# Web Programming

Lecture 1 – Introduction to XHTML and HTML5

## A Quick Overview of HTML and XHTML

- HTML is based on Standard Generalized Markup language (an ISO standard for text-formatting language).
- It was developed by Tim Berners-Lee sometime before late 1991.
- The most recent formally adopted version was 4.01
- XHTML 1.0 is a reformulation of HTML 4.01
- XHTML is being supplanted by HTML5

## HTML vs. XHTML

- HTML is easier to write with a more lax syntax. It is supported by all browsers and will most likely continue to be supported.
- XHTML forces the web designer to maintain discipline in how web documents are written.
- XHTML syntax can be checked by an XML browser or a validation tool (several exist online).

## A First XHTML Document

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
 "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<!-- the first.html
   A Basic first web page
  -->
<html xmlns = "http://www.w3.org/1999/xhtml">
<head><title>The First Web Page</title>
</head>
<body>
<p>Mary had a little lamb,<br />
little lamb, little lamb</p>
<p>his fleece was white as snow.</p>
</body>
</html>
```

## Beginning an XHTML Document

*Identifies this document as based on XML*

```
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE html PUBLIC "-//w3c//DTD XHTML 1.1//EN"
 http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>
```

*Identifies the document as conforming to XHTML 1.1*

```
<!-- the first.html
   A Basic first web page
  -->
```

*A comment explaining what the document is*

## Basic XHTML Structure

*Identifies the following as an HTML document*

*Identifies this as the document's head*

```
<html  … … >
<head>
<title> The title goes here</title>
</head>
```

*Identifies text for the title bar*

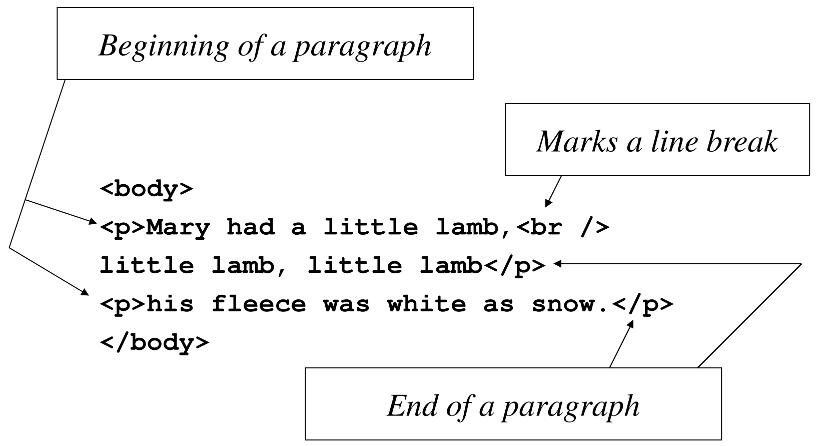
*End of the head*

```
<body>Body of the text goes here</body>
</html>
```

*Body of the document*

*End of the document*

## The Body of the XHTML Document



```
<body>
<p>Mary had a little lamb,<br />
little lamb, little lamb</p>
<p>his fleece was white as snow.</p>
</body>
```

## How The Document Looks in the Browser

Mary had a little lamb,
little lamb, little lamb

his fleece was white as snow.

## greet.html

```
<!DOCTYPE html>

<!-- greet.html
   A Basic first HTML5 web page
  -->
<html lang = "en">
  <head>
    <title> Our First HTML5 Document </title>
    <meta charset = "utf-8">
  </head>
```

*Document's language is English*

*Indicate attributes of document elements*
*Unicode Transformation format 8 bit*

```
  <body>
    <p>
      Greetings from your Webmaster!
    </p>
  </body>
</html>
```

## Line Breaks

- Line breaks allow text to be split between two lines without having to start a new paragraph.
- Example

```
<p>This text is split between <br />
two line.</p>
```

- This is displayed as:

```
This text is split between
two line.
```

## Headings

- There are six different sizes of headings:

```
<h1>Heading size 1</h1>
<h2>Heading size 2</h2>
<h3>Heading size 3</h3>
<h4>Heading size 4</h4>
<h5>Heading size 5</h5>
<h6>Heading size 6</h6>
```

## How headings might look in your browser

# Heading size 1

## Heading size 2

### Heading size 3

#### Heading size 4

##### Heading size 5

###### Heading size 6

## **align** Attribute

- Many tags support **align** attributes.
- The **align** attribute is placed in the opening tag before the **>**.
- Examples:

```
<h4 align="left">Left align</h4>
<h4 align="center">Centered</h4>
<h4 align="right">Right align</h4>
```

Left align

Centered

Right align

## Block Quotations

- Longer quotations usually are set apart with their own indentation. This makes it easier to read than if we placed it inside quotation marks.
- This is done in XHTML using the **`<blockquote>`** tag.

## **`blockquote`** Tag – An Example

