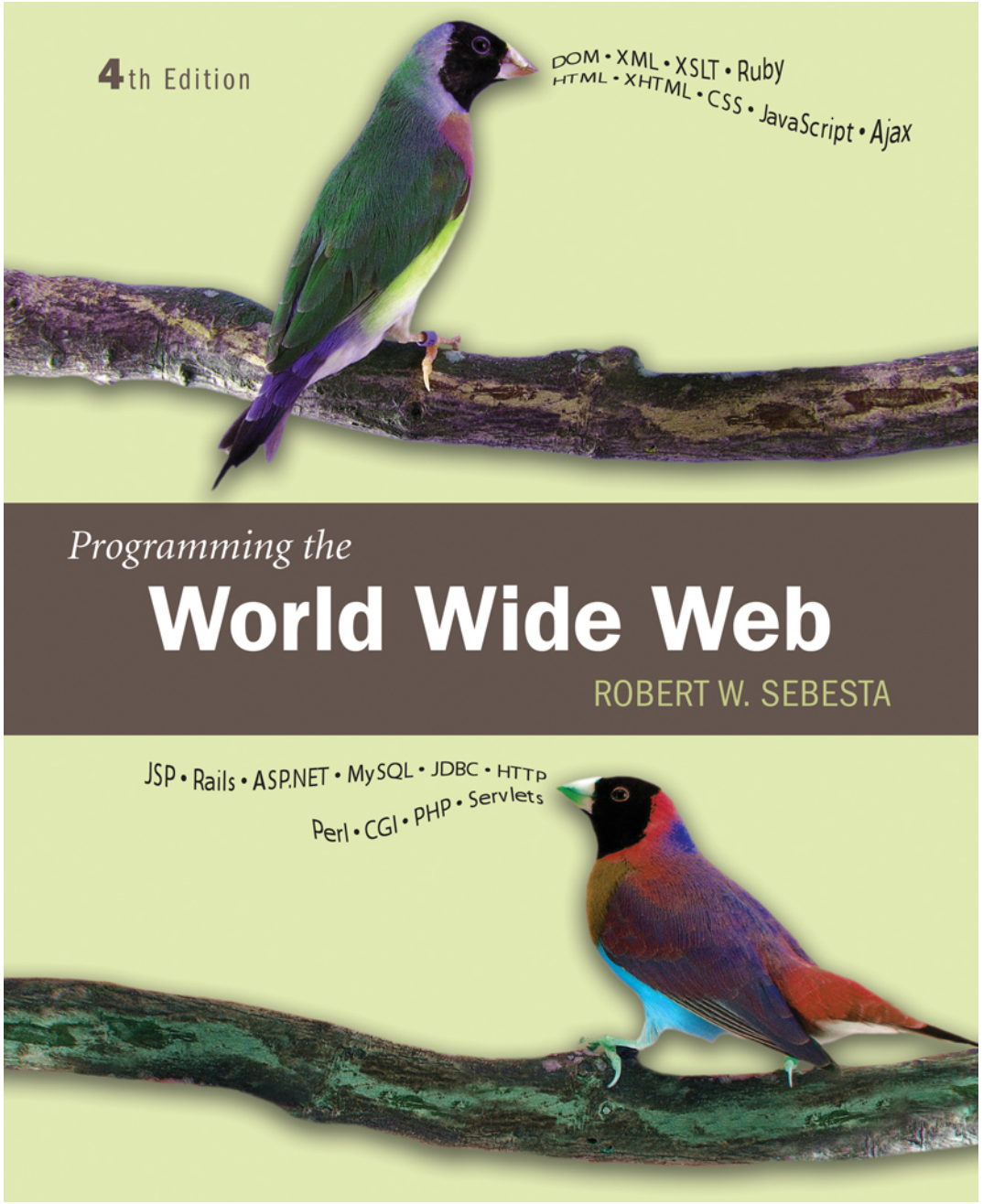


Chapter 2
Introduction to XHTML

PEARSON Addison Wesley

Topics

- Origins and Evolution of HTML and XHTML
- Basic syntax
- Standard XHTML Document structure
- Basic text markup
- Images
- Hypertext links
- Lists
- Tables
- Forms
- Frames
- Syntactic differences between HTML and XHTML

Origins and Evolution of HTML and XHTML

- HTML was defined with SGML
- SGML is an ISO standard notation for describing text-formatting languages.
- Original intent of HTML: General layout of documents that could be displayed by a wide variety of computers, using different browsers.

- Original version of HTML was designed, in conjunction with the structure of the web.
- Mosaic – was commercialized and marketed by Netscape.
- Internet Explorer – was developed by Microsoft.

Versions of HTML and XHTML

- HTML 2.0 – released in 1995
- HTML 3.2 – early 1997.
- After 1997, Evolution of HTML was dominated by W3C.
- Recent versions:
  - HTML 4.0 – 1997
    - » Introduced many new features and deprecated many older features
  - Latest version HTML 4.01 - 1999 - A cleanup of 4.0
  - XHTML 1.0 - 2000
    - » Just 4.01 defined using XML, instead of SGML
    - » Redefinition of HTML 4.0.1 using XML.
  - XHTML 1.1 – 2001
    - » Modularized 1.0, and drops frames
    - » We'll stick to 1.1, except for frames

HTML versus XHTML

- Reasons to use XHTML, rather than HTML:

1. HTML has lax syntax rules, leading to sloppy and sometime ambiguous documents.

   – XHTML syntax is much more strict, leading to clean and clear documents in a standard form.

2. HTML processors do not even enforce the few syntax rule that do exist in HTML.
3. The syntactic correctness of XHTML documents can be validated.
4. Quality and consistency .
5. Syntactic correctness of a XHTML document can be checked, either by an XML browser or by a validation tool.

   - Otherwise errors go undetected until the document is posted on a site.

- Tidy – is a tool to convert legacy HTML documents to XHTML documents.

