COMP519
Web Programming

Autumn 2015

HTML and CSS

Hypertext and HTML

The Hypertext Markup Language (HTML) is the language for specifying the (typically static) content of Web pages (originally based on SGML, the Standard Generalized Markup Language).

Hypertext refers to the fact that Web pages are more than just text. It can, for example, contain multimedia and provide links for jumping within the same document and/or to other documents.

“Markup” refers to the fact that HTML works by augmenting text with special symbols (tags) that identify the document structure and content type.

Hypertext and HTML (cont.)

HTML is an evolving standard (as new technology/tools are added).

- HTML 1 (Berners-Lee, 1989): Very basic, limited integration of multimedia in 1993, Mosaic added many new features (e.g. integrated images).
- HTML 2.0 (IETF, 1994): Tried to standardize these & other features, but late in 1994–96, Netscape & IE added many new, divergent features.
- HTML 3.2 (W3C, 1996): Attempted to unify into a single standard but didn't address newer technologies like Java applets & streaming video.
- HTML 4.0 (W3C, 1997): Attempted to map out future directions for HTML, not just react to vendors.
- XHTML 1.0 (W3C, 2000): HTML 4.01 modified to conform to XML standards.

Hypertext and HTML (cont.)

- XHTML 1.1 (W3C, 2001): “Modularization” of XHTML 1.0.
- HTML5 (Web Hypertext Application Technology Working Group, W3C, 2006): New version of HTML4, XHTML 1.0, and DOM 4 (still a work in progress), no longer based on SGML, but “backward compatible” with parsing of older versions of HTML.

Hypertext and HTML (cont.)

- XHTML 1.1 (W3C, 2001): “Modularization” of XHTML 1.0.
- HTML5 (Web Hypertext Application Technology Working Group, W3C, 2006): New version of HTML4, XHTML 1.0, and DOM 4 (still a work in progress), no longer based on SGML, but “backward compatible” with parsing of older versions of HTML.

  HTML5 was released as a “stable W3C Recommendation” on October 28, 2014.

  Many features of HTML5 were developed with the idea of being run on low-powered devices like smartphones, and include new methods for handling multimedia and graphical context without APIs and plugins. Also, many features have been redefined or standardized, and defines how invalid documents should be treated by browsers that conform to the new HTML5 standards.

In this course. . .

My intention is that you design your webpages to the HTML 5 specification.

I will be enforcing this standard in your assessments (as much as possible).

More will be said when I discuss Document Type Declarations (soon).

Web Development Tools

Many high-level tools exist for creating Web pages.
e.g. Microsoft FrontPage, Netscape Composer, Adobe PageMill, Macromedia DreamWeaver, HotDog, . . .

Also, many applications have “save to HTML” options (e.g. Microsoft Word).

Dont use these tools!!

For some users who want to develop basic, static Web pages, these might work (but many of these programs produce very poorly structured HTML code).

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as “vim” on the Departmental Unix system)?

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as “vim” on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as “vim” on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as “vim” on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.
- May want to “borrow” components from other pages and integrate into existing pages. (**Note: May sure you acknowledge your use of code from other sites.**)

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as "vim" on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.
- May want to "borrow" components from other pages and integrate into existing pages. (**Note: May sure you acknowledge your use of code from other sites.**)
- May want dynamic features such as scripts or applets.

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as "vim" on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.
- May want to "borrow" components from other pages and integrate into existing pages. (**Note: May sure you acknowledge your use of code from other sites.**)
- May want dynamic features such as scripts or applets.
- We can avoid certain problems that can occur using Windows editors when we get to CGI programming (later).

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as "vim" on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.
- May want to "borrow" components from other pages and integrate into existing pages. (**Note: May sure you acknowledge your use of code from other sites.**)
- May want dynamic features such as scripts or applets.
- We can avoid certain problems that can occur using Windows editors when we get to CGI programming (later).
- Remote editing of web pages may only be possible using a basic text editor.

Web Development Tools (cont.)

Why do I encourage learning how to edit HTML using a basic text editor (such as "vim" on the Departmental Unix system)?

- May want low-level control of features not available with other webpage creation tools.
- May care about size and readability of pages.
- May want to "borrow" components from other pages and integrate into existing pages. (**Note: May sure you acknowledge your use of code from other sites.**)
- May want dynamic features such as scripts or applets.
- We can avoid certain problems that can occur using Windows editors when we get to CGI programming (later).
- Remote editing of web pages may only be possible using a basic text editor.
- Sticking to (internationally and industrially) agreed upon web standards will help ensure your web documents are rendered as you intend them to look, and operate as you desire.

