that your site always appears as the top window.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML base Tag Example</title>
<base target="_top">
```

```
</head>
<body>
...
</body>
</html>
```

TAG: bdo

<bdo> ... </bdo>

The **bdo** tag is used to change the direction that the text reads, from left to right (the default) over to right to left, or vice versa. The purpose of this tag is to allow HTML to correctly display a great variety of languages, such as Chinese and Hebrew (which are read right to left), in the same document.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML bdo Tag Example</title>
</head>
<body>
<bdo dir="rtl"">
Very few browsers support this tag.
<br />
.gat siht proppus sresworb wef yreV
</bdo>
</body>
</html>
```

TAG: big

<big> ... </big>

The **big** tag causes the designated text to be displayed in a larger font size. The actual amount of change will be determined by the browser. If the text is already at maximum size, this attribute will have no effect.

The separate closing tag is mandatory.

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML big Tag Example</title>
</head>
<body>
<small>think</small> very <big>BIG</big>
</body>
</html>
```

TAG: blockquote

<blockquote> ... </blockquote>

The **blockquote** tag is used to offset a block of text (a citation) from the rest of the document. The exact appearance will be browser dependent. However, the citation is usually indented on both the right and left margins, may be in italics, and is preceded and followed by paragraph breaks. You will still need to use the **br**, **p**, and possibly other HTML tags to effect the textual display and appearance that you desire. In fact, you can use any other HTML tag inside a **blockquote**, including **cite** and **q**.

It is recommended that you use the **blockquote** tag for long citations and the **cite** tag for short citations.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

cite
The **cite** attribute is used to provide a reference to the URL that is the source of the citation.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML blockquote Tag Example</title>
</head>
<body>
Some writers make you think:
<blockquote cite="http://www.hermanmelville.com">
Push not off from that isle,<br />
thou canst never return...<p />
         - Herman Melville
</blockquote>
</body>
</html>
```

Output:

Some writers make you think:

> Push not off from that isle,
> thou canst never return...
>
> - Herman Melville

TAG: body

<body> ... </body>

The **body** tag is used to signify the start and stop of the contents of your HTML document. A well-formed XHTML document must contain properly nested and closed **html, head, title,** and **body** tags.

If you are using frames, then you use the **frameset** tag in place of the **body** tag.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

This example demonstrates the code structure for creating an XHTML document.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML body Example</title>
</head>
<body>
... place document contents between the body tags
</body>
</html>
```

TAG: br

**
**

The **br** tag is used to create a line break and is analogous to a carriage return. The flow of the display of the text and any images is halted on the current line, and the flow resumes on the next line starting at the left margin.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibility.

Attributes and Events

id **class** **style** **title**

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML br tag</title>
</head>
<body>
My pseudotrophus "Acei" cichlid came from Lake Malawi in Africa.
<br />
The Showa Koi in my pond came from Japan.
</body>
</html>
```

Output:

My pseudotropheus "Acei" cichlid came from Lake Malawi in Africa.
The Showa Koi in my pond came from Japan.

TAG: button

<button> ... </button>

The **button** tag is used to create a button form control (field). Control is a technical term that refers to the various elements (buttons, check boxes, radio buttons, text areas) that can be used inside a form to gather information.

The four tags that can be used to build a **form** are:

button input select textarea

Specifically, the **button** tag is used to create a button in a **form** that can have content, consist of an image, and have the illusion of 3-dimensions. The buttons created using the **input** tag, including the type="image", cannot have content or display with a 3-D appearance. (Otherwise, there is little difference.)

Any text or images placed between the opening and closing **button** tags will be displayed on the button. The text can include most HTML tags and they will perform as expected. The one major exception is that you cannot display an image map using the **map** and **area** tags.

A **button** element should not contain other **button** tags, nor should it contain **fieldset, form, input, label, select,** or **textarea** tags.

The separate closing tag is mandatory.

Attributes and Events

accesskey class dir id lang ondblur onclick ondblclick onfocus onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style tabindex title

disabled
The **disabled** attribute is a Boolean value that, if present, prevents the **button** from functioning. In some browsers, the **button** will appear to be greyed out.

name
The **name** attribute is used to assign a unique string of characters as the name of the **button**.

type
The **type** attribute defines the type of button. The permitted values are **button, reset,** or **submit**.

value
The **value** attribute assigns a value to the **button** element. This can be changed later using script code.

When you click on either button with the mouse, an alert is displayed.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
```

```
<head>
<title>DevGuru XHTML button Tag Example</title>
</head>
<body>
<form action="html_button.html" method="get">
<button type="button" onclick="JavaScript: alert('Image Button')">
<img src="images/guru.gif">
</button>
<p />
<button type="button" onclick="JavaScript: alert('Text Button')">
Please Click This Button!
</button>
</form>
</body>
</html>
```

TAG: input

<input />

The **input** tag is used to create individual form controls (fields). Control is a technical term that refers to the various elements (buttons, check boxes, radio buttons, text areas) that can be used inside a **form** to gather information.

The four tags that can be used to build a **form** are:

button input select textarea

The **form** is the heart of an e-commerce page and the **input** tag is the heart of a **form**. There are ten types of form controls that can be created using the **input** tag. Appropriately, you must use the **type** attribute to determine the specific type of the control (field). The types are:

button
checkbox
file
hidden
image
password
radio
reset
submit
text

Any number of **input** tags can be placed anywhere between a pair of opening and closing **form** tags to create the desired appearance of the form.

This tag is self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

accept
The **accept** attribute is used with the **type="file"** control to specify what kind of files (including the path) that can be attached to the **form**. It is a comma-delimited list. For example: **accept="images/*.gif"**

accesskey
The **accesskey** attribute allows you to designate a key on the keyboard that when pressed, along with the alt or meta key, will bring focus to the **input** form element.

alt
The **alt** attribute displays an alternative text for browsers that cannot display an image.

checked

The **checked** attribute, if present, allows a checkbox or radio button to be preselected (and have a preselected **value**).

disabled
The **disabled** attribute is a Boolean value that, if present, prevents the form control (field) from functioning. In some browsers, the control (field) will appear to be greyed out.

ismap
The **ismap** attribute is a Boolean value that, when present, signifies that the image is being used as a mouse-clickable server-side image map.

maxlength
The **maxlength** attribute sets the maximum number of characters that can be entered into a one-line text window.

name
The **name** attribute is a string of characters that is used to label a form control (field) with a name. The name must be unique to that document and cannot be reused.

onblur
The **onblur** event allows the execution of JavaScript code when an element loses focus (for example, the mouse was clicked onto another element or a tab navigation directed the cursor elsewhere).

onchange
The **onchange** event allows the execution of JavaScript code when the form control (field) has been changed and loses focus.

onfocus
The **onfocus** event allows the execution of JavaScript code when an element comes into focus (for example, the mouse was clicked onto the element or a tab navigation brought the cursor to the element).

onselect
The **onselect** event allows the execution of JavaScript code when the form control (field) gains focus.

readonly
The **readonly** attribute allows you to display a value in a form control (field) that cannot be changed by the user.

size
The **size** attribute sets the width of a one-line text window by defining how many characters can fit in the window.

src
The **src** attribute is the URL address or directory/file (path/name) where an image file is being stored.

tabindex
The **tabindex** attribute specifies an integer that defines the rank in the tabbing order for the specified element when you use the keyboard to navigate (tab through) a page. This attribute is poorly supported.

type
The **type** attribute determines the specific control.

Note that by using style sheets you can effect the appearance of the various types of controls. However, the exact appearance will be browser dependent.

type="button"
The **type="button"** control creates a rectangular button that can be clicked when you wish a desired action to occur. For example, you could use buttons to navigate through a **recordset** data source being displayed on a web page. (See the RDS objects in the ADO Quick Reference.) If you want to reset or submit the **form**, you should use the **type="reset"** and **type="submit"** controls.

A value for the **name** attribute is required. You can use the **value** attribute to assign a text that will be displayed on the face of the button. The default color is gray, but that can be changed using style sheets (see example at bottom of page). The text displayed on the button will set the minimum size, but you can set a different size using style sheets or the **size** attribute.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<input type="button" name="btn1" value="click me" />
</body>
</html>
```

type="checkbox"
The **type="checkbox"** control creates a small, square button that can be selected or unselected by the user with a click of the mouse. When the checkbox is selected, a check mark appears within the box.

A value for the **name** attribute is required. You must use the **value** attribute to assign a value that is associated with the checkbox. You can also use the **checked** attribute to precheck the checkbox. When the checkbox has been selected, the **value** is saved as part of the form information that is passed to the appropriate file and/or database when the form is submitted. However, the **value** for any unselected checkbox is not saved or passed.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="checkbox" name="ckbx1" value="true" />
</form>
</body>
```

```
</html>
```

type="file"
The **type="file"** control is used to attach a file to a **form**. When the **form** is submitted, the file is passed along with all the other contents of the **form**.

This control displays an input window and a browse button. By clicking on the browse button, a user can search your directories for the desired file. When you find it, a double mouse click will select the file. Alternately, you can enter the path and name of the file into the input window. A value for the **name** attribute is required. You do not use the **value** attribute.

To use the **type="file"**, you must set both the **enctype** and **method** attributes of the **form** tag exactly as shown in the following code snippet.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form method="post" enctype="multipart/form-data" action="next.asp">
<input type="file" name="file1" />
</form>
</body>
</html>
```

type="hidden"
The **type="hidden"** control is used to add content to the **form** that cannot be viewed or changed by the user. When the **form** is submitted, the hidden content is passed along with all the other contents of the **form**.

The hidden content is essentialy a name/value pair. You must provide a name using the **name** attribute. You also must provide a **value**. This value can be any textual or numeric information.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="hidden" name="SocSecNum" value="123-45-6789" />
</form>
</body>
</html>
```

type="image"
The **type="image"** control allows you to use a graphic image or a picture as the equivalent of a **type="submit"** button. For example, you could create your own submit gif. Indeed, the

specified image will perform exactly as a submit button, when you click the image, the **form** is submitted.

The **type="image"** also allows you to display a mouse-sensitive image map within a **form**. (This requires that the browser be able to capture and process the X/Y coordinates of the mouse.)

You must use the **src** attribute to list the location of the image file. You can use the **alt** attribute to provide text for browsers that cannot display an image. Some browsers, by default, place a border around the image.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="image" src="images/submit.gif" onclick="submitform()" name="submitgif" />
</form>
</body>
</html>
```

type="password"
The **type="password"** control displays a one line input window which is used to collect a password as a masked text. A masked text means that when the user types in the password, the actual characters being typed do not appear in the window. Rather an asterisk * is displayed in place of each character in the password.

You can use a **type="text"** control to gather the user name.

The default window size is browser dependent and is usually 20-30 characters in length. However, you can set the size using style sheets or the **size** attribute. The **maxlength** attribute can be used to set the maximum number of characters that will be accepted by the control (field).

This is only a semi-secure method for gathering a password. While the password is not visible on screen, when the form is submitted, the password is passed unencrypted as part of the contents of the form.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="password" />
</form>
```

```
</body>
</html>
```

type="radio"
The **type="radio"** control is used to create a small round circular button with a white center that can be selected or unselected by the user with a click of the mouse. When the radio button is selected, a black dot appears in the center of the circle.

The concept is to present the user with a selection of choices, but only to allow one choice to be made, for example, picking the color of a tee-shirt being ordered on a web site from a selection of twelve different colors.

Each related group of radio buttons must have the same **name**. Further, each radio button in a related group must have a different **value**. When a specific radio button is selected, only the **value** for the selected button is passed as part of the contents of the **form** upon submission. The values for the unselected radio buttons are not passed.

You cannot change the size or color of a radio button. You can use the **checked** attribute to preselect one of the radio buttons.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="radio" name="radio1" value="one" checked="checked" />
<input type="radio" name="radio1" value="two" />
<input type="radio" name="radio1" value="three" />
<input type="radio" name="radio1" value="four" />
</form>
</body>
</html>
```

type="reset"
The **type="reset"** control creates a rectangular button that can be clicked when the user wants to clear all of the fields in a **form**. The **form** is redisplayed in its original presentation.

By default the reset button is a gray color and has the text "Reset" displayed inside the button. However, you can change the color using style sheets and you can use the **value** button to provide a different text for the button. The text displayed on the button will set the minimum size, but you can set a different size using style sheets or the **size** attribute.

A **name** is not required, but it is becoming a very common coding practice to give a unique name to every control (field) in a **form**.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
```

```
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="reset" value="Clear The Form" name="reset1" />
</form>
</body>
</html>
```

type="submit"
The **type="submit"** control creates a rectangular button that can be clicked when the user wants to submit the contents of a **form** for processing.

A **form** can be submitted to the same page or to another URL. This is set inside the **form** tag by using the **action** attribute. However, if it is important that certain contents of the **form** be filled out correctly, then you could use the **onclick** core attribute in the submit button to direct the submission to a JavaScript function that proofs the contents. Alternatively, you could use an **onsubmit** attribute inside the **form** tag to direct the submission to the JavaScript function. This is called client-side form verification.

By default the submit button is a gray color and has the text "Submit Query" displayed inside the button. However, you can change the color using style sheets and you can use the **value** button to provide a different text for the button. The text displayed on the button will determine the minimum size, but you can set a different size using style sheets or the **size** attribute.

A **name** is not required, but it is becoming a very common coding practice to give a unique name to every control in a **form**.

If you assign a **value**, when the **form** is submitted, that value will be passed as part of the contents.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="submit" name="submit" onclick="submitform()" value="Submit" />
</form>
</body>
</html>
```

type="text"
The **type="text"** control displays a one-line text input window into which the user can type the requested information (such as an address or phone number). If the requested information will require more than one line to enter, you should use a **textarea** tag rather than a **type="text"** control.

The default window size is browser dependent and is usually 20-30 characters in length. It is strongly recommended that you set the size of a text window rather than using the default. This

will allow you to effect the appearance of the **form** in a pleasing and useful manner. Further, the appearance of the **form** will be fairly independent of the browser being used.