HTML versus XHTML

HTML	XHTML
HTML is lot easier to write as the rules are not strict	XHTML is not so easier to write as there are a lot of rules
All browsers support HTML	There are a few older versions which do not support XHTML
There are lot of loop holes like they are not required to have compulsory closing tags	They enforce a lot of constraints like each of these tags have to have a closing tags mandatory)
Whenever a document is in HTML there are no tools to validate its syntactic correctness	Whenever a document is in XHTML there are tools to validate its syntactic correctness which helps in error identification.

Basic syntax

- The fundamental syntactic units of HTML are called *tags* .
- syntax of tag is the tag's name enclosed within the "<>"(angled brackets) .
- Tag names should be in *lowercase letter.*
- Each of the tags have to have a corresponding closed tag.
- For e.g. something here
- Tag format:

  - Opening tag: <name>

  - Closing tag: </name>

  - The opening tag and its closing tag together specify a container for the *content* they enclose.

- The container and its content together are called an element.

For e.g. This is extremely simple

- Not all tags have content
  - If a tag has no content, its form is <name />.

- If a tag has attributes, they appear between its name and the right bracket of the opening tag.

  - Attribute names, like tag names, are written in lowercase letters.

  - Attribute values must by delimited by double quotes.

- Comment form: <!– anything except two adjacent dashes… -->

- Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs

- Tags are suggestions to the browser, even if they are recognized by the browser.

Standard XHTML Document Structure

- Every XHTML document must begin with:
  - XML declaration element that identifies the document as being one based on XML.
  - This element includes an attribute that specifies the version number, 1.0
  - second attribute encoding specifies Unicode encoding , utf-8.

  <?xml version = “1.0” encoding = “utf-8”?>

  - Following XML declaration element is an SGML DOCTYPE command, which specifies the particular SGML document-type definition (DTD) with which document complies.

<!DOCTYPE html PUBLIC ″-//w3c//DTD XHTML 1.1//EN″
http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>

- Four tags <html>, <head>, <title>, and <body> are required in every document.
- The whole document must have <html> as its root.

- XHTML documents always have an <html> tag immediately following DOCTYPE command, end with closing html tag.
  - html must have the xmlns XHTML namespace attribute:

    <html xmlns = ″http://www.w3.org/1999/xhtml″

Example of HTML code:

- **<!DOCTYPE html>**
- **<html>**
- **<head>**
- **<title>Page Title</title>**
- **</head>**
- **<body>**
- **<h1>This is a Heading</h1>**
- ** This is a paragraph. **
- **</body>**
- **</html>**

- An XHTML document consists of 2 parts: a head and a body.
- <head> element : provides info about document.
- <body> element: provides content of the document.
- The <title> tag is used to give the document a title, which is normally displayed in the browser's window title bar (at the top of the display).

- Prior to XHTML 1.1, a document could have either a body or a frameset.

- There are several other compulsory tags which we define in the following example the text version can be found in the following link.

- Another simple example is as shown here.
- W3C provides a convenient Web-based way to validate XHTML documents against its standards.

- URL of the W3C HTML Validation Service

  http://validator.w3.org/sample.html

Basic Text Markup

- Text is normally organized into paragraphs in the body of document.

Paragraph Elements

Textual paragraph appear as the content of a paragraph element.

- The <p> tag breaks the current line and inserts a blank line - the new line gets the beginning of the content of the paragraph.
- The browser puts as many words of the paragraph's content as will fit in each line.

The textThe text and the html for paragraph tag is as shown here.

The multiple spaces in the source paragraph text are replaced by single space.

Another example for the usage of paragraph tag is as shown with the text Another example for the usage of paragraph tag is as shown with the text and html here.

Line breaks

- The effect of the break tag is the same as that of <p>, except for the blank line
  - » No closing tag!

- The break tag is specified as <br />.
  Slash indicates that tag is both an opening and closing tag.

  The text The text and html is as shown here.

Preserving whitespaces:

- To preserve the browser from eliminating multiple spaces and ignoring embedded line breaks.

- This can be specified with the <pre> tag.

- The textThe text and html is as shown here.

Headings

- Six levels of headings, 1 - 6, specified by the tags <h1> to <h6>
- 1, 2, and 3 use font sizes that are larger than the default font size
- 4 uses the default size
- 5 and 6 use smaller font sizes
- <h1> is the highest-level heading.

- The heading tags always break the current line, so their content always appear in the new

  line.
- Browsers usually insert some vertical space before and after the headings.

- The text The text and html is as shown here.

Blockquotes

- If we want a block of text to be set off from the normal flow in a document.
  - Tag designed is <blockquote>.
  - Browsers decide how to make the content look different and often indent, and sometimes italicize.
  - The text The text and html is as shown here.

- XHTML tags are categorized as either block or inline.
- The content of the inline tag appears on the current line.
- An inline tag does not implicitly include a line break, exception is
 (inline).
- A block tag breaks the current line so that its content appears on a new line.
- Block tags: Heading, block quote. Inline tags: , .

- In XHTML, block tag can never be nested directly in an inline tag.
- Also inline tags and text cannot be nested in body or form elements.
- Only block tag can be nested directly in body or form elements.

Font Styles and Sizes (can be nested)

- Content based style tags.
  - Boldface - <b>
  - Italics - <i>
  - Larger - <big>
  - Smaller - <small>
  - Emphasis -
  - Strong -
    - » These tags are not affected if they appear in the content of a <blockquote>, unless there is a conflict (e.g., italics).
  - *Superscripts and subscripts*
    - » Subscripts with <sub>
    - » Superscripts with <sup>

    Example: x₂³

    Display: x_2^3

The textThe text The text and html is as shown here.

Character Entities

- The special characters are defined as entities, which are names for the characters by the browser.
- An entity in a document is replaced by its associated character by the browser.

Char.	*Entity*	*Meaning*
&	&amp;	Ampersand
<	&lt;	Less than
>	&gt;	Greater than
”	&quot;	Double quote
’	&apos;	Single quote
¼	&frac14;	One quarter
½	&frac12;	One half
¾	&frac34;	Three quarters
°	&deg;	Degree
(space)		Non-breaking space

Horizontal rules

– <hr /> draws a line across the display, after a line break

The text The text and html is as shown here.

