Creating Pages with HomeSite and review of HTML 4.0

Structure of a web document

Create a Basic HTML Page

- Doctype Statement
- Required Tags
- Comment Tag

Stand Text Formatting

- Headings
- Paragraphs
- Breaking Lines of Text
- Horizontal Rules
- Emphasizing Text
  - EM
  - Strong
- Lists
  - Ordered
  - Unordered
- Hyperlinks
  - Relative and Absolute
  - Text Inks
- Text Alignment
  - DIV
- Blockquote
- Pre

IMAGES

- Width
- Height
- SRC
- Alt

Tables

- Create Tables
- Border
- Table and Cell Attributes
- Column and Row Span
- Table, Row and Cell Alignment Options

HTML Basics

HTML documents are made up of text, media content such as graphics, audio, and video, and HTML tags that determine how a Web browser displays the content. If a text document has no “markup”–no HTML codes–a Web browser won’t know where to begin a paragraph, where to break a line, and where to place headlines. The browser will display the words in a jumbled block of text. Today we will learn how to insert these structural codes into text documents for browsers to interpret.

Tags and content

HTML tags are enclosed in brackets made of the less-than and greater-than signs on your keyboard, (< >). Most HTML tags consist of a start tag, <TAG NAME>, and an end tag, </TAG NAME>. The end tag is identified by a forward slash (/) before the tag name. Some tags, such as the Break <BR> and Horizontal Rule <HR> tags, do not require an end tag.

Tag attributes and their values are entered in the start tag, <TAG NAME ATTRIBUTE="value">. Attributes are separated from one another by a single space. There should be no spaces after the left bracket (<) or before the right bracket (>).

Tag and attribute names are not case-sensitive, so that <HTML> and <html> are interchangeable. Some attributes values, particularly file names, are case-sensitive, however.

Many tags can be nested, that is, placed inside other tags, but tags should never overlap:

```
Correct: <P><FONT SIZE="2">text</FONT></P>
Incorrect: <P><FONT SIZE="2">text</P></FONT>
```

The actual page content, text, links, images, etc., is placed between the start and end tags.

Let’s Start

The fastest way to get up and running is to create a basic Web page template.

To begin, we must create a directory to store all of our web files.

1. Click on the Start button, go to Programs, and select Windows Explorer and navigate to the desktop. We will using this to store your web pages. Go the File menu, select New, select Folder, and create a new directory for your Web Documents.

2. Go to the Start button, then select Programs and then Homesite 4.5. Go the File menu and select New to open a new document.

Now we are ready to create our basic Web page template.

Building the Document Structure

We will be using HTML 4.0 as our standard HTML language. An HTML 4.0 document consists of elements nested inside other elements. Each elements starts with a start-tag <...> and end with an end-tag</...>. Most tags are place in symmetrical “on-off” pairs. These are called **container tags** because they contain text that should all be structured a particular way. Tags are not case-sensitive, but as a best practice for coding, it is useful to type them in all capital letters to differentiate them from page content.

At the highest level, a document consists of one element, call HTML. Inside that are two elements, HEAD and BODY.

The HTML start tag (<HTML>) marks the beginning of an HTML document, and the HTML end tag (</HTML>) marks the end.

The two primary sections of an HTML document are the HEAD an BODY. They are held together by the HTML element, which identifies the document to browsers as containing HTML code.

A legal HTML 4.0 document also includes the HTML version and language information at the very top of the document.

Homesite creates a basic html document shell.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>


<HEAD>

<TITLE>     <TITLE>


</HEAD>

<BODY>


</BODY>


</HTML>
```

<TITLE>...</TITLE>
Every HTML document must have a Title element. The title should identify the contents of the document and in a global context, and may be used in history lists, bookmarks and as a label for the windows displaying the document. Titles are not typically rendered in the text of a document itself. Normally, browsers will render the text contained within the <TITLE> ... </TITLE> elements in the title bar of the browser window.

The Title element must occur within the head of the document and may not contain anchors, paragraph elements, or highlighting. Only one title is allowed in a document.

NOTE : The length of a title is not limited, however, long titles may be truncated in some applications. To minimize the possibility, titles should be kept as succinct as possible. Also keep in mind that a short title, such as 'Introduction' may be meaningless out of context. An example of a meaningful title might be 'Introduction to HTML elements'

W3C WAI-A WCAG 1.0

1.1.4 The !DOCTYPE statement
Checkpoints in this section:
3.2 Create documents that validate to published formal grammars. [Priority 2]

Validating to a published formal grammar and declaring that validation at the beginning of a document lets the user know that the structure of the document is sound. It also lets the user agent know where to look for semantics if it needs to. The W3C Validation Service validates documents against a whole list of published grammars.

NOTE *This empty, formatted document is referred to as an HTML “shell” and can be used to develop your custom templates.*

Requirements of an HTML document

We have just completed a basic html shell and has the elements required for a valid HTML document.

The default document template contains these elements:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>

<HEAD>

<TITLE>    <TITLE>

</HEAD>

<BODY>


</BODY>

</HTML>
```