The default window size is browser dependent and is usually 20-30 characters in length. However, you can set the size using style sheets or the **size** attribute. The **maxlength** attribute can be used to set the maximum number of characters that will be accepted by the control (field). You can use the **value** attribute to display a text inside the window, but this is optional.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>
</head>
<body>
<form>
<input type="text" name="text1" size="50" />
</form>
</body>
</html>
```

value
The **value** attribute assigns an initial value, such as a text or number, to a form control (field). This can be changed later using script code.

usemap
This **usemap** attribute specifies the name of an image map that you wish to associate with the element. This indicates that the image is being used as a mouse-clickable client-side image map.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML input Tag Example</title>

<script type="text/javascript">
function submitform()
{ alert("Thank You For Submitting The Form") }
</script>

</head>
<body>
MY SAMPLE FORM:
<p />
<form method="post" enctype="multipart/form-data" action="next.asp">

<input type="button" name="btn2" value="click me" style="background-color:yellow;" /> Button
<p />
<input type="checkbox" name="ckbx2" value="true" /> Check Box
<p />
<input type="file" name="file2" /> Attach a File
```

```
<p />
<input type="hidden" name="SocSecNum" value="123-45-6789" />
<p />
<input type="password" /> Password
<p />
<input type="radio" name="radio2" value="one" checked="checked" /> First Radio Button
<br />
<input type="radio" name="radio2" value="two" /> Second Radio Button
<br />
<input type="radio" name="radio2" value="three" /> Third Radio Button
<br />
<input type="radio" name="radio2" value="four" /> Forth Radio Button
<p />
<input type="text" name="submit" name="text2" size="50" /> Text
<br />
<input type="text" name="submit" name="text3" size="25" /> Text
<br />
<input type="text" name="submit" name="text4" size="2" /> Text
<p />
<input type="reset" value="Clear The Form" style="color:white; background-color:blue;" />
Reset Button
<br />
<input type="submit" name="submit" name="submit2" onclick="submitform()"
value="Submit" style="color:white; background-color:red;" />
Submit Button Or Use An Image Submit Button
<input type="image" src="images/submit.gif" onclick="submitform()" name="submitgif" />

</form>
</body>
</html>
```

TAG: select

<select> ... </select>

The **select** tag is used to create a pull-down list form control (field). Control is a technical term that refers to the various elements (buttons, check boxes, radio buttons, text areas) that can be used inside a form to gather information.

The four tags that can be used to build a **form** are:

button input select textarea

The **select** tag is used to delimit the start and stop of a multiple choice pull-down menu. The default for this tag is to create a one-line window with a button on the left side. (Optionally, you can display a multi-line window.) When you single click onto the button with a mouse, the one-line window expands to display a list of items. You can then select one item from the list by simply clicking on the item with the mouse. (Optionally, you can allow more than one item to be selected.) Note that the user can select or unselect the item(s) any number of times. When an item(s) is selected, it is added to the contents of the **form** and when the **form** is submitted, the selected item(s) will be passed along with all of the other contents.

The beauty of the **select** form control is that it can occupy as little as one line of space on a **form**, yet allow the user to choose from a very large selection of items. For example, your pull-down could be a list of all fifty states. As another example, note the pull-down list at the top of this HTML Quick Reference page. When you select an item, it acts as a link and directs you to another page in this HTML Quick Reference.

You must use the **option** or the **optgroup** tags to define each item in the menu.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

disabled
The **disabled** attribute is a Boolean value that, if present, prevents the form control (field) from functioning. In some browsers, the control (field) will appear to be greyed out.

multiple
The **multiple** attribute allows the user to choose more than one item in the pull-down. The default is to be able to choose only one item.

name
The **name** attribute is a string of characters that is used to label a form control (field) with a name. The name must be unique to that document and cannot be reused.

onblur
The **onblur** event allows the execution of JavaScript code when an element loses focus (for example, the mouse was clicked onto another element or a tab navigation directed the cursor elsewhere).

onchange
The **onchange** event allows the execution of JavaScript code when the form control (field) has been changed and loses focus.

onfocus
The **onfocus** event allows the execution of JavaScript code when an element comes into focus (for example, the mouse was clicked onto the element or a tab navigation brought the cursor to the element).

size
The **size** attribute allows the user to have more than one item on display in the pull-down window. Up and down buttons will allow you to scroll through the list. The default is to display only one item.

tabindex
The **tabindex** attribute specifies an integer that defines the rank in the tabbing order for the specified element when you use the keyboard to navigate (tab through) a page. This attribute is poorly supported.

This is a simple pull-down that allows the choice of one item.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML select Tag Example</title>
</head>
<body>
Gold Fish Types
<br />
<select>
<option selected>Comet</option>
<option>Oranda</option>
<option>Ranchu</option>
<option>Ryukin</option>
</select>
</body>
</html>
```

Output:

Here is the code for the pull-down at the top of this page.

Code:

```
...
<select name="SeeAlsoSelect" onChange=ViewCrossReference(this); style="height: 22px; width: 240px">
<option value="xhtml_form.html" selected>form tag</option>
<option value="xhtml_input.html">input tag</option>
<option value="xhtml_button.html">button tag</option>
<option value="xhtml_optgroup.html">optgroup tag</option>
<option value="xhtml_option.html">option tag</option>
```

```
<option value="xhtml_textarea.html">textarea tag</option>
</select>
<a href="javascript:ViewCrossReference(document.MainForm.SeeAlsoSelect)" border=0>
<img src="images/go.gif" border="0" height=22 width="20" value="Go">
</a>
...
```

This JavaScript function is placed in the head element.

```
...
<script type="text/javascript">
function ViewCrossReference (selSelectObject)
{
  if (selSelectObject.options[selSelectObject.selectedIndex].value != "")
  {
    location.href=selSelectObject.options[selSelectObject.selectedIndex].value
  }
}
</script>
...
```

TAG: textarea

<textarea> ... </textarea>

The **textarea** tag is used to create a multi-line text input window form control (field). Control is a technical term that refers to the various elements (buttons, check boxes, radio buttons, text areas) that can be used inside a form to gather information.

The four tags that can be used to build a **form** are:

button input select textarea

The **textarea** control should be used when the user needs to input more than one line of data. If you only expect the user to input a few words, you should use the **input** tag with **type="text"** control that creates a one-line text input window.

Any text or HTML code that occurs between the opening and closing **textarea** tags will appear inside the textarea window.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

accesskey
The **accesskey** attribute allows you to designate a key on the keyboard that when pressed, along with the alt or meta key, will bring focus to the **textarea** element. This attribute is poorly supported.

cols
The **cols** attribute sets how many characters will be visible across the width of the **textbox** window.

disabled
The **disabled** attribute is a Boolean value that, if present, prevents the form control (field) from functioning. In some browsers, the control (field) will appear to be greyed out.

name
The **name** attribute is a string of characters that is used to label a form control (field) with a name. The name must be unique to that document and cannot be reused.

onblur
The **onblur** event allows the execution of JavaScript code when an element loses focus (for example, the mouse was clicked onto another element or a tab navigation directed the cursor elsewhere).

onchange
The **onchange** event allows the execution of JavaScript code when the form field or element has been changed and loses focus.

onfocus

The **onfocus** event allows the execution of JavaScript code when an element comes into focus (for example, the mouse was clicked onto the element or a tab navigation brought the cursor to the element).

onselect

The **onselect** event allows the execution of JavaScript code when the form field or element gains focus.

readonly

The **readonly** attribute allows you to display a text or value in a form control (field) that cannot be changed by the user.

rows

The **rows** attribute sets how many rows will be displayed in the **textarea** window.

tabindex

The **tabindex** attribute specifies an integer that defines the rank in the tabbing order for the specified element when you use the keyboard to navigate (tab through) a page. This attribute is poorly supported.

This example shows two **textarea** windows, the first is not sized and the second is sized.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML textarea Tag Example</title>
</head>
<body>
<form action="html_textarea.html" method="post">
<textarea name="textarea12">
The default size is 2 rows, 20 columns
</textarea>
<br />
<textarea cols="45" rows="4" name="textarea13">
This is 4 rows, 45 columns
</textarea>
</form>
</body>
</html>
```

EVENT: onselect

onselect="action"

The **onselect** event occurs when a single character in a text is highlighted by using the mouse to drag over the character. This is called a select. Whenever the select occurs, the script code associated with the **onselect** event is executed. This script can call JavaScript functions that contain code that you want to run when the select occurs.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit**- runs a script when the form is submitted.

In this example, every time you use the mouse to highlight the text in the input window, an alert appears.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onselect Example</title>
</head>
<body>
<form>
Please use the mouse to highlight the text in the input window
<br />
Note that as soon as you highlight one character, the alert appears
<br />
<input type="text" value="DevGuru" id="text1" name="text1"
onselect="JavaScript:alert('Thank you for choosing DevGuru!')" />
</form>
</body>
</html>
```

TAG: head

<head> ... </head>

The **head** tag is the HTML document header. It serves as a container for other tags that control the contents and appearance of the main body of the document.

A well-formed XHTML document must contain properly nested and closed **html, head, title,** and **body** tags. The **head** tag is inserted immediately after the **html** tag, but before the **body** or **frameset** tags.

The **head** element can contain any of the following HTML tags in any order:

base	set base URL
link	set document link
meta	document keywords
script	script code
style	set style sheet rules
title	name of document

The separate closing tag is mandatory.

Attributes and Events

dir lang

profile
The **profile** attribute is a list of one or more (comma separated) URL addresses of the meta data profiles.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML head Tag Example</title>
<base href="http://www.devguru.com/" />
<link rel="stylesheet" type="text/css" href="include/StylesDefinitions.css" />
</head>
<body>
...
</body>
</html>
```

TAG: link

<link>

The **link** tag is used to establish a relationship between the current document and one or more other related documents. For example, if you have a Cascading Style Sheets .css file that defines the style for an entire web site, then each file in the site would be linked to the .css file using the **link** tag.

This tag can be used more than once (i.e., one link per tag). Each occurrence of this tag must be placed inside the **head** element. Remember, a value for the **href** attribute is required.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

charset
The **charset** attribute is used to specify the character encoding used on the page that is the target of the link. Character encoding defines how a sequence of bytes is to be converted into characters for display.

href
The **href** attribute is a valid URL address for another document. This attribute is required.

hreflang
The **hreflang** attribute can only be used when the href attribute is also being used. It specifies the language of the page that is the target of the link.

media
The **media** attribute specifies which medium the links apply to. Permitted values include: **all, aural, braille, handheld, print, projection, screen, tty,** and **tv**.

rel
The **rel** attribute is currently not fully supported by most browsers. It is a space-separated list of one or more values that specify the relationship from the source page to the target for a link. Some of the proposed values are, **appendix, bookmark, chapter, contents, copyright, glossert, help, index, next, prev, section, stylesheets,** and **subsection**.

rev
The **rev** attribute is currently not fully supported by most browsers. It is a space-separated list of one or more values that specify the relationship from the target page to the source for a link. Perhaps the most useful value is **relation**.

target
The **target** attribute specifies the name of the frame or window in which the target page should appear when a link is clicked. The four reserved names are **_blank, _parent, _self,** and **_top**.

type
The **type** attribute specifies the MIME (Multipurpose Internet Mail Extension) type of the linked

document. Two permitted type values include: **text/css** and **text/javascript**.

This example demonstrates the code for creating two different types of links.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title> XHTML link Tag Example </title>
<link rel="stylesheet" type="text/css" href="include/StylesDefinitions.css" />
<link rev="relation" href="http://www.fly-by-day-computing.com" />
</head>
<body>
...
</body>
</html>
```

TAG: meta

<meta />

The **meta** tag is primarily used to list information (meta data) about the current HTML document. This information can be anything you want users to know about your site including author names, keywords, subject matter, and topics of interest. This meta data is used extensively by the various Internet search engines.

The **meta** tag can also be used with the **http-equiv** attribute to simulate HTTP headers.

This tag presents the information as name/content paired values by using the **name** and **content** attributes. You can have more than one **meta** element per page.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

dir **lang**

content
The **content** attribute is the content portion of the name/content value. There are no defined values for this attribute. You may choose any words or text you desire. For example, you could choose "keywords" for the **name** and then use the **content** attribute to list keywords that you feel are important to your site.

http-equiv
The **http-equiv** attribute can be used in place of the **name** attribute to create an http-equiv/content value pair. It provides information used to generate HTTP headers (however, most proxy servers ignore this attribute). The permitted values are **content-type, expires, refresh,** and **set-cookie**.

name
The **name** attribute is the name portion of the name/content value. There are no defined values for this attribute. You may choose any word you desire. For example, you could choose "keywords" for the **name** and then use the **content** attribute to list keywords that you feel are important to your site.

scheme
The **scheme** attribute sets the scheme (the procedure) for interpreting the meta data. This must agree with the profiles listed by the **profile** attribute of the **head** tag.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML meta Tag Example</title>
</head>
<body>
<head>
<meta name="author" content="The Guru" />
```

```
<meta name="description" content="DevGuru Cascading Style Sheets Quick Reference is a 40
page reference that discusses the style properties that set specific visual parameters for a Web
page." />
<meta name="keywords" content="DevGuru, CSS, css, Cascading Style Sheet, cascading
style sheet, style sheet, background, border, floating elements, font, margin, padding, text" />
<title>DevGuru Cascading Style Sheets Quick Reference</title>
</head>
<body>
...
</body>
</html>
```

The **http-equiv** attribute can be used to update a page. In this example, the update occurs every thirty seconds.

```
<meta http-equiv="refresh" content="30" />
```

TAG: script

<script> ... </script>

The **script** tag is used to place script code inside the **head** and **body** elements of an HTML code document. For all practical purposes, this tag is primarily used to allow the execution of JavaScript code inside HTML code (and, to a lesser extent, VBScript).

The separate closing tag is mandatory.

Attributes and Events

charset
The **charset** attribute is used to define the character set that was used to write the script.

defer
The **defer** attribute informs the server that the script will not alter the contents of the page on the server-side. This is intended to allow the server to load the page faster. (In JavaScript, this means no document.write statements.)

src
The **src** attribute is the URL of a file that contains a script code. This is useful when you need to use repeatedly a long script program. The browser will load and execute the script as a separate file.

type
The mandatory **type** attribute is used to define the script language being used. Permitted values include **text/css**, **text/javascript**, and **text/vbscript**. There is no default value.