Tags and elements

HTML specifies a set of tags that identify the structure of the document and the content type.

Tags are enclosed in angle brackets, i.e. < > (with text in between them).

Tags and elements

HTML specifies a set of tags that identify the structure of the document and the content type.

Tags are enclosed in angle brackets, i.e. < > (with text in between them).

<img src="image.gif"> specifies an image element.

Tags and elements

HTML specifies a set of tags that identify the structure of the document and the content type.

Tags are enclosed in angle brackets, i.e. < > (with text in between them).

<img src="image.gif"> specifies an image element.

Most tags come in pairs, marking a beginning and ending of an element.

The pair <title> and </title> enclose the title of a webpage (what is displayed at the very top of the web browser.

Tags and elements (cont.)

An HTML element is an object, typically enclosed by a pair of tags.

<title>My Home Page</title> is a “title” element.

This text appears bold. is a “bold” element.

Part of this text is bold. is a “paragraph” element that contains a “bold” element.

Structural Elements

A standard HTML document has two main structural elements.

<head> contains setup information for the browser and the Web page, e.g. the title for the browser window, style definitions, JavaScript code, ...

<body> contains the actual content to be displayed in the Web page.

Structural Elements

A standard HTML document has two main structural elements.

<head> contains setup information for the browser and the Web page, e.g. the title for the browser window, style definitions, JavaScript code, . . .

<body> contains the actual content to be displayed in the Web page.

```
<html>
  <!-- File: page01.html
     -- Creation: 2015.09.24
     -- Description: introductory page
   -->
  <head>
  <title>My first HTML document</title>
  </head>
       <body>
       <p> Hello world! </p>
       </body>
</html>
```

view page

Structural Elements (cont.)

HTML documents begin and end with <html> and </html> tags. (Ok, the Document Type Declaration comes before the <html> tag.)

Comments appear between <!-- and --> pairs.

The “head” section is enclosed between <head> and </head> tags, etc.

The <head> element

The <head> element is where you include a <title> element (that appears in the title bar of the browser, or the “tab title” of a tab in the browser).

You can also include lots of other type of information in the <head> element such as:

- Cascading Style Sheet (CSS) definitions and/or a link to an external style sheet (or several).
- “Meta” data, such as who authored the page, the type of content, and clues that search engines may (or may not) use to help categorize your webpage.
- JavaScript code.

The <body> element

The <body> element contains the main bulk of the material to be displayed on the webpage such as:

- Paragraphs.
- Tables and lists.
- Images.
- JavaScript code.
- PHP code can be included here too (if passed through a PHP parser before being served to the client's browser).
- Other embedded objects (<canvas> element, Java applets, videos, etc).

Text Layout (or lack of it)

```
<html>
<!-- COMP519 page02.html 2015.09.24 -->
<head>
  <title>Text Layout</title>
</head>

<body>
  <p>
  This is a paragraph of text<br>
  made up of two lines.
  </p>

  <p>
  This is another paragraph with a
    GAP   between
  some of the words.
  </p>

  <p>
     This paragraph is<br>
  indented on the first line<br>
  but not on subsequent lines.
  </p>
</body>
</html>
```

For the most part, the layout of the text is left to the browser.

Every (ok, almost every) sequence of whitespace is interpreted as a single space.

The web browser automatically wraps the text to fit into the window size.

view page

Text Layout (cont.)

Generally, you specify the parts of the document, such as what text constitutes a paragraph (using the and tags), that there is a list of items (in one of several different styles of lists), and so forth. The exact (default) layout is left to the browser.

A line break can be forced through the use of the
 tag.

A space character can be forced by using   for a "non-breaking" space.

Headers and Separating Blocks of Text

```
<html>
<!-- COMP519  page03.html  2015.09.24 -->
<head>
  <title>Blocks of Text</title>
</head>

<body>
  <h1>Major heading 1</h1>
  <p>
  Here is some text.
  </p>

  <h2>Subheading</h2>
  <p>
  Here is some subtext.
  </p>

  <hr>

  <h1>Major heading 2</h1>
  <p>
  Here is some more text.
  </p>
</body>
</html>
```

Headings can be specified.

<h1>...</h1> tags produce a large, bold heading.

<h2>...</h2> tags produce a slightly smaller heading.