These tags must be included in every Web page. The following table provides a brief description of each of these required elements.

Tag	Description
DOCTYPE	Declares the version of HTML that you are using.
HTML	Marks the beginning and end of the HTML code.
HEAD	Contains data about the document, including the title, metadata, scripts, and a style block or style sheet link.
TITLE	Contains the title of the document, which is displayed in the title bar of the browser and is important for indexing documents.
BODY	Contains everything that will be rendered by the browser.

Using Comment Tags to Denote Sections

Most of your document will consist of elements. However, you can also insert HTML comments into the document. A comment is anything you want to say about what is going on with your document a nd to provide helpful information to anyone using your code.

Comments begin and end with less-than and great-than symbols, but contain an exclamation point and dashes, as follows:
<!-- your comment here -->

Your comment will go in between the dashes. You can put as much information as you like within the opening and closing tags. You can even add images and, as long as they are within the comment tags, they will not appear on the page.

Comment tags do not follow a specific placement rule. You can use them wherever it's logical or necessary to do so.

1. Open your HTML shell (template.htm).

2. Above the opening <HEAD> tag, add a comment designating when the document was last updated.

3. Now add a logical comment about the <BODY> tag

4. Save your file.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>
<!-- Updated: 00/00/00 -->

<HEAD>

<TITLE>     <TITLE>


</HEAD>


<!-- Begin Body Section -->
<BODY>


</BODY>

</HTML>
```

NOTE *Because comment tags hide all the information that appears inside them, you also should use comment tags to hide scripts and style sheet information (usually found in the HEAD section of a document) from older non-compliant browsers.*

Standard Text Formatting

HTML began as a document formatting language. It was never meant to be the language of design. Rather, its concern was to enable document creators to format basic headers, paragraphs, line breaks and list of information.

This unit takes a look at the most basic elements of text formatting. As an HTML coder seeking the most control over your documents, it's very important to know how to manage text before embarking on HTML-based style and design.

Text formatting can be divided into two types of groups:

1. **Block-level**
   Block formatting elements are used for the formatting of whole blocks of text within a HTML document, rather than single characters.

   A block-level element is an element that has a line break before and after its content. Examples of block-level elements are Headers (H1) and paragraphs <P>

2. **Inline**
   Inline elements are use for formatting single characters within a block-level element. An inline element is an element that does not begin and end a line. Examples of inline elements are STRONG and EM.

Creating Section Headers

HTML has six levels of headings, numbered 1 through 6, with 1 being the most prominent. Headings are displayed in larger and/or bolder fonts than normal body text. The first heading in each document should be tagged <H1>.

The syntax of the heading element is: <H*x*>Text of heading </H*x*> where *x* is a number between 1 and 6 specifying the level of the heading. Do not skip levels of headings in your document. For example, don't start with a level-one heading (<H1>) and then next use a level-three (<H3>) heading.

<H*1-6*> are elements that convey structure rather than presentation.

1. To add a section header, open the template file, and then save as homepage.htm.

2. In the new htm file, withing the TITLE tag, type in the title of the file.

3. Within the BODY section of the HTML file, add the text you want to use for your level 1 header.

4. Place your cursor at the beginning of the title and style as <H1>.

5. Save the file and view in the browse mode.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>
<!-- Updated: 00/00/00 -->

<HEAD>

<TITLE>U.S. Embassy NetDiplomacy</TITLE>


</HEAD>


<!-- Begin Body Section -->

<BODY>
<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<H2>The Embassy</H2>
<H2>Information and Culture</H2>

<H3>Calendar of Events</H3>


</BODY>


</HTML>
```

W3C WAI-A WCAG 1.0

1.2.1 Section headings
Checkpoints in this section:
3.5 Use header elements to convey document structure and use them according to specification. [Priority 2]

Sections should be introduced with the HTML heading elements (H1-H6).