Images

- Image is stored in a file, which is specified by an XHTML request. The image is inserted into the display of the document by the browser.

- Common methods of representing images:
- GIF (Graphic Interchange Format)
  - 8-bit color (256 different colors)
- JPEG (Joint Photographic Experts Group)
  - 24-bit color (16 million different colors)

- Both use compression, but JPEG compression is better.

- This compression process actually loses some of the color accuracy of the image, but is rarely discernable by the user.

- Third image format – Portable Network Graphics (PNG).
  - It has the best characteristics of both GIF and JPEG. Relatively new.
  - Should eventually replace both gif and jpeg

• Images are inserted into a document with the <img /> tag, which is an inline tag.

• 2 attributes :

src attribute : specifies the file containing the image.

alt attribute: specifies text to be displayed when it is not possible to display the image.

» Purposes of alt attribute:

1. Non-graphical browsers.
2. Browsers with images turned off.

Eg:

• The <img> tag has 30 different attributes, including width and height (in pixels)

• Two optional attributes: width and height

Can be used to specify (in pixels) the size of the rectangle for the image.

The text The text and htmlThe text and html is as shown here.

```
<!-- image.html
     An example to illustrate an image
     -->
<html xmlns = ″http://www.w3.org/1999/xhtml″>
  <head> <title> Images </title>
  </head>
  <body>
    <h1> Aidan's Airplanes </h1>
    <h2> The best in used airplanes </h2>
    <h3> "We've got them by the hangarful"
    </h3>
    <h2> Special of the month </h2>
    <p>
      1960 Cessna 210 <br />
      577 hours since major engine overhaul
      <br />
      1022 hours since prop overhaul
      <br /><br />
      <img src = "c210new.jpg"
           alt = "Picture of a Cessna 210"/>
      <br />
      Buy this fine airplane today at a
      remarkably low price <br />
      Call 999-555-1111 today!
    </p>
  </body>
</html>
```

Aidan's Airplanes

The best in used airplanes

"We've got them by the hangarful"

Special of the month

1960 Cessna 210
577 hours since major engine overhaul
1022 hours since prop overhaul

Buy this fine airplane today at a remarkably low price
Call 999-555-1111 today!

Hypertext Links

- Hypertext is the essence of the Web!
- A Hypertext link in an XHTML document (link) is usually a pointer to some resource these resource can be

  - any web resource.
  - any place with in the same document.
  - any place with in the other document.

- A link that points to a different document specifies the address of that document.
- Such address can be a filename, a directory path and a filename, or a complete URL.

- Links are specified in an attribute of an anchor tag (<a>), which is an inline tag.

- The anchor tag that specifies a link is called the source of that link.
- The document whose address is specified in a link is called the target of that link.

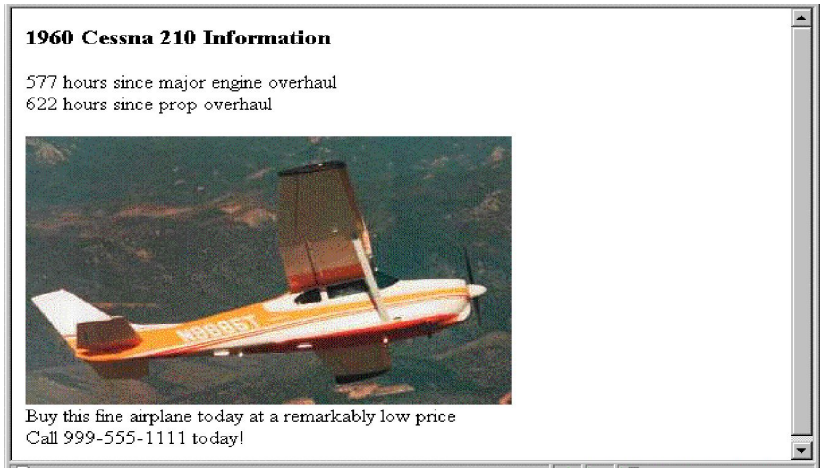
- A link is specified with the href (*h*ypertext *ref*erence) attribute of <a> (the anchor tag).
- The value assigned to href specifies the target of the link.

- Target if is a document in the same directory then specify just its filename.
- Target if is in some other directory UNIX pathname conventions are used.

Note: Relative addressing of targets is easier to maintain and more portable than absolute addressing (entire pathname is given).

e.g. optional text to be displayed in the place of link

```
<!-- link.html
     An example to illustrate a link
     -->
<html xmlns = ″http://www.w3.org/1999/xhtml″>
  <head> <title> Links </title>
  </head>
  <body>
    <h1> Aidan's Airplanes </h1>
    <h2> The best in used airplanes </h2>
    <h3> "We've got them by the hangarful"
    </h3>
    <h2> Special of the month </h2>
    <p>
      1960 Cessna 210 <br />
      <a href = "C210data.html"> Information on the Cessna 210 </a>
    </p>
  </body>
</html>
```



Usually the

e.g. optional text to be displayed in the place of link

- any web resource text text html

- place within the same doc text - place within the same doc text html

- place within other documents text - place within other documents text html

- If the target is not at the beginning of the document, the target spot must be marked.
- Target labels can be defined in many different tags with the id attribute, as in

  <h1 id = "baskets"> Baskets </h1>
- The link to an id must be preceded by a pound sign (#); If the id is in the same document, this target could be

  What about baskets?

- If the target is in a different document, the document reference must be included

   Baskets
- Style note: a link should blend in with the surrounding text, so reading it without taking the link should not be made less pleasant.

- Links can have images:

   Info on C210

Lists

- XHTML provides simple and effective ways for arranging the elements in a systematic fashion via lists

  There are several types of lists

  - Ordered Lists
  - Unordered Lists
  - Definition Lists

Unordered Lists

The <ul> tag which is a block tag creates an unordered list. Each item in a list is specified with an <li> tag .