```
<body>
<p>Abraham Lincoln is generally regarded as one of
  the greatest presidents in the history of the
  United States.  His most famous speech was
  delivered at the dedication of the Soldier’s
  Cemetary at Gettysburg. It began
<blockquote>
<p> "Fourscore and seven years ago, our forefathers
  brought forth upon this continent a new nation,
  conceived in Liberty and dedicated to the
  proposition that all men are created equal."</p>
</blockquote>
<p>Great speech, isn’t it?</p>
</body>
```

## Font Styles and Sizes

- There are occasions when text needs to be distinguished from the text surrounding it.
- We can emphasize or set off text in several ways:
  - use of boldface or italics
  - using different sizes
  - superscripts and subscripts
  - monospace fonts

## Inline and Block Tags

- XHTML tags are classified as either inline or block tags.
- Inline tags appear within a current line of text.
- Block tags cause the affected text to appear on a new line.
- Block tags cannot appear in the content of inline tags directly.

## Adding Emphasis

- The easiest way to emphasize text is by use of the inline tags **`<b>`** (bold face) and **`<i>`**(italics). However, these tags are deprecated.
- One can use **`<strong>`** for boldface (or an equivalent) and **`<em>`** for italics (or an equivalent).

## Other Font Style Tags

| Tag | Source | Output |
|---|---|---|
| `<em>` | `<em>emphasized text</em>` | *emphasized text* |
| `<strong>` | `<strong>strong text</strong>` | **Strong text** |
| `<dfn>` | `<dfn>definition</dfn>` | *definition* |
| `<code>` | `<code>computer code</code>` | computer code |
| `<samp>` | `<samp>sample computer code</samp>` | sample computer code |
| `<kbd>` | `<kbd>keyboard text</kbd>` | keyboard text |
| `<var>` | `<var>variable</var>` | *variable* |
| `<cite>` | `<cite>citation</cite>` | *citation* |

## `<big>` and `<small>`

- The `<big>` and `<small>` tags allow for relative sizing of text.
- Example

```
  Mary <big>had <big>a <big>little <big>lamb
</big> </big> </big> </big>
```

is displayed as

```
Mary had a little lamb
```

## `<small>`

- Example

```
  Mary <small>had <small>a <small>little
  <small>lamb</small> </small> </small>
  </small>
```

is displayed as

```
Mary had a little lamb
```

## Monospace Fonts

- A monospace font is a font where all the characters are the same width.
  - E.g., compare “w” with “i”.
- **`<tt>`** is used to specify monospace font. **`<big>`** and **`<small>`** can be used on text so specified.
- Example

```
<tt>Mary <big>had <big>a</big>little</big>
lamb.</tt>
```

## Character Entities

- XHTML and HTML provide a means of displaying characters that cannot be written as themselves within the XHTML document. They are defined as entities.
- These entities are replaced when the document is displayed by the specified character.

## Commonly Used Character Entities

| **Character** | **Entities** | **Meaning** |
|---|---|---|
| & | &amp;amp; | Ampersand |
| < | &amp;lt; | Less than |
| > | &amp;gt; | Greater than |
| " | &amp;quot; | Quotation Mark |
| ' | :&amp;apos; | Apostrophe (single quotes) |
| ¼ | &amp;frac14; | One quarter |
| ½ | &amp;frac12; | One half |
| ¾ | &amp;frac34; | Three quarters |
| ° | &amp;deg; | Degrees |
| (space) | &amp;nbsp; | Nonbreaking space |

## Horizontal Rules

- A horizontal rule can be used to separate two areas of text in the browser window.
- The tag is **`<hr />`**
- **`<hr />`** is a block tag, separating the text on either side. Its thickness, length and exact horizontal placement is determined by the browser.
- There is no closing tag.

## The **`meta`** element

- The **`meta`** element provides additional information about a document.
- Whatever information it provides through its attributes **`name`** and **`content`**.
- Example:

```
<meta name="keywords"
content="binary tree, linked lists,
stacks" />
```

## Images

- Images are an important additionalto any web page.
- Images are typically stored in a file in the same directory as the XHTML file that references them.

## Image Formats

- There are three image formats that are most commonly used on the World Wide Web:
  - GIF
  - JPEG
  - PNG
- The advantage in using standard file formats is that browsers will be able to open these files.

## GIF

- GIF (Graphic Interchange Format) was developed by CompuServe specifically for moving images.
- It uses 8-bit color representation, limiting pictures to 256 colors (a smaller number than you would imagine).

## JPEG

- JPEG (Joint Photographic Experts Group) uses 24-bit color representation, allowing for over 16 millions colors.
- JPEG uses a lossy compression method to save pictures, which leads to the loss of some color information.
- JPEG files will usually be smaller than Gif files of the same image.

## PNG

- PNG (Portable Network Graphics) was designed in 1996 to replace GIF when its patent holder, Unisys), suggested that it might start charging for its use.
- PNG has advantages over both GIF and JPEG in that it combines the best features of both.
- PNG supports transparency like GIF and a large number of colors.
- Because its compression algorithm is not lossy, its fles are bigger than JPEG files for the same image.

## `<image>` Tag

- The standard format for an image tag is:
  **`<image src="`**_FileName_**`"`**
  **`alt=`** " _A description of the picture_ " **`/>`**
- Example