Other markup may complement these elements to improve presentation, but visual presentation is not sufficient to identify document sections.

Since some users skim through a document by navigating its headings, it is important to use them appropriately to convey document structure. Users should order heading elements properly. For example, in HTML, H2 elements should follow H1 elements, H3 elements should follow H2 elements, etc.

Content developers should not "skip" levels (e.g., H1 directly to H3). Do not use headings to create font effects; use style sheets to change font styles for example.

NOTE *HTM headers run from level 1, being the largest to level 6, being the smallest. It's very important to remember that level 1 is larger than level 6. Remember to think of priority, and you're get the results you need.*

Designating Paragraphs

The Paragraph element indicates a paragraph of text.

Unlike most HTML tags, the paragraph tag is often used alone (no closing tag), but if you intend to use CSS you must put in a closing tag </P> at the end of your text.

Because consistency it is important to choose the method of using <P> tag and stick to that method. The recommendation is to use the closing </P> tag.

1. In your open document, type your first paragraph of text underneath the text NetDiplomacy, FSI.

2. Select the paragraph of text and insert the <P> tag.

3. Save your file and load it into your browser

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>
<!-- Updated: 00/00/00 -->

<HEAD>

<TITLE>U.S. Embassy NetDiplomacy</TITLE>


</HEAD>


<!-- Begin Body Section -->

<BODY>
<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<P>Welcome to the Embassy of the United States at
NetDiplomacy, FSI.</P>

<H2>The Embassy</H2>

<H2>Information and Culture</H2>

<H3>Calendar of Events</H3>


</BODY>

</HTML>
```

NOTE *Just remember that one paragraph tag equals two breaks <BR>. A big "however" here–you can't use more than one paragraph tag in a row. The browser will ignore everything but the first paragraph tag.*

If you are using the opening and closing paragraph <P>...</P> tag, keep them on the save line as the text. Some browsers will add extra lines if you don't follow this rule carefully. With the single paragraph tag method, you can put it on a separate line without causing browsers to misinterpret the command.

Breaking Lines of Text

So far all HTML elements had content. HTML also has some elements that do not have content; they are called empty elements. One example is the BR element who sole purpose is to force a line break. Since empty elements do not have any content they don't need any end tags either.

The Line Break element specifies that a new line must be started at the given point. The amount of line space used is dependent on the particular browser, but is generally the same as it would use when wrapping a paragraph of text over multiple lines.

Unlike HTML paragraphs, HTML line breaks do not add an extra line below the return.

It is especially handy for creating white space or anything where you need to control exactly where a line breaks

Let's create some white space between our headings and the page footer.

1. Add the comment tag for the footer section and then add the text with your paragraphs tags surround the text.

2. Now add the <BR> tag at the beginning of the footer section - just above the comment tag.

3. Continue to add your desired amount of breaks

4. View the results in your browser.

Another example of an empty element is the HR element which inserts a horizontal rule in the document. Horizontal rules are used to break up sections of text.

1. Place your cursor after the footer paragraphs and type in the <HR> tag.

2. View your work in your browser.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0
Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>
<!-- Updated: 00/00/00 -->

<HEAD>

<TITLE>U.S. Embassy NetDiplomacy</TITLE>

</HEAD>

<!-- Begin Body Section -->

<BODY>
<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<P>Welcome to the Embassy of the United States at
NetDiplomacy, FSI.</P><H2>The Embassy</H2>

<H2>Information and Culture</H2>

<H3>Calendar of Events</H3>

<BR>
<BR>
<BR>

<!-- Begin Footer Section -->

<P>This site is produced and maintained by the Public Affairs
Office of the Embassy of the United State in NetDiplomacy,
FSI. Links to other Internet sites should not be construed as
an endorsement of the views contained therein.</P>

<HR>


</BODY>
</HTML>
```

NOTE *You can stack as many line breaks as you like, You can either combine a single paragraph tag with any number of break tags to get your results, or simple stack break tags.*

Emphasizing Text

Text emphasis tags are considered inline elements, as they emphase a word relative to the surrounding text. Several HTML elements can express this.

Bold <B>, italic <I>, and underline <U> are the most common, but they are part of the HTML 3.2 standard and are considered presentation elements. In strict HTML 4.0, text emphasis should be controlled with style sheets. If you choose to use these tags, be sure that your HTML 4.0 document is identified as being transitional.