This example uses the **script** tag to place a JavaScript function in the **head** element to see if two form fields (firstname and lastname) have been left blank. The **form** portion of the code is not included, but may be viewed on the form tag page.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru JavaScript Example</title>
<script type="text/javascript">
<!--
function checksubmit()
{
  if (document.formname.firstname.value == "")
  {
    alert("Please enter your full name")
    document.formname.firstname.focus()
    return false
  }
  if (document.formname.lastname.value == "")
  {
    alert("Please enter your email address")
    document.formname.lastname.focus()
```

```
      return false
    }
    return true
)
-->
</script>
</head>
<body>
... form goes here ...
</body>
</html>
```

TAG: form

<form> ... </form>

The **form** tag is used to delimit the start and stop of a form element and serves as a container for form controls (fields). Control is a technical term which refers to the various elements that can be used inside a **form** to gather information. The information gathered is referred to as the contents or parameters of the **form** and is a collection of name/value pairs.

The four tags that can be used to build a **form** are:

button input select textarea

The general concept is that the user fills out the appropriate sections of the **form** as a response to a request for information (such as a shipping address). The user then clicks the submit button. The contents of the **form** are then submitted for processing, usually to another page on the web site. However, you can also submit to the same page, or to a window or frame.

This ability to use a web site to display products and information, to gather a response via a **form** from a user, to store and manipulate the gathered information, and then to have the web site respond dynamically to the user input, is the heart of the e-commerce/e-business industry.

There are two required attributes for the **form** tag. The **action** attribute dictates where the form contents will be submitted. The **method** attribute specifies how to send the submission.

It is recommended that you should only place one **form** per page. It may prove best to divide a long **form** between two or more pages.

A **form** element should not contain other **form** tags. The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

accept
The **accept** attribute is a comma-separated list of file types that can be accepted by the server when using the **type="file"** content.

accept-charset
The **accept-charset** attribute is a comma-separated or space-separated list of the character types that the server must be able to support (process) when the contents of a **form** are submitted.

action
The mandatory **action** attribute sets the URL of the page that the contents (name/value pairs) of the **form** will be submitted to for processing.

enctype
The **enctype** attribute specifies the MIME type used to encode the contents of a **form**. The default is **application/x-www-form-urlencoded**. When using **type="file"** content, this attribute must be set to **multipart/form-data**.

method
The **method** attribute specifies which of two permitted HTTP methods will be used to pass the contents of a **form**. The more commonly used **post** method submits the contents of the **form** in a two step procedure. First the URL specified by the **action** attribute is contacted. Second, if the contact is successful, the contents are transmitted to the URL. The **get** method appends the contents to the end of the URL specified by the **action** attribute. (A question mark is used to delimit the end of the URL and the start of the contents.)

name
The **name** attribute is a string of characters that is used to label a form control (field) with a name. The name must be unique to that document and cannot be reused.

onreset
The **onreset** attribute is an event that allows the execution of JavaScript code when the user resets (clears) the **form**.

onsubmit - also see code example below
The **onsubmit** attribute is an event that allows the execution of JavaScript code when the user submits the **form**. For example, you could call a JavaScript function that performs client-side form verification to ensure that the **form** is filled out correctly.

target
The **target** attribute is used to submit the contents of the **form** to a specified window or frame.

Here is a sample **form** that includes client-side verification of fields.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>Form Example</title>
<script type="text/javascript">
// checksubmit() has to be a function, not a subroutine, because the
// onsubmit attribute in the form tag (in red text below) which calls
// this function requires a Boolean return value of true or false
// if the return is true, the form is automatically submitted without
// the need to call document.formname.submit()
// if the return is false, the form submission will be aborted and the client is sent
// back to the form to fill out the missing field
function checksubmit()
{
  // check to see if name field is empty
  // note that you use the name of the form and the field
  if (document.formname.fullname.value == "")
  {
    // tell client that you need the name
    alert("Please enter your full name")
    // send the cursor to the fullname field
    document.formname.fullname.focus()
    // field is empty, so return false to abort the form submission
    // the client is returned to the form
    return false
  }
```

```
  // check to see if email address field is empty
  if (document.formname.emailaddress.value == "")
  {
    alert("Please enter your email address")
    document.formname.emailaddress.focus()
    return false
  }
  // if both fields are filled out, return true
  // this triggers the form submission
  return true
}
</script>
</head>
<body onload="document.formname.fullname.focus()">
If you wish to receive information about upgrades to dgCharge,
please fill out this form.
<br />
<form method="get" name="formname" id="formname"
    action="xhtml_form.htm"
    onsubmit="return checksubmit()">
Full Name (required) <input type="text" name="fullname" size="30" />
<br />
EmailAddress (required) <input type="text" name="emailaddress" size="30" />
<br />
Phone Number (optional) <input type="text" name="phonenumber" size="15" />
<br />
<input type="reset" value="Clear" />
<input type="submit" name="submitbtn" value="Submit" />
</form>
</body>
</html>
```

EVENT: onreset

onreset="action"

The **onreset** event occurs when a user resets or clears a form, rather than submitting. The event causes a script to execute that performs a desired task. For example, you could call a function or display an alert box offering additional information to the user.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit**- runs a script when the form is submitted.

In this example, clicking the reset button causes an alert box to be displayed with a message.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onreset Example</title>
</head>
<body>
<form onreset="alert('Default color restored')">
Please enter your favorite color:
<br />
<input type="text" name="fullname" size="20" value="lavender">
<br />
<input type="reset" value="Reset">
</form>
</body>
</html>
```

EVENT: onsubmit

onsubmit="action"

The **onsubmit** event occurs when the user submits a form. There are many possible courses of action when using an **onsubmit** event. For example, before proceding on to the next page, you could call a function to check the various entries on the form for correctness.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit**- runs a script when the form is submitted.

In this form example, the **onsubmit** event is used in the **form** tag to call the JavaScript checksubmit function.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru Test Form</title>
<script type="text/javascript">
function checksubmit()
{
  if (document.formname.fullname.value == "")
  {
    alert("Please enter your full name")
    document.formname.fullname.focus()
    return false
  }
  if (document.formname.emailaddress.value == "")
  {
    alert("Please enter your email address")
    document.formname.emailaddress.focus()
    return false
  }
  return true
}
</script>
</head>
<body onload="document.formname.fullname.focus()">
<b>
FORM EXAMPLE
<br /><br />
Please leave one or both of the required fields blank and click the submit button.
</b>
<br />
<hr />
```

```
<br />
If you wish to receive information about upgrades to dgCharge,<br />
please fill out this form.
<br /><br />
<form method="post" name="formname" action="html_form_example.asp" onsubmit="return
checksubmit()">
Full Name (required)
<input type="text" name="fullname" size="30" />
<br /><br />
Email Address (required)
<input type="text" name="emailaddress" size="30" />
<br /><br />
Phone Number (optional)
<input type="text" name="phonenumber" size="15" />
<br /><br />
<input type="submit" name="submitbtn" value="Submit" />
<input type="reset" value="Clear" />
</form>
</body>
</html>
```

TAG: style

<style> ... </style>

The **style** tag is used to create document-level style sheet rules. It is referred to as document-level because the style rules will apply to the entire HTML page. In contrast, the **style** core attribute behaves as an inline style rule and only effects a small portion the HTML document.

The **style** tag must appear inside the **head** element. The code, contained between the opening and closing **style** tags, is not HTML, but is CSS (Cascading Style Sheets). For more information about Style Sheets, please visit the CSS2 Quick Reference.

The separate closing tag is mandatory.

Attributes and Events

dir lang title

media
The **media** attribute is a comma-separated list of one or more types of media in which the HTML document may appear (**all, aural, braille, handheld, print, projection, screen, tv,** and **tty**). The default is **screen**.

type
The mandatory **type** attribute is used to define the type of style being used. The most common value would be **text/css**. There is no default value.

Here is a simple example of creating document-level style rules. A series of CSS properties defining font face, color, and size is assigned to the **br, pre,** and **code** tags.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML style Tag Example</title>

<style type="text/css">
<!--
q { font-face: arial; color: black; font-size: 18px }
pre { font-face: arial; color: red; font-size: 24px }
code { font-face: courier; color: blue; font-size: 36px }
-->
</style>

</head>
<body>
<q>DevGuru is great!</q>
<p />
<pre>D e v G u r u i s g r e a t !</pre>
<p />
<code>DevGuru is great!</code>
</body>
</html>
```

TAG: title

<title> ... </title>

The **title** tag is used to provide a name or a title for your HTML document. There can only be one **title** element per document. The title is usually displayed as a text header by the browser in a special title or status window. If you do not provide a title, many browsers will display a default title. However, it is to your advantage to provide a meaningful and descriptive title for your site. For example, titles are sometimes used as the default text for links.

A well-formed XHTML document must contain properly nested and closed **html, head, title,** and **body** tags. The **title** must appear inside the **head** element. It cannot contain any other HTML tags. It can only contain text or be empty. However, you can use character entities (such as ** ** for spacing).

The separate closing tag is mandatory.

Attributes and Events

dir **lang**

Code:

```
<html>
<head>
<base href="http://www.devguru.com/">
<link rel="stylesheet" type="text/css" href="include/StylesDefinitions.css">
<title> DevGuru </title>
</head>
<body>
...
</body>
</html>
```

EVENT: onchange

onchange="action"

The **onchange** event occurs when a select input element has a selection made or when a text input element has a change in the text. Whenever the select or change occurs, the script code associated with the **onselect** event is executed. This script can call JavaScript functions which contain code that you want to run when the select occurs.

These six events are useful when dealing with forms and form elements:

- **onblur** - runs a script when the element loses focus.
- **onchange** - runs a script when the element changes.
- **onfocus** - runs a script when the element gets the focus.
- **onreset** - runs a script when the form is reset.
- **onselect** - runs a script when the element is selected.
- **onsubmit** - runs a script when the form is submitted.

In this example, an alert message appears when a select is made.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onchange Example</title>
</head>
<body>
<form>
Please make a selection.
<br />
Note that an alert appears every time that you make a new selection.
<p />
<select onchange="JavaScript:alert('A new selection has been made')">
<option>Angel fish</option>
<option>Cat fish</option>
<option>Discus</option>
<option>Neon tetra</option>
</select>
</p>
</form>
</body>
</html>
```

TAG: caption

<caption> ... <caption>

The **caption** tag allows you to place a comment (caption) with a table. The caption does not appear inside the table, but rather it appears outside the table on either the top (default) or bottom. By default, the caption will be centered on the table. The **caption** tag must occur immediately after the **table** tag. You can only have one caption per table. The text that composes the caption can contain HTML tags and style sheet properties.

Note that the **table** tag has the **summary** attribute which is used with non-visual media such as braille and speech to provide information about a table.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

This example demonstrates the basic table tags:

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML caption Tag Example</title>
</head>
<body>

<table width="80%" border="10" cellpadding="3" cellspacing="5" align="center">
<caption>
<b>How To Code A Table</b>
<br />
Use the caption tag to name the table and to provide useful information about the contents
</caption>
<tr>
<th colspan'"2">Use the <b>th</b> tag to create the header which will display in bold</th>
</tr>
<tr>
<td style="width:50%">The <b>tr</b> tag creates the row</td>
<td style="width:50%">The <b>td</b> tag creates individual cells for each row</td>
</tr>
<tr>
<td style="width:50%">You should always use the closing tags</td>
<td style="width:50%">You can use a variety of XHTML tags inside the cell</td>
</tr>
<tr>
<td style="width:50%">The three new table tags are <b>tbody, tfoot,</b> and
<b>thead</b></td>
<td style="width:50%">The <b>colspan</b> attribute allows you span more than one cell</td>
```

```
</tr>
</table>

</body>
</html>
```

Output:

How To Code A Table

Use the caption tag to name the table and to provide useful information about the contents

Use the th tag to create the header which will display in bold	
The **tr** tag creates the row	The **td** tag creates individual cells for each row
You should always use the closing tags	You can use a variety of XHTML tags inside the cell
The three new table tags are **tbody, tfoot,** and **thead**	Setting **colspan** allows you span more than one cell

TAG: cite

<cite> ... </cite>

The **cite** tag takes the selected block of text (a citation) and displays it in italics. Unlike the **blockquote** tag, the citation is not preceded or followed by paragraph breaks. So the citation remain inline. You may need to use other HTML tags and style sheets to effect the textual display and appearance that you desire.

It is recommended that you use the **blockquote** tag for long citations and the **cite** tag for short citations.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML cite Tag Example</title>
</head>
<body>
As for what was lost, I simply refer you to the photography by Eliot Porter in
<cite> The Place No One Knew: Glen Canyon on the Colorado River.</cite>
</body>
</html>
```

Output:

As for what was lost, I simply refer you to the photography by Eliot Porter in
The Place No One Knew: Glen Canyon on the Colorado River.

TAG: code

<code> ... </code>

The **code** tag is designed to display computer code in a special font, usually in a mono spaced font style. However, the exact appearance will be browser dependent. You may still need to use various HTML tags, such as **br** and **p**, to create a presentation and appearance to suit your needs.

An HTML code example cannot use the HTML delimiters **<** and **>**, because they will be executed by the browser as real HTML code. Rather you must substitute a **<** for the **<** and a **>** for the **>**.

Usually, the display font appearance for the **code**, **kbd**, **samp**, and **tt** tags is the same for a specific browser.