```
<image src="cessna.jpg"
 alt= " Picture of a Cessna 210" />
```

- Attributes include height and width which can be expressed in pixels or as a percentage of its size.

```
<img src="rms.jpg" align="right"
height="240" alt="My Picture">
```

## image.html

```
<!DOCTYPE html>

<!-- image.html
   A Basic first HTML5 web page
  -->
<html lang = "en"> <!-- Document's language is
English -->
  <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 airplan today at a remarkably
      low price  <br />
      Call 999-555-1111 today!
    </p>
  </body>
</html>
```

## Hypertext Links

- A link points to a different document that should share some logical connection to the document with which it is linked.
- Links all use the inline anchor tag **`<a>`**, which connects to the source document (with the tag) to the target document (the one to which it is linked).
- There are three main types of links:
  - Links/ to other pages
  - Links within the page
  - Links to external sites

## Links to other pages

- The most basic link within a document is to another document within the same directory on the same site:

```
<a href="myotherpage.html">My Other
  Page</a>
```

- You can also link it to a document in another directory:

```
<a href="../someotherpage.html">Some
  Other Page</a>
```

## Links to other pages

- It is possible to set up a link that takes the browser from one part of a document to another.
- The anchor is set to mark the spot using

  **`<a name=`** " *SpotName*"**>**

- You can jump to that other spot in the document by writing

  **`<a href=`**"**#***SpotName*"**>**

## Links to External Sites

- Linking to pages elsewhere on the World Wide Web is a little more involved:

```
<p>This is a link to
<a
 href="http://www.myothersite.com">
 My Other Page</a></p>
```

## Lists

- Lists are a common way of organizing information, and for this reason, XHTML provides ways to set up lists.
- XHTML provides tags for three different types of lists:
  - Unordered Lists
  - Ordered Lists
  - Definition Lists

## Unordered Lists

- Unordered lists have their items marked with a “bullet”.
- They are marked off by the block tags **`<ul>`** and **`</ul>`**.
- Each individual item on the list is marked with the block tags **`<li>`** and **`</li>`**.

## Unordered Lists – An Example

```
<ul><li>This is the first item.</li>
<li>This is second item.</li>
<li>This is third item.

<ul><li>Subitem A</li>
<li>Subitem B</li
</ul></li>

<li>The fourth item</li>
</ul>
```

## How the Unordered List Appears

- `This is the first item.`
- `This is second item.`
- `This is third item.`
  - `Subitem A`
  - `Subitem B`
- `The fourth item`

## Ordered Lists

- Ordered lists have their items preceded by a number or letter.
- They are marked off by the block tags **`<ol>`** and **`</ol>`**.
- Each individual item on the list is marked with the block tags **`<li>`** and **`</li>`**.

## Ordered Lists – An Example

```
<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></ol><br />
<li>Commercial Aircraft</li></ol></li></ol>
```

## Ordered Lists – How It Appears

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


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


2.  Commercial Aircraft
```

## Definition Lists

- Definition lists are used to specify lists of terms and their definition.
- There are three tags that are used:
  - `<dl>` - definition list, which begins the list
  - `<dt>` - defined term, which indicates the term being defined
  - `<dd>` - definition description, which indicates the definition of the term.
- All three tags are block tags, with their corresponding closing tags.

## Definition Lists – An Example

```
<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>
```

## Definition List – How It Appears

**Single-engine Cessna Airplanes**

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

## Tables

- Tables are commonplace in documents, books and web pages.
- Tables provide a means of laying out content on a web page.
- A table is a matrix of rows and columns, with each entry called a ***cell***.

## Tables (continued)

- While some cells contain labels for the row or columns, most hold data. This data can be text, images or even nested tables.
- In HTML5, tables do not have lines between the rows or between the columns – they are added using CSS.

## Basic Table Tags

- **`<table>`** marks the beginning of the table; **`</table>`** marks the end.
- Many **`<table>`** tags contain the attribute **`border`**, which indicates the thickness of the border in pixels.
- There is frequently a **`<caption>`** tag which follows the table tag, which contains the table's caption.

