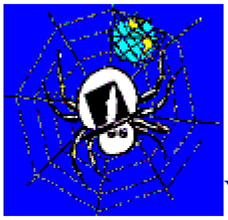


Webber™ Validation

The Hypertext Mark-up Language is a simple mark-up language that has some complex rules. The rules governing the proper usage of HTML are specified in the HTML Document Type Definition (dtd). Webber™ comes with a built-in validation system that helps you check your document against the rules laid out in the dtd. While it would be nice to just give you a list of the rules, it is not that simple. The rules of HTML are context-sensitive and what is true in one case may not be in another. The goal of this tutorial is to help you learn more about HTML and how the rules are applied. When you have worked with Webber for a while, it will all become second nature and you will learn a lot about HTML in the process. Webber's validation feature gives you complete control over the mark up in your document. As you learn more about HTML and what constitutes a valid document, you will be able to choose which rules are important to you.

Some of the links in this help file are connected to the Webber Help file. This file should be in the same directory as webber.hlp in order for it to work properly.

HTML as a Structured Language
Structure vs. Layout
How Our Validation Works
Selecting a dtd
The Validation Window
Common Errors
Validation Messages
Tutorial

Feedback?? We are constantly striving to improve our documentation. If you found this tutorial to be particularly good or bad, please let us know. E-mail your comments to webber@csdcorp.com.

HTML as a Structured Language

HTML stands for HyperText Markup Language. Based on SGML, it is a simple markup language used to define the structure and organization of a document in order to create documents which are portable from one environment to another.

Many people think of HTML Tags as a form of 'styles' such as those used in word processors. HTML Tags differ from styles because styles are usually 'flat' which means they are allowed anywhere in a document, while tags are only allowed in certain places or 'contexts'. A context is just a set of surrounding tags, for example the HTML and HEAD tags would form a context where the TITLE tag is allowed, this tag is not allowed anywhere else in the document. The relationship between the tags is at least as important as the way in which they are rendered (formatted) by browsers.

Browsers rely on the tags that mark-up a document to tell them how the document is structured and how they should appear. There are many cases where tags make sense only within a specific context.

Some of the context-sensitivity of HTML is obvious. For example HTML provides the capability of creating a form into which a user can type information and then submit the information by clicking a button. What happens to the information submitted is determined by the attributes specified within the FORM start tag. Imagine an INPUT field that was not contained inside a form. While the user might be able to type into the field there would be no way to submit that information because there are no instructions as to what should be done with it. Therefore in this example it is obvious that an INPUT field is only valid within the context of a form.

In other cases, the rules governing the context of tags in relation to each other are more subtle. The IETF is responsible for designing the HTML specification (dtd) in order to provide a consistent guide for both authors of HTML and developers of software such as editors and browsers. It is part of the specification that browsers should be very forgiving in what they can display while authors (and editing tools) should be very strict in what they produce. This is why browsers will often display properly even if your document is not valid. See Structure vs. Layout for examples why this doesn't make it unneccessary for you to have a valid document. This rule is simple yet important - it may be all that stands between the Web as a powerful medium for information distribution and the web as a chaotic collection of sporadically decipherable documents.

While HTML is a simple mark-up language, the dtd for HTML is far from simple to draft and to get agreement on. It is difficult to come up with a specification that satisfies the many requirements of the marketplace while maintaining the simplicity and consistency that is required in order for the Web to flourish.

Structure vs. Layout

All right, so we have established that HTML was originally intended to define the structure of a document, so what? Should you sacrifice the appearance of your document just to conform with a standard?

The two aspects of HTML page creation do not have to exclude each other. Many Web page designers design creative and interesting pages while working within the HTML spec.

However there are others who feel that the specification is too restrictive and choose to use HTML and extensions made to it in ways that do not fit with the dtd.

Your choice will vary depending on the intended use and life span of the document. HTML documents generally fall along a continuum of life span and purpose. At one end there are pages that will have a limited life span and where appearance is critical (an advertisement for example). At the other end there are pages that will have exceptionally long life spans and where the content is critical (a research paper for example). For the former it might make sense to 'flout' the standard but in the latter it does not.

Consider the following examples of HTML:

```
<HTML><HEAD><TITLE>Heading Example</TITLE></HEAD><BODY>
<H1>This is a Heading 1. It is usually rendered in a large bold
font.</H1>
</BODY></HTML>
```

```
<HTML><HEAD><TITLE>Font Size Example</TITLE></HEAD><BODY>
<FONT size="+3"><B>This is text also rendered in a large bold
font.</B></FONT>
</BODY></HTML>
```

The two examples will look almost identical in your browser, so what difference does it make which one you use? Web browsers are continuing to develop new, powerful features. It is conceivable that in the very near future a browser might be able to automatically generate a table of contents for a document based on the headings found in it. An H1 tag would be meaningful to such an algorithm while FONT, B would not. Or perhaps they will be able to display the page in a collapsible outline style based on headings, lists, etc. Or you may be able to search a set of documents for a specific heading, table, or list. These types of searches would be made possible by consistent formatting and tagging of documents where full text searches of the content would not be practical.

