WEB TECHNOLOGIES

Module2:

Introduction to HTML/XHTML : Origins and Evolution of HTML and XHTML, Basic Syntax of HTML, Standard HTML Document Structure, Basic Text Markup, Images, Hypertext Links, Lists, Tables, Forms,HTML5, Syntactic Differences between HTML and XHTML.

1.Origins and Evolution of HTML and XHTML

HTML stands for Hyper Text Markup Language .Hyper Text is text which contains links to other texts .It is a markup language consists of a set of markup tags .HTML uses markup tags to describe web pages .HTML markup tags are usually called HTML tags .HTML tags are keywords surrounded by angle brackets like <html> .HTML tags come in pairs like <b> and </b> .The first tag in a pair is the start tag, the second tag is the end tag .Start and end tags are also called opening tags and closing tags

XHTML stands for E**X**tensible **H**yper**T**ext **M**arkup **L**anguage.XHTML is almost identical to HTML 4.01.XHTML is a stricter and cleaner version of HTML 4.01.

1.1Versions of HTML and XHTML

Original intent of HTML: General layout of documents that could be displayed by a wide variety of computers.

HTML 1.0

- HTML 1.0 was the **first release of HTML** to the world
- The language was very limiting.

HTML 2.0

- HTML 2.0 included everything from the original 1.0 specifications
- HTML 2.0 was the **standard** for website design until **January 1997**
- Defined many **core HTML features** for the first time.

HTML 3.2 - 1997

- The **browser-specific tags** kept coming.
- First version developed and standardized exclusively by the W3C.

HTML 4.0 – 1997

- Introduced many new features and deprecated many older features.
- Support for HTML's new supporting presentational language, CSS.

HTML 4.01 - 1999

- A cleanup of 4.0,
- it faces 2 problems
  - it specifies loose syntax rules

- its specification does not define how a user agent (browser) is to recover when erraneous code is encountered.

XHTML 1.0 – 2000

- Redefinition of HTML4.01 using XML, instead of SGML& HTML with stronger syntactic rules
- XHTML 1.0 is three standards: Strict, Transitional and Frameset.
  - **Strict** standard requires all of the syntax rules of XHTML1.0 to be followed
  - **Transitional** standard allows deprecated features of XHTML1.0 to be included.
  - **Frameset** standard allows the collection of frame elements and attributes to be included although they have been deprecated.

XHTML 1.1 – 2001

- Modularization of XHTML 1.0 , drops some of the features of its predecessor such as frames.

HTML 5 – January 2008

- Latest specification of the HTML.
- HTML5 was published as a Working Draft by the W3C.

2.Basic Syntax of HTML

Fundamental synctactic units of HTML are called tags.Elements are defined by tags.

Tag format:

- Opening tag: <name>
- Closing tag: </name>

Most tags appear in pairs.The opening tag and its closing tag together specify a container for the *content* they enclose. Not all tags have content.If a tag has no content, its form is <name /> .The container and its content together are called an *element.*

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

<a href="default.html"> this is a link </a>

Comment form: <!-- … -->.Comments increases the readability of programs.Browsers ignore comments, unrecognizable tags, line breaks, multiple spaces, and tabs.Tags are suggestions to the browser.

3.Standard HTML Document Structure

- The first line of every HTML document is a DOCTYPE command.

<!DOCTYPE html>

An HTML document must include the four tags <html>, <head>, <title>, and <body>. The <html> tag identifies the root element of the document. HTML documents always have an <html> tag immediately following the DOCTYPE command, and they always end with the closing html tag, </html>.

The html element includes an attribute *lang* .Which specifies the language in which the document is written.

<html lang = “en”>

en means english . An html document consist of 2 parts: head and body.

The head element provides information about the document . The head element always contains 2 simple elements, a *title* element and *meta* element. The meta element is used to provide additional information about a document. The meta element has no content, all information is specified with attributes. The meta tag specifies the character set used to write the document.

<meta charset = “utf-8” />

The slash at the end of the tag specifies it has no closing tag,It has a combined opening and closing tag . The content of the title element is displayed by the browser in the browser window’s title bar. The body of a document provides content of the document .

```
<!DOCTYPE html>

<!- - File name and document purpose - ->

<html lang = “en”>

<head>

<title> A title for the document </title>

<meta charset = “utf-8” / >

</head>

<body>

……

</body>

</html>
```

4. Basic Text Markup

Describes how the text content of an XHTML document can be formatted with XHTML tags .

1. Paragraphs
2. Line Breaks
3. Preserving white space
4. Headings
5. Block Quotes
6. Font styles & size
7. Character Entities
8. Horizontal Rules
9. Meta Element

1.Paragraph Element

Text is normally placed in paragraph elements . 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 closing tag is required in XHTML, not in HTML. Also multiple spaces in the text are replaced by single space.

```
<p>

Mary

    had  a little

                lamb, its fleece    was white as snow.</p>
```

OUTPUT

Mary had a little lamb, its fleece was white as snow.

```
<!DOCTYPE html>

<!-- greet.hmtl

     A trivial document

     -->

<html lang = "en">

  <head> <title> Our first document </title>

<meta charset= "utf-8" />

  </head>

  <body>

    <p>

      Greetings from your Webmaster!

    </p>

  </body>

</html>
```

2.Line breaks

The effect of the <br /> tag is the same as that of <p>, except for the blank line.It can have no content therefore no closing tag!

```
<p>

Mary had a little lamb, <br /> its fleece was white as snow.

</p>
```

- Typical display of this text:

Mary had a little lamb,

its fleece was white as snow.