...

<h6>...</h6> tags produce a tiny heading.

<hr> draws a horizontal line across the window.

view page

A Basic Webpage – A Worked Example

```
<html>
<!-- COMP519  page22.html  2015.09.24 -->
   <head>
      <title> Bill Smiggins Inc. </title>
   </head>
   <body>
      <h1>Bill Smiggins Inc.</h1>
      <h2>About our Company... </h2>
      <p>This Web site provides clients, customers,
         interested parties and our staff with all of
         the information that they could want on
         our products, services, success and failures.
      </p>
      <hr>
      <h3> Products </h3>
      <p> We are probably the largest
      supplier of custom widgets, thingummybobs, and bits
      and pieces in North America. </p>
      <hr>
   </body>
</html>
```

view page

A Basic Webpage – Another Example

```
<html>
 <!-- COMP519  page25.html  2015.09.24 -->
  <head> <title>Text Variations and
       Escape Sequences</title> </head>
  <body>
     <h1>Text Variations</h1>
     <p>We can use <b>simple</b> tags to
     <i>change</i> the appearance of
     <strong>text</strong> within
     <tt>webpages</tt>.
     Even super<sup>script</sup>
     and sub<sub>scripts</sub> are
     <em>supported</em>.</p>
    <h1>Text Escape Sequences</h1>
  <p>
   &amp; &lt; &gt; &quot; &copy;  &#261;
  </p>
 <h1>Preformatted text</h1>
      <pre>
          University of Liverpool
          Department of Computer Science
          Ashton Building, Ashton St
          Liverpool, L69 3BX, UK
      </pre>
 </body>
</html>
```

You can specify some styles for fonts.

... specify bold.

... specify italics.

<tt>...</tt> specify typewriter-like (fixed-width) font.

view page

Styling Text and Escape Sequences

<small>. . . </small> decreases the size of the font.

. . . puts emphasis.

. . . puts even more emphasis.

_{. . .} specify a subscript.

^{. . .} a superscript.

<pre>. . . </pre> specify ready-formatted text.

Escape characters exist for specifying all sorts of mathematical symbols, special characters, accented letters, etc.

£	£	€	€
ü	ü	ñ	ñ
©	©	and many more. . .	

Lists

There are three different types of lists defined in HTML.

 . . . specifies an ordered list (i.e. using numbers or letters to label each list item).

- . . . is used to denote each item in the list. (A list element contains one or more <li> elements inside of it.)

  With Cascading Style Sheets (CSS), you can specify how to enumerate the list (e.g. numbers or letters). For some browsers, you can even specify with which number (for example) to start the list.

Lists

There are three different types of lists defined in HTML.

 . . . specifies an ordered list (i.e. using numbers or letters to label each list item).

- . . . is used to denote each item in the list. (A list element contains one or more <li> elements inside of it.)

  With Cascading Style Sheets (CSS), you can specify how to enumerate the list (e.g. numbers or letters). For some browsers, you can even specify with which number (for example) to start the list.

. . . is used for an unordered list (using a bullet point for each item). elements are contained in the <ol> element.

With CSS you can change the symbol used for the list.

Lists

There are three different types of lists defined in HTML.

 . . . specifies an ordered list (i.e. using numbers or letters to label each list item).

- . . . is used to denote each item in the list. (A list element contains one or more <li> elements inside of it.)

  With Cascading Style Sheets (CSS), you can specify how to enumerate the list (e.g. numbers or letters). For some browsers, you can even specify with which number (for example) to start the list.

. . . is used for an unordered list (using a bullet point for each item). elements are contained in the <ol> element.

With CSS you can change the symbol used for the list.

<dl>. . . </dl> specifies a “definition list”. A <dt>. . . </dt> pair denotes each term in the list, and <dd>. . . </dd> identifies its definition.

Lists (cont.)

```
<html>
<!-- COMP519 page07.html  2015.09.24 -->
<head>  <title>Simple Lists</title>  </head>
<body>
<ul style="list-style-type: square;">
 <li> ... first list item... </li>
 <li> ... second list item... ... </li>
</ul>   <hr>
<dl>   <dt> Dweeb </dt>
   <dd> young excitable person who may
  mature into a <em>Nerd</em> </dd>
 <dt> Hacker </dt>
      <dd> a clever programmer </dd>
 <dt> Nerd </dt>  <dd> technically bright but
        socially inept person </dd>
</dl>   <hr>
 <ol >
  <li>This is the first item.</li>
  <li>Item number two.</li>
</ol>
</body>
</html>
```

view page

Hyperlinks

Perhaps the most important HTML element (the one that makes the Web what it is) is the hyperlink, or “anchor”.