Another point to consider is that browsers are offering more and more control to the user over how they would like a document to appear. Some might like headings to be in a 14 point font while others might like 24. If you use a heading they will be able to control what they see. If you use a font size you are taking that control away from them.

How Our Validation Works

Webber's validation engine uses an SGML compliant parser to check your document. Like a grammar checker, you call it up whenever you wish to check your document for errors in the HTML. The validator will start going through your document, checking for mistakes, syntax errors etc. When it finds a problem it highlights the offending line and displays a message indicating the nature of the error in it's message window.

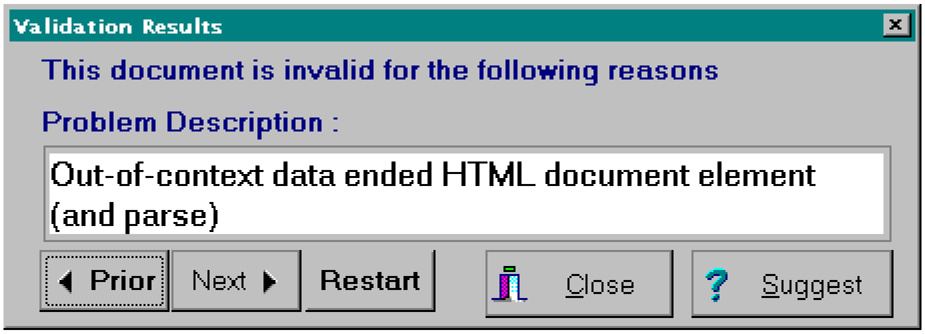
Errors can be caused by several kinds of mistakes. It is important to remember that each HTML dtd has its own allowed grammar and rules. If the validator tells you that there is a problem with your document the first thing you should check is which dtd you have selected in your preferences. It may be that you are using the HTML 2 dtd to validate a document containing elements from the Netscape or Explorer Extensions, or the HTML 3 dtd. Select the dtd you want to use and try the validation again.

When you are sure you are using the rule set that you want to use for your document, read the error message carefully. If you do not understand what it means, click the suggest button. This will open the Webber help file on a topic which lists the most common messages. Find the message you see displayed in the validation window and click on it to see an explanation of the message and suggestions on what to do. Unfortunately the validator may not be able to tell you exactly what you are doing wrong. Because of the complexity of dtds, several different mistakes can cause the same problem.

For more details on how to use the validation, see the tutorial.

The Validation Window

The validation window appears when you start document validation by pressing F2, right-clicking and selecting Validate Document, or by selecting HTML|Document|Validate from the menu. If it finds an error, the validator will highlight the line in your document where the error occurs, and will display a message describing the nature of the mistake. Click on the areas of the window below to find out what they do.



Validation Results

This document is invalid for the following reasons

Problem Description :

Out-of-context data ended HTML document element (and parse)

This area displays a message describing the error found in the document. The line where the error occurs will be highlighted. For hints and suggestions about this error message, click on the suggest button.

When enabled this button will cause the highlight to return to the line where the previous error was found and display the error message relating to it. This button will be *grayed-out* if there is no previous error.

When enabled this button will cause the highlight to move to the line where the next error is found and display the error message relating to it. This button will be *grayed-out* if there are no more errors.

This button will close the validation window.

This button will open the Webber help file at a topic that lists the most common validation problems and gives some suggestions on the types of problems to look for and how they might be resolved. When the help topic appears, scan the list for the error message displayed and click on it for suggestions.

This button starts the parsing (validation) of your document over from the beginning. You can use this to check whether a fix has resolved the problem.

Selecting a dtd

Which dtd you should use for your documents will depend on several factors. Only you can decide how strict you want to be. Please go through the tutorial for examples on the differences between the specifications.

If you want your pages to be accessible to the majority of Web users on any platform, then you should stick to the HTML 2 dtd in Recommended mode.

If you want to take advantage of new features such as tables and alignment options on paragraphs and headings, then you can use the HTML 3 dtd. Since the HTML 3 dtd is still in draft mode, it has been suggested in some of the HTML newsgroups, that you should use the Deprecated mode rather than the Recommended mode for HTML 3. See the tutorial for more information. In Webber, all tags that appear in the HTML 3 dtd, and not in HTML 2 are marked with a * on the tag lists. These tags are also clearly marked in the Webber help file.