The best practice for proper HTML text emphasis is to use the structural tags <EM> and <STRONG>. STRONG elements are usually displayed in **bold**, and the EM elements are displayed in *italic.*

1. Place your cursor at the beginning of the words or words you want to appear in bold using the STRONG tag.

2. Now put your cursor in front of the next group of words and format as STRONG and then the EM tag.

3. Save and then view the file in the browser.

```
...

<BODY>
<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<P>Welcome to the <STRONG>Embassy of the United
States</STRONG> at <STRONG><EM>NetDiplomacy,
FSI.</EM></STRONG></P>

<H2>The Embassy</H2>

<H2>Information and Culture</H2>

<H3>Calendar of Events</H3>

...
```

W3C WAI-A WCAG 1.0

Checkpoints in this section:
3.3 Use style sheets to control layout and presentation. [Priority 2]

The proper HTML elements should be used to mark up emphasis: EM and STRONG. The B and I elements should not be used; they are used to create a visual presentation effect. The EM and STRONG elements were designed to indicate structural emphasis that may be rendered in a variety of ways (font style changes, speech inflection changes, etc.)

NOTE *Emphasis should be applied with a light touch. If there are too many incidents of emphasized text on a page, you can dilute the importance of the text.*

Lists

Lists are very common in HTML documents. HTML has three elements that create lists and they are Block-level elements.

OL, which creates an ordered list. Each list item has a label that indicates the order (1, 2, 3 or I, II, II). Referred to as numbered lists.

UL, which creates an unordered list. Each list item has a mark (bullet) that does not indicate order. Referred to as bulleted lists.

DL, which creates a definition list. A list of terms with their corresponding definitions. An example of a DL is a dictionary or glossary

To make an unordered list, you work with two different tags. The UL element tells the document that each individual list item (denoted with the LI tag) get a bulleted beside it.

1. Homesite offers a number of ways to format lists. Let explore.

Notice that you do not have to change the LI elements to change the list from unordered to ordered: both the UL and OI use LI as the list item element. But since the LI elements are now inside the UL element, they will look different.

```
...
<BODY>

<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<P>Welcome to the <STRONG>Embassy of the United
States</STRONG> at <STRONG><EM>NetDiplomacy,
FSI.</EM></STRONG></P>

<H2>The Embassy</H2>

<H2>Information and Culture</H2>

<UL>
<LI>Embassy Services and Offices
<LI>Contact Information
</UL>
```

W3C WAI-A WCAG 1.0 Checkpoints in this section:
3.6 Mark up lists and list items properly. [Priority 2]

The HTML list elements DL, UL, and OL should only be used to create lists, not for formatting effects such as indentation.

Avoid using images as bullets in definition lists created with DL, DT, and DD. However, if this method is used, be sure to provide a text equivalent for the images.

To change the "bullet" style of unordered list items created with the LI element, use style sheets. In CSS, it is possible to specify a fallback bullet style (e.g., 'disc') if a bullet image cannot be loaded.

NOTE *You can nest lists (place one element with another) to achieve an outline effect.*

Adding Hyperlinks

Hyperlinks are the essence of Web text, also referred to as hypertext.

To make a hyperlink, you use the A (anchor) element. When the user clicks on the a element, the browser retrieves the document at the other end of the hyperlink. The browser needs to be told where it can find the other document, and this information goes into an attribute on the A element. An Attribute is a characteristic quality of the element, other than the type or content of an element. The a element uses an attribute called HREF (hypertext reference) to add a hyperlink.

1. In your open text window, go back to the list that we just created and select the text you wish to use as your link.

2. Let's explore Homesite's options for adding Hyperlinks.

```
<UL>
<LI><A HREF="embassy/services.htm" title="Embassy Services and
Offices">Embassy Services and Offices</A>
<LI><A HREF="embassy/contact.htm" title="embassy contact
information">Contact Information</A>
</UL>
```

The A start tag is a bit more complicated than the other start tags we have seen so far; in addition to the element name it includes an attribute. Most attributes need a value: the HREF attributes always takes a URL as a value. A URL, Universal Resource Locator, is a web address that the browser uses to locate the hyperlinked document. When URL are used as values on the HREF attribute, they should always be quoted ("...").

There are two primary methods for creating links.