## Table Content Tags

- Table content is specified one row at a time.
- **`<tr>`**… **`</tr>`** - specifies content for one row.
- **`<th>`**… **`</th>`** - specifies the content for a heading in one cell of the table.
- **`<td>`**… **`</td>`** - specified data contained in one cell of the table.

## Tables – An Example

```
<table border="border">

<caption>Fruit Juice Drinks</caption>

<tr> <th /> <th>Apple</th> <th>Orange</th>
  <th>Strawberry</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>
```

## How the Table Appears

Fruit Juice Drinks

| | **Apple** | **Orange** | **Strawberry** |
|---|---|---|---|
| **Breakfast** | 0 | 1 | 0 |
| **Lunch** | 1 | 0 | 0 |
| **Dinner** | 0 | 0 | 1 |

## **`rowspan`** and **`colspan`** Attributes

- There are occasions when we will want information in a table to extend over more than one cell.
- The **`rowspan`** attribute specifies that the cell’s contents will extend over more than one row.
- The **`colspan`** attribute specifies that the cell’s contents will extend over more than one column.

## `rowspan` – An Example

```
<table border="border">

<caption>Fruit Juice Drinks</caption>

<tr> <th /> <th>Apple</th> <th>Orange</th>
  <th>Banana/th> </tr>

<tr> <th>Breakfast</th> <td>0</td> <td>1</td>
<td rowspan="3">Yes, we have no bananas0</td> </tr>

<tr> <th>Lunch</th> <td>1</td> <td>0</td> </tr>

<tr> <th>Dinner</th> <td>0</td> <td>0</td> </tr>
</table>
```

## `rowspan` - How the Table Appears

Fruit Juice Drinks

| | Apple | Orange | Banana |
|---|---|---|---|
| **Breakfast** | 0 | 1 | Yes, we have no bananas |
| **Lunch** | 1 | 0 | |
| **Dinner** | 1 | 0 | |

## `colspan` – An Example

```
<table border="border">

<caption>Fruit Juice Drinks</caption>

<tr> <th /> <th>Apple</th> <th>Orange</th>
   <th>Strawberry</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 rowspan= "3">No juice with
   dinner</td> </tr>
</table>
```

## `colspan` – How the Table Appears

Fruit Juice Drinks

| | Apple | Orange | Strawberry |
|---|---|---|---|
| **Breakfast** | 0 | 1 | 0 |
| **Lunch** | 1 | 0 | 0 |
| **Dinner** | No juice with dinner | | |

## **`align`** and **`valign`** Attributes

- Most elements in an XHTML document have default alignment.
  - Headings and paragraphs are left-aligned.
  - Table headings are center-aligned, while other entries are left-aligned.
- Many of these elements will allow the use of the align attribute, which changes the alignment to left, center or right depending on the programmer's choice.

## **`align`** and **`valign`** Attributes

- The **`align`** attribute can be used within the **`<tr>`** to realign data in the entire row.
- The **`align`** attribute can also be used with the **`<th>`** and **`<td>`** tags to realign data within the cell.
- Simiarly, the **`valign`** attribute can be used to vertically align a row or an individual cell.

## **`align`** and **`valign`** – An Example

```
<table border="border">
<caption>Fruit Juice Drinks</caption>

<tr> <th /> <th>Apple</th> <th>Orange</th>
  <th>Banana</th> </tr>

<tr> <th>Breakfast</th> <td align="center">0</td>
  <td>1</td>
<td valign="top" rowspan="3">Yes, we have no
  bananas</td> </tr>

<tr> <th>Lunch and <br />lunch some more<br />
 and lunch still until</th> <td
  valign="bottom">1</td>
 <td>0</td> </tr>

<tr> <th>Dinner</th> <td>0</td> <td>0</td> </tr>
</table>
```

## **`rowspan`** and **`colspan`** – How They Appear

Fruit Juice Drinks

|  | **Apple** | **Orange** | **Banana** |
|---|---|---|---|
| **Breakfast** | 0 | 1 | Yes, we have no bananas |
| **Lunch and lunch some more and lunch still until** | 1 | 0 |  |
| **Dinner** | 1 | 0 |  |

## **`cellpadding`** and **`cellspacing`**

- **`cellpadding`** is an attribute of the table tag that specifies the spacing between the cell's contents and its inner wall.
- **`cellspacing`** is an attribute that specifies the spacing between cells within the table.