3.Preserving whitespaces

It is used to preserve white spaces in text.Text in a <pre> element preserves both spaces and line breaks. </pre> .It blocks the browser from eliminating multiple spaces

```
<p><pre>

Mary

         had a

                  little

                         lamb

</pre></p>
```

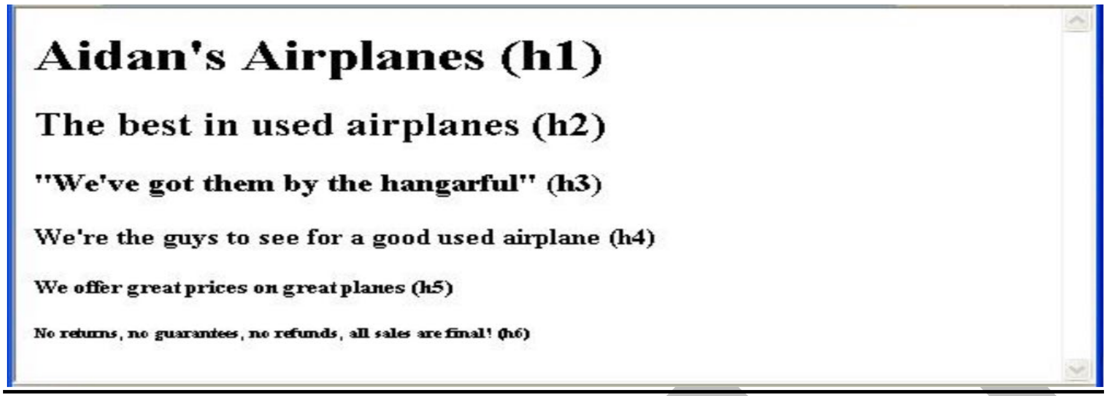
Display(output)

```
Mary

    had a

        little

           lamb
```

4. Headings

Six sizes, 1 - 6,are used for specifying heading. It is specified with <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.

```
 < !DOCTYPE  html >

<!-- headings.html

     An example to illustrate headings  -->

<html lang = “en”>

  <head> <title> Headings </title>

<meta charset = “utf-8” />

  </head>

  <body>

    <h1> Aidan’s Airplanes (h1) </h1>

    <h2> The best in used airplanes (h2) </h2>

    <h3> "We’ve got them by the hangarful" (h3)

    </h3>

    <h4> We’re the guys to see for a good used

         airplane (h4) </h4>
    <h5> We offer great prices on great planes

         (h5) </h5>

    <h6> No returns, no guarantees, no refunds,

         all sales are final! (h6) </h6>

  </body>

</html>
```



Aidan's Airplanes (h1)

The best in used airplanes (h2)

"We've got them by the hangarful" (h3)

We're the guys to see for a good used airplane (h4)

We offer great prices on great planes (h5)

No returns, no guarantees, no refunds, all sales are final! (h6)

5.Blockquotes

The Content of <blockquote> can be made to look different from the surrounding text .It set a block of text off from the normal flow and appearance of text.Browsers often indent, and sometimes italicize.

```
<!DOCTYPE html>

<!-- blockquote.html

An example to illustrate a blockquote -->

<html lang = "en">

<head><title> Blockquotes </title>

<meta charset= "utf-8"/>

</head>

<body>

<p>

Abraham Lincoln is generally regarded as one of the greatest presidents
of the U.S. His most famous speech was delivered in Gettysburg,
Pennsylvania, during the Civil War. This speech began with </p>

<blockquote><p>

"Fourscore and seven years ago our fathers brought forth on this
continent, a new nation, conceived in Liberty, and dedicated to the
proposition that all men are created equal."

 </p>

</blockquote>
```

```
<p>

Whatever one's opinion of Lincoln, no one can deny the enormous and
lasting effect he had on the U.S. </p>

</body></html>
```

OUTPUT

Abraham Lincoln is generally regarded as one of the greatest presidents of the U.S. His most famous speech was delivered in Gettysburg, Pennsylvania, during the Civil War. This speech began with

> "Fourscore and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal."

Whatever one's opinion of Lincoln, no one can deny the enormous and lasting effect he had on the U.S.

6.Font Styles and Sizes (can be nested)

- Boldface - <b>
- Italics - <i>
- Larger - <big>
- Smaller - <small>

```
The <big> sleet <big> in <big> <i> Crete

</i><br /> lies </big> completely </big>

in </big> the street
```

OUTPUT

The sleet in ***Crete***

lies completely in the street

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: x<sub>2</sub><sup>3</sup>

Display: x_2^3

There are few tags for fonts that are still in widespread use, called content based style tags.They are called content based because the tag indicates the particular kind of text that appear in their content.

- The emphasis tag- <em>=Most browsers use italics for such content
- The strong tag -<strong>= Browsers often set the content of strong elements in bold

- The <code> = used to specify a monospace font usually used for program code

cost = quantity * price

- tags are categorized into two

1**)Block 2)Inline**

- ***Block tags*** breaks the current line so that its content appears in a new line
- Eg: heading and blockquote tag
- ***Inline tags*** does not implicitly include a line break except <br />.The content of the inline tag appears on the current line
- Eg: <em> and <strong>
- Inline versus block elements
  - Block elements CANNOT be nested in inline elements.

<u>7.Character Entities</u>

- To include special characters

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)	&nbsp;	Non-breaking space

<u>8.Horizontal rules</u>

<hr />

Used to separate the parts of a document from each other making the document easier to read by placing horizontal line between them. This causes a line break and draws a line across the screen .

<u>9. The meta element</u>

The meta element (for search engines) Used to provide additional information about a document, with attributes, not content .Two attributes that are used to provide information are *name* and *content*

```
<meta name="Title"  content="World wide web"/>

<meta name="Author" content="Sebesta"/>

<meta name="keywords" content="html,xml,javascript,ajax,xhtml" />
```

Web search engines use the information provided with the meta element to categorize the web documents in their indices .So if the author of a document seeks widespread exposure for the document ,one or more meta elements are included to ensure that it will be found by at least some web searches .

5. Images

Image Formats:

1.GIF (Graphic Interchange Format)

8-bit color (256 different colors)

2.JPEG (Joint Photographic Experts Group)

It has 24-bit color representation (16 million different colors). Both use compression, but JPEG compression is better.

3.Portable Network Graphics (PNG)

It is relatively new. Should eventually replace both gif and jpeg

Images are inserted into a document with the **<img />** tag with the **src** attribute. Src means source of the image. The **alt** attribute means" Alternate Text" which is required by XHTML where the image cannot be loaded by the browser. Purpose of **alt** attribute is for non graphical browsers and browsers with images turned off.