The other option is to use either the Netscape or Microsoft Extensions. You might choose to use one of these dtds if you want to use features that are not defined in HTML. You should be aware that many of these extensions are not likely to find their way into the standard and we would recommend using HTML 2 or HTML 3 equivalents where possible. (For example: to center a paragraph use <P ALIGN="Center"></P> rather than the Netscape <CENTER> tag.) Some extensions do not have equivalents however. In Webber, all tags that appear in the Extensions, and not in HTML 2 or 3 are marked with a ** on the tag lists. These tags are also clearly marked in the Webber help file where we have tried to identify their status with regards to the HTML 3 specification.

Deprecated

Deprecated features are features that are either already obsolete or that the IETF (who design the standard), feel will become obsolete in the near future. Use of deprecated features will mean that your document is likely to become non-compliant in the future.

Recommended
Using the Recommended setting removes all deprecated features from the selected dtd so that they can't be used.   It is the strictest form of validation, and enforces a more standardized document so that future, more advanced web utilities will have an easier time understanding and making use of the document's structure.

Validation Tutorial Page 1

⊟
⊞

In this tutorial we will take a simple document and validate it using the HTML 2 and the HTML 3 dtd. The goal of this tutorial is to show you how to use Webber's validation feature and also to point out the most common mistakes when creating HTML. Some of the errors will be found no matter which dtd you are using to validate against while some are dtd specific.

The sample document that we will use is given below. To save yourself some typing, use the 'copy' feature of the help system and paste the sample into a new blank document in Webber. Once you have your document prepared, save it and move to the next page of the tutorial.

We have used the following symbols to guide you through the tutorial section:

⊟ takes you back one page in the tutorial
⊞ takes you to the next page in the tutorial

Webber gives you the option of validating your HTML against two versions of three different specifications. You can use either the recommended or deprecated features of HTML 2.0, HTML 3.0, Netscape Extensions, or Internet Explorer Extensions.

⊞ Next

Sample Document:

```
<HTML>
<HEAD><TITLE>Webber's Validation Tutorial Example
</TITLE></HEAD>
<BODY><H1><CENTER><IMG SRC="image.gif"> Webber's Validation Tutorial
Example</CENTER></H1>
<HR>
The goal of this tutorial is to show you how to use Webber's
validation feature and also to point out the most common mistakes when
creating HTML. (<A HREF=#$1>Jump to item one</A>)
<UL><H2>This is a three item list</H2>
<LI><H3><A HREF="test.htm" NAME="$1">Item one in the
list</A></H3></LI>
<HR><LI>Item two in the list</LI>
<OL><LI>Item one in the second list</LI></OL>
</UL>
<HR><ADDRESS><!--Webber_Auto_StampDNR-->Page maintained by <A
HREF="mailto:webmaster@yourcompany.com">webmaster@yourcompany.com</A>.
Copyright &copy; 1995 <A HREF="http://www.yourcompany.com/">Your
Company</A>.</ADDRESS></BODY></HTML>
```

Validation Tutorial Page 2

⊟
⊞

To prepare for this tutorial, be sure that you have the sample document loaded in Webber. The document has been installed with Webber into the Webber directory, you can open it from there. Or you can use the help file copy feature to copy the sample and then paste it into a new blank Webber document.

First, let's preview the document with your browser. Be sure that you have your browser configured in the preferences, your document saved, and click the 'preview' button on the toolbar. Depending on your browser, you will most likely find that the document loads and looks much like you would expect it to. This is lesson number one. A browser does not care if a document is valid. Some are more robust than others and can handle (almost) anything you throw at them. It is part of the HTML specification that this be so. But this is not always the case and it is a mistake to assume that just because one browser can display a page correctly, that they all can. It is also part of the specification that editors and authors of HTML be much more strict with what they produce than browsers are with what they will display.

You can choose to work through the validation of the sample document with:

⊞HTML 2

⊞HTML 3

We have also included a brief summary of the Netscape and Microsoft Extensions

Tutorial for HTML 2.0

Part 1

⊟
⊞

We will start with validation against the HTML 2.0 Recommended [HTML 2 (R)] dtd. Be sure that you have this selected on the Status Line at the bottom of the Webber Window. If you need to change it, do so by right-clicking on the button and selecting HTML 2 (R) from the list.

This dtd is the one which is supported the most fully by the majority of browsers. Some browsers can handle more complex documents but all browsers will be able to handle HTML 2 as a minimum.

To begin the Validation:

When you have your sample document open and saved, right click and select 'Validate Document' from the pop-up menu.

The first message that we see says "Undefined CENTER start tag GI ignored; not used in dtd". This tells us that the CENTER tag is not used in the dtd that we are validating against so it will be ignored. You will notice that the line containing the center tag is highlighted to show you where the error has been found. If we click the 'Next' button we ill see a similar message for the end tag.