The list is the content of the <ul> tag. List elements are the content of the <li> tag

- ul - *unordered list.*
- li - *list item.*

The example for the unordered list is shown here The example for the unordered list is shown here. The example for the unordered list is shown here and text

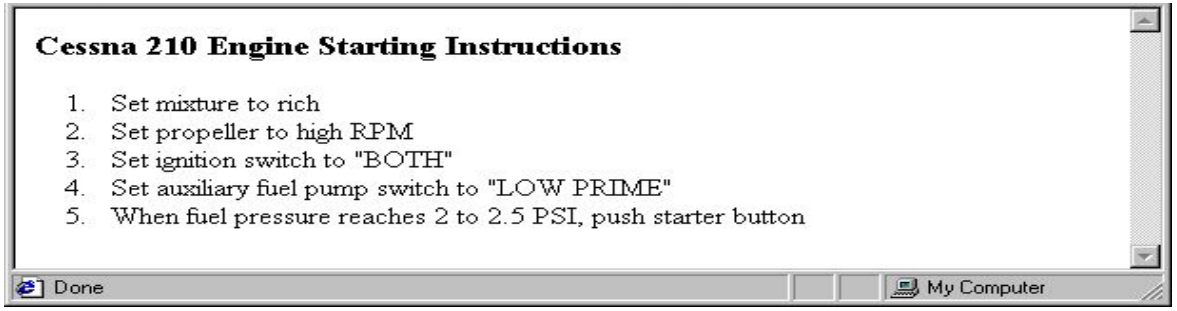
```
<h3> Some Common Single-Engine Aircraft </h3>
 <ul>
   <li> Cessna Skyhawk </li>
   <li> Beechcraft Bonanza </li>
   <li> Piper Cherokee </li>
 </ul>
```

Ordered Lists

- These type of lists are used to create lists which are sequential.These lists are usually numbered.
- An ordered list is created within the block tag <ol>.The list is the content of the <ol> tag.
- Each item in the display is preceded by a sequence value.

The example for the ordered list is shown here The example for the ordered list is shown here and text

```
<h3> Cessna 210 Engine Starting Instructions
</h3>
<ol>
  <li> Set mixture to rich </li>
  <li> Set propeller to high RPM </li>
  <li> Set ignition switch to "BOTH" </li>
  <li> Set auxiliary fuel pump switch to
       "LOW PRIME" </li>
  <li> When fuel pressure reaches 2 to 2.5
       PSI, push starter button </li>
</ol>
```



- Nested lists
  - Any type list can be nested inside any type list.
  - The nested list must be in a list item.

We will see the html We will see the html and source pages here

Definition lists (for glossaries, etc.)

This type of list is usually used to list the element and provide its definition to it .

The tags used is

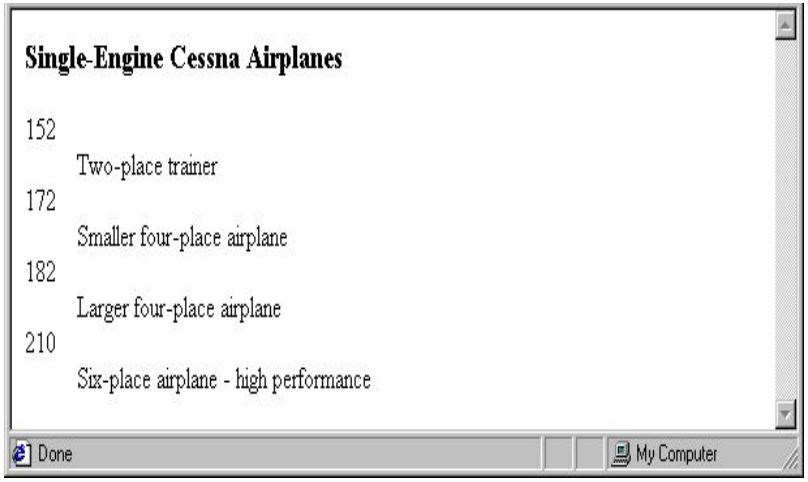
<dl> and </dl> - To define a definition list.

<dt> and </dt> - List the element.

<dd> and </dd>- Provide the description.

The example is listed here The example is listed here The example is listed here and the source The example is listed here and the source

```
<h3> Single-Engine Cessna Airplanes </h3>
<dl >
  <dt> 152 </dt>
  <dd> Two-place trainer </dd>
  <dt> 172 </dt>
  <dd> Smaller four-place airplane </dd>
  <dt> 182 </dt>
  <dd> Larger four-place airplane </dd>
  <dt> 210 </dt>
  <dd> Six-place airplane - high performance
  </dd>
</dl>
```



Lists

- XHTML provides simple and effective ways for arranging the elements in a systematic fashion via lists

  There are several types of lists

  - Ordered Lists
  - Unordered Lists
  - Definition Lists

Unordered Lists

The <ul> tag which is a block tag creates an unordered list. Each item in a list is specified with an <li> tag .

The list is the content of the <ul> tag. List elements are the content of the <li> tag

- ul - *unordered list.*
- li - *list item.*

The example for the unordered list is shown here The example for the unordered list is shown here and text

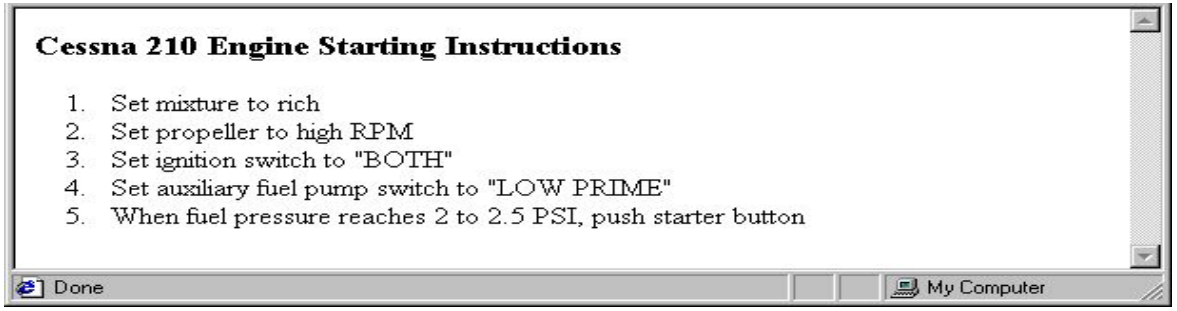
```
<h3> Some Common Single-Engine Aircraft </h3>
 <ul>
   <li> Cessna Skyhawk </li>
   <li> Beechcraft Bonanza </li>
   <li> Piper Cherokee </li>
 </ul>
```