The **pre** tag is more commonly used to display code.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML code Tag Example</title>
</head>
<body>
<code>
if (document.getElementById) blnDOM = true;<br />
else if (document.layers) blnNN4 = true;<br />
else if (document.all) blnIE4 = true;<br />
</code>
</body>
</html>
```

Output:

```
if (document.getElementById) blnDOM = true;
else if (document.layers) blnNN4 = true;
else if (document.all) blnIE4 = true;
```

TAG: col

<col />

The **col** tag is used to assign attribute values to the individual columns within the **colgroup** elements.

The **col** tag can only be used inside a **colgroup** element. However, this tag must be omitted if you are using the **span** attribute of the **colgroup** tag.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents. The possible values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of the cells in a column. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

span
The **span** attribute is used to set the number of columns that are associated with each column group. If the columns are dissimilar, however, use the **col** tag to create each column, rather than this attribute.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents. The possible values are **baseline, bottom, middle,** and **top**.

width
The **width** attribute is used to set the width of each of the spanned columns. The value can be a percentage of the total width or an integer number of pixels.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML col Tag Example</title>
</head>
```

```
<body>

<table width="90%" border="10" cellspacing="5">
<caption>
<b>Column Group Table</b>
<br />
This table uses the <b>colgroup</b> and <b>col</b> tags
</caption>
<colgroup>
<col width="20%" />
<col width="30%" />
<col width="15%" />
</colgroup>
<colgroup>
<col width="15%" />
<col width="20%" />
</colgroup>
<tr>
<th>First Column Header</th>
<th>Second Column Header</th>
<th>Third Column Header</th>
<th>Fourth Column Header</th>
<th>Fifth Column Header</th>
</tr>
<tr>
<td>First Column First Row</td>
<td>Second Column First Row</td>
<td>Third Column First Row</td>
<td>Fourth Column First Row</td>
<td>Fifth Column First Row</td>
</tr>
<tr>
<td>First Column Second Row</td>
<td>Second Column Second Row</td>
<td>Third Column Second Row</td>
<td>Fourth Column Second Row</td>
<td>Fifth Column Second Row</td>
</tr>
</table>

</body>
</html>
```

Output:

Column Group Table

This table uses the **colgroup**and **col**tags

First Column Header	Second Column Header	Third Column Header	Fourth Column Header	Fifth Column Header

First Column First Row	Second Column First Row	Third Column First Row	Fourth Column First Row	Fifth Column First Row
First Column Second Row	Second Column Second Row	Third Column Second Row	Fourth Column Second Row	Fifth Column Second Row

TAG: colgroup

<colgroup> ... </colgroup>

The **colgroup** tag allows you to create a column-centric table as compared to the standard HTML table which is row-centric. The columns in the **table** are assigned to column groups using this tag. You can have one or more groups of columns.

If you are not using the **span** attribute of this tag, then you can use the **col** tag to assign attribute values to the individual columns within the **colgroup** elements.

The **colgroup** tag can only be used inside a **table** element.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents. The possible values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of the cells in a column. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

span
The **span** attribute is used to set the number of columns that are associated with each column group. If the columns are dissimilar, however, use the **col** tag to create each column, rather than this attribute.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents. The possible values are **baseline, bottom, middle,** and **top**.

width
The **width** attribute is used to set the width of each of the spanned columns. The value can be a percentage of the total width or an integer number of pixels.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
```

```
<title>DevGuru XHTML colgroup Tag Example</title>
</head>
<body>

<table width="90%" border="10" cellspacing="5">
<caption>
<b>Column Group Table</b>
<br />
This table uses the <b>colgroup</b> and <b>col</b> tags
</caption>
<colgroup>
<col width="20%" />
<col width="30%" />
<col width="15%" />
</colgroup>
<colgroup>
<col width="15%" />
<col width="20%" />
</colgroup>
<tr>
<th>First Column Header</th>
<th>Second Column Header</th>
<th>Third Column Header</th>
<th>Fourth Column Header</th>
<th>Fifth Column Header</th>
</tr>
<tr>
<td>First Column First Row</td>
<td>Second Column First Row</td>
<td>Third Column First Row</td>
<td>Fourth Column First Row</td>
<td>Fifth Column First Row</td>
</tr>
<tr>
<td>First Column Second Row</td>
<td>Second Column Second Row</td>
<td>Third Column Second Row</td>
<td>Fourth Column Second Row</td>
<td>Fifth Column Second Row</td>
</tr>
</table>

</body>
</html>
```

Output:

Column Group Table

This table uses the **colgroup**and **col**tags

First Column Header	Second Column Header	Third Column Header	Fourth Column Header	Fifth Column Header

First Column First Row	Second Column First Row	Third Column First Row	Fourth Column First Row	Fifth Column First Row
First Column Second Row	Second Column Second Row	Third Column Second Row	Fourth Column Second Row	Fifth Column Second Row

TAG: dd

<dd>

The **dd** tag is used to create a definition list which is a type of listing where a term or phrase is followed by a definition or explanation. The **dd** tag presents the definition. It must be used with the **dt** tag which presents the term and the **dl** tag which delimits the start and stop of the definition list.

The exact appearance of a definition list can be browser dependent. Most browsers place the term on the left margin and then the definition is placed on the next line and the entire text is indented. However, if the term is short (3 characters or less) the definition may start on the same line as the term. The list is usually separated from preceding and following text by paragraph breaks.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML dd Tag Example</title>
</head>
<body>
Definition Lists:
<dl>
<dt>dl tag</dt>
<dd> definition list start and stop delimiter.</dd>
<dt>dt tag</dt>
<dd> presents the term or phrase to be defined.</dd>
<dt>dd tag</dt>
<dd> presents the definition for the term or phrase.</dd>
</dl>
</body>
</html>
```

Output:

Definition Lists:

dl tag
    definition list start and stop delimiter.
dt tag
    presents the term or phrase to be defined.
dd tag
    presents the definition for the term or phrase.

TAG: del

<del> ... </del>

The **del** tag is used with the **ins** tag to display editorial changes involving additions and deletions to the text of a document. The **del** tag marks the text to be deleted by striking a horizontal line through the characters. The **ins** tag designates the text to be added with an underline. This can also be accomplished using style sheets.

A **del** tag cannot occur between any opening and closing **ins** tags, and vice versa.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

cite
The **cite** attribute is used to list one reference URL related to the editorial changes.

datetime
The **datetime** attribute is used to present the date and time that the editorial changes occurred. It must be of the format **YYYY-MM-DDThh:mm:ssTZD** where the required **T** denotes the break between the date, the time, and the time zone.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML del Tag Example</title>
</head>
<body>
The gentlemen of the first party agrees to pay <del>$1,047,300</del> <ins>$953,150</ins> to the gentlemen of the second party no later than <del>August 1, 2001</del> <ins>June 1, 2002</ins>.
</body>
</html>
```

Output:

The gentlemen of the first party agrees to pay ~~$1,047,300~~ $953,150 to the gentlemen of the second party no later than ~~August 1, 2001~~ June 1, 2002.

TAG: dfn

<dfn> ... </dfn>

The **dfn** tag applies an oblique font style to the enclosed word or phrase. It is common practice in engineering, scientific, and technical articles to specially mark the first occurrence of a term.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML dfn Tag Example</title>
</head>
<body>
<dfn>HTML</dfn> is the acronym for the Hyper Text Markup Language.
</body>
</html>
```

Output:

HTML is the acronym for the Hyper Text Markup Language.

TAG: div

** ... **

The **div** tag is used to designate a block-line portion of an HTML document as a **div** element and to apply any of the core attributes of this tag against the contents of that element. The behavior of this tag is block-line, because this tag causes a line break (the equivalent of a **br** tag) to occur both before and after the **div** element. **div** tags, which can be nested, allow a significant degree of control and manipulation of each individual block section of your web page.

For example, you could use the **class** or **style** core attributes to apply the effects of Cascading Style Sheets, or you could use the **lang** attribute to display a foreign language text, or you could assign an **id** to each block of code for reference by a hyperlink.

In contrast, you should use the **span** tag when you want to apply attributes to an inline portion of a page. Note that the behavior of the **span** tag is not block-line, but rather inline, since the effects occur in the normal flow of the text and images (without the tag inserting any additional linefeeds or carriage returns).

With the advent of Cascading Style Sheets, both the **div** and **span** tags have gained importance and usefulness.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML div Tag Example</title>
</head>
<body>
<div style="text-align: left;" id="div_example1">
The <b>div</b> offers exciting possibilities for the developer.
</div>
<div style="text-align: center;" id="div_example2">
You should take time to learn about this tag.
</div>
<div style="text-align: right;" id="div_example3">
Also check out the <b>span</b> tag.
</div>
</body>
</html>
```

Output:

The **div** offers exciting possibilities for the developer.

You should take time to learn about this tag.

Also check out the **span** tag.

TAG: dl

<dl> ... </dl>

The **dl** tag delimits the start and stop of a definition list. This is a type of list where a term or phrase is followed by a definition or explanation.

This **dl** tag must be used with the **dt** tag which presents the term and the **dd** tag which presents the definition.

The exact appearance of a definition list can be browser dependent. Most browsers place the term on the left margin and then the definition is placed on the next line and the entire text is indented. However, if the term is short (three characters or less) the definition may start on the same line as the term. Also, the list is usually separated from any preceding and following text by paragraph breaks.

You can use the **ol** tag to create an ordered list and the **ul** tag to create an unordered list.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML dl Tag Example</title>
</head>
<body>
Definition Lists:
<dl>
<dt>dl tag</dt>
<dd>definition list start and stop delimiter.</dd>
<dt>dt tag</dt>
<dd>presents the term or phrase to be defined.</dd>
<dt>dd tag</dt>
<dd>presents the definition for the term or phrase.</dd>
</dl>
</body>
</html>
```

Output:

Definition Lists:

dl tag
    definition list start and stop delimiter.
dt tag
    presents the term or phrase to be defined.

dd tag

presents the definition for the term or phrase.

DTD TAG: DOCTYPE

<!DOCTYPE ... >

The **DOCTYPE** tag is used to declare the DTD (Document Type Definition) for an XHTML document.

On a broader scale, XHTML, like HTML, is a subset of the SGML (Standardized Generalized Markup Language). SGML, and its various family members, use DTDs to define the context of the language. The W3C has defined a DTD to be:

> "...a collection of declarations that, as a collection, defines the legal structure, elements, and attributes that are available for use in a document that complies to the DTD."

Specifically, the XHTML DTD precisely defines the grammar, rules, and syntax that will apply to a document that has been created using XHTML. To repeat this in a different way, to be valid XHTML, the XHTML code that creates the XHTML document must obey all of the grammar, rules, and syntax in the XHTML DTD.

This tag is mandatory and must appear at the top (on the first line) of all XHTML code. If the **DOCTYPE** DTD tag is not present, then it is not XHTML code.

The exclamation mark (!) is required. This is the only tag in an XHTML document that is not closed. Obey the case and syntax.

Currently, there are three types of DTDs that apply to XHTML: **Frameset, Strict,** and **Transitional**. You must obey the syntax.

Frameset

This is declared when you have partitioned the HTML document into two or more frames (with or without using Cascading Style Sheets).

Syntax:
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "DTD/xhtml-frameset.dtd">

Strict

This is declared when you use Cascading Style Sheets (CSS) to enhance the appearance and style of your HTML document. In general, you should only use this type of DTD if you are certain that your viewer has access to modern browsers that recognize CSS.

Syntax:
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "DTD/xhtml-strict.dtd">

Transitional

This is declared when you are simply using HTML to create the appearance of the web page, rather than using Cascading Style Sheets (CSS). This type of DTD would ensure the widest viewing audience to your XHTML document.

Syntax:

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">

The basic XHTML document is composed of **DOCTYPE, html, head, title,** and **body** tags. Note that all tags in XHTML are closed except for the **DOCTYPE**.

Code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional/EN" "DTD/xhtml-transitional.dtd">
<html>
<head>
<title>DevGuru</title>
</head>
<body>
DevGuru is great!
<br />
<img src="images/xhtmlvalid.gif" />
</body>
</html>
```

Output:

DevGuru is great!

TAG: dt

<dt> ... </dt>

The **dt** tag is used to create a a definition list. This is a type of list where a term or phrase is followed by a definition or explanation. The **dt** tag presents the term to be defined. It must be used with the **dd** tag which presents the definition and the **dl** tag which delimits the start and stop of the definition list.

The exact appearance of a definition list can be browser dependent. Most browsers place the term on the left margin and then the definition is placed on the next line and the entire text is indented. However, if the term is short (3 characters or less) the definition may start on the same line as the term. Also, the list is usually separated from preceding and following text by paragraph breaks.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML dt Tag Example</title>
</head>
<body>
Definition Lists:
<dl>
<dt>dl tag</dt>
<dd>definition list start and stop delimiter.</dd>
<dt>dt tag</dt>
<dd>presents the term or phrase to be defined.</dd>
<dt>dd tag</dt>
<dd>presents the definition for the term or phrase.</dd>
</dl>
</body>
</html>
```

Output:

Definition Lists:

dl tag
    definition list start and stop delimiter.
dt tag
    presents the term or phrase to be defined.
dd tag
    presents the definition for the term or phrase.

TAG: em

** ... **

The **em** tag is used to provide emphasis when displaying the enclosed word or phrase. Most browsers do this by rendering the selected text in italics.

The **i** tag also renders in italics (or oblique), or you could use the style sheets **font-style** property to display in italics.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML em Tag Example</title>
</head>
<body>
Did you know that the <b>Guru</b> is over <em>800 years old</em>?
<br />
Did you know that the <b>Guru</b> is over <i>800 years old</i>? </body>
</html>
```

Output:

Did you know that the **Guru** is over 800 years old?
Did you know that the **Guru** is over 800 years old?

TAG: fieldset

<fieldset> ... </fieldset>

The **fieldset** tag is used to group related **form** elements together with a caption and labels. The caption is created using the **legend** tag. One or more labels are created using the **label** tag. The browser is supposed to render the appearance of the caption and labels in a special manner.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML fieldset Tag Example</title>
</head>
<body>
<form>
<fieldset>
<legend>Please Enter Your Name</legend>
<label for="firstname">First Name</label><input type="text" id="firstname" />
<br />
<label for="lastname">Last Name</label><input type="text" id="lastname" />
</fieldset>
</form>
</body>
</html>
```

TAG: frame

<frame />

The **frame** tag is used to create one frame, which is simply a window within another window. As an example, if you want three frames (windows) to appear on an HTML page, you must have three **frame** tags and the **frameset** element must set the spacing for three frames.

The **frame** tags can only occur between an opening and closing **frameset** element.

A **frame** element displays content, including forms, images, multimedia, other frames, tables, etc. It is very important to understand that the only way to assign content to a **frame** is to assign a URL using the **src** attribute. Whatever content is displayable at the referenced URL will in turn be displayed inside the **frame**.

The frames can have either a column (vertical) or a row (horizontal) configuration, but not both. The configuration must be set using either the **cols** or the **rows** attribute of the **frameset** tag, but not both. The order of the **frame** tags sets the order of the frames. If they are in a **rows** configuration, the first **frame** element is the top window, the second **frame** element is second from the top, etc. For the **cols** configuration, the order is left to right.

Note that you can nest **frameset** tags inside of **frameset** tags. Therefore, an HTML document can contain numerous frames. This opens the possibility of creating complex and imaginative page displays (however, always remember to keep the page user friendly).