- **A *relative* URL gives the location of the document relative to the document where it is referenced (like our example). These links are used when you are linking within the local server, from document to document or directory to directory.**

- **An *absolute URL* gives the location of the document independent of any other document. You must use absolute URLs when you link to a document on a different server. (A HREF="http://usinfo.state.gov/"</A>**

W3C WAI-A WCAG 1.0

6.1 Link text
Checkpoints in this section:
13.1 Clearly identify the target of each link. [Priority 2]

Good link text should not be overly general; don't use "click here." Not only is this phrase device-dependent (it implies a pointing device) it says nothing about what is to be found if the link if followed. Instead of "click here", link text should indicate the nature of the link target, as in "more information about sea lions" or "text-only version of this page". Note that for the latter case (and other format- or language-specific documents), content developers are encouraged to use content negotiation instead, so that users who prefer text versions will have them served automatically

NOTE *To use links successfully; Keep links descriptions short and sweet; Link descriptions should be descriptive, they should make sense to the user accessing the link; Avoid overused and unclear terminology such as "click here" or "click this."*

Aligning Sections of Text

Text alignment enables you to gain control over the way text falls on a page.

The <DIV> element is a block element that is basically used as a 'container' to contain other HTML elements and display them discretely from the rest of the document by adding paragraph breaks before and after the contents of the <DIV> element.

To align text, you use the ALIGN attribute combined with a value: center, left, or right.

```
...
<BODY>

<DIV align="center">

<H1>Embassy of the United States</H1>

<H1>NetDiplomacy, FSI</H1>

<P>Welcome to the <STRONG>Embassy of the United
States</STRONG> at <STRONG><EM>NetDiplomacy,
FSI.</EM></STRONG></P>

</DIV>

<H2>The Embassy</H2>

<H2>Information and Culture</H2>

<UL>
<LI>Embassy Services and Offices
<LI>Contact Information
</UL>
```

What Else

We have covered the most important text Block-level formatting tags, there are two more that should be mentioned. Since we are not using these elements in our sample web page, there will not to a walk through.

Below is a brief description.

BLOCKQUOTE.

The <BLOCKQUOTE>..</BLOCKQUOTE> tag set indents the left and right margins on most browsers. Under the Accessibility guides this tag should only be used for quotation markup not for formatting effect such as indentation.

PRE

The <PRE></PRE> Preformatted Text element presents blocks of text in fixed-width font, and so is suitable for text that has been formatted on screen, or formatted for a mono-spaced font.

NOTE *The DIV approach is more elegant and flexible than the CENTER tag, particularly under transitional HTML 4.0.*

Adding Images

Images are added to your documents with the IMG element - IMG is short for image.

IMG is a peculiar element. First, it's empty (quick reminder: an empty element is an element without content, e.g., HR, BR). Second, it's a replaced element. A replaced element is a placeholder for some other content that is being pointed to from the element. In the case if IMG, it points to an image that is retrieved by the browser when the IMG element is encountered. Unlike the A element, which give the user the option of jumping to a link or not, the browser will automatically retrieve the image IMG points to. IMG uses an attribute called SRC to point to the image.

The SRC attribute on IMG is similar to the HREF attribute on A: they both take a URL as a value.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN"
"http:///www.w3org/TR/REC-html40/loose.dtd">

<HTML>

<!-- Updated: 00/00/00 -->

<HEAD>

<TITLE>U.S. Embassy NetDiplomacy</TITLE>

</HEAD>

<!-- Begin Body Section-->

<BODY>


<IMG SRC="images/seal.gif" WIDTH="111" HEIGHT="112" ALT="U.S.
Department of State Seal">

<!-- Begin Head Section -->
```

W3C WAI-A WCAG 1.0

7.1 Short text equivalents for images ("alt-text")
Checkpoints in this section:

1.1 Provide a text equivalent for every non-text element (e.g., via "alt", "longdesc", or in element content). This includes: images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video. [Priority 1] .

When using IMG, specify a short text equivalent with the "alt" attribute. Note. The value of this attribute is referred to as "alt-text".

7.2 Long descriptions of images
Checkpoints in this section:

When a short text equivalent does not suffice to adequately convey the function or role of an image, provide additional information in a file designated by the "longdesc" attribute:
Example.