Ordered Lists

- These type of lists are used to create lists which are sequential.These lists are usually numbered.
- An ordered list is created within the block tag <ol>.The list is the content of the <ol> tag.
- Each item in the display is preceded by a sequence value.

The example for the ordered list is shown here The example for the ordered list is shown here and text

```
<h3> Cessna 210 Engine Starting Instructions
</h3>
<ol>
  <li> Set mixture to rich </li>
  <li> Set propeller to high RPM </li>
  <li> Set ignition switch to "BOTH" </li>
  <li> Set auxiliary fuel pump switch to
       "LOW PRIME" </li>
  <li> When fuel pressure reaches 2 to 2.5
       PSI, push starter button </li>
</ol>
```



- Nested lists
  - Any type list can be nested inside any type list.
  - The nested list must be in a list item.

We will see the html We will see the html and source pages here

Definition lists (for glossaries, etc.)

This type of list is usually used to list the element and provide its definition to it .

The tags used is

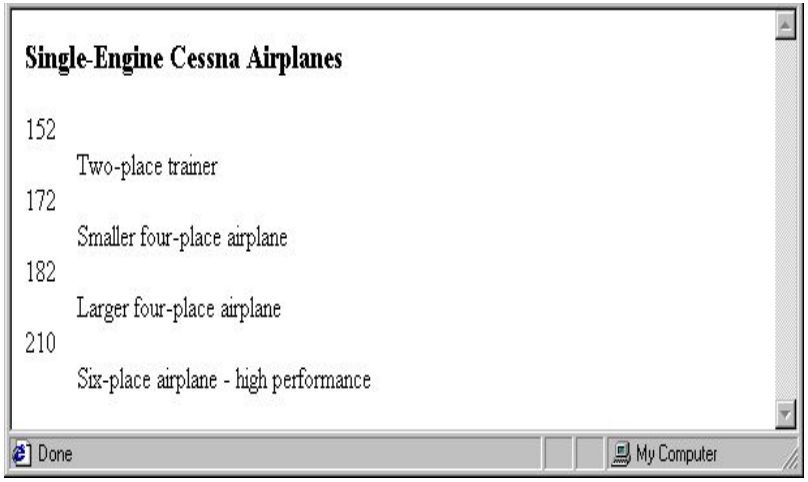
<dl> and </dl> - To define a definition list.

<dt> and </dt> - List the element.

<dd> and </dd>- Provide the description.

The example is listed here The example is listed here and the source

```
<h3> Single-Engine Cessna Airplanes </h3>
<dl >
  <dt> 152 </dt>
  <dd> Two-place trainer </dd>
  <dt> 172 </dt>
  <dd> Smaller four-place airplane </dd>
  <dt> 182 </dt>
  <dd> Larger four-place airplane </dd>
  <dt> 210 </dt>
  <dd> Six-place airplane - high performance
  </dd>
</dl>
```



Tables

- Table is a collection of a set of rows and columns. Each of the intersection of these rows and columns are *cells* .
- A table is a matrix of cells, each possibly having content.
- The cells in the top row contain column labels and leftmost column contain row labels, rest of the cells contain data.

- A table is specified as the content of block tag<table>.
- A **border** attribute in the <table> tag specifies a border between the cells i.e., line around the outside of whole table, lines that separate the cells from each other are called **rules**.

- If border is set to "border", the browser's default width border and rules are used.
- The border attribute can be set to a number, which will be the border width in pixels, example border="3" specifies a border 3 pixels wide.
- Without the border attribute, the table will have no lines!

- Tables are given titles with the <caption> tag, which can immediately follow <table> tag.
- The cells of a table are specified one at a time.

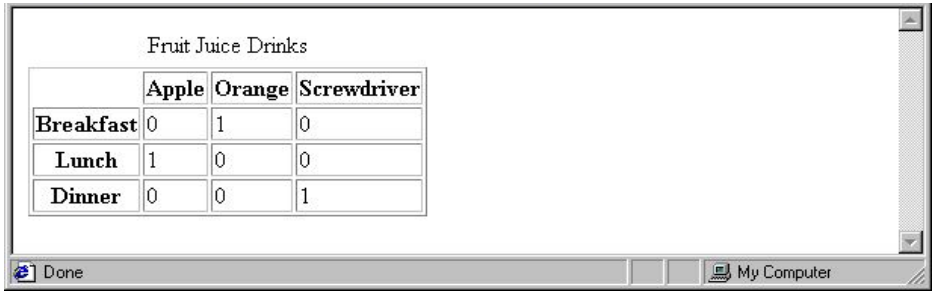
- Each row is specified with a row tag,<tr>.
- The row label is specified by the table heading tag, <th>.
- Each data cell of a row is specified with the table data tag, <td>.