- . . . is the way to make a hyperlink, where URL is the Web address of the page to be displayed when the user clicks on the link.

Hyperlinks

Perhaps the most important HTML element (the one that makes the Web what it is) is the hyperlink, or “anchor”.

- . . . is the way to make a hyperlink, where URL is the Web address of the page to be displayed when the user clicks on the link.

  The text between the start and end tags is what the user sees on the screen.

Hyperlinks

Perhaps the most important HTML element (the one that makes the Web what it is) is the hyperlink, or “anchor”.

- . . . is the way to make a hyperlink, where URL is the Web address of the page to be displayed when the user clicks on the link.

  The text between the start and end tags is what the user sees on the screen.
- . . . will open the URL in a new window (or tab).

  Note: It is probably unwise to (over)use the method to open new windows/tabs. (Many web users don’t like that...)

Hyperlinks (cont.)

```
<html>
<!-- COMP519 page08.html  2015.09.24 -->
<head>
  <title>Hyperlinks</title>
</head>
<body>
 <p>Links can, more or less, be included anywhere in
 a document.  <br>   You can talk about <a
 href="http://cgi.csc.liv.ac.uk/~martin/teaching/comp519">COMP519</a>
 being taught at the
 <a href="http://www.liv.ac.uk">University of Liverpool</a>.
 <br><br>
  Links can appear in lists:
 </p>
  <ul>
    <li><a href="page07.html" target="_blank">Open page07
     in a new window</a></li>
    <li>And, of course, not every item needs to be a link.</li>
  </ul>
</body>
</html>
```

view page

Hyperlinks (cont.)

For long documents, you can have links to other locations in that same document.

Use syntax like <xxx id="ident"> . . . </xxx> where “ident” is a variable for identifying this location.

"xxx" can, in principle, be any HTML element.

. . . will then jump to that location within the file.

. . . can jump into the middle of another webpage in the same way.

Hyperlinks (cont.)

```
<html>
<!-- COMP519  page09.html  2015.09.24 -->
<head>   <title>Internal Links in a Page</title>   </head>
<body>
  <p>[ <a href="#HTML">HTML</a> |
    <a href="#HTTP">HTTP</a> |
    <a href="#IP">IP</a> |
    <a href="#TCP">TCP</a> ]      </p>
  <p>
  Computer acronyms:
  <dl>
   <dt id="HTML">HTML</dt>
   <dd>HyperText Markup Language
   <dt id="HTTP">HTTP</dt>
   <dd>HyperText Transfer Protocol ... </dd>
   <dt id="IP">IP</dt>
   <dd>Internet Protocol ... </dd>
   <dt id="TCP">TCP</dt>
   <dd>Transfer Control Protocol ... </dd>
  </dl>
  </p>
</body>
</html>
```

view page

Images

Images are included with the <img> tag. (Note that this tag has no “ending” tag.)

By default, browsers can (or should!) display GIF, JPG, and PNG formats of images. Support for other image formats varies...

Images

Images are included with the <img> tag. (Note that this tag has no “ending” tag.)

By default, browsers can (or should!) display GIF, JPG, and PNG formats of images. Support for other image formats varies...

Typical syntax is as follows:

<img src="URL (or filename)" height="n" width="n" alt="text" title="text">

```
<html>
<!-- COMP519  page10.html  2015.09.25 -->
<head>
  <title>Image example</title>
</head>
<body>
<img src="http://www.csc.liv.ac.uk/~martin/teaching/comp519/HTML/Cathedral.jpg"
title="Liverpool’s Anglican cathedral"
alt="image of Liverpool’s Anglican Cathedral" width="400" />

<p>The Anglican Cathedral of Liverpool</p> </body>

</html>
```

view page

Images (cont.)

Usual parameters to use with the <img> tag include:

- src - Specifies the file name (and can include a URL).

Images (cont.)

Usual parameters to use with the <img> tag include:

- src - Specifies the file name (and can include a URL).
- width and/or height - Dimensions in pixels. Often only need to specify one of them and the other is automatically scaled to match. Where possible pictures should be resized using other programs to save on bandwidth and problems that some (older) browsers might have with resizing images.

Images (cont.)