<CENTER> is a Netscape extension so we need to remove it for this to be a valid HTML 2 document. Take out the start and end center tags and restart the validation.

⊞ Next

Tutorial for HTML 2.0

Part 2

⊟

⊞

The next message states that "Out-of-context data ended HTML document element and parse". This tells you that you have used data (text, content) in your document in a place where it was not allowed. When the validator comes across an instance like this it stops parsing (reading) the document because it has lost the context of what is allowed.

The recommended HTML 2 dtd states that all text has to be contained inside a block element such as a paragraph or heading. Text formatting tags such as bold and emphasis do not count as a block element and are not allowed here either.

Place a paragraph tag <P></P> around the text beginning with `"The goal of this tutorial..."` to `"Jump to item one</A>)"` and then restart the validation.

If we had been using the deprecated setting for the dtd rather than recommended, we would have found that the validator would not display this error. Placing text in the body of a document not within a block element was a feature of previous versions of HTML and has been made obsolete in HTML 2 and higher. Be sure that you still have HTML 2 (R) selected as your dtd before continuing.

⊞ Next

Tutorial for HTML 2.0

Part 3

⊞
⊞

The validator will now highlight the line containing the anchor `<A HREF=#$1>Jump to item one</A>` with a message that says "Incorrect character in mark-up; markup terminated". This message is telling you that the document contains illegal mark-up. In this case it is caused by missing quotation marks around the value of the HREF because the # is not a legal character in an attribute value unless it is surrounded by quotation marks. Place double quotes before and after the #$1 ("#$1") and restart the validation.

⊞ Next

More on illegal and incorrect characters:

Messages similar to this can be caused by the use of non-sgml characters in the text. For example, if you have used a word processor such as Microsoft Word to create a document, it may use 'smart quotes' instead of double quotes, or 'smart apostrophes' instead of standard ascii apostrophes. Both of these will cause an error and should be replaced with the standard ascii characters available on your keyboard.

Sometimes you may inadvertently place a character in your document that is not a valid sgml character. Such characters include trademark and copyright symbols, and characters with accents. Some of these characters have sgml entities defined for them and can be found in the entity assistant. Not all entities are available in all dtd's. Depending upon the font that you are using, the character might appear as a box or even be a non-printing character (not visible in the document). If you can not determine what the problem is, you may have to remove the line of text and type it in again.

Tutorial for HTML 2.0

Part 4

⊞
⊞

The next message states "LI start-tag implied by H2 start-tag; not minimizable". This message tells you two things.
1/ that the LI start tag is not minimizable. In other words that you must indicate the start of a list item by using the start tag <LI>.
2/ that the H2 tag is not allowed in this location. Since the parser was expecting an LI here, it tells you that a list item tag is all that is allowed after a <UL>

Move the H2 tag and its content above the unordered list and restart the validation.

Your sample document should now look like this:

```
<HTML>
<HEAD><TITLE>Webber's Validation Tutorial Example
</TITLE></HEAD>
<BODY><H1><IMG SRC="image.gif"> Webber's Validation Tutorial
Example</H1>
<HR>
<P>The goal of this tutorial is to show you how to use Webber's
validation feature and also to point out the most common mistakes when
creating HTML. (<A HREF="#$1">Jump to item one</A>)</P>
<H2>This is a three item list</H2>
<UL>
<LI><H3><A HREF="test.htm" NAME="$1">Item one in the
list</A></H3></LI>
<HR><LI>Item two in the list</LI>
<OL><LI>Item one in the second list</LI></OL>
</UL>
<HR><ADDRESS><!--Webber_Auto_StampDNR-->Page maintained by <A
HREF="mailto:webmaster@yourcompany.com">webmaster@yourcompany.com</A>.
Copyright &copy; 1995 <A HREF="http://www.yourcompany.com/">Your
Company</A>.</ADDRESS></BODY></HTML>
```

⊞ Next

Tutorial for HTML 2.0

Part 5

⊞
⊞

The next message states "UL end-tag implied by H3 start-tag; not minimizable". This message tells you two things.
1/ that the UL end tag is not minimizable. In other words that you must indicate the end of an unordered list by using the end tag </UL>.
2/ that validator thinks that the <UL> is ended because it has encountered the <H3> tag. This tells you that the H3 tag is not allowed in this location.

Headings are not allowed within a list item. To find out what tags are allowed within a list item, place your cursor after the LI start tag, select HTML|Tag... from the menu. Click the 'Tags in Context' check box in the lower left corner, and click on the drop-down list. All tags allowed within a list item in HTML 2 recommended will be listed here. As you click on a tag name you will see the attributes available for each tag with this dtd. The Tags in Context feature will only work if the cursor is precisely positioned. In this case it must be immediately after the LI start tag and before any text.