The basic table tags

TAGS	EXPLANATION
<table> and </table>	Are the tags which are used to indicate the beginning and ending of the table. There are 2 attributes to this tag 1 . Border - one around the table e.g. border=2 : defines that the border with pixel size 2 2 . Rules - one between the cells e.g. rules=2 : defines the thickness in pixels between cells
<tr> and </tr>	Are the tags which indicate the row in the table. So nesting takes place these tags are placed within the table tags
<td> and </td>	Are the table data tags which are used to place the values within the cells
<th> and </th>	Are the table heading tags which are also known as *labels*

The html The html and the corresponding source.

```
<table border = "border">
  <caption> Fruit Juice Drinks </caption>
    <tr>
      <th> </th>
      <th> Apple </th>
      <th> Orange </th>
      <th> Screwdriver </th>
    </tr>
    <tr>
      <th> Breakfast </th>
      <td> 0 </td>
      <td> 1 </td>
      <td> 0 </td>
    </tr>
    <tr>
      <th> Lunch </th>
      <td> 1 </td>
      <td> 0 </td>
      <td> 0 </td>
    </tr>
  </table>
```



- A table can have multiple levels of row or column labels.
  - If so, the colspan attribute must be set in the <th> tag to specify that the label must span some number of columns

```
<tr>
  <th colspan = "3"> Fruit Juice Drinks </th>
</tr>
<tr>
  <th> Orange </th>
  <th> Apple </th>
  <th> Screwdriver </th>
</tr>
```

- The colspan attribute specification in a table header or table data tag tells the browser to make the cell as wide as the specified number of rows below it in the table.

- If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using rowspan.

```
<table border = "border">
  <tr>
    <td rowspan = "2"> </td>
    <th colspan = "3"> Fruit Juice Drinks
    </th>
  </tr>
  <tr>
    <th> Apple </th>
    <th> Orange </th>
    <th> Screwdriver </th>
  </tr>
        …
</table>
```

- The placement of the content within a table cell can be specified with the **align** and **valign** attributes in the <tr>,<th> and <td> tags.

- The align attribute controls the horizontal placement of the contents in a table cell.
  - Values are left, right, and center (default).
  - align is an attribute of <tr>, <th>, and <td> elements.
  - The default alignment for <th> cells is center; for <td> cells, it is left.

    - If align is specified in a <tr> tag, it applies to all cells in the row, If it is included

      in a <th> or <td> tag, it only applies to that cell.

- The valign attribute controls the vertical placement of the contents of a table cell.
  - Values are top, bottom, and center (default).
  - valign is an attribute of <th> and <td> elements.

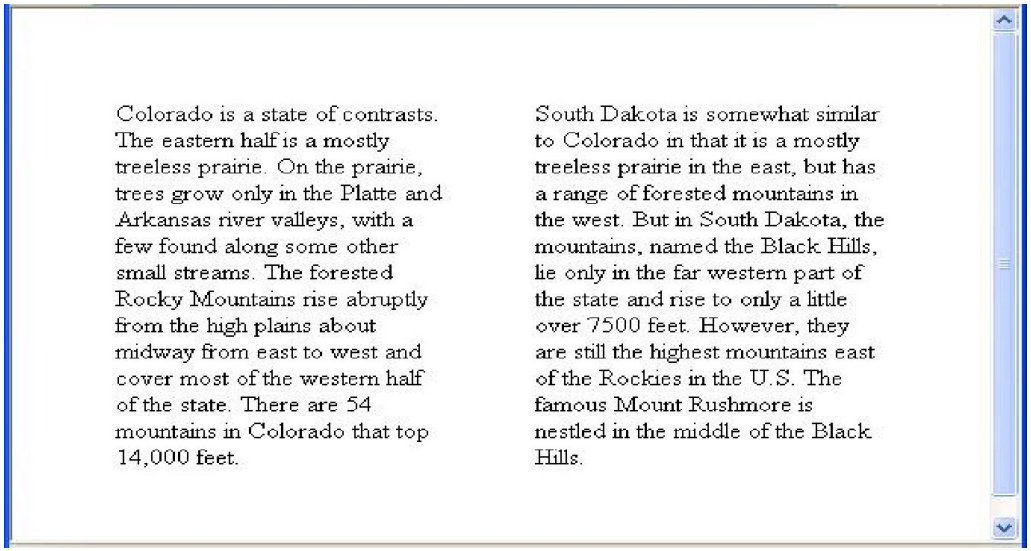
- The table tag has two attributes that can be used to specify the spacing between the content of a table cell and the cell's edge and the spacing between adjacent cells.

- The cellspacing attribute of <table> is used to specify the distance between cells in a table.

- The cellpadding attribute of <table> is used to specify the spacing between the content of a cell and the inner walls of the cell.

- Source codeSource code and the html to demonstrate use of padding and spacing is as shown.

```
<table cellspacing = "50">
    <tr>
      <td> Colorado is a state of …
      </td>
      <td> South Dakota is somewhat…
      </td>
    </tr>
 </table>
```



- Another example illustrating use of table tag and other attributes are shown with sourceAnother example illustrating use of table tag and other attributes are shown with source and html as shown.

Forms

- A form is the common way for a user to communicate information from a web browser to the server.
- XHTML has tags to create a collection of objects that implement this information gathering.
  - The objects are called *controls* or *widgets* (e.g., radio buttons and checkboxes).
  - All control tags are inline tags.

- Each control can have a value given in a form called the form data.
- When the Submit button of a form is clicked, the form's values are sent to the server.
- When the user clicks the Submit button, the form data is encoded and sent to the web server for processing.

- All of the widgets, or components of a form appear in the content of a <form> tag, which is a block tag.
- The only required attribute of <form> is *action*, which specifies the URL of the application on the web server that is to be called when the Submit button is clicked.

  action = "http://www.cs.ucp.edu/cgi-bin/survey.pl"

  If the form has no action, the value of action is the empty string.

- The *method* attribute of <form> specifies one of the two techniques: get and post, used to pass the form data to the server.

- get is the default, used to pass the form data to the server. If no method attribute is given in the <form> tag, get will be used.
- The alternative technique is post.

Widgets

- Many of the commonly used controls are created with the inline tag, <input> which is used for text, passwords, checkboxes, radio buttons and the action buttons Reset, Submit and plain.

- The type attribute of <input> is required for all of the controls which specifies the kind of widget or controls being created.