This tag is self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class **id** **style** **title**

frameborder
The **frameborder** attribute is used to add or remove a border around a frame. The value of **yes** or **1** allows a border. The value of **no** or **0** forbids a border.

longdesc
The **longdesc** attribute is the URL address of an HTML document that contains a long textual description of the **frame**. This is designed to provide more information than a **title** element would display.

marginheight
The **marginheight** attribute is used to set the amount of white space in pixels that will appear along the top and bottom margins between the frame border and the contents. If you provide an unrealistic value, the browser will simply ignore this attribute.

marginwidth
The **marginwidth** attribute is used to set the amount of white space in pixels that will appear along the left and right margins between the frame border and the contents. If you provide an unrealistic value, the browser will simply ignore this attribute.

name
The **name** attribute is a string of characters that is used to label a frame with a name. The name must be unique to that document and cannot be reused.

noresize
The **noresize** attribute prevents the user from changing the **frame** size.

scrolling
The **scrolling** attribute is used to determine if horizontal and vertical scrolling bars will appear in the **frame**. If the content is larger than the frame, this permits scrolling up and down or left and right, as needed, to see the entire contents. There are three permitted values: **yes** which always displays scroll bars; **no** which never displays scroll bars; and **auto** which displays scroll bars only when needed. If a value is not provided, the default is **auto**.

src
The **src** attribute provides the URL address of any valid HTML, ASP, or other displayable page. The contents of the referenced URL will be displayed inside the **frame**.

This example creates five frames, each with a different background color. Note that there is no **body** tag.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML frame Tag Example</title>
</head>
<body>
<html>
<head>
<title>frames example</title>
</head>
<frameset cols="50%, 50%">
  <noframes>
    To be viewed properly, this page requires frames.
  </noframes>
  <frameset rows="33%, 33%, *">
    <frame src="xhtml_frames_red.html" />
    <frame src="xhtml_frames_orange.html" />
    <frame src="xhtml_frames_yellow.html" />
  </frameset>
  <frameset rows="50%, 50%">
    <frame src="xhtml_frames_green.html" />
    <frame src="xhtml_frames_blue.html" />
  </frameset>
</frameset>
</html>
```

TAG: frameset

<frameset> ... </frameset>

The **frameset** tag serves as a container element to hold a collection of frames. This tag determines the number of frames, whether the frames are to be of a column or of a row configuration, and sets the spacing for either case. In fact, you must specify either a column or row configuration using either the **rows** or **cols** attribute (but not both).

When you are using the **framset** tag, you absolutely must not use a **body** tag. The **frameset** tag is used in place of the **body** tag.

The only tags that are valid inside the **frameset** tag are the **frame, frameset, head,** and **noframe**. Note that you can nest **frameset** tags inside of **frameset** tags. Therefore, an HTML document can contain numerous frames. This opens the possibility of creating complex and imaginative page displays (however, always remember to keep the page user friendly).

The separate closing tag is mandatory.

Attributes and Events

id **title**

cols
The **cols** attribute is used to set the number of column frames in a **frameset** element. This is done indirectly since you never provide the actual number of columns. Rather, this attribute is a quote-enclosed, comma-separated list of values that specify the width assigned to each **frame**. The number of values in the list determines the number of frames. For example, if you have three values, then you are expected to have three frames. These values can be:

- a percentage of the window width
- an integer number of pixels
- either of the above methods plus the asterisk *

The asterisk can be used to proportionally assign spacing. For example:
<frameset cols="*, *, 50%">
will create three frame whose proportional spacing will be 25%, 25%, and 50%. A 3* value has three times the proportion than a single *.

Possible examples are:
Code:

<frameset cols="40%, 60%" >	for two column frames
<frameset cols="20%, 50%, 30%" >	for three column frames
<frameset cols="20%, 20%, 20%, 20%, 20%" >	for five column frames
<frameset cols="100, 100" >	for two column frames
<frameset cols="50, 75, 150" >	for three column frames
<frameset cols="150, * ">	for two column frames
<frameset cols="*, 50%, *, * ">	for four column frames
<frameset cols="3*, *" >	for two column frames
<frameset cols="20%, *, 2*" >	for three column frames

If you use the **cols** attribute, you cannot use the **rows** attribute.

onload
The **onload** attribute is an event that allows a JavaScript code to execute when the **frame** is first loaded by the browser.

onunload
The **onunload** attribute is an event that allows a JavaScript code to execute when the **frame** is unloaded by the browser.

rows
The **rows** attribute is used to set the number of row frames in a **frameset** element. This is done indirectly since you never provide the actual number of rows. Rather, this attribute is a quote-enclosed, comma-separated list of values that specify the height assigned to each **frame**. The number of values in the list determines the number of frames. For example, if you have three values, then you are expected to have three frames. These values can be:

- a percentage of the window height
- an integer number of pixels
- either an integer of the above methods plus the asterisk *

The asterisk is used to proportionally assign leftover spacing. A 3* value has three times the proportion than a single *.

Possible examples are:
Code:

```
<frameset rows="40%, 60%" >                        for two row frames
<frameset rows="20%, 50%, 30%" >                   for three row frames
<frameset rows="20%, 20%, 20%, 20%, 20%" >         for five row frames
<frameset rows="100, 100" >                        for two row frames
<frameset rows="50, 75, 150" >                     for three row frames
<frameset rows="150, * ">                          for two row frames
<frameset rows="*, 50%, *, * ">                    for four row frames
<frameset rows="3*, *" >                           for two row frames
<frameset rows="20%, *, 2*" >                      for three row frames
```

If you use the **rows** attribute, you cannot use the **cols** attribute.

This example creates five frames, each with a different background color. Note that there is no **body** tag.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML frameset Tag Example</title>
</head>
<body>
<html>
<head>
<title>frameset example</title>
</head>
<frameset cols="50%, 50%">
  <noframes>
```

```
    To be viewed properly, this page requires frames.
  </noframes>
  <frameset rows="33%, 33%, *">
    <frame src="xhtml_frames_red.html" />
    <frame src="xhtml_frames_orange.html" />
    <frame src="xhtml_frames_yellow.html" />
  </frameset>
  <frameset rows="50%, 50%">
    <frame src="xhtml_frames_green.html" />
    <frame src="xhtml_frames_blue.html" />
  </frameset>
</frameset>
</html>
```

TAG: h1, h2, h3, h4, h5, h6

<h#> ... </h#> where # = 1, 2, 3, 4, 5, or 6

The **h1, h2, h3, h4, h5, h6** tags are used to create text headers for a document. They can display the header text in one of six different sizes. **h1** is the largest size and, as the integer number increases to **h6**, the text size becomes progressively smaller. The exact display appearance of the header text is browser dependent.

This tag also automatically inserts a carriage return and line feed after the closing tag (i.e., a paragraph break).

The true purpose of this tag is only to create headers and titles, and not to set text font size. You are to use style sheets to set the font properties for a text.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML h Tag Example</title>
</head>
<body>
<h1>DevGuru in h1</h1>
<h2>DevGuru in h2</h2>
<h3>DevGuru in h3</h3>
<h4>DevGuru in h4</h4>
<h5>DevGuru in h5</h5>
<h6>DevGuru in h6</h6>
</body>
</html>
```

Output:

DevGuru in h1

DevGuru in h2

DevGuru in h3

DevGuru in h4

DevGuru in h5

DevGuru in h6

TAG: hr

<hr />

The **hr** tag is used to render a horizontal rule (line). This is a very useful and commonly used tag.

The flow of the text and images is halted on the current line, the rule is rendered on the next line, and then the flow of text and images resumes on the following line. The exact amount of vertical space between the rule and any elements above and below it is browser dependent.

You are to use style sheets to effect the appearance of the rule.

This is a self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML hr Tag Example</title>
</head>
<body>
<hr>
<hr style="color:red; height:15px; width:350px;">
</body>
</html>
```

Output:

TAG: html

<html> ... </html>

The **html** tag informs the browser that this is an HTML encoded program. The **html** opening and closing tags delimit the start and stop of the document.

The only tag that can precede the opening **html** tag is the **DOCTYPE** tag. It is used to select the XHTML DTD (Document Type Definition) that sets the rules and syntax for the XHTML document.

A well-formed XHTML document must contain properly nested and closed **html, head, title,** and **body** tags.

Although it is not a requirement, it is strongly recommended that the opening **html** tag contain information defining the language being used for both the XHTML document and for XML. This is set respectively with the **lang** and the **xml:lang** attributes. In addition, the **html** tag should also contain the XML namespace declaration (**xmlns**). This is demonstrated for English with the following code.

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">

The separate closing tag is mandatory.

Attributes and Events

dir lang

This example demonstrates the basic structure of an XHTML program.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML html Tag Example</title>
</head>
<body>
<html>
body content goes here...
</body>
</html>
```

Output:

body content goes here...

TAG: i

** ... **

The **i** tag causes the designated text to be displayed as an italic or oblique font style. If italic or oblique is not available for a particular font, the browser may impose some other physical change to the text, such as underline.

You can use the **em** tag to render in italics, the **dfn** tag to render in oblique, or use the style sheets **font-style** property to display in italics.

The separate closing tag is mandatory.

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML i Tag Example</title>
</head>
<body>
Did you know that the <b>Guru</b> is over <i>800 years old</i>?
<br />
Did you know that the <b>Guru</b> is over <em>800 years old</em>?
</body>
</html>
```

Output:

Did you know that the **Guru** is over 800 years old?
Did you know that the **Guru** is over 800 years old?

TAG: img

<img />

The **img** tag is used to insert a graphic or photographic image directly into the flow of text and other images. No line breaks or carriage returns are automatically inserted before or after the image. You can use attributes and style sheets to effect an appearance and presentation of images and text that suits your needs.

This tag can be placed in an **a** element to provide a clickable image for a hyperlink. A border will be automatically drawn around all four sides of an image used as a link. You can remove the border using style sheets or the **border** attribute.

The exact display of an image is very browser dependent.

This tag is self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

alt
The **alt** attribute provides a text message that will be displayed (in place of the image) on browsers that cannot display a graphic image or picture.

border
The **border** attribute places a border around all four sides of the image. The thickness of the border is set in pixels. If you are using an image as a hyperlink and desire no border, set this attribute to zero.

height
The **height** attribute is used to set the vertical extent of the image in pixels.

hspace
The **hspace** attribute is used to add additional blank space in pixels to each side of the image.

ismap
The **ismap** attribute is a Boolean value that, when present, signifies that the image is being used as a mouse-clickable server-side image map.

longdesc
The **longdesc** attribute is the URL of a source that contains a long description of the image. This is useful for server-side image maps and nonvisual browsers.

name
The **name** attribute sets the name of the image. (This is required when using JavaScript.)

src
The mandatory **src** attribute is the URL of the location of the image file. The URL must include the file name and can also include a relative or absolute path.

usemap
This **usemap** attribute specifies the name of an image map that you wish to associate with the element. This indicates that the image is being used as a mouse-clickable client-side image map.

vspace
The **vspace** attribute is used to add additional blank space in pixels both above and below the image.

width
The **width** attribute is used to set the horizontal extent of the image in pixels.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML img Tag Example</title>
</head>
<body>
The Guru
<img src="images/guru.gif" align="middle" alt="The Guru" border="2" />
</body>
</html>
```

TAG: ins

<ins> ... </ins>

The **ins** tag is used with the **del** tag to display editorial changes involving additions and deletions to the text of a document. The **del** tag marks the text to be deleted by striking a horizontal line through the characters. The **ins** tag designates the text to be added with an underline. This can also be accomplished using style sheets.

An **ins** tag cannot occur between any opening and closing **del** tags, and vice versa.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

cite
The **cite** attribute is used to list one reference URL related to the editorial changes.

datetime
The **datetime** attribute is used to present the date and time that the editorial changes occurred. It must be of the format **YYYY-MM-DDThh:mm:ssTZD** where the required **T** denotes the break between the date, the time, and the time zone.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML ins Tag Example</title>
</head>
<body>
The gentlemen of the first party agrees to pay <del>$1,047,300</del> <ins>$953,150</ins> to the gentlemen of the second party no later than <del>August 1, 2001</del> <ins>June 1, 2002</ins>.
</body>
</html>
```

Output:

The gentlemen of the first party agrees to pay ~~$1,047,300~~ $953,150 to the gentlemen of the second party no later than ~~August 1, 2001~~ June 1, 2002.

TAG: kbd

<kbd> ... </kbd>

The **kbd** tag is used to signify that the indicated text is to be typed by the user on the keyboard. For example, this can be used in training manuals.

The exact appearance will be browser dependent. However, most browsers render the indicated text in a mono spaced font.

Usually, the display font appearance for the **code**, **kbd**, **samp**, and **tt** tags is the same.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML kbd Tag Example</title>
</head>
<body>
Please type <kbd>Request.Write strVar</kbd> in the third window.
</body>
</html>
```

Output:

Please type `Request.Write strVar` in the third window.

TAG: label

<label> ... </label>

The **label** tag is used to associate text labels with a specific element (control) inside a **form** (such as an **input** element). This is done by having the values of the **for** attribute of the **label** tag and the **id** attribute of the form element be the same. Note that you can attach more than one label to the same element. The browser is supposed to render the labels in a special manner to accentuate their appearance.

This tag is one of three tags implemented in 4.0 that help set the appearance of a **form**. The other two are **fieldset** and **legend**.

However, these three tags are poorly implemented by most browsers.

A **label** element should not contain other **label** tags. The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

accesskey
The **accesskey** attribute allows you to designate a key on the keyboard that when pressed, along with the alt or meta key, will bring focus to the form element associated with the **label**. This attribute is poorly supported.

for
The **for** attribute is used to relate the **label** tag to a specific form control. It must have the exact same value as the **id** attribute of the form control. However, if the form control is placed inline between opening and closing **label** tags, the **for** and **id** attributes are not needed since the form control and the **label** are implicitly joined.

onblur
The **onblur** attribute is an event that allows a JavaScript code to execute when an element loses focus (for example, the mouse was clicked onto another element or a tab navigation directed the cursor elsewhere).

onfocus
The **onfocus** attribute is an event that allows a JavaScript code to execute when an element comes into focus (for example, the mouse was clicked onto the element or a tab navigation brought the cursor to the element).

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML label Tag Example</title>
</head>
<body>
<form>
```

```
<fieldset>
<legend>Please Enter Your Name</legend>
<label for="firstname">First Name</label><input type="text" id="firstname" />
<br />
<label for="lastname">Last Name</label><input type="text" id="lastname" />
</fieldset>
</form>
</body>
</html>
```

TAG: legend

<legend> ... </legend>

The **legend** tag is used to create a caption for the **form** elements associated with a **fieldset** group. Therefore, this tag can only be used inside the **fieldset** element. The browser is supposed to render the **legend** caption in a special manner to accentuate the appearance.