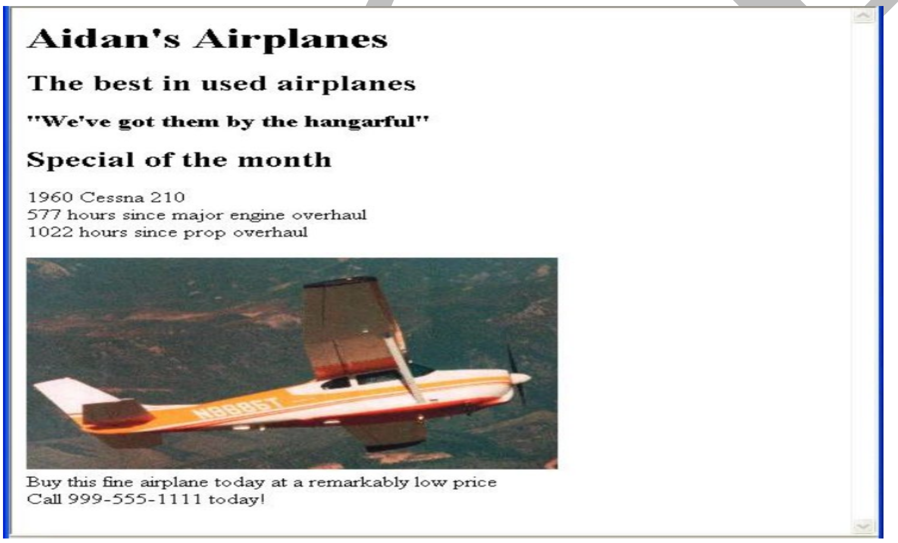
<img src = "icet.jpg" alt = "Picture of Icet " />

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

```
<!DOCTYPE html>

<!-- image.html

     An example to illustrate an image

     -->

<html lang = "en">

  <head> <title> Images </title>

     <meta charset="utf-8">

  </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>
```

OUTPUT



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!

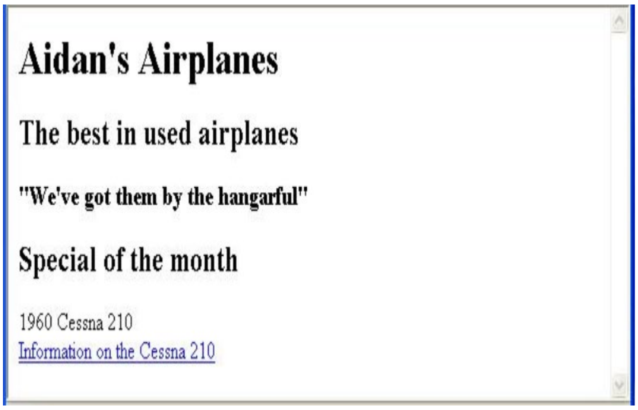
6. Hypertext Link

Link is the convenient way for the browser user to get from one document to any logically related document. A hypertext link acts as a pointer to some resource. The resource can be an HTML document anywhere on the web, or it may be another place in the same document or be a specific place (rather than the top) in some other document.

A link is specified with the href (*h*ypertext *ref*erence) attribute of <a> (the anchor tag).<a> tag is inline tag.

The content of <a> is the visual link in the document (clickable link). Links are implicitly rendered in a different color than the surrounding text. Sometimes they underlined .

```
<!DOCTYPE html>

<!-- link.html

     An example to illustrate a link

     -->

<html lang = “en”>

  <head>

<title> Links </title>

<meta charset=“UTF-8” />

  </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>
```



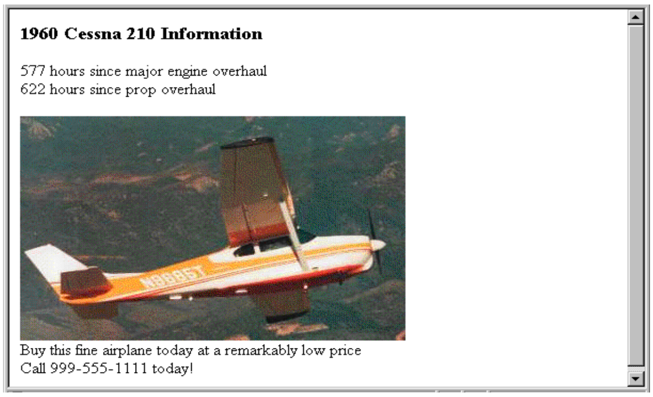
Aidan's Airplanes

The best in used airplanes

"We've got them by the hangarful"

Special of the month

1960 Cessna 210
Information on the Cessna 210



1960 Cessna 210 Information

577 hours since major engine overhaul
622 hours since prop overhaul

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

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 value of the id attribute must be unique within the document. The link to an id must be preceded by a pound sign (#); If the id is in the same document, this target could be

```
<a href = "#baskets"> What about baskets? </a>
```

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

```
<a href = "myAd.html#baskets”> Baskets </a>
```

Links can have images:

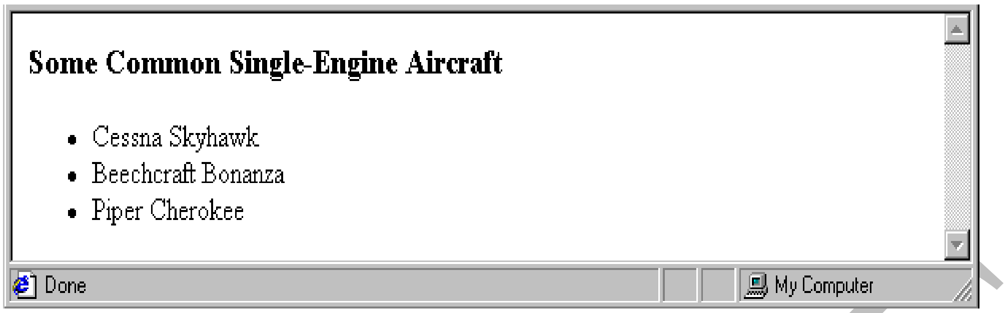
```
 <a href = "c210data.html“>
<img src = "smallplane.jpg" alt = "Small picture of an airplane " />
Info on C210 </a>
```