## **`cellpadding`** and **`cellspacing`** – An example

```
<p><strong>Table 1 (space = 10,
padding = 30)</strong> </p>
<table border="5" cellspacing="10" cellpadding="30">

<tr> <td>Small spacing</td> <td>Large padding</td>
  </tr>
</table>

<p><strong>Table 2 (space = 30,
padding = 10)</strong> </p>

<table border="5" cellspacing="30" cellpadding="10">
<tr> <td>Large spacing</td> <td>Small padding</td>
  </tr>
</table>
```

## **`cellpadding`** and **`cellspacing`** – How It Appears

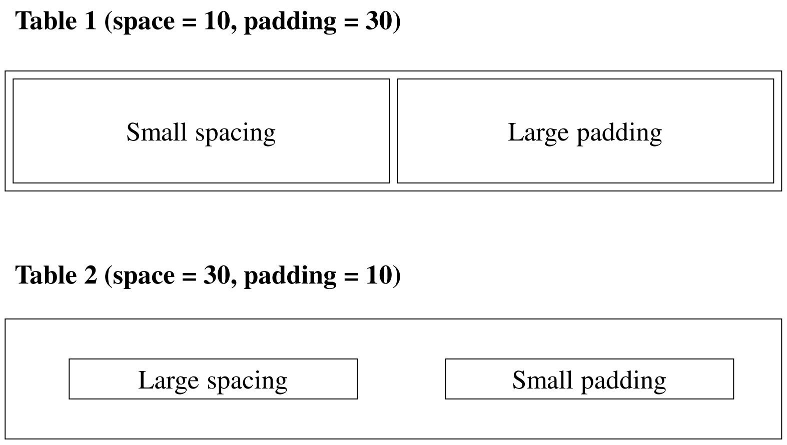


**Table 1 (space = 10, padding = 30)**

| | |
|---|---|
| Small spacing | Large padding |

**Table 2 (space = 30, padding = 10)**

| | |
|---|---|
| Large spacing | Small padding |

## Table Sections

- Table sections can include header, body, and footer, and they are indicated by the tags: **`<thead>`**, **`<tbody>`**, and **`<tfoot>`**
- If a document has multiple **`<tbody>`** elements, they are separated by thicker horizontal lines

## Use of Tables

- In the past, tables were used to align elements in rows and columns – general layout
- That use of tables is now frowned upon
- Use Cascading Style Sheets to place elements in rows and columns – general layout
- Use tables only when the information is naturally tabular

## Forms

- HTML forms are modeled on paper forms that people can fill out when providing information.
- HTML uses tags to specify objects through which information is inputted. These objects are commonly called ***controls*** or ***widgets***.
- The values entered using these controls are called the ***form data***.

## Form Controls

- Form controls include:
  - Single- and multiple-line text collections
  - Checkboxes
  - Radio buttons
  - Menus
- Most controls are used to gather information. Each control can have a value. Taken together, these are called ***form data***.

## The **`<form>`** Tag

- All the components of a form appear within the **`<form>`** tag, which is a block tag.
- While there are several attributes that it may have, its only required attribute is action, which specifies the URL of the application on the web server to be called when the user clicks submit.
- For the moment we will set action as an empty string, although we could use **`"mailto:someone@somewhere.com "`**

## **`<input>`** Tab

- Most of the commonly used controls are specified using the **`<input>`** tab. These include:
  - Text box
  - Password box
  - Checkboxes
  - Radio buttons
  - Submit and Reset buttons
- While there are a variety of attributes that may be used, one attribute that is always required is **`type`**, which specifies which type of control.

## **`text`**

- A text control creates a horizontal box in which the user can insert text.
- While the default size is usually 20 characters, this is not always the case, so it may be a good idea to set it explicitly.
- A maximum length may be specified if you wish to limit the number of characters in the text input.
- Since there are no restriction on which characters can be entered, the application using the form should do its own validity checking.

## `text` - Examples

```
<form action="">
<p> <input type = "text" name="Name" size="25" />
</p>
</form>
<! This will scroll if you enter more than 25 characters >

<form action="">
<p> <input type = "text" name="Name" size="25"
maxlength="25" />
</p>
</form>
<! This will not scroll - If you enter more than 25
   characters you will not be able to enter more >
```

## `password`

- This works like a text control, except the text that you type will only show up as a series of astericks.
- Example

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

## `checkbox`

- Checkboxes are used to collect multiple-choice responses from the user.
- If the box is checked, the value assocaited with the boxes name is assigned to its value attribute.

## `checkbox` – An Example

```
<form action="">
<p>Grocery Checklist</p>
<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>
```

Grocery Checklist

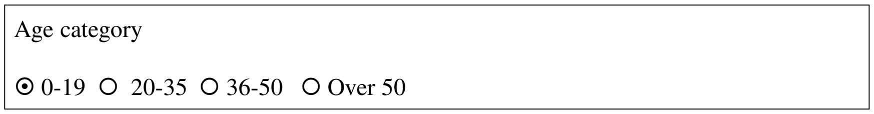
☑ Milk ☐ Bread ☐ Eggs

## **`radio`**

- Unlike checkbox, where any combination of boxes may be checked (or even none may be checked), radio buttons allow for one and only one button to be selected.