This tag is one of three tags implemented in HTML 4.0 that help set the appearance of a **form**. The other two are **fieldset** and **label**.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

accesskey
The **accesskey** attribute allows you to designate a key on the keyboard that when pressed, along with the alt or meta key, will bring focus to the form element associated with the **legend**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML legend Tag Example</title>
</head>
<body>
<form>
<fieldset>
<legend>Please Enter Your Name</legend>
<label for="firstname">First Name</label><input type="text" id="firstname" />
<br />
<label for="lastname">Last Name</label><input type="text" id="lastname" />
</fieldset>
</form> </body>
</html>
```

TAG: li

** ... **

The **li** tag is used to list an item in a list. This tag is required for both the ordered list **ol** tag and the unordered list **ul** tag.

For ordered and unordered lists, there are essentially no restrictions as to the text that may be placed in a list, such as length.

In an unordered list, each item is preceded by a bullet, such as •. In an ordered list, each item is labeled with a number, letter, or Roman numeral that increments with each following item.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

This is an unordered list.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML li Tag Example</title>
</head>
<body>
Three Major Types Of Lists:
<ul>
<li> Definition List</li>
<ul style="list-style-type: square;">
<li> dd tag</li>
<li> dl tag</li>
<li> dt tag</li>
</ul>
<li> Ordered List</li>
<ul style="list-style-type: circle;">
<li> ol tag</li>
<li> li tag</li>
</ul>
<li> Unordered List
<ul style="list-style-type: disc;">
<li> ul tag</li>
<li> li tag</li>
</ul>
</ul>
</body>
</html>
```

Output:
Three Major Types Of Lists:

- Definition List
  - dd tag
  - dl tag
  - dt tag
- Ordered List
  - ol tag
  - li tag
- Unordered List
  - ul tag
  - li tag

This is an ordered list.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML li Tag Example</title>
</head>
<body>
Three Major Types Of Lists:
<ol>
<li> Definition List</li>
<ol style="list-style-type: lower-alpha;">
<li> dd tag</li>
<li> dl tag</li>
<li> dt tag</li>
</ol>
<li> Ordered List</li>
<ol >
<li> ol tag</li>
<li> li tag</li>
</ol>
<li> Unordered List
<ol style="list-style-type: upper-roman;">
<li> ul tag</li>
<li> li tag</li>
</ol>
</ol>
</body>
</html>
```

Output:
Three Major Types Of Lists:

1. Definition List
   1. dd tag
   2. dl tag
   3. dt tag

2. Ordered List
    1. ol tag
    2. li tag
3. Unordered List
    1. ul tag
    2. li tag

TAG: map

<map> ... </map>

The **map** tag is used to create a client-side image map which is divided into two or more mouse-sensitive regions. For example, when you click onto a specific region of the image map, a hyperlink could send you to a target page or a pop-up window could display some information.

The **map** element essentially serves as a container for the code that creates and implements the image map. For example, this code could be a series of **area** tags that are used to set image coordinates and to assign a hyperlink to each region of the image map. Also, since the image map is mouse-sensitive, you could also use any of the onmouse core attributes to capture a mouse event and execute a JavaScript code.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

name
The **name** attribute is a string of characters that is used to label the **map** with a name. The name must be unique to that document and cannot be reused. The value for this attribute is case-sensitive.

The **area** and **map** tags are used on the **DevGuru** site to create image mapped links for the indexes to the various Quick References. For example, this is the code used for the ASP Quick Reference. These links are active and will take you to the ASP indexes.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML map Tag Example</title>
</head>
<body>
<img border="0" src="images/asp_method_event_property.gif" usemap="#methodproperty" />
<br />
<map name="methodproperty">
<area shape="rect" coords="192,0,248,26" href="/technologies/asp/quickref/properties.html" />
<area shape="rect" coords="133,0,191,26" href="/technologies/asp/quickref/collections.html" />
<area shape="rect" coords="90,0,134,22" href="/technologies/asp/quickref/events.html" />
<area shape="rect" coords="43,0,89,22" href="/technologies/asp/quickref/methods.html" />
</map>
</body>
</html>
```

TAG: noframes

<noframes> ... </noframes>

The **noframes** tag is used to display an alternative message on browsers that do not recognize frames. Usually, this message advises the user that frames are needed to see the contents of this HTML document. If the browser does recognize frames, then the contents of the **noframes** element are not displayed.

The **noframes** tag must be placed immediately following the first use of the **frameset** tag. The separate closing tag is required.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

This example creates five frames, each with a different background color.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML noframes Tag Example</title>
</head>
<frameset cols="50%, 50%">
  <noframes>
    To be viewed properly, this page requires frames.
  </noframes>
  <frameset rows="33%, 33%, *">
    <frame src="html_frames_red.html" />
    <frame src="html_frames_orange.html" />
    <frame src="html_frames_yellow.html" />
  </frameset>
  <frameset rows="50%, 50%">
    <frame src="html_frames_green.html" />
    <frame src="html_frames_blue.html" />
  </frameset>
</frameset>
</html>
```

TAG: noscript

<noscript> ... </noscript>

The **noscript** tag is used to display alternative content for browsers that do not recognize the **script** tag, or for those occasions where the user has purposely disabled **script**.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML noscript Tag Example</title>
</head>
<body>
The following message will only appear on broswers that
do not support scripting or have scripting turned off:
<noscript>
Script is disabled.
</noscript>
</body>
</html>
```

Output:

The following message will only appear on broswers that
do not support scripting or have scripting turned off:

TAG: object

<object> ... </object>

The **object** tag is used to insert an object (such as ActiveX components, applets, image maps, media players, and plug-ins) into an HTML document and to provide all of the neccessary information to implement and run the object. For example, you can give the location and type of the executable code of the object, and the location and type of any associated data files. You have the option to declare and instantiate the object at the same time, or declare the object and instantiate it, one or more times, elsewhere in the program.

This tag can be placed in the **head** and the **body** element. If it is placed in the **head**, the object cannot be used to render (display) content onto the page. (No tag in the **head** is permitted to render content.)

You can insert text between the opening and closing tags that will be displayed as the default message if the object cannot be displayed by the browser.

The **param** tag is used to pass any parameters (run-time values) required to implement the object. This tag must appear immediately after the opening **object** tag, but before any other content.

You are to use the **object** tag (and the **classid** attribute) to insert a Java applet into a web page.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

archive
The **archive** attribute is a comma-separated list of URLs of object resources, including resources specified by the **classid** and **data** attributes, that must be preloaded into the local user cache before the object can be displayed. If the URLs are relative, you must provide the base URL using the **codebase** attribute.

classid
The **classid** attribute is used to provide the URL location of the object. If the URL is relative, you must provide a base URL using the **codebase** attribute. When using this attribute, you may also wish to specify the program code type of the object by using the **codetype** attribute.

codebase
The **codebase** attribute is used to specify the base URL address of the directory where the object is stored. This allows the use of relative URLs in other attributes. If the directory is different from where the HTML code is stored, you must use this attribute.

codetype
The **codetype** attribute is used to specify the program code type (MIME type) of the object. The default is to allow the browser to determine the program code type. The data files code type is specified by using the **type** attribute.

data
The **data** attribute is used to specify the URL location of the data files associated with the object (such as images). If the URL is relative, you must provide a base URL using the **codebase** attribute.

declare
The **declare** attribute is a Boolean value that, if present, signifies that the object is a declaration only and prevents the browser from downloading and executing the object. (When you declare an object, you should also assign a value to the **id** attribute. This **id** can be referenced at a later time when you instantiate the object.)

height
The **height** attribute is used to set the vertical extent of the object in pixels.

hspace
The **hspace** attribute is used to add additional blank space in pixels to each side of the object.

name
The **name** attribute is used to assign a unique string of characters as the name of the object (which allows the object to be referenced).

standby
The **standby** attribute is used to display a message while the object is loading. This is a wise precaution for slow-loading objects and should be considered a courtesy to the user.

tabindex
The **tabindex** attribute specifies an integer that defines the rank in the tabbing order for the specified element when you use the keyboard to navigate (tab through) a page. This attribute is poorly supported.

type
The **clear** attribute is used to specify the data files code type (MIME type) of the object. The default is to allow the browser to determine the data file code type. The program code type is specified by using the **codetype** attribute.

usemap
The **usemap** attribute is used to indicate that the object is a client-side image map.

vspace
The **vspace** attribute is used to add additional blank space in pixels both above and below the object.

width
The **width** attribute is used to set the horizontal extent of the object in pixels.

This example displays the Microsoft MSCAL calendar.

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML object Tag Example</title>
</head>
```

```
<body>
<object classid="clsid:8E27C92B-1264-101C-8A2F-040224009C02" ID="calFormDate"
width="372" height="200">
Sorry, your browser does not display the Microsoft MSCAL Calendar
</object>
</body>
</html>
```

TAG: ol

** ... **

The **ol** tag is used to delimit the start and stop of an ordered list. An ordered list is a collection of items (typically related somehow) that need to be listed in particular order. For example, an ordered list could be an index, table of contents, or a set of instructions. The default is to list each item in numeric order (starting with the number 1). However, you can also specify Roman numerals or alphabetic characters.

Most browsers separate the list from any preceding and following text by paragraph breaks. You can nest ordered lists and the nested lists will also be in ordered value.

You must use the **li** tag to display an item in the list.

You can use the **ul** tag to create an unordered list and the **dl** tag to create a definition list.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML ol Tag Example</title>
</head>
<body>
Three Major Types Of Lists:
<ol>
<li> Definition List</li>
<ol>
<li> dd tag</li>
<li> dl tag</li>
<li> dt tag</li>
</ol>
<li> Ordered List</li>
<ol style="list-style-type: upper-roman;">
<li> ol tag</li>
<li> li tag</li>
</ol>
<li> Unordered List</li>
<ol style="list-style-type: lower-alpha;">
<li> ul tag</li>
<li> li tag</li>
</ol>
</ol>
</body>
```

```
</html>
```

Output:

Three Major Types Of Lists:

1. Definition List
    1. dd tag
    2. dl tag
    3. dt tag
2. Ordered List
    1. ol tag
    2. li tag
3. Unordered List
    1. ul tag
    2. li tag

TAG: optgroup

<optgroup> ... </optgroup>

The **optgroup** tag is used to group together related items in a pull-down form control. An item can only be added to the group by using the **option** tag. Both the **optgroup** and **option** tags must be used inside the opening and closing **select** tags.

You cannot nest this tag.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

disabled
The **disabled** attribute is used to disable the caption group. The only permitted value is diabled.

label
The **label** attribute is used to specify a short caption.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML optgroup Tag Example</title>
</head>
<body>
<select>
<optgroup label="Mexican tropical fish">
<option>Molly</option>
<option>Platy</option>
<option>Sword Tail</option>
</optgroup>
<optgroup label="Goldfish">
<option>Comet</option>
<option>Oranda</option>
<option>Ryukin</option>
</optgroup>
</select>
</body>
</html>
```

TAG: option

<option> ... </option>

The **option** tag is used to insert one item in a pull-down form control. You may repeat this tag an unlimited number of times in a pull-down list. (For example, you could list all fifty states.) By default, when an item is chosen, the value that is added to the contents of the **form** is the text that appears immediately after the opening **option** tag. However, you can specify a different value by using the **value** attribute.

This tag must be placed between the opening and closing **select** tags. The order of the **option** tags determines the default order of the list. However, you can use the **selected** attribute to specify one item always to appear at the top of the list.

In general, a pull-down list should be treated as a menu. It is recommended that the text for each item in the list be short, such as one or two words. A long text can be far too unwieldy to read and will probably annoy most users.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

disabled
The **disabled** attribute is a Boolean value that, if present, prevents the form control (field) from functioning. In some browsers, the control (field) will appear to be greyed out.

label
The **label** attribute is used to create a short text version of the text that appears after the opening **option** tag.

selected
The **selected** attribute is used to specify which one item in the pull-down will appear at the top of the list. Some browsers highlight the selected item. The default is that the list is in the same order as the **option** tags and there is no highlighting

value
The **value** attribute is used when you need to set a different value for the item in the pull-down that what is listed in the **option** tag.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML option Tag Example</title>
</head>
<body>
<select size="2">
Gold Fish
```

```
<br />
<option selected>Bubble-Eye</option>
<option>Comet</option>
<option>Common</option>
<option>Oranda</option>
<option>Pearl Scale</option>
<option>Ranchu</option>
<option>Ryukin</option>
</select>
```

Output:

Gold Fish

TAG: p

** ... **

The ** ** tag is used to signify the beginning of a paragraph and is analogous to having both a line break and a carriage return occur. The flow of the display of the text and any images is halted on the current line, an entire line is skipped (or a certain amount of white vertical space), and then the flow resumes starting at the left margin (by default).

The closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML p Tag Example</title>
</head>
<body>
My favorite fish!
<p />
The pseudotrophus "Acei" cichlid came from Lake Malawi in Africa...
<p />
...while the Showa Koi in my pond came from Japan.
</body>
</html>
```

Output:

My favorite fish!

The pseudotropheus "Acei" cichlid came from Lake Malawi in Africa...

...while the Showa Koi in my pond came from Japan.

TAG: param

<param ... />

The **<param>** tag is used to set the name/value pair that provides a parameter required by a Java object in an **applet** tag or by some object in an **object** tag. Each **param** tag can provide only one parameter. However, there can be any number of parameters (from zero to many). These parameters, including the name and acceptable range of values (defaults, etc.), are established by the author of the applet or object. For example, they could set the color of the applet.

This tag is placed immediately after the opening **applet** or **object** tag.

This tag is self-closing tag. The blank space before the slash (/) is recommended for maximum browser compatibilty.

Attributes and Events

id

name
The **name** attribute is the name portion of the name/value pair. You must provide a value for this attribute. The exact value (spelling, etc.) of the **name** will be determined by the author of the object.

type
The **type** attribute defines the MIME media type of the value when the **valuetype** attribute is set to **ref**.

value
The **value** attribute provides the value required by the parameter for the name/value pair. You must provide a value for this parameter.

valuetype
The **valuetype** attribute defines the type of the value. There are three permitted values **data, object,** and **ref**. The default is **data**. The **ref** is the URL source that contains the value.

This example demonstrates that even a simple Microsoft Active-X button can have numerous parameters.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML param Tag Example</title>
</head>
<body>
<object id="CommandButton" classid="CLSID:D7053240-CE69-11CD-A777-00DD01143C57" style="width:200pt; height:25pt;">
<param name="Caption" value="Command Button" />
<param name="FontEffects" value="1073741825" />
<param name="FontHeight" value="200" />
```

```
<param name="FontCharSet" value="0" />
<param name="FontPitchAndFamily" value="2" />
<param name="FontWeight" value="700" />
<param name="ParagraphAlign" value="3" />
<param name="Size" value="2540;847" />
</object>
</body>
</html>
```

TAG: pre

<pre> ... </pre>

The **pre** tag is used to display pre-formatted text. The output will mimic exactly how the text is rendered inside the **pre** element including white space, tabs, and line breaks. This allows you to maintain the appearance of data in rows and columns, or have extra white spaces in text such as poems. Perhaps one of the most common uses of this tag is to display computer code and output.

The browser will display the text in a mono spaced font. However, you can use style sheets, including the **style** attribute, to effect the appearance of the text to suit your desires.

The following tags may not appear inside a **pre** element: **big, img, object, small, sub,** and **sup**. Therefore, images, which require an **img** tag, cannot appear in a **pre**.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

The example is shown twice, with and without the use of style sheets. (Note that HTML does not show the extra white spaces that are present in the code.)

Code without style:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML pre Tag Example</title>
</head>
<body>
<pre>
function CheckFrames()
{
var strPage, strURL
strPage = location.pathname + location.search
strURL = location.protocol + "//" + location.hostname +
"/index.asp?page=" + escape(strPage)

if ((window.name != "body") || (window.parent.name != "frameset") ||
(window.parent.parent.parent != window.parent.parent))
{
window.top.location.replace(strURL)
}
}
</pre>
</body>
</html>
```

Code with style:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
```

```
<head>
<title>DevGuru XHTML pre Tag Example</title>
</head>
<body>
<pre style="font-family:arial; color:purple; font-size:12px;">
function CheckFrames()
{
var strPage, strURL
strPage = location.pathname + location.search
strURL = location.protocol + "//" + location.hostname +
"/index.asp?page=" + escape(strPage)

if ((window.name != "body") || (window.parent.name != "frameset") ||
(window.parent.parent.parent != window.parent.parent))
{
window.top.location.replace(strURL)
}
}
</pre>
</body>
</html>
```

Output without style:

```
function CheckFrames()
{
        var strPage, strURL
        strPage = location.pathname + location.search
        strURL = location.protocol + "//" + location.hostname +
                        "/index.asp?page=" + escape(strPage)

        if ((window.name != "body") || (window.parent.name != "frameset") ||
                (window.parent.parent.parent != window.parent.parent))
        {
                window.top.location.replace(strURL)
        }
}
```

Output with style:

```
function CheckFrames()
{
      var strPage, strURL
      strPage = location.pathname + location.search
      strURL = location.protocol + "//" + location.hostname +
                  "/index.asp?page=" + escape(strPage)

      if ((window.name != "body") || (window.parent.name != "frameset") ||
            (window.parent.parent.parent != window.parent.parent))
      {
            window.top.location.replace(strURL)
      }
}
```

TAG: q

<q> ... </q>

The **q** tag takes the selected block of text and displays it as a quote which is enclosed by a pair of double quotes. Unlike the **blockquote** tag, the **q** quote is not preceded or followed by paragraph breaks. So the quote remains inline. You may still need to use other HTML tags to effect the textual display and appearance that you desire.

It is recommended that you use the **blockquote** tag for long citations and the **cite** tag for short citations.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

cite
The **cite** attribute is used to provide a reference to the URL that is the source of the quote.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML q Tag Example</title>
</head>
<body>
<q>His enthusiasm was only exceeded by his lack of enthusiasm!</q>
</body>
</html>
```

Output:

His enthusiasm was only exceeded by his lack of enthusiasm!

TAG: samp

<samp> ... </samp>

The **samp** tag is designed to place emphasis on a short string of special characters. The designated text is displayed in a teletype or mono space font face. The actual appearance will be determined by the browser.

For many browsers, the display font appearance for the **code**, **kbd**, **samp**, and **tt** tags is the same.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML samp Tag Example</title>
</head>
<body>
<samp>DevGuru</samp>
</body>
</html>
```

Output:

DevGuru

TAG: small

<small> ... </small>

The **small** tag causes the designated text to be displayed in a smaller font size. The actual amount of change will be determined by the browser. If the text is already at the minimum size, this attribute will have no effect.

This can also be done using the style sheets **font-size** property.

The separate closing tag is mandatory.

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML small Tag Example</title>
</head>
<body>
<big>THINK</big> very <small>small</small>
</body>
</html>
```

Output:

THINK very small

TAG: span

** ... **

The **span** tag is used to designate an inline portion of an HTML document as a **span** element and to apply any of the core attributes of this tag against the contents of that element. The behavior of this tag is inline because the effects occur in the normal flow of the text and images (without the tag inserting any additional linefeeds or carriage returns). **span** tags, which can be nested, allow a significant degree of control and manipulation of a localized part of your web page.

For example, you could use the **class** or **style** core attributes to apply the effects of Cascading Style Sheets or you could use the **lang** attribute to display a foreign language text.

In contrast, you should use the **div** tag when you want to apply attributes to a block of code. Note that the behavior of the **div** tag is not inline, but rather is block-line, since this tag causes a line break to occur both before and after the **div** element.

With the advent of Cascading Style Sheets, both the **div** and **span** tags have gained importance and usefulness.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

As an example, all of the code samples on the **DevGuru** site are in blue colored text. This is done by enclosing the code sample inside a pair of **span** tag elements and designating a **class** attribute of "CODE". In turn, the "CODE" class has been assigned a text color of blue in the style sheets file (a .css file) that is located in the Include/ directory on the **DevGuru** web site. Thus, only one file is needed to define the appearance of the entire site.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML span Tag Example</title>
</head>
<body>
<span class="CODE">
This line is blue.
</span>
</body>
</html>
```

Output:

This line is blue.

However, if you do not wish to create a .css file, you can still get the exact same results by

using the **style** attribute for the **span** tag and the style sheet **color** property.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML span Tag Example</title>
</head>
<body>
<span style="color: blue;">
This line is blue.
</span>
</body>
</html>
```

Output:

This line is blue.

TAG: strong

** ... **

The **strong** tag is is used to provide a strong emphasis when displaying the enclosed word or phrase. Most browsers do this by rendering the selected text in bold which makes the characters thicker and slightly bigger in appearance than the surrounding text. Of course, you can also do this using the **b** tag.

If you wish to have less emphasis, you can use the **em** tag which usually renders in italics, rather than in bold.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML strong Tag Example</title>
</head>
<body>
Did you know that the <b>Guru</b> is over <strong>800</strong> years old?
<br />
Did you know that the <b>Guru</b> is over <em>800</em> years old?
</body>
</html>
```

Output:

Did you know that the **Guru** is over **800** years old?
Did you know that the **Guru** is over 800 years old?

TAG: sub

_{...}

The **sub** tag is used to insert subscript into a text. This allows the correct display of chemical equations, footnotes, math formulas, and scientific notation on a web page.

Most browsers reduce the text size by half for the subscript (and superscript), but use the same font face as used for the rest of the text. However, you can use style sheets and the **style** attribute to effect the appearance of the text to suit your desires.

The similar **sup** tag is used to create superscripts.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML sub Tag Example</title>
</head>
<body>
2H<sub style="color:green;">2</sub>+O<sub style="color:green>2</sub> -> 2H<sub style="color:green;">2</sub>O
</body>
</html>
```

Output:

$2H_2+O_2 \rightarrow 2H_2O$

TAG: sup

^{...}

The **sup** tag is used to insert superscript into a text. This allows the correct display of footnotes, math formulas, and scientific notation on a web page. Most browsers reduce the text size by half for the superscript (and subscript), but use the same font face as used for the rest of the text.

However, you can use style sheets and the **style** attribute to effect the appearance of the text to suit your desires.

The similar **sub** tag is used to create subscripts.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML sup Tag Example</title>
</head>
<body>
E = MC<sup>2</sup>
</body>
</html>
```

Output:

E = MC^2

TAG: table

<table> ... </table>

The **table** tag is used to designate that this element is a table. A table is a structural presentation of data or information using rows and columns.

The default is that the flow of any text and images is stopped in the HTML document, the **table** element is inserted starting on the next line, and after the end of the **table** element is reached, the flow of text and images resumes on the following line. However, the use of attributes and style sheet properties will allow you to embed the **table** element within the text and images.

The insides of the **table** are constructed using the **tr, th, td,** and the **caption** tags. There are also three new table tags that were released with HTML version 4.0 and are included in XHTML version 1.0. They are **tbody, tfoot,** and **thead**.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

border
The **border** attribute informs the browser to draw lines around the entire table and all of the rows and cells. The lines are shaded to give an embossed appearance. You declare the thickness of the border line in pixels as an integer. The default is no border (zero pixels).

cellpadding
The **cellpadding** attribute sets the amount of white space to place between a cell wall and the contents of a cell. You declare the amount in pixels as an integer. The default is one pixel.

cellspacing
The **cellspacing** attribute sets the amount of white space to place between each adjacent cell and between each cell and the outer border of the table. You declare the amount in pixels as an integer. The default is two pixels.

frame
The **frame** attribute is used to determine which of the four lines that make up the outside border of a table will be visible. The permitted values are **above, below, border, box, hsides, lhs, rhs, void,** and **vsides**.

rules
The **rules** attribute is used to control whether the border lines will appear around cells inside the table. You can specify **row** only to have borders between rows, **col** for only between columns, and **none** for no internal borders.

summary
The **summary** attribute is used with non-visual media such as braille and speech to provide information about a table.

width
The **width** attribute is used to set the width of a table. It can be declared either as an integer

number of pixels or as a percentage of the width of the display window.

This example demonstrates the basic table tags:

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML table Tag Example</title>
</head>
<body>

<table width="80%" border="10" cellpadding="3" cellspacing="5" align="center">
<caption>
<b>How To Code A Table</b>
<br />
Use the caption tag to name the table and to provide useful information about the contents
</caption>
<tr>
<th colspan="2">Use the <b>th</b> tag to create the header which will display in bold</th>
</tr>
<tr>
<td style="width:50%">The <b>tr</b> tag creates the row</td>
<td style="width:50%">The <b>td</b> tag creates individual cells for each row</td>
</tr>
<tr>
; <td style="width:50%">You should always use the closing tags</td>
<td style="width:50%">You can use a variety of XHTML tags inside the cell</td>
</tr>
<tr>
<td style="width:50%">The three new table tags are <b>tbody, tfoot,</b> and
<b>thead</b></td>
<td style="width:50%">The <b>colspan</b> attribute allows you span more than one cell</td>
</tr>
</table>

</body>
</html>
```

Output:

How To Code A Table

Use the caption tag to name the table and to provide useful information about the contents

Use the th tag to create the header which will display in bold	
The **tr** tag creates the row	The **td** tag creates individual cells for each row

You should always use the closing tags	You can use a variety of XHTML tags inside the cell
The three new table tags are **tbody, tfoot,** and **thead**	Setting **colspan** allows you span more than one cell

TAG: tbody

<tbody> ... </tbody>

The **tbody** tag is one of three tags that divide a **table** into three distinct sections.

The **tbody** tag defines the body portion where the data is displayed, the **thead** tag defines the header section, and the **tfoot** tag defines the footnote section of the **table**. The purpose of this division is to allow scrolling through the body of a large table while both the header and footnote portions remain visible.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents for all of the cells in a single row. The five permitted values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of all of the rows in a cell. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents for all of the cells in a single row. The possible values are **baseline, bottom, middle,** and **top**

Note the the correct order for using these tags is **thead,** then **tfoot**, and last **tbody**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML tbody Tag Example</title>
</head>
<body>
<table border="1">
<caption>
<b>TABLE</b>
<br />
thead, tfoot & tbody
</caption>
```

```
<thead>
<tr><th>Header 1</th><th>Header 2</th></tr>
</thead>
<tfoot>
<tr><td colspan="2">footnote</td></tr>
</tfoot>
<tbody>
<tr><td>data cell 1</td><td>data cell 2</td></tr>
</tbody>
</table>
</body>
</html>
```

Output:

TABLE

thead, tfoot & tbody

Header 1	Header 2
footnote	
data cell 1	data cell 2

TAG: td

<td> ... </td>

The **td** tag is used to create cells that contain the data, information, or text that you wish to display in the **table**. You may place as many data cells in a row as you desire.

The **th** tag is used to create a header cell for the cells in a row in a **table** element.

The separate closing tag is mandatory.

The coding sequence is:
<tr><td> ... place data here ... </td></tr>

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

abbr
The **abbr** attribute is not recognized by most browsers. You use it to provide an abbreviated summary of the data cell text.

align
The **align** attribute is used to set the horizontal alignment of the cell contents. The possible values are **center, char, justify, left,** and **right**.

axis
The **axis** attribute is not recognized by most browsers. This attribute is used to create a list of category names which form the basis of a query. This query is executed against the cells that form the table.

char
The **char** attribute permits you to select a character that will be used to align the contents of the cell. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

colspan
The **colspan** attribute allows a header to span horizontally two or more columns (cells). This may be combined with the **rowspan** attribute to form data cells that encompass several rows and columns.

headers
The **headers** attribute is not recognized by most browsers. The value of this attribute is a list of cell names. The names are the values of the cells' **id** attributes. This attribute is used with speech capable browsers.

rowspan

The **rowspan** attribute allows a cell to extend down two or more rows. This may be combined with the **colspan** attribute to form data cells that encompass several rows and columns.

scope
The **scope** attribute is not recognized by most browsers. It is used to assign a set of data cells to one header cell. The four possible values are **col, colgroup, row,** and **rowgroup**.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents. The possible values are **baseline, bottom, middle,** and **top**.