Remove the start and end tags for the heading and restart the validation.

⊞ Next

Tutorial for HTML 2.0

Part 6

⊞
⊞

The next message highlights the line containing the list item with the anchor. The error message states that "NAME = "$1" attribute defaulted; wrong token type". Name and ID attribute values must be a sequence of alphanumeric characters for a maximum of 72 characters. Replace the $ with the letter A in front of the 1 to make this NAME attribute value valid. Since this is the target for the anchor in the first paragraph, that anchor   needs to read "#A1" in order to work.

⊞ Next

Tutorial for HTML 2.0

Part 7

⊞
⊞

The next error will highlight the line containing the HR tag within the list. The message tells you that the HR start tag implies a UL end tag. This means that the HR is not allowed in this position. If you were to check the 'Tags in Context' feature in the Tag Assistant (HTML|Tags... from the menu), you would find that only the LI tag is allowed inside an unordered (or ordered) list. Remove the HR and restart the validation.

⊞ Next

Your sample document should now appear like this:

```
<HTML>
<HEAD><TITLE>Webber's Validation Tutorial Example
</TITLE></HEAD>
<BODY><H1><IMG SRC="image.gif"> Webber's Validation Tutorial
Example</H1>
<HR>
<P>The goal of this tutorial is to show you how to use Webber's
validation feature and also to point out the most common mistakes when
creating HTML. (<A HREF="#A1">Jump to item one</A>)</P>
<H2>This is a three item list</H2>
<UL>
<LI><A HREF="test.htm" NAME="A1">Item one in the list</A></LI>
<LI>Item two in the list</LI>
<OL><LI>Item one in the second list</LI></OL>
</UL>
<HR><ADDRESS><!--Webber_Auto_StampDNR-->Page maintained by <A
HREF="mailto:webmaster@yourcompany.com">webmaster@yourcompany.com</
A>.Copyright &copy; 1995 <A HREF="http://www.yourcompany.com/">Your
Company</A>.</ADDRESS></BODY></HTML>
```

Tutorial for HTML 2.0

Part 8

⊞
⊞

The next error will highlight the line containing the OL tag within the list. The message tells you that the OL start tag implies a UL end tag. This means that the OL is not allowed in this position. Lists can be embedded within other lists but must be contained inside a list item.

```
<LI>Item two in the list</LI>
<OL><LI>Item one in the second list</LI></OL>
</UL>
```

Notice that the <OL> above is not inside a list item. Type a <LI> before the <OL> and a </LI> after the </OL>. Therefore this section should look like this:

```
<UL>
<LI><A HREF="test.htm" NAME="A1">Item one in the list</A></LI>
<LI>Item two in the list
<OL><LI>Item one in the second list</LI></OL></LI>
</UL>
```

Note that the ordered list is now inside the second list item (marked in red above). After making this change, restart the validation.

⊞ Next

Tutorial for HTML 2.0

Part 9

⊞
⊞

The last error in this document will highlight the line containing the copyright notice in the address. The message states "No declaration for entity "copy"; reference ignored". This message tells you that the entity for the copyright symbol (&copy;) is not defined in the current dtd. You can open the entity assistant (HTML|Entity) to see what entities are available in HTML 2. If you were validating against either Netscape dtd, Explorer dtd, or HTML 3, you would find that you do not get this error message since the copyright entity is defined for them. You can remove the copyright entity and replace it with its ascii equivalent (c). Restart the validation. You should now get a beep and a message in your status bar telling you that the document is now valid.

Congratulations, you have validated the document against the Recommended HTML 2 dtd. For illustration on how the dtds can differ, you can follow the next couple of lessons as we validate the valid HTML 2 page against the Recommended HTML 3 dtd. If you would like more detail on validating against the HTML 3 dtd, try the HTML 3 tutorial, where we will go through the sample document from the start.

⊞ Next

DTD Differences

HTML 2.0 Tutorial Part 10

⊞
⊞

For illustration on how the dtds can differ, we will now use the valid HTML 2 page and validate it against the Recommended HTML 3 dtd. Open your preferences by selecting Options| Preferences from the menu. Select HTML 3 and Recommended as the dtd settings. Click Ok. Start the validation by right clicking and selecting Validate Document from the pop-up menu.