## `radio` – An Example

```
<form action = "">
<p>Age category</p>
<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>
```



Age category

◉ 0-19 ○ 20-35 ○ 36-50 ○ Over 50

## **`<select>`** Tag

- If there are too many options, it may make more sense to use a menu (or dropdown box). This is achieved in an XHTML form by using the **`<select>`** tag.

## **`select`** – An Example

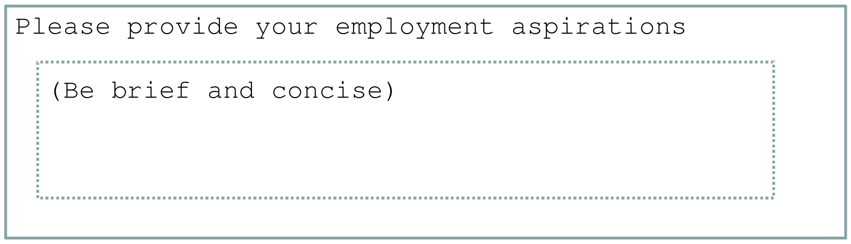
```
<p>Grocery Menu - milk, bread, eggs,
  cheese</p>
<form action="">
<p>With size=1 (the default)
<select name="groceries">
<option>milk</option>
<option>bread</option>
<option>eggs</option>
<option>cheese</option>
</select>
```

## **`<textarea>`** Tag

- A textarea is a multiline text box.
- There is no limit in length and scrolling is assumed to happen both horizontally and vertically.
- We usually specify the number of rows and columns to ensure enough space.

## **`<textarea>`** – An Example

```
<p>Please provide your employment aspirations</p>

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



```
Please provide your employment aspirations

  (Be brief and concise)
```

## Submit And Reset Buttons

- The Reset button clears all of the controls in the form to their initial states.
- The Submit button encodes the form data and sends it to the server, and requests that the program named in the action attribute be executed using the encoded data.
- Every form needs a Submit button.

## Reset and Submit Buttons – An Example

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

## A Complete Form

```
<!DOCTYPE html>
<!-- popcorn.html
      This describes popcorn sales form page
  -->

<html lang = "en">
  <head> <title> Popcorn Sales Form </title>
    <meta charset = "utf-8" />
  </head>

<body>
    <h2> Welcome to Millenium Gymnastics Booster
  Club Popcorn Sales
    </h2>
```

```
    <!-- the next line gives the address of the CGI
         program -->
    <form action = "">

    <table>
      <!-- Text widgets for name and address -->
       <tr>
         <td> Buyer's Name: </td>
         <td> <input type = "text" name = "name"
                     size= "30"> </td>
       </tr>

       <tr>
         <td> Street Address: </td>
         <td> <input type = "text" name = "street"
                                   size= "30"> </td>
       </tr>
       <tr>
         <td> City, State,Zip: </td>
         <td> <input type = "text" name = "city"
                                   size= "30"> </td>
       </tr>
    </table>
    <br />

    <table border = "border">
    <!-- First, the columns headings -->
       <tr>
         <th> Product Name </th>
         <th> Price </th>
         <th> Quantity </th>
       </tr>
```

```
      <tr>
        <th> Unpopped Popcorn (1 lb.) </th>
        <td> $3.00 </td>
        <td> <input type = "text" name = "unpop"
                            size = "2" />
        </td>
      </tr>
      <tr>
        <th> Caramel Popcorn (2 lb. cannister)</th>
        <td> $3.50 </td>
        <td> <input type = "text" name = "caramel"
                                size = "2" /> </td>
      </tr>
```

```
          <tr>
            <th> Caramel Nut Popcorn (2 lb. cannister)
            </th>
            <td> $4.50 </td>
            <td> <input type = "text"
                        name = "caramelnut"
                        size = "2" />
            </td>
          </tr>
          <tr>
            <th> Toffey Nut Popcorn (2 lb. cannister)
            </th>
            <td> $5.00 </td>
            <td> <input type = "text‟
                        name = "toffetnut"
                        size = "2" /> </td>
          </tr>
```

```
        </table>
        <br />
        <!-- The radio buttons for the payment
             method -->
          <h3> Payment Method: </h3>
          <p>
            <label>
              <input type = "radio" name = "payment"
                 value = "visa" checked = "checked" />
             Visa<br />
            </label>

            <label>
              <input type = "radio" name = "payment"
               value = "mc" />  Master Card<br />
            </label>
            <label>
              <input type = "radio" name = "payment"
               value = "discover" /> Discover<br />
            </label>
            <label>
              <input type = "radio" name = "payment"
                 value = "check" />     Check <br />
            </label>
          </p>
       <!-- The submit and reset buttons -->
          <p>
            <input type = "submit"
                        value = "Submit Order" />
            <input type = "reset"
                        value = "Clear Order Form" />
          </p>
       </form>
```