```
<IMG src="97sales.gif" alt="Sales for 1997"
  longdesc="sales97.htm">
```

NOTE *Twenty words is the maximum words to use in the Alternative text description. If you feel 20 words isn't enough, use the D link (longdesc).*

Tables

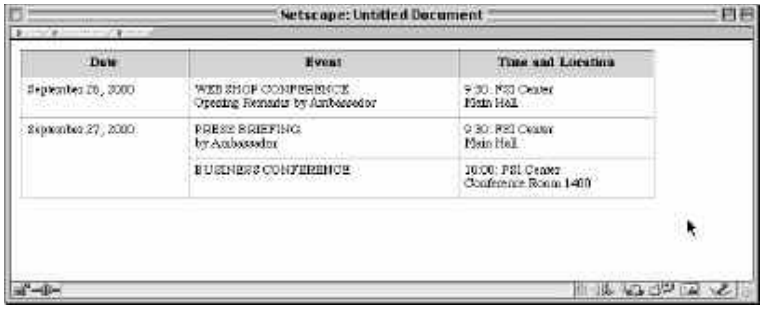
Tables are a way to visually show relations between pieces of data. Designers rely on the use of tables to create a grid upon which they create designs.

Tables <TABLE>...</TABLE> >are made up of rows <TR>...</TR>. Inside table rows are table cells, marked with table data <TD>...</TD>. That is the general idea, but in practice can be a bit more complex.

By default, the table has certain characteristics. its width is determined by the longest element in each cell. It has no visible borders. All content is left-justified and i the vertical middle of the cell.

To override these default values, each tag has attributes. Attributes are always written with the start tag and let you customized your table for your specific need.

```
<H2>Calendar of Events</H2>

<TABLE WIDTH="90%" BORDER="1" CELLSPACING="0" CELLPADDING="5"
ALIGN="left" summary="This table charts the calendar of
events at the webshop training class">









</TR>
</TABLE>
```



Creating a Dynamic Table

The width of the table can be controlled through a variety of ways. The width can be fixed (pixels) or it can be dynamic using percentage that allows their contents to flow into the available set percentage of the space of any browser, platform and screen resolution.

To build our table, we are going to use percentages to control the width of the table.

W3C WAI-A WCAG 1.0 5.1 Tables of data

Content developers may make HTML 4.01 data tables more accessible in a number of ways:

Providing summary information
*Identifying row and column information

NOTE *If you are using fixed tables, you must add the amount of cell padding and spacing values to the individual cell widths to achieve the fixed table width.*

Entering Text in Tables

Content for HTM tables is entered in the table cells using the table data<TD> and table header <TH> tags. Table text is formatted in the same way as paragraph text, by coding the <P> tag.

Before we can add the content to the <TD> tag, we must contain that the <TD> inside a row <TR>.

1. When you are finished, you will have constructed a table with 4 rows and 3 columns (TD cells).

You can cause table data or table headers to span rows or columns by using the ROWSPAN and COLSPAN attributes, which produce effects similar to merging cells in a spreadsheet or work processor table.

2. TD, TH, TR have the following attributes.

ALIGN=Left, Right, or Center. Aligns the contents of the data cell or row horizontally.

VALIGN=Top, Middle, or Bottom. Aligns the contents of the data cell or row vertically.

BGCOLOR=color. Specifies a background color for the data cell or row.

3. Save your file and preview in browser.

```
<H2>Calendar of Events</H2>

<TABLE WIDTH="90%" BORDER="1" CELLSPACING="0" CELLPADDING="5"
ALIGN="left" summary="This table charts the calendar of
events at the webshop training class">

<TR BGCOLOR="#CCCCCC">
<TH>Date</TH>

<TH>Event</TH>

<TH>Time and Location</TH>
</TR>

<TR ALIGN="left" VALIGN="top">
<TD>September 26, 2000</TD>

<TD><P>WEBSHOP CONFERENCE<BR>
Opening Remarks by Ambassador</P></TD>

<TD><P>9:30: FSI Center<BR>
Main Hall</P></TD>
</TR>

<TR ALIGN="left" VALIGN="top">
<TD ROWSPAN="2">September 27, 2000</TD>
<TD>PRESS BRIEFING<BR>
by Ambassador</TD>

<TD><P>9:30: FSI Center<BR>
Main Hall</P></TD>
</TR>

<TR ALIGN="left" VALIGN="top">
<TD>BUSINESS CONFERENCE</TD>
<TD><P>10:00: FSI Center<BR>
Conference Room 1400</P></TD>
</TR>
</TABLE>
```

NOTE *Table Header tags <TD> are identical to table data tags except that the text is bolded and centered horizontally in the data cell by default.*