Usual parameters to use with the <img> tag include:

- src - Specifies the file name (and can include a URL).
- width and/or height - Dimensions in pixels. Often only need to specify one of them and the other is automatically scaled to match. Where possible pictures should be resized using other programs to save on bandwidth and problems that some (older) browsers might have with resizing images.
- title - Displayed when the mouse is “hovered” over the picture.

Images (cont.)

Usual parameters to use with the <img> tag include:

- src - Specifies the file name (and can include a URL).
- width and/or height - Dimensions in pixels. Often only need to specify one of them and the other is automatically scaled to match. Where possible pictures should be resized using other programs to save on bandwidth and problems that some (older) browsers might have with resizing images.
- title - Displayed when the mouse is "hovered" over the picture.
- alt - Text that is displayed when the image is missing, can't be loaded (e.g. if file permissions aren't set correctly), or if the client has disabled loading images in his/her browser.

  Proper web programming practice is to include meaningful alt text for all images.

Image maps

HTML5 includes the <map> tag to define an image map. An image map is an image that has clickable areas (that are typically links, say, to other images).
The following example is taken from the W3Schools website:

```
<!DOCTYPE html>
<html>
<head><title>Image map example</title></head>

<body>
<p>Click on the sun or on one of the planets to watch it closer:</p>

<img src="planets.gif" alt="Planets" usemap="#planetmap"
       style="width:145px;height:126px;">

<map name="planetmap">
  <area shape="rect" coords="0,0,82,126" alt="Sun"
          href="sun.htm">
  <area shape="circle" coords="90,58,3" alt="Mercury"
          href="mercur.htm">
  <area shape="circle" coords="124,58,8" alt="Venus"
          href="venus.htm">
</map>

</body>
</html>
```

Tables

Tables are a common way of displaying data and other information.

A table's contents is divided into rows and columns.

By default, in a web browser, column entries are left-justified, so you must provide for your own alignment when needed (using Cascading Style Sheets, for example).

<table>. . .</table> specify a table element.

<tr>. . .</tr> specify a row in the table.

<td>. . .</td> specify table data (i.e. each column entry in the table).

Tables

Tables are a common way of displaying data and other information.

A table's contents is divided into rows and columns.

By default, in a web browser, column entries are left-justified, so you must provide for your own alignment when needed (using Cascading Style Sheets, for example).

<table>. . .</table> specify a table element.

<tr>. . .</tr> specify a row in the table.

<td>. . .</td> specify table data (i.e. each column entry in the table).

Each <table> element can contain a number of <tr> elements, where each <tr> element typically contains several <td> elements (usually the same number).

Tables (cont.)

```
<html>
<!-- COMP519  page11.html  2015.09.25 -->
<head>
  <title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
    <table>
       <tr>  <td> Left Column </td>
                 <td> Right Column </td>  </tr>
       <tr>  <td> Some data </td>
                 <td> Some other data </td>
       </tr>
   </table>
</body>
</html>
```

view page

Tables (cont.)

```
<html>
<!-- COMP519  page11.html  2015.09.25 -->
<head>
  <title>Tables</title>
</head>
<body>
<h2>A Simple Table</h2>
    <table>
       <tr>  <td> Left Column </td>
                <td> Right Column </td>  </tr>
       <tr>  <td> Some data </td>
                <td> Some other data </td>
       </tr>
   </table>
</body>
</html>
```

view page

Note that the browser (by default) doesn't provide lines to separate rows and columns, and no particular formatting that helps to identify the table as a table (in the way that we might typically expect to see). Use CSS to provide these effects.

Layout in a Table

```
<html>
<!-- COMP519  page12.html  2015.09.25 -->
<head>
  <title>Table Layout</title>
</head>
<body>
  <table style="border: 1px solid;">
     <tr style="text-align: center;">
     <td style="border: 1px solid;">
            Left<br>Column</td>
     <td style="border: 1px solid;
         vertical-align: top;">
            Right Column</td>
    </tr>
    <tr>
      <td style="border: 1px solid;">
           Some data</td>
      <td style="border: 1px solid;">
           Some data</td>
    </tr>
  </table>
</body>
</html>
```

view page

One can use “style” attributes to provide borders.

<table style=“border: 1px solid;”>

Can control the horizontal and vertical layout within cells.

<td style=“text-align: center”>

Can apply layout to an entire row.

<tr style=“text-align: center”>

We will explore this more with CSS.