```
  </body>
</html>
```

## Frames

- A browser window can display more than one document at a time if you divide into rectangular areas called ***frames***.
- Frames are typically used to organize the display in the browser window.
- A common usage is to have a frame for a title (and/or a menu) and another frame for the actual content.
- Frames are not part of XHTML version 1.1. Among their problems is lack of support by older browsers and they are inaccessible for the blind.

## **`<frameset>`** Tag

- The frames and their layout on the screen are specified using **`<frameset>`**.
- **`<frameset>`** must include the attribute **`rows`** or **`cols`**, and may even have both.
- Examples

```
<frameset rows = "200, 300, 400">
<frameset rows = "22%, 33%, 45%">
<frameset rows = "22%, 33%, *">
<frameset rows = "33%, 33%, 33%"
          cols = "25%, *>
```

## Frame layout

| | |
|---|---|
| *Content of Frame 1* | *Content of Frame 2* |
| *Content of Frame 3* | *Content of Frame 4* |
| *Content of Frame 5* | *Content of Frame 6* |

## **`<frame>`** Tag

- The <frame> tag specifies the content of a frame, associating it with a particular source file.
- There may also be a name attribute, which makes it possible to change the frame's content, usually through a hypertext link.
- Example:

```
<frame src = "apples.html" name = apple>
```

## frames.html

```
<!DOCTYPE html>

<!-- frames.html
   An example to illustrate frames
  -->

<html lang = "en">
  <head> <title> Frames </title>
    <meta charset = "utf-8" />
  </head>

  <frameset cols = "20%, 80%">
    <frame src = "contents.html" />
    <frame src = "fruits.html" name = "descriptions" />
  </frameset>
</html>
```

## contents.html

```
<!DOCTYPE html>

<!-- contents.html
   The contents of the first frame of frames.html,
   which is the table of contents for the second
   frame
  -->
<html>
  <head> <title> Frames </title>
  <meta charset = "utf-8" />
  </head>
```

```
<body>
    <h4> Fruits </h4>
    <ul>
      <li><a href = "apples.html"
                  target = "descriptions">
          apples </a>
      </li>
      <li><a href = "bananas.html"
                  target = "descriptions">
          bananas </a>
      </li>
      <li><a href = "oranges.html"
                  target = "descriptions">
          oranges </a>
      </li>

    </ul>
  </body>
</html>
```

## fruits.html

```
<!DOCTYPE html>

<!-- fruits.html
  The initial contents of the second frame
  of frames.html - a general description of fruit
  -->

<html>
  <head> <title> General information on fruits
  </title>
 <meta charset = "utf-8" />
  </head>
```

```
<body>
    <p>
  A fruit is the mature ovary in a flowering plant.
  Fruit is classified by several characteristics,
  the
  most important being the number of ovaries
  included.
  If only a single ovary is included, it is called a
  simple fruit.
    </p>

    </ul>
  </body>
</html>
```

## bananas.html

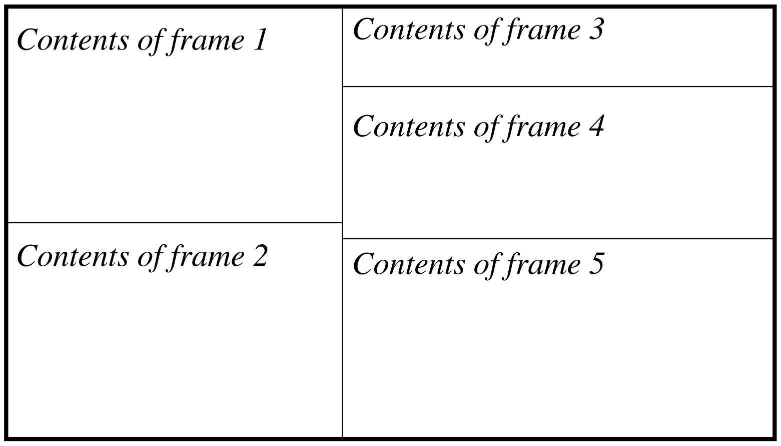
```
<!DOCTYPE html>

<!-- fruits.html
  The initial contents of the second frame
  of frames.html - a general description of fruit
  -->

<html>
  <head> <title> Bananas</title>
    <meta charset = "utf-8" />
  </head>
  <body>
    <p>
  Banana is the common name for tropical herbs of
  the genus Musa, family Musaceae, as well as for
  their fruit.  Bananas plants are native to
  Southeast Asia.
    </p>

    </ul>
  </body>
</html>
```

## nested_frames.html

```
<!DOCTYPE html>

<!-- nested_frames.html
  An example to illustrate nested frames
  -->

<html xmlns = "http://www.w3.org/1999/xhtml">
  <head> <title> Frames </title>
     <meta charset = "utf-8" />
  </head>
```