1. Text : To create a box.

- A text control, referred to as a text box, creates a horizontal box into which the user can type a line of text.
    - Default size is 20 characters; it can be changed with the size attribute.
    - If more characters are entered than will fit, the box is scrolled.
    - If you do not want the box to be scrolled, include the maxlength attribute to specify the maximum number of characters that the browser will accept.

- Controls cannot directly appear in the form content – they must be placed in the block container, such as a paragraph.

```
<form action = " ">
 <p>
      <input type = "text" name = "Name" size = "25" maxlength = "25" />
  </p>
</form>
```

- If the contents of a text box should not be displayed when it is entered by the user, a password control can be used.

```
<input type = "password" name = "myPassword" size = "10" maxlength = "10" />
```

- For text and password textFor text and password text html

- A control and its label can be connected by putting the control and its label in the content of a label element.

```
<label>Phone: < input type = "text" name = "phone" />
</label>
```

2. Checkboxes - to collect multiple choice input

- Checkbox and radio controls are used to collect multiple-choice input from the user.
- A checkbox control is a single button that is either on or off.
- Every checkbox button requires a name and value attribute in its <input> tag.
- For form processing on the server, the name identifies the button and the value is its value. (if checked).
  - Every checkbox requires a value attribute, which is the widget's value in the form data when the checkbox is 'checked'
    - » A checkbox that is not 'checked' contributes no value to the form data.
  - By default, no checkbox is initially 'checked'.
  - To initialize a checkbox to 'checked', the checked attribute must be set to "checked".
  - Checkbox elements must also appear in label elements.

For checkboxes the source code For checkboxes the source code and the html is as shown.

```
Grocery Checklist
<form action = "">
  <p>
  <input type = "checkbox"  name ="groceries"
        value = "milk"  checked = "checked">
   Milk
  <input type = "checkbox"  name ="groceries"
        value = "bread">
   Bread
  <input type = "checkbox"  name = "groceries"
        value= "eggs">
   Eggs
  </p>
</form>
```

3. Radio Buttons

- Collections of checkboxes in which only one button can be ‘checked’ at a time
  - »Every button in a radio button group MUST have the same name.
  - »Every time a radio button is pressed, the button in the group that was previously on is turned off.
  - »The type value for radio buttons is radio.

  - •If no button in a radio button group is ‘pressed’, the browser often ‘presses’ the first one.

  - •A radio button can include a checked attribute, set to the value checked in the input tag of
    the button’s definition.

  - •The source code The source code and html are as shown.

```
Age Category
<form action = "">
  <p>
  <input type = "radio"  name = "age"
   value = "under20" checked = "checked"> 0-19
  <input type = "radio"  name = "age"
        value = "20-35"> 20-35
  <input type = "radio"  name = "age"
        value = "36-50"> 36-50
  <input type = "radio"  name = "age"
        value = "over50"> Over 50
  </p>
</form>
```

4. The <select> tag – To create menus

- Checkboxes and Radio buttons – methods for collecting multiple choice data from a user.
- If a number of possible choices is large – display becomes too large.

- A menu is selected with a <select> tag.
- There are two kinds of menus, those that behave like checkboxes(multiple menu items can be selected at a time) and those that behave like radio buttons (one menu item can be selected at a time).

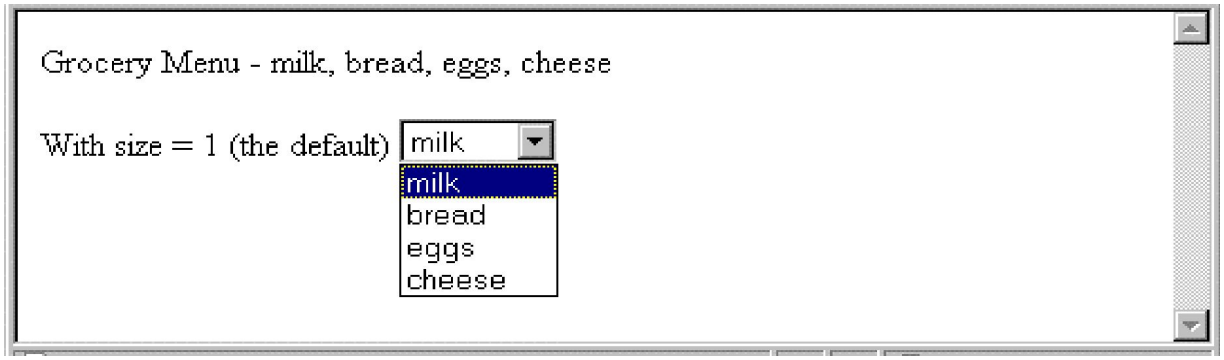
- The default option is the one related to radio buttons.
  - Menus that behave like checkboxes are specified by including the multiple attribute, which must be set to "multiple".
- The name attribute of <select> is required.
- The size attribute of <select> can be included to specify the number of menu items to be displayed (the default is 1).
  - If size is set to > 1 or if multiple is specified, the menu is displayed as a pop-up menu.

- Each of the menu items is specified with an <option> tag, nested in the select element.
- Content of option tag is the value of the menu item, just text.

- The <option> tag can include the selected attribute, which specifies that the item is preselected.
- The value assigned to selected is “selected”.

```
Grocery Menu - milk, bread, eggs, cheese
<form action = "">
  <p>
    With size = 1 (the default)
    <select name = "groceries">
      <option> milk </option>
      <option> bread </option>
      <option> eggs </option>
      <option> cheese </option>
    </select>
  </p>
</form>
```

- After clicking the menu:



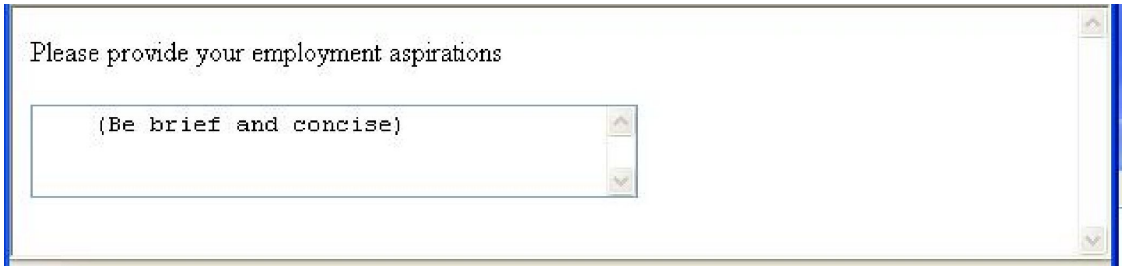
- After changing size to 2:

•The source code The source code and html are as shown.