This example demonstrates the basic table tags:

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML td Tag Example</title>
</head>
<body>
<table width="80%" border="10" cellpadding="3" cellspacing="5" align="center">
<caption>
<b>How To Code A Table</b>
<br />
Use the caption tag to name the table and to provide useful information about the contents
</caption>
<tr>
<th colspan="2">Use the <b>th</b> tag to create the header which will display in bold</th>
</tr>
<tr>
<td style="width:50%>The <b>tr</b> tag creates the row</td>
<td style="width:50%">The <b>td</b> tag creates individual cells for each row</td>
</tr>
<tr>
<td style="width:50%">You should always use the closing tags</td>
<td style="width:50%">You can use a variety of HTML tags inside the cell</td>
</tr>
<tr>
<td style="width:50%">The three new table tags are <b>tbody, tfoot,</b> and
<b>thead</b></td>
<td style="width:50%">The <b>colspan</b> attribute allows you span more than one cell</td>
</tr>
</table>
</body>
</html>
```

Output:

How To Code A Table

Use the caption tag to name the table and to provide useful information about the contents

Use the th tag to create the header which will display in bold	
The **tr** tag creates the row	The **td** tag creates individual cells for each row
You should always use the closing tags	You can use a variety of HTML tags inside the cell
The three new table tags are **tbody, tfoot,** and **thead**	Setting **colspan** allows you span more than one cell

TAG: tfoot

<tfoot> ... </tfoot>

The **tfoot** tag is one of three tags that divide a **table** into three distinct sections.

The **tbody** tag defines the body portion where the data is displayed, the **thead** tag defines the header section, and the **tfoot** tag defines the footnote section of the **table**. The purpose of this division is to allow scrolling through the body of a large table while both the header and footnote portions remain visible.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents for all of the cells in a single row. The five permitted values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of all of the rows in a cell. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents for all of the cells in a single row. The possible values are **baseline, bottom, middle,** and **top**.

Note the the correct order for using these tags is **thead,** then **tfoot**, and last **tbody**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML tfoot Tag Example</title>
</head>
<body>
<table border="1">
<caption>
<b>TABLE</b>
<br />
thead, tfoot & tbody
</caption>
```

```
<thead>
<tr><th>Header 1</th><th>Header 2</th></tr>
</thead>
<tfoot>
<tr><td colspan="2">footnote</td></tr>
</tfoot>
<tbody>
<tr><td>data cell 1</td><td>data cell 2</td></tr>
</tbody>
</table>
</body>
</html>
```

Output:

TABLE

thead, tfoot & tbody

Header 1	Header 2
footnote	
data cell 1	data cell 2

TAG: th

<th> ... </th>

The **th** tag is used to create a header cell for the cells in a row in a **table** element. The purpose of the header is to provide a caption, name, or information about the column of cells. The header text will be displayed in a bold font style.

The **td** tag is used to create cells that contain the data, information, or text that you wish to display in the **table**.

The separate closing tag is mandatory.

The coding sequence is:
<tr><th> ... place header here ... </th></tr>

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

abbr
The **abbr** attribute is not recognized by most browsers. You use it to provide an abbreviated summary of the header text.

align
The **align** attribute is used to set the horizontal alignment of the cell contents. The possible values are **center, char, justify, left,** and **right**.

axis
The **axis** attribute is not recognized by most browsers. This attribute is used to create a list of category names which form the basis of a query. This query is executed against the cells that form the table.

char
The **char** attribute permits you to select a character that will be used to align the contents of the cell. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

colspan
The **colspan** attribute allows a header to span horizontally two or more columns (cells). This may be combined with the **rowspan** attribute to form data cells that encompass several rows and columns.

headers
The **headers** attribute is not recognized by most browsers. The value of this attribute is a list of cell header names. The names are the values of the cells' **id** attributes. This attribute is used with speech capable browsers.

rowspan
The **rowspan** attribute allows a cell to extend down two or more rows. This may be combined with the **colspan** attribute to form data cells that encompass several rows and columns.

scope
The **scope** attribute is not recognized by most browsers. It is used to assign a set of data cells to one header cell. The four possible values are **col, colgroup, row,** and **rowgroup**.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents. The possible values are **baseline, bottom, middle,** and **top**.

This example demonstrates the basic table tags:

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML th Tag Example</title>
</head>
<body>
<table width="80%" border="10" cellpadding="3" cellspacing="5" align="center">
<caption>
<b>How To Code A Table</b>
<br />
Use the caption tag to name the table and to provide useful information about the contents
</caption>
<tr>
<th colspan="2">Use the th tag to create the header which will display in bold</th>
</tr>
<tr>
<td style="width:50%">The <b>tr</b> tag creates the row</td>
<td style="width:50%">The <b>td</b> tag creates individual cells for each row</td>
</tr>
<tr>
<td style="width:50%">You should always use the closing tags</td>
<td style="width:50%">You can use a variety of HTML tags inside the cell</td>
</tr>
<tr>
<td style="width:50%">The three new table tags are <b>tbody, tfoot,</b> and
<b>thead</b></td>
<td style="width:50%">The <b>colspan</b> attribute allows you span more than one cell</td>
</tr>
</table>
</body>
</html>
```

Output:

How To Code A Table

Use the caption tag to name the table and to provide useful information about the contents

Use the th tag to create the header which will display in bold	
The **tr** tag creates the row	The **td** tag creates individual cells for each row
You should always use the closing tags	You can use a variety of HTML tags inside the cell
The three new table tags are **tbody, tfoot,** and **thead**	Setting **colspan** allows you span more than one cell

TAG: thead

<thead> ... </thead>

The **thead** tag is one of three tags that divide a **table** into three distinct sections.

The **tbody** tag defines the body portion where the data is displayed, the **thead** tag defines the header section, and the **tfoot** tag defines the footnote section of the **table**. The purpose of this division is to allow scrolling through the body of a large table while both the header and footnote portions remain visible and fixed in place.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents for all of the cells in a single row. The five permitted values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of all of the rows in a cell. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents for all of the cells in a single row. The possible values are **baseline, bottom, middle,** and **top**.

Note the the correct order for using these tags is **thead,** then **tfoot**, and last **tbody**.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML thead Tag Example</title>
</head>
<body>
<table border="1">
<caption>
<b>TABLE</b>
<br />
thead, tfoot & tbody
</caption>
```

```
<thead>
<tr><th>Header 1</th><th>Header 2</th></tr>
</thead>
<tfoot>
<tr><td colspan="2">footnote</td></tr>
</tfoot>
<tbody>
<tr><td>data cell 1</td><td>data cell 2</td></tr>
</tbody>
</table>
</body>
</html>
```

Output:

TABLE

thead, tfoot & tbody

Header 1	Header 2
footnote	
data cell 1	data cell 2

TAG: tr

<tr> ... </tr>

The **tr** tag is used to create a row in a **table** element. You can create as many rows as you wish.

This row can contain one or more cells where each cell can contain data, information, or text that you wish to display in the **table**. These cells are created by using either the **td** or **th** tags. The **tr** tag should never appear inside of a **th**, **td**, or other **tr** tags.

The separate closing tag is mandatory.

The coding sequence is:
<tr><th> ... place header here ... </th></tr>
<tr><td> ... place data here ... </td></tr>

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

align
The **align** attribute is used to set the horizontal alignment of the cell contents for all of the cells in a single row. The five permitted values are **center, char, justify, left,** and **right**.

char
The **char** attribute permits you to select a character that will be used to align the contents of all of the rows in a column. This was designed to allow the decimal points in a column of numbers to be in vertical alignment. The default value is country dependent. In the U.S., it is the first occurrence of a period or a decimal point.

charoff
The **charoff** attribute is an integer used to specify if the value designated in the **char** attribute is to be the first, second, or whatever occurrence.

valign
The **valign** attribute is used to set the vertical alignment of the cell contents for all of the cells in a single row. The possible values are **baseline, bottom, middle,** and **top**.

This example demonstrates the basic table tags:

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML tr Tag Example</title>
</head>
<body>
<table width="80%" border="10" cellpadding="3" cellspacing="5" align="center">
<caption>
```

```
<b>How To Code A Table</b>
<br />
Use the caption tag to name the table and to provide useful information about the contents
</caption>
<tr>
<th colspan="2">Use the <b>th</b> tag to create the header which will display in bold</th>
</tr>
<tr>
<td style="width:50%">The <b>tr</b> tag creates the row</td>
<td style="width:50%">The <b>td</b> tag creates individual cells for each row</td>
</tr>
<tr>
<td style="width:50%">You should always use closing tags</td>
<td style="width:50%">You can use a variety of HTML tags inside the cell</td>
</tr>
<tr>
<td style="width:50%">The three new table tags are <b>tbody, tfoot,</b> and
<b>thead</b></td>
<td style="width:50%">The <b>colspan</b> attribute allows you span more than one cell</td>
</tr>
</table>
</body>
</html>
```

Output:

How To Code A Table

Use the caption tag to name the table and to provide useful information about the contents

Use the th tag to create the header which will display in bold	
The **tr** tag creates the row	The **td** tag creates individual cells for each row
You should always use the closing tags	You can use a variety of HTML tags inside the cell
The three new table tags are **tbody, tfoot,** and **thead**	Setting **colspan** allows you span more than one cell

TAG: tt

<tt> ... </tt>

The **tt** tag causes the designated text to be displayed in a teletype or mono space font face. The actual appearance will be determined by the browser.

For many browsers, the same font is used for the **code**, **kbd**, **samp**, and **tt** tags.

The separate closing tag is mandatory.

Core attributes:

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML tt Tag Example</title>
</head>
<body>
<tt>The tt tag is named teletype.</tt>
</body>
</html>
```

Output:

```
The tt tag is named teletype.
```

TAG: ul

** ... **

The **ul** tag is used to delimit the start and stop of an unordered list. An unordered list is a collection of items (typically related somehow) that are in no particular order. Each item in the list is preceded by a bullet with the default being a round disc: •

Most browsers separate the list from any preceding and following text by paragraph breaks. You can nest unordered lists and each nested list can have a different bullet.

You must use the **li** tag to display an item in the list.

You can use the **ol** tag to create an ordered list and the **dl** tag to create a definition list.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML ul Tag Example</title>
</head>
<body>
Three Major Types Of Lists:
<ul>
<li> Definition List</li>
<ul style="list-style-type: square>
<li> dd tag</li>
<li> dl tag</li>
<li> dt tag</li>
</ul>
<li> Ordered List</li>
<ul style="list-style-type: circle;">
<li> ol tag</li>
<li> li tag</li>
</ul>
<li> Unordered List</li>
<ul style="list-style-type: disc">
<li> ul tag</li>
<li> li tag</li>
</ul>
</ul>
</body>
</html>
```

Output:

Three Major Types Of Lists:

- Definition List
  - dd tag
  - dl tag
  - dt tag
- Ordered List
  - ol tag
  - li tag
- Unordered List
  - ul tag
  - li tag

TAG: var

<var> ... </var>

The **var** tag is used to signify that the designated word is a variable name in a listing of computer code. This tag is used inside **code** and **pre** tag elements. On most browsers, the word or text is rendered in an italic or oblique font style.

The separate closing tag is mandatory.

Attributes and Events

class dir id lang onclick ondblclick onkeydown onkeypress onkeyup onmousedown onmousemove onmouseout onmouseover onmouseup style title

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML var Tag Example</title>
</head>
<body>
<pre>
if (document.getElementById) <var>blnDOM<>/var> = true;
</pre>
</body>
</html>
```

Output:
```
if (document.getElementById) blnDOM = true;
```

Or:

Code:
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML var Tag Example</title>
</head>
<body>
<code>
if (document.getElementById) <var>blnDOM</var> = true;
</code>
</body>
</html>
```

Output:
```
if (document.getElementById) blnDOM = true;
```

EVENT: onload

onload="action"

The **onload** event occurs when the HTML document is loaded into a window for viewing. When the event happens, the code calls a function that performs a desired action. For example, you may want an alert box to appear that provides useful information to the user.

It can be used with either the **body** or **frameset** tags.

In contrast, the **onunload** event occurs just before the HTML document is unloaded or removed from viewing.

In this form example, the **onload** event is used in the **body** tag to set the focus to the Full Name input box.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru Test Form</title>
<script type="text/javascript">
function checksubmit()
{
  if (document.formname.fullname.value == "")
  {
    alert("Please enter your full name")
    document.formname.fullname.focus()
    return false
  }
  if (document.formname.emailaddress.value == "")
  {
    alert("Please enter your email address")
    document.formname.emailaddress.focus()
    return false
  }
  return true
}
</script>
</head>
<body onload="document.formname.fullname.focus()">
<b>
FORM EXAMPLE
<br /><br />
Please leave one or both of the required fields blank and click the submit button.
</b>
<br />
<hr />
<br />
If you wish to receive information about upgrades to dgCharge,<br />
please fill out this form.
<br /><br />
<form method="post" name="formname" action="html_form_example.asp" onsubmit="return
```

```
checksubmit()">
Full Name (required)
<input type="text" name="fullname" size="30" />
<br /><br />
Email Address (required)
<input type="text" name="emailaddress" size="30" />
<br /><br />
Phone Number (optional)
<input type="text" name="phonenumber" size="15" />
<br /><br />
<input type="submit" name="submitbtn" value="Submit" />
<input type="reset" value="Clear" />
</form>
</body>
</html>
```

EVENT: onunload

onunload="action"

The **onunload** event occurs just before the HTML document is unloaded or removed from viewing (also when the page is refreshed). When this event happens, the code calls a function that performs a desired action. For example, you may want an alert box to appear that provides useful information to the user.

It can be used with either the **body** or **frameset** tags.

In contrast, the **onload** event occurs just after the HTML document is loaded for viewing.

In this example, the **onunload** event causes an alert box with a message to appear when you exit the page.

Code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="eng">
<head>
<title>DevGuru XHTML onunload Tag Example</title>
</head>
<body>
<body onunload="JavaScript:alert('Thank you for\nvisiting DevGuru!')">
Please exit or refresh this page to see output.
<br />
The alert box should appear.
</body>
</html>
```