Layout in a Table (A Better Way)

```
<html>
<!-- COMP519  page12a.html  2015.09.25 -->
<head>
  <title>Table Layout</title>
  <style type="text/css">
     table {  border: 1px solid;  }
     tr    {  text-align: center;  }
     td    {  border: 1px solid;  }
  </style>
</head>
<body>
  <table>
     <tr>  <td>Left<br>Column</td>
           <td style="vertical-align: top;">
           Right Column</td>
    </tr>
    <tr>  <td>Some data</td>
          <td>Some data</td>
    </tr>
  </table>
</body>
</html>
```

view page

Note the Style Sheet in the <head> element which takes care of most of the formatting of the table.

The HTML in the <table> element is much "cleaner" and easier to read.

Other Table Attributes

You can control the space between cells and margins within cells.

This is the “padding” attribute in the <table>, <th>, and <td> (and other) style sheet declarations. (More on this with Cascading Style Sheets.)

Other Table Attributes

You can control the space between cells and margins within cells.

This is the “padding” attribute in the <table>, <th>, and <td> (and other) style sheet declarations. (More on this with Cascading Style Sheets.)

You can add table headings.

<th> is similar to <td> but displays heading centered in bold.

Other Table Attributes

You can control the space between cells and margins within cells.

This is the “padding” attribute in the <table>, <th>, and <td> (and other) style sheet declarations. (More on this with Cascading Style Sheets.)

You can add table headings.

<th> is similar to <td> but displays heading centered in bold.

A table can have data that spans more than one column.

<td colspan="2">. . . </td>

Similarly, a table cell can span more than one row.

<td rowspan="2">. . . </td>

The next example uses CSS commands in the page <head>.

Other Table Attributes (cont.)

```
<html>
<!-- COMP519  page14.html  2015.09.25 -->
<head>
  <title>Table Formatting</title>
 <style type="text/css" media="screen">
     table { border: 1px solid; padding: 1px;}
     th, td {  border: 1px solid; padding: 10px;
               text-align: center; }
  </style>
</head>
<body>
  <table>
    <tr>  <th>HEAD1</th> <th>HEAD2</th> <th>HEAD3</th>  </tr>
    <tr>  <td>one</td> <td>two</td> <td>three</td>  </tr>
    <tr>
      <td rowspan="2"> four </td>
      <td colspan="2"> five </td>
    </tr>
    <tr>  <td> six </td> <td> seven </td>  </tr>
</table>
</body>
</html>
```

view page

Other HTML5 elements

Some other HTML5 elements that I haven't discussed here include:

- <canvas> - A container that can be used to hold graphics, typically using JavaScript to do the drawing.

Other HTML5 elements

Some other HTML5 elements that I haven't discussed here include:

- <canvas> - A container that can be used to hold graphics, typically using JavaScript to do the drawing.
- <figure> - A stylized container (with pre-defined CSS directives) for holding images, together with <figcaption> element to provide a caption for the <figure> element.

Other HTML5 elements

Some other HTML5 elements that I haven't discussed here include:

- <canvas> - A container that can be used to hold graphics, typically using JavaScript to do the drawing.
- <figure> - A stylized container (with pre-defined CSS directives) for holding images, together with <figcaption> element to provide a caption for the <figure> element.
- <embed> - A container for an external application or interactive content (such as a Flash (boo!!) animation).

Other HTML5 elements

Some other HTML5 elements that I haven't discussed here include:

- <canvas> - A container that can be used to hold graphics, typically using JavaScript to do the drawing.
- <figure> - A stylized container (with pre-defined CSS directives) for holding images, together with <figcaption> element to provide a caption for the <figure> element.
- <embed> - A container for an external application or interactive content (such as a Flash (boo!!) animation).
- <nav> - Can be used for a set of "navigation links" (say, at the top of a webpage).

Other HTML5 elements

Some other HTML5 elements that I haven't discussed here include:

- <canvas> - A container that can be used to hold graphics, typically using JavaScript to do the drawing.
- <figure> - A stylized container (with pre-defined CSS directives) for holding images, together with <figcaption> element to provide a caption for the <figure> element.
- <embed> - A container for an external application or interactive content (such as a Flash (boo!!) animation).
- <nav> - Can be used for a set of "navigation links" (say, at the top of a webpage).
- etc....

More help...

One of the best sources for more information on specific HTML tags is the w3schools.com.

And there are obviously many other online sources...

Remember... You should acknowledge sources that you use in developing your webpages (especially, if you use code found online).