7. Lists

- **Unordered lists**

- Unordered list is represented using <ul> tag which is block tag.
- Each item in a list is specified with the content of the <li> tag.
- It can include nested lists.
- When displayed each item is implicitly preceded with a bullet.

```
<!DOCTYPE html>
<!- - unordered.html -->
<html lang=“en”>
<head>
<title> Unordered list </title>
<meta charset = “utf-8” />
</head>
<body>
<h3> Some Common Single-Engine Aircraft </h3>
 <ul>
   <li> Cessna Skyhawk </li>
   <li> Beechcraft Bonanza </li>
   <li> Piper Cherokee </li>

 </ul>

</body>

</html>
```



Some Common Single-Engine Aircraft

- Cessna Skyhawk
- Beechcraft Bonanza
- Piper Cherokee

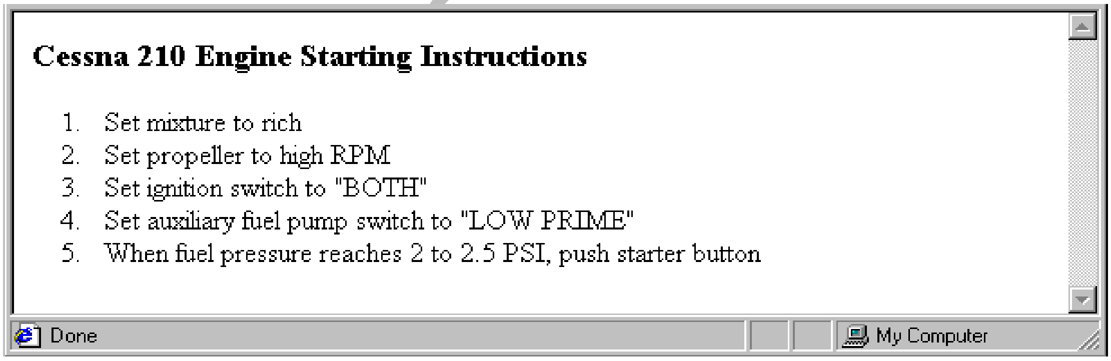
- **Ordered lists**

Ordered lists are those in which the order of items is important.Each item in the display is preceded by a sequence value

-The list is the content of the <ol> tag.

-The default sequential values are Arabic numerals beginning with 1.

```
<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>
```

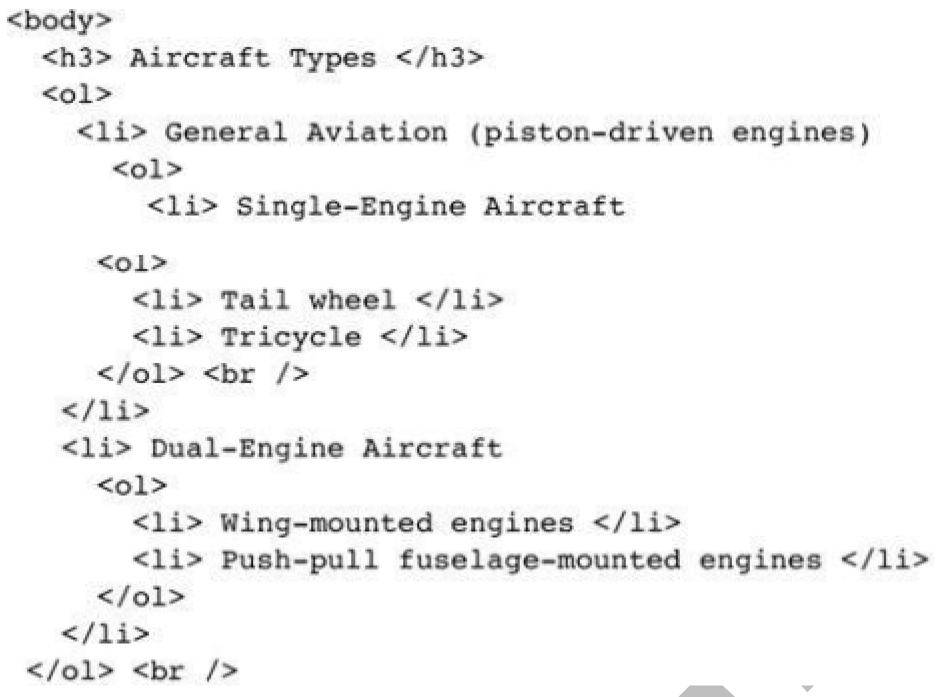


Cessna 210 Engine Starting Instructions

1. Set mixture to rich
2. Set propeller to high RPM
3. Set ignition switch to "BOTH"
4. Set auxiliary fuel pump switch to "LOW PRIME"
5. When fuel pressure reaches 2 to 2.5 PSI, push starter button

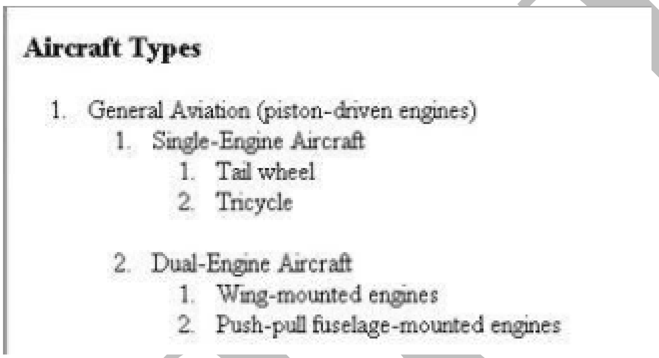
- **Nested lists**
  - Any type list can be nested inside any type list

- The nested list must be in a list item



```
<body>
  <h3> Aircraft Types </h3>
  <ol>
    <li> General Aviation (piston-driven engines)
      <ol>
        <li> Single-Engine Aircraft

     <ol>
       <li> Tail wheel </li>
       <li> Tricycle </li>
     </ol> <br />
   </li>
   <li> Dual-Engine Aircraft
     <ol>
       <li> Wing-mounted engines </li>
       <li> Push-pull fuselage-mounted engines </li>
     </ol>
   </li>
 </ol> <br />
```