Your sample document should now appear like this:

```
<HTML>
<HEAD><TITLE>Webber's Validation Tutorial Example
</TITLE></HEAD>
<BODY><H1><IMG SRC="image.gif"> Webber's Validation Tutorial
Example</H1>
<HR>
<P>The goal of this tutorial is to show you how to use Webber's
validation feature and also to point out the most common mistakes when
creating HTML. (<A HREF="#A1">Jump to item one</A>)</P>
<H2>This is a three item list</H2>
<UL>
<LI><A HREF="test.htm" NAME="A1">Item one in the list</A></LI>
<LI>Item two in the list
<OL><LI>Item one in the second list</LI></OL></LI>
</UL>
<HR><ADDRESS><!--Webber_Auto_StampDNR-->Page maintained by <A
HREF="mailto:webmaster@yourcompany.com">webmaster@yourcompany.com</A>.
Copyright (c)1995 <A HREF="http://www.yourcompany.com/">Your
Company</A>.</ADDRESS></BODY></HTML>
```

DTD Differences

HTML 2.0 Tutorial Part 11

⊞
⊞

The first message that we encounter says "NAME = "A1" attribute ignored: not defined for this element". This is because the NAME attribute has been deprecated and replaced with the ID attribute in the recommended version of the HTML 3 dtd. To make it valid for HTML 3 we could change the word NAME to ID so that the line reads `<A HREF="test.htm" ID="A1">`. However if we do so the internal link that is defined by this tag will not work since most browsers do not yet understand the ID attribute. Since the message says that the attribute will be ignored, we will leave it as is. Click the next button to move on to the next error.

A good question to ask here might be 'Now why did the IETF replace NAME with ID'? The change was made because all tags have ID attributes in HTML 3, so now every tag can be the target of a jump reference. This is a good idea, but it causes real problems for legacy documents and authors because it is such a switch. Also the word NAME is misleading for an attribute of this type because names are usually human-readable and meaningful while jump targets merely have to be unique. This conflict between a clear improvement to the standard and the vast collection of existing pages on the Web illustrates the difficulties of developing new standards.

⊞ Next

DTD Differences

HTML 2.0 Tutorial Part 12

⊞
⊞

The next message that we encounter says "UL end-tag implied by A start-tag; not minimizable". HTML 3 requires that within a list item, content (including text, text formatting tags, images and anchors) be contained inside a block element such as a paragraph. Place a paragraph tags around the contents of each list item so that the list looks like this:

```
<UL>
<LI><P><A HREF="test.htm" NAME="A1">Item one in the list</A></P></LI>
<LI><P>Item two in the list</P>
<OL><LI><P>Item one in the second list</P></LI></OL></LI>
</UL>
```

Restart the validation. You will now find that the only error it finds is the 'NAME' attribute which we ignored earlier.

Congratulations, you have now finished this section of the tutorial. Please see the HTML 3 validation tutorial for more details on this topic.

Tutorial for HTML 3.0

Part 1

⊞
⊞

The HTML 3 dtd is still in draft and there is no anticipated date of acceptance. Many of the elements from it (such as tables and alignment attributes on headings and paragraphs) are being supported by the newest browsers. In recommended mode it is very strict, particularly with where it will allow raw text. There has been some suggestions in the HTML Authoring newsgroups that this dtd should not be used in its recommended form at this time. This does make things complicated since some aspects of it are useful. We will use the HTML 3.0, Recommended dtd to illustrate what it entails in it's strictest form. Be sure that you have this selection made on the Status Line.

To begin the Validation:

When you have your sample document open and saved, right click and select 'Validate Document' from the pop-up menu.

The first message that we see says "Undefined CENTER start tag GI ignored; not used in dtd". This tells us that the CENTER tag is not used in the dtd that we are validating against so it will be ignored. You will notice that the line containing the center tag is highlighted to show you where the error has been found. If we click the 'Next' button we ill see a similar message for the end tag.

<CENTER> is a Netscape extension so we need to remove it for this to be a valid HTML 3 document. Take out the start and end center tags and restart the validation. We can achieve the same effect by adding an alignment attribute to the heading. Add the align="center" attribute so that your H1 tag looks like this: `<H1 ALIGN="Center"><IMG SRC="image.gif"> Webber's Validation Tutorial Example</H1>`
This will cause the heading to be centered in most browsers. Restart the validation when you have done this.

⊞ Next

Tutorial for HTML 3.0

Part 2

⊞
⊞ (If you have done the HTML 2 tutorial you may wish to skim through this part briefly as a review)

The next message states that "Out-of-context data ended HTML document element and parse". This tells you that you have used data (text, content) in your document in a place where it was not allowed. When the validator comes across an instance like this it stops parsing (reading) the document because it has lost the context of what is allowed.

The recommended HTML 3 dtd states that all text has to be contained inside a block element such as a paragraph or heading. Text formatting tags such as bold and emphasis do not count as a block element and are not allowed here either.

If you change the setting from recommended to deprecated you will find that the validator does not display this error. Placing text in the body of a document not within a block element was a feature of previous versions of HTML and has been made obsolete in HTML 2 and higher.