5. Text areas - created with <textarea> tag

- A multiline text area can be created with the <textarea> tag.
  - Usually include the rows and cols attributes to specify the size of the text area.
  - Default text can be included as the content of <textarea>.
  - Scrolling is implicit if the area is overfilled.

```
Please provide your employment aspirations
<form action = "">
  <p>
  <textarea name = "aspirations"  rows = "3”
          cols = "40">
  (Be brief and concise)
  </textarea>
  </p>
</form>
```



6. Reset and Submit buttons

- The Reset button clears all of the controls in the form to their initial states.
- The Submit button has two actions:
  1. Encode the data of the form.
  2. Request that the server execute the server-resident program specified as the value of the action attribute of <form>.
     - A Submit button is required in every form.
     - Both are created with <input> tag.

<input type = "reset" value = "Reset Form">

<input type = "submit” value = "Submit Form">

Frames

- Frames are rectangular sections of the display window, each of which can display a different document.
- Because frames are no longer part of XHTML 1.1, you cannot validate a document that includes frames.

- Most common use of frames: Having a table of contents displayed in one frame and parts of main document displayed in another.
- The table of contents can include links that, when followed, lead to the targeted parts of the main document being displayed in the other frame.

- The <frameset> tag specifies the number of frames and their layout in the browser window.
  - A <frameset> element takes the place of <body> element in a document.
  - Cannot have both!
  - A <frameset> element must have either a rows attribute or a cols attribute, or both.

  - The rows attribute specifies the number of rows of frames that will occupy the window.
  - The possible values for rows and cols are numbers, percentages, and asterisks
    - A number value specifies the row height in pixels.
    - A percentage specifies the percentage of total window height for the row.

    - Example: <frameset rows = "200, 300, 400">
      This frameset will have three rows of frames.
      <frameset rows = "22%, 33%, 45%" >
      <frameset rows = "22%, 33%, *">
      If 2 asterisks are given, they get half of what remains as the height of the window.

    - The cols attribute specifies the number of columns of frames that will occupy the window.
    - If the window is to have six frames in three equal-height rows and two columns:
      <frameset rows = "33%, 33%, 33%" cols="25%, "*">

  - An asterisk after some other specification gives the remainder of the height of the window
  - Examples:

```
<frameset rows = "150, 200, 300">
<frameset rows = "25%, 50%, 25%">
<frameset rows = "50%, 20%, *" >
<frameset rows = "50%, 25%, 25%" cols = "40%, *">
```

- The <frame> tag specifies the content of a frame which can appear only within the content of a frameset element.

- The first <frame> tag in a <frameset> specifies the content of the first frame, etc.
  - Row-major order is used
  - Frame content is specified with the src attribute.
  - Without a src attribute, the frame will be empty (such a frame CANNOT be filled later).

- If <frameset> has fewer <frame> tags than frames, the extra frames are empty.

- Scrollbars are implicitly included if needed (they are needed if the specified document will not fit), attribute scrolling can be set to yes.
- If a name attribute is included, the content of the frame can be changed later (by selection of a link in some other frame that specifies that name).

- Note: the Frameset standard must be specified in the DOCTYPE declaration

- The source code The source code of frames.html consists of left frame with contents.html and right frame displays fruits.html.
- Source code Source code for contents.html.
- The contents.html document is a list of links to the fruit description documents. Each link must both give an href attribute for the document filename and a target attribute to specify the name of the frame in which the name of the document is to be displayed.

- Source code Source code for fruits.html.

- Source code Source code for apples.html.
- Source code Source code for bananas.html.
- Source code Source code for oranges.html.

- Nested frames - to divide the screen in more interesting ways, source code Nested frames - to divide the screen in more interesting ways, source code and html.

Syntactic Differences between HTML & XHTML

- *Case sensitivity:*

  In HTML, tag and attribute names are case insensitive.

  Eg: <form>, <FORM>, <Form> are equivalent.

  In XHTML, all tags and attribute names must be in lowercase.

- *Closing tags:*

  In HTML, closing tags may be omitted.

  Eg: Paragraph elements in HTML often do not have closing tags. The appearance of another opening tag is used to infer closing on previous paragraph tag.

  In XHTML, all elements must have closing tags. For elements that do not include content, a slash (preceeded by a space) can be included at the end of opening tag.

- *Quoted attribute values:*

  In HTML, attribute values must be quoted only if there are embedded special characters or whitespace characters.

  In XHTML, all attribute values must be double quoted, regardless of what characters are included in their value.

- *Explicit attribute values:*

  In HTML, some attribute values are implicit.

  Eg: If the border attribute appears in the <table> tag without a value, it specifies a default width border.

  In XHTML, attribute values have to be explicitly stated.

- *id and name attributes:*

  HTML markup often uses name attribute for elements.

  In XHTML, the use of id is encouraged and name is discouraged. But name is used in form elements for processing the data.

- *Element nesting:*

  Although HTML has rules against improper nesting of elements, they are not enforced.

  In XHTML, nesting rules are strictly enforced.

  All of the XHTML syntactic rules are checked by the W3C validation software.

Examples of nesting rules are:

1. An anchor element cannot contain another anchor element, and a form element cannot contain another form element.
2. If an element appears inside another element, the closing tag of the inner element must appear before the closing tag of the outer element.
3. Block elements cannot be nested in inline elements.
4. Text cannot be directly nested in body or form elements.
5. List elements cannot be directly nested in list elements.