</li></body>



Aircraft Types

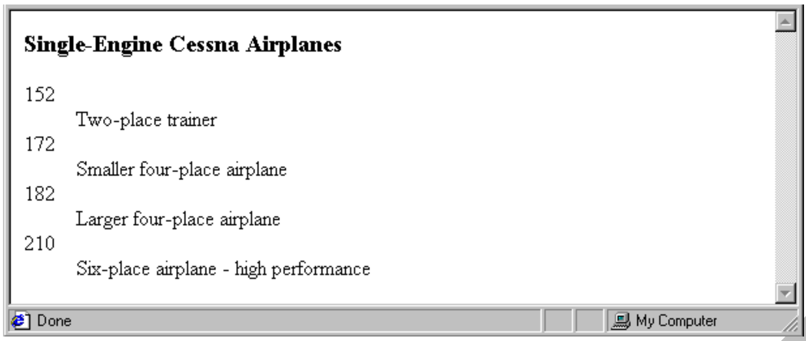
1. General Aviation (piston-driven engines)
    1. Single-Engine Aircraft
        1. Tail wheel
        2. Tricycle

    2. Dual-Engine Aircraft
        1. Wing-mounted engines
        2. Push-pull fuselage-mounted engines

- **Definition lists**

It is used for glossaries, etc. It represents set of terms and their definitions. List is the content of the <dl> tag.Terms being defined are the content of the <dt> tag. The definitions themselves are the content of the <dd> tag.

```
<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>
```



Single-Engine Cessna Airplanes

```
152
      Two-place trainer
172
      Smaller four-place airplane
182
      Larger four-place airplane
210
      Six-place airplane - high performance
```

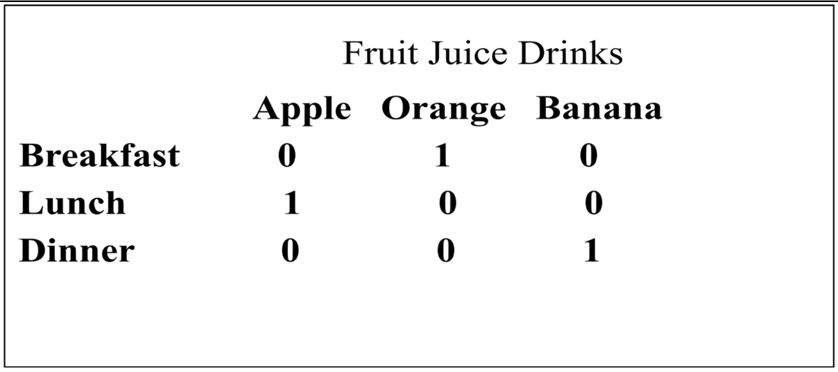
8. Tables

A table is a matrix of cells, each possibly having content. The cells can include almost any element .Some cells have row or column labels and some have data. The cells in the top row often contain columns labels , those in the left most column often contains row labels and most of the rest of the cells contain the data of the table. The content of a cell can be text, a heading, a horizontal rule , an image or a nested table. A table is specified as the content of a <table> tag. A border attribute in HTML 4.01 and XHTML 1.0 in the <table> tag specifies a border between the cells. This attribute is not present in HTML 5.The styles of the border and rules in a table are specified in HTML 5 with stylesheets. Tables are given titles with the <caption> tag, which can immediately follow <table> tag

- Each row of a table is specified as the content of a <tr> tag
- The row labels(headings) are specified as the content of a <th> tag
- The contents of a data cell is specified as the content of a <td> tag

```
<!DOCTYPE html>
<! - - table.html - ->
<html lang = “en”>
<head>
<title> A simple table </title>
<meta charset = “utf-8” />
</head>
<body>
<table>
```

```
   <caption> Fruit Juice Drinks </caption>
     <tr>
       <th> </th>
       <th> Apple </th>
       <th> Orange </th>
       <th> Banana </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>
<tr>
       <th> Dinner </th>
       <td> 0 </td>
       <td> 0 </td>
       <td> 1 </td>
     </tr>
   </table>
</body>
</html>
```



Fruit Juice Drinks

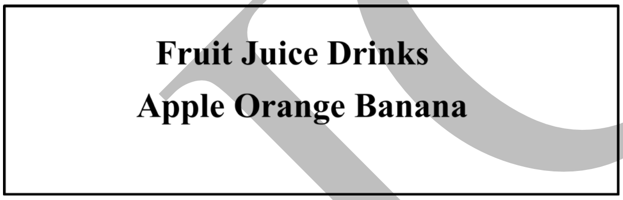
	Apple	Orange	Banana
Breakfast	**0**	**1**	**0**
Lunch	**1**	**0**	**0**
Dinner	**0**	**0**	**1**

<u>rowspan and colspan attribute</u>

- A table can have two levels of 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> Apple </th>

  <th> Orange </th>

  <th> Banana </th>

</tr>
```

OUTPUT



Fruit Juice Drinks

Apple Orange Banana

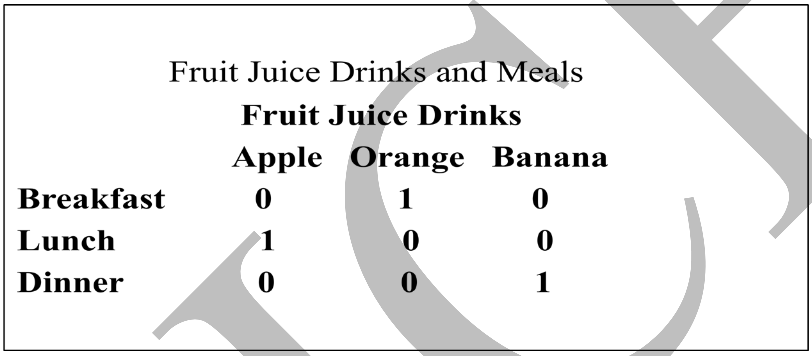
- If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using rowspan

```
<table border = "border">

  <tr>

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

    <th colspan = "3"> Fruit Juice Drinks

    </th>

  </tr>

  <tr>

    <th> Apple </th>
    <th> Orange </th>
    <th> Banana </th>
  </tr>
        …
</table>
```

Example