Place a paragraph tag <P></P> around the text beginning with `"The goal of this tutorial..."` to `"Jump to item one</A>)"` and then restart the validation.

⊞ Next

Tutorial for HTML 3.0

Part 3

⊞

⊞ (If you have done the HTML 2 tutorial you may wish to skim through this part briefly as a review)

The validator will now highlight the line containing the anchor `<A HREF=#$1>Jump to item one</A>` with a message that says "Incorrect character in mark-up; markup terminated". This message will occur in all dtd's and tells you that the document contains illegal mark-up. In this case it is caused by missing quotation marks around the value of the HREF. Place double quotes before and after the #$1 and restart the validation.

⊞ Next

More on illegal and incorrect characters

Tutorial for HTML 3.0

Part 4

⊞
⊞

The next message states "LI start-tag implied by H2 start-tag; not minimizable". This message tells you two things.
a/ that the LI start tag is not minimizable. In other words that you must indicate the start of a list item by using the start tag <LI>.
b/ that the H2 tag is not allowed in this location.
To find out what tags are allowed here, place your cursor after the <UL> tag and open the Tag Assistant (HTML|Tag...). Check the 'Tags in Context' button. Click on the drop-down list to see what tags area available given this context. You will find that only a list header (LH) and a list item (LI) are allowed inside a UL (or OL). Change the H2 tag to a LH so that the line looks like this: `<UL><LH>This is a three item list</LH>`. Restart the validation.

⊞ Next

Tutorial for HTML 3.0

Part 5

⊞

⊞ (If you have done the HTML 2 tutorial you may wish to skim through this part briefly as a review)

The next message states "UL end-tag implied by H3 start-tag; not minimizable". This message tells you two things.

a/ that the UL end tag is not minimizable. In other words that you must indicate the end of an unordered list by using the end tag </UL>.

b/ that validator thinks that the <UL> is ended because it has encountered the <H3> tag. This tells you that the H3 tag is not allowed in this location.

Headings are not allowed within a list item. To find out what tags are allowed within a list item, select HTML|Tag... from the menu. Click the 'Tags in Context' check box in the lower left corner, and click on the drop-down list. All tags allowed within a list item in HTML 3 recommended will be listed here. As you click on a tag name you will see the attributes available for each tag with this dtd.

Remove the start and end tags for the heading and restart the validation.

⊞ Next

Your document should now look like this:

```
<HTML>
<HEAD><TITLE>Webber's Validation Tutorial Example
</TITLE></HEAD>
<BODY><H1 ALIGN="Center"><IMG SRC="image.gif"> Webber's Validation
Tutorial Example</H1>
<HR>
<P>The goal of this tutorial is to show you how to use Webber's
validation feature and also to point out the most common mistakes when
creating HTML. (<A HREF="#$1">Jump to item one</A>)</P>
<UL><LH>This is a three item list</LH>
<LI><A HREF="test.htm" NAME="$1">Item one in the list</A></LI>
<HR><LI>Item two in the list</LI>
<OL><LI>Item one in the second list</LI></OL>
</UL>
<HR><ADDRESS><!--Webber_Auto_StampDNR-->Page maintained by <A
HREF="mailto:webmaster@yourcompany.com">webmaster@yourcompany.com</A>.
Copyright &copy; 1995 <A HREF="http://www.yourcompany.com/">Your
Company</A>.</ADDRESS></BODY></HTML>
```

Tutorial for HTML 3.0

Part 6

⊞

⊞ (If you have done the HTML 2 tutorial you may wish to skim through this part briefly as a review)

The next message highlights the line containing the list item with the anchor. The error message states that "NAME = "$1" attribute defaulted; wrong token type". Name and ID attribute values must be a sequence of alphanumeric characters for a maximum of 72 characters. Replace the $ with the letter A in front of the 1 to make this NAME attribute value valid. Since this is the target for the anchor in the first paragraph, that anchor needs to read "#A1" in order to work. This will validate under the deprecated setting of the dtd.

⊞ Next

Tutorial for HTML 3.0

Part 7

⊞

⊞

The next message that we encounter says "NAME = "A1" attribute ignored: not defined for this element". This is because the NAME attribute has been deprecated and replaced with the ID attribute in the recommended version of the HTML 3 dtd. To make it valid for HTML 3 we could change the word NAME to ID so that the line reads `<A HREF="test.htm" ID="A1">`. However if we do so the internal link that is defined by this tag will not work since most browsers do not yet understand the ID attribute. Since the message says that the attribute will be ignored, we will leave it as is. Click the next button to move on to the next error.