```
<!DOCTYPE html>
<! - - cellspan.html - ->
<html lang = “en”>
<head>
<title> A simple table </title>
<meta charset = “utf-8” />
</head>
<body>
<table>
   <caption> Fruit Juice Drinks and Meals </caption>
<tr>
    <td rowspan = "2"> </td>
    <th colspan = "3"> Fruit Juice Drinks  </th>
  </tr>
     <tr>
       <th> </th>
       <th> Apple </th>
       <th> Orange </th>
       <th> Banana </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>

<tr>

       <th> Dinner </th>

       <td> 0 </td>

       <td> 0 </td>

       <td> 1 </td>

     </tr>

   </table>

</body>

</html>
```



Fruit Juice Drinks and Meals

	Fruit Juice Drinks		
	Apple	**Orange**	**Banana**
Breakfast	**0**	**1**	**0**
Lunch	**1**	**0**	**0**
Dinner	**0**	**0**	**1**

Table Sections

Tables naturally occur in two and sometimes three parts: header, body, and footer. (Not all tables have a natural footer.) These three parts can be respectively denoted in HTML with the thead, tbody, and tfoot elements.

- The header includes the column labels, regardless of the number of levels in those labels.
- The body includes the data of the table, including the row labels.
- The footer, when it appears, sometimes has the column labels repeated after the body. In some tables, the footer contains totals for the columns of data above .

9. Forms

- A form is the usual way information is gotten from a browser to a server.
- HTML has tags to create a collection of objects that implement this information gathering
  - The objects are called *widgets or controls* (e.g., radio buttons ,checkboxes,text ,password , reset, submit)
  - All control tags are inline tags

  - Most controls are used to gather information from the user in the form of either text or button selections
  - Every form data requires submit button
- When the Submit button of a form is clicked, the form data is encoded and are sent to the server for processing

1. The form Element

All of the widgets, or components of a form are defined in the content of a <form> 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 possible techniques of transferring the form data to the server, *get* and *post*. The method attribute specifies how to send form-data to the server. Default is get.In both techniques the form data is coded into a text string when the user clicks the Submit button. This text string is often called *query string* .

2. The< input> tag

Many of the commonly used controls such as text ,passwords checkboxes, Reset, Submit ,radio buttons etc. are created with the <input> tag.

The type attribute of <input> specifies the kind of widget being created such as checkbox, text, password, radio button etc.

All the previously listed controls except Reset and submit also require a name attribute which becomes the name of the value of the control within the form data.

The controls for checkboxes and radio buttons require value attributes which initializes the value of the control.

Textboxes as well as most other control elements should be labeled.

Eg: `<label> Phone: <input type = “text”  name = “phone” /> </label>`

3.Text

Text control is often used to gather information from the user such as the user’s name or address.Creates a horizontal box for text input. Default size is 20; it can be changed with the size attribute.If more characters are entered than will fit, the box is scrolled (shifted) left.

```
<input type = "text" name = "Phone" size = “25“ />
```

If you don’t want to allow the user to type more characters than will fit, set maxlength, which causes excess input to be ignored

```
<input type = "text" name = "Phone” size = “25“ maxlength =“25” />
```

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

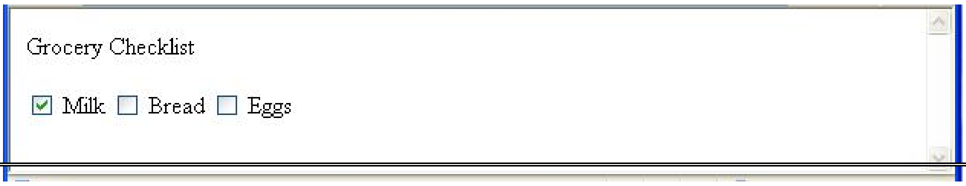
```
<input  type= “password” name = “myPassword” size = “10” maxlength = “10”
/>
```

In this case, regardless of what characters are typed into a password control, only bullets or asterisks are displayed by the browser.

4.Checkboxes

It is used to collect multiple choice input.A checkbox control is a single button that is either on/off (checked or not) .Every checkbox requires a *name* attribute and *value* attribute, which is the widget’s value in the form data when the checkbox is ‘checked’. A checkbox that is not ‘checked’ (off) contribute no value to the form data. To initialize a checkbox to ‘checked’, the checked attribute of checkbox must be set to "checked“. Check box should appear in the label elements.

```
<!DOCTYPE html>

<! - -checkbox.html - ->

<html lang = “en”>

<head>

<title> Checkboxes</title>

<meta charset = =“utf-8” />

</head>

<body>

<p>

Grocery Checklist

</p>

<form  action = "">

  <p>

 <label>  <input type = "checkbox"  name ="groceries"

         value = "milk"  checked = "checked‟ />    Milk </label>

  <label><input type = "checkbox"  name ="groceries"

         value = "bread‟ />   Bread </label>

 <label> <input type = "checkbox"  name = "groceries"

         value= "eggs‟ />  Eggs </label>

  </p>

</form>

</body>

</html>
```



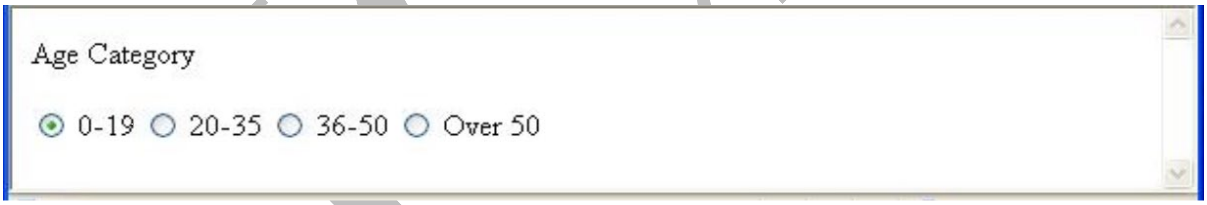
Grocery Checklist

☑ Milk ☐ Bread ☐ Eggs

5. Radio Buttons

Closely related to checkbox buttons except that only one radio button can be ‘checked’ at a time. Every time a radio button is pressed , the button in the group that was previously on turned off.The type value for the radio buttons is radio.Every button in a radio button group MUST have the same name.