A good question to ask here might be 'Now why did the IETF replace NAME with ID'? The change was made because all tags have ID attributes in HTML 3, so now every tag can be the target of a jump reference. This is a good idea, but it causes real problems for legacy documents and authors because it is such a switch. Also the word NAME is misleading for an attribute of this type because names are usually human-readable and meaningful while jump targets merely have to be unique. This conflict between a clear improvement to the standard and the vast collection of existing pages on the Web illustrates the difficulties of developing new standards.

⊞ Next

Tutorial for HTML 3.0

Part 8

The next message that we encounter says "UL end-tag implied by A start-tag; not minimizable". HTML 3 requires that within a list item, content (including text, text formatting tags, images and anchors) be contained inside a block element such as a paragraph. Place a paragraph tags around the contents of each list item so that the list looks like this:

```
<UL><LH>This is a three item list</LH>
<LI><P><A HREF="test.htm" NAME="A1">Item one in the list</A></P></LI>
<LI><P>Item two in the list</P>
<OL><LI><P>Item one in the second list</P></LI></OL></LI>
</UL>
```

There are some browsers which do not properly display this HTML and which will place the bullet on one line and the list item below it something like this:

-

  Item one in the list
-

  Item two in the list
  -

    Item one in the second list

According to discussions in the newsgroups that we have seen, this occurs with text-based browsers such as lynx. These discussions seem to suggest that you should not strictly adhere to HTML 3 Recommended in this regard. HTML 3 Recommended also requires a block element around text in table cells (i.e. `<TD><P>Content of cell</P></TD>`) This can cause browsers to add a lot of extra space around the contents which may not be desirable.

Restart the validation.

Next

Tutorial for HTML 3.0

Part 9

⊞

Except for the line

```
<LI><P><A HREF="test.htm" NAME="A1">Item one in the list</A></P></LI>
```

where HTML 3 Recommended will complain about the use of the NAME attribute, your document is now valid. If you open your preferences and switch the dtd to 'Deprecated' you will find that the document validates.

Since the HTML 3 dtd is still being developed and undergoing many changes, it may not be wise to adhere to its strictest rules. Certainly you should continue to use the NAME attribute for internal anchors until such time that browsers support this internal linking through the ID attribute.

Netscape and Microsoft Extensions

⊞

Netscape Communications and Microsoft Corp. have developed several extensions to the HTML language for use with their browsers. It is not the intention of this tutorial to spawn a debate over whether these extensions should be used or not.

We have not included a tutorial for the extensions as we felt that it would be fairly repetitive after the HTML 2 and HTML 3 tutorials. The types of errors would be similar. Test your new found skills at validation by running the sample document against either the Netscape Extensions or the Microsoft Extensions.

The extensions include new elements and new attributes used with existing elements. If you are not sure which tags and attributes are extensions, see the Webber Help topics on Netscape Extensions and Internet Explorer Extensions.

The dtds for Netscape and Explorer Extensions were developed by CSD Corp. based on the documentation available on the Netscape Web site and the Microsoft Web site. We have added their elements and attributes to the HTML 2 dtd. In some cases the documentation available was not clear as to how the extension was intended to fit within HTML. In these cases we have applied our knowledge of SGML to come up with a dtd that is consistent with common usage and good practice of SGML. We can not guarantee that Netscape or Microsoft would agree with our interpretation.

Common Errors

There are a few very common errors that are worth summarizing here. For details on these and other common errors, try the validation tutorials.

1 Using elements (tags or entities) that are not defined in the currently selected dtd.
For example, using the FONT tag, or the BGCOLOR attribute on the body tag and validating the document with either the HTML 2 or HTML 3 dtd. Neither of these elements are defined except as Netscape extensions. Alignment attributes (LEFT, RIGHT) are defined in the Netscape Extensions, Explorer Extensions and HTML 3, but not in HTML 2.

2 Using text in your document in places that it is 'out-of-context', or not allowed.
All of the Recommended dtds require text to be enclosed within a block element such as a paragraph or a heading. It can't be placed inside the body except inside a block tag. Images, anchors and text formatting tags are also required to be enclosed in block elements. The HTML 3, Recommended dtd has further complicated this by requiring that text be enclosed in a block element when it is in a list or table cell. (i.e. `<LI><P>Here is the text</P></LI>` rather than `<LI>Here is the text</LI>`

3 Using tags in your document in places where they are out-of-context.
As discussed in HTML as a Structured Language, tags are context sensitive and are allowed only in certain places in a document. By placing your cursor after a start tag and using the 'Tags in Context' feature of the Tag Assistant, you will be able to determine what tag is allowed in the position that the cursor is. This is a great way to find out what is wrong when you get a message telling you that a `tagname` has implied the end of another tag.

dtd
Document Type Definition - a set of SGML definitions (rules) that define the structure of a document of a particular type. HTML documents use a dtd (i.e. HTML 2.0 or HTML 3.0) which defines the structure that all HTML documents must follow.