```
<p>Age Category </p>

<form action = "">

  <p>

  <label><input type = "radio"  name = "age"

   value = "under20" checked = "checked‟ /> 0-19 </label>

 <label> <input type = "radio"  name = "age"

         value = "20-35‟ /> 20-35 </label>

<label>  <input type = "radio"  name = "age"

         value = "36-50‟ /> 36-50 </label>


<label><input type = "radio"  name = "age"

         value = "over50‟ /> Over 50 </label>

  </p>

</form>
```



Age Category

● 0-19 ○ 20-35 ○ 36-50 ○ Over 50

6.Menus

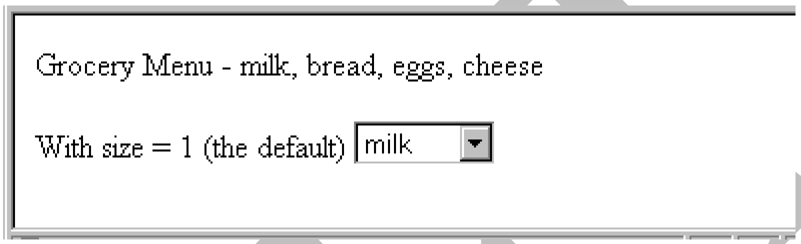
Menu is created with <select> tags.There are two kinds of menus, those that behave like checkboxes and those that behave like radio buttons (the default). A menu is specified with <select> tag. The *name* attribute of <select> is required.The *size* attribute of <select> can be included to specify the number of menu items to be displayed (the default is 1) in the <select> tag. If size is set to > 1 or if multiple is specified, the menu is displayed as a pop-up menu. If no *size* attribute is specified , the value 1 is used. Each item of a menu is specified with an <option> tag, whose pure text content is the value of the item.

```
<body>

<p>Grocery Menu - milk, bread, eggs, cheese</p>

<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>

</body>
```

OUTPUT

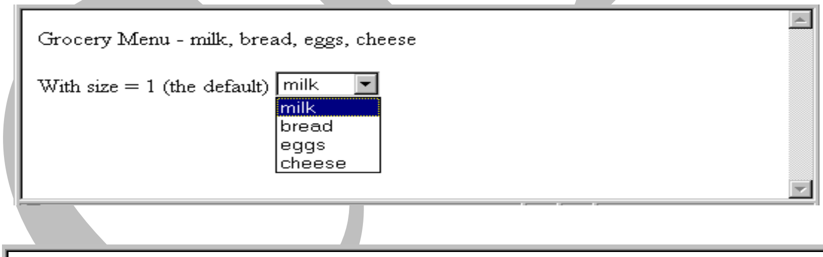
Default size of 1 :



Grocery Menu - milk, bread, eggs, cheese

With size = 1 (the default) milk

After clicking scroll arrow:

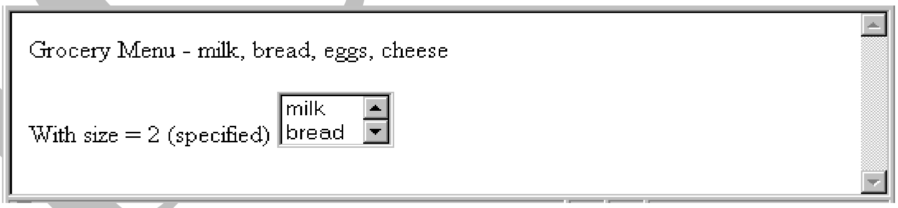


Grocery Menu - milk, bread, eggs, cheese

With size = 1 (the default) milk

milk
bread
eggs
cheese

After changing size to 2:



Grocery Menu - milk, bread, eggs, cheese

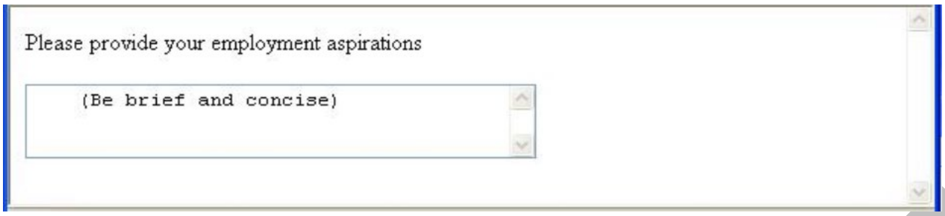
With size = 2 (specified) milk
bread

7.Text areas

Textarea is created with <textarea> tag. It is used when multiline text area is needed.It 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>.

Example

```
<body> <p>

Please provide your employment aspirations

<form action = "">

   <p>
   <textarea name = "aspirations"  rows = "3″ cols = "40">
   (Be brief and concise)
   </textarea>
   </p>
</form>
```



Please provide your employment aspirations

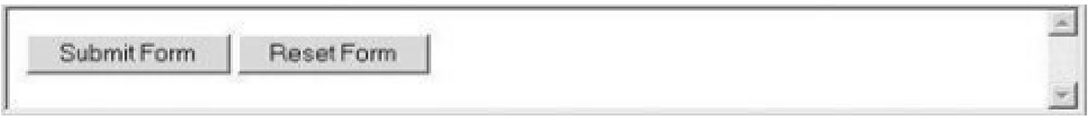
(Be brief and concise)

8. Reset and Submit buttons (The Action Buttons)

Both are created with <input> tag.

```
<input type = "reset"  value = "Reset Form /">
<input type = "submit"  value = "Submit Form /">
```

- Submit has two actions:
    1. Encode the data of the form and send to the server.
    2. Request that the server execute the server-resident program specified as the value of the action attribute of <form>
    3. A Submit button is required in every form
- Reset button is used to reset the form data to its initial value.



10.HTML 5

The audio Element

Prior to HTML5, a plug-in was required to play sound while a document was being displayed. Audio information is coded into digital streams with encoding algorithms. Audio encoding algorithms are called audio codecs – e.g., MP3, Vorbis and WAV. Coded audio data is stored in containers (like zip file).It is a way to pack data into a file .There are three different audio container types. e.g., Ogg, MP3, and Wav. The type of the container is indicated by the extension on the file’s name. Eg.Ogg container file has the .ogg filename extension.

- Vorbis codec is stored in Ogg containers

- MP3 codec is stored in MP3 containers

- Wav codec is stored in Wav containers

General syntax of audio element:

```
      <audio attributes>
         <source src = "filename 1" >
         ...
         <source src = "filename n" >
       Your browser does not support the audio element
       </audio>
```

Browser chooses the first audio file it can play and skips the content; if it cannot play any of the audio files that appear in the source elements , it displays content. Different browsers have different audio capabilities. The controls attribute in the audio element, which is set to controls", creates a start/stop button, a clock, a progress slider, total time of the file, and a volume slider .

Example

```
<!DOCTYPE html>

  <html lang = "en" >

    <head>     <title> Test audio element </title>

      <meta charset = "utf-8" />    </head>

    <body>     This is a test of the audio element

      <audio controls = "controls" >

        <source src = “test.ogg" />

        <source src = “test.wav" />

        <source src = “test.mp3" />

      Your browser does not support the audio element

      </audio>

    </body>

  </html>
```

The video Element

Prior to HTML5, there was no standard way to play video clips while a document was being displayed. Video information must be digitized into datafiles before it an be played by a browser. The algorithm used for encoding video are called video codec . Video codecs are stored in containers

Video codecs:

- H.264 (MPEG-4 AdvancedVideoCoding) – can be stored in MPEG-4 containers
- Theora – can be stored in any container
- VP8—can be stored in any container

Different browsers support different codecs.The width and height attributes set the size of the screen for the video in pixels. The autoplay attribute, set to "autoplay", specifies that the video should play as soon as it is ready. The preload attribute, set to "preload", specifies that the video should be loaded as soon as the document is loaded. The controls attribute, set to "controls", is like that of the audio element.

```
<!DOCTYPE html>

<html>
```

```
<body>

<video width="400‶ height =‶500″ autoplay=‶autoplay″ controls=‶controls″
preload=‶preload″>

  <source src="internet.mp4" type="video/mp4">

  <source src="internet.ogg‶  type="video/ogg">

  Your browser does not support HTML5 video.

</video>

</body>

</html>
```

Organization Elements

One of the deficiencies of HTML 4.01(and XHTML 1.0)is that it is difficult to organize displayed information in meaningful ways.HTML has a collection of new elements that assist in organizing documents and outlines of documents .

header Elements

- header element was designed to encapsulate the whole header of a document.

- *hgroup* element can be used to enclose the header and other information that precedes the body .hgroup – a container for header information

```
       <hgroup>

      <header>

       <h1> The Podunk Press </h1>

        <h2> ″All the news we can fit″ </h2>

      </header>

-- table of contents - -

 </hgroup>
```

Footer Elements

footer – a container for footer information such as author and copyright data

```
<footer>

  &copy; The Podunk Press, 2012

  <br />

   Editor in Chief: Squeak Martin   </footer>
```

The **section** Element – a container for section.eg.chapters of a book.

The **article** Element – a container for self-contained part of a document (from another source) eg:a newspaper article .

The **aside** Element – a container for tangential info(contents often placed in a sidebar.

The **nav** Element – navigation sections (list of links). The **nav** elements clearly marks the parts of a document that are used to get to other documents.

```
<!Doctype html>
<!- - organized.html- ->
<html lang= “en”>
<head>
<title>organization elements</title>
<meta charset = “utf-8” />
</head>
<body>
<hgroup>
<header>
<h1> The Podunk Press </h1>
<h3> “All the news we can fit” </h3>
</header>
<ol>
<li>Local News</li>
<li>National news</li>
<li>sports </li>
<li>Entertainment</li>
</ol>
</hgroup>
<p> -- Put the paper’s contents here - -  </p>
<footer>
&copy;The Podunk Press, 2012 <br />
Editor in Chief: Squeak Martin   </footer>
</body>
</html>
```

OUTPUT

The Podunk Press

“All the news we can fit”

1. Local News

2. National news
3. sports
4. Entertainment

-- Put the paper’s contents here - -

©The Podunk Press, 2012

Editor in Chief: Squeak Martin

The time Element

For putting a time stamp on a document. Two parts, text and machine-readable (datetime). datetime attribute (optional) is a machine-readable part. Date part: 4-digit year, a dash, 2-digit month, a dash, 2-digit day of the month (″2012-08-29”). Time (optional) format: T09:00. Text part is given as the content of time.

```
<time datetime = ″2012-08-29T09:00″>
    August 8, 2012 9:00 am
    </time>
```

11.Syntactic Differences between HTML and XHTML

- ***Case sensitivity***. In HTML, tag and attribute names are case insensitive, meaning that <FORM>, <form>, and <Form> are equivalent. In XHTML, all tag and attribute names must be all lowercase.

- ***Closing tags*****.** In HTML, closing tags may be omitted if the processing agent (usually a browser) can infer their presence. For example, in HTML, paragraph
  elements often do not have closing tags. The appearance of another opening paragraph tag is used to infer the closing tag on the previous paragraph.
  <p>
  During Spring, flowers are born. ...<p>
  During Fall, flowers die. .

- In XHTML, all elements must have closing tags

- ***Quoted attribute values***. In HTML, attribute values must be quoted only if there are embedded special characters or white-space characters. Numeric attribute values are rarely quoted in HTML.

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

- ***Explicit attribute values***. In HTML, some attribute values are implicit; that is, they need not be explicitly stated. For example, if the border attribute appears in a <table> tag without a value, it specifies a default-width border on the table. Thus,
  <table border>
  specifies a border with the default width.

In XHTML, in which such an attribute is assigned a string of the name of the
attribute:
<table border = “border”>

- ***id and name attributes***. HTML markup often uses the name attribute for elements. This attribute was deprecated for some elements in HTML 4.0, which added the id attribute to nearly all elements.

  In XHTML, the use of id is encouraged and the use of name is discouraged.

- ***Element nesting.*** Although HTML has rules against improper nesting of elements, they are not enforced. Examples of nesting rules are an anchor element cannot contain another anchor element etc

In XHTML, these nesting rules are strictly enforced.