```
   <frameset cols = "40%, *">
     <frameset rows = "50%, *">
       <frame src = "frame1.html" />
       <frame src = "frame2.html" />
     </frameset>

     <frameset rows = "20%, 35%, *">
       <frame src = "frame3.html" />
       <frame src = "frame4.html" />
       <frame src = "frame5.html" />
     </frameset>

   </frameset>
 </html>
```

## Layout of **`nexted_frames.html`**



## New Tags in HTML5

- HTML5 introduced several new tags, whose purpose is to allow the inclusion of multimedia elements without needing plug-ins.
- These tags are ignored by browsers that do not support HTML5.
- These elements include:
  - **`<audio>`**
  - **`<video>`**
  - **`<time>`**

## Including audio in HTML5

- Audio information is coded into digital streams with encoding algorithms called ***audio codecs*** (***co***der-***dec***oder). The most common codecs include MP3, Vorbis and Wav.
- Coded audio data is packages in containers. An audio container can contain MP3 or Vorbis coded audio.
- The three container types are Ogg, MP3 and Wav.

## Audio Containers

- Ogg container files have the **`.ogg`** extension and contain Vorbis coded audio data.
- MP3 container files have the **`.mp3`** extension and contain MP3 codec audio data.
- Wav containers have the extension **`.wav`** and contain Wav codec audio data.

## The **`<audio>`** Element

- The basic syntax for the <audio> tag is:

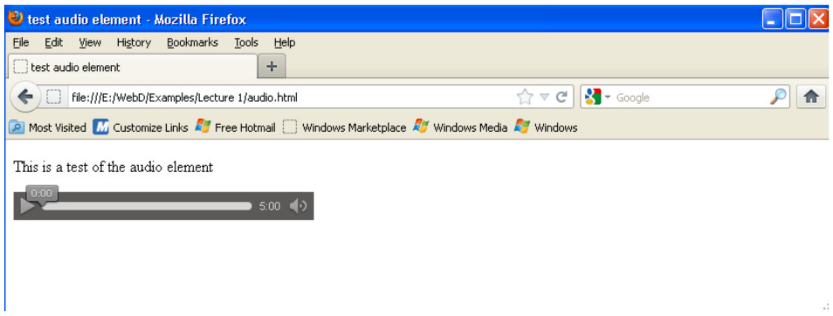
```
<audio attributes>
  <source src = "filename1">
  <source src= "filename2">
   …
  Your browser does not support the
  audio element
</audio>
```

## <audio.html>

```
<!DOCTYPE html>
<!-- audio.html
     test the audio element
     -->

<html lang = "en">
  <head>
    <title> test audio element </title>
    <meta charset = "utf-8" />
  </head>
```

```
  <body>
    <p>This is a test of the audio element</p>
    <audio controls = "controls" >
      <source src = "Live.mp3" />
      <source src = "Live.ogg" />
      Your browser does not support the audio
      element
    </audio>
  </body>
</html>
```

## Running **`audio.html`**


test audio element - Mozilla Firefox
File Edit View History Bookmarks Tools Help
test audio element
file:///E:/WebD/Examples/Lecture 1/audio.html
Google
Most Visited Customize Links Free Hotmail Windows Marketplace Windows Media Windows
0:00
5:00

This is a test of the audio element

## Browser Support for **`<audio>`**

- The **`<audio>`** tag is supported in Internet Explorer 9, Firefox, Opera, Chrome, and Safari.

## Browser Support for Audio File Formats

| Browser | OGG Vorbis | MP3 | WAV |
|---|---|---|---|
| FireFox 3.6+ | √✓ | | ✓ |
| Safari 5+ | | ✓ | ✓ |
| Chrome 6 | ✓ | ✓ | |
| Opera 10.5+ | ✓ | | ✓ |
| Internet Explorer 9 | | ✓ | ✓ |

## Attributes for **`<audio>`**

| Attribute | Value | Description |
|---|---|---|
| **`autoplay`** | **`autoplay`** | Specifies that the audio will start playing as soon as it is ready |
| **`controls`** | **`controls`** | Specifies that audio controls should be displayed (such as a play/pause button etc). |
| **`loop`** | **`loop`** | Specifies that the audio will start over again, every time it is finished |
| **`preload`** | **`auto`** **`metadata`** **`none`** | Specifies if and how the author thinks the audio should be loaded when the page loads |
| **`src`** | *URL* | Specifies the URL of the audio file |

## Values for `preload`

| **Value** | **Description** |
|---|---|
| `auto` | The browser should load the entire audio file when the page loads |
| `metadata` | The browser should load only metadata when the page loads |
| `none` | The browser should NOT load the audio file when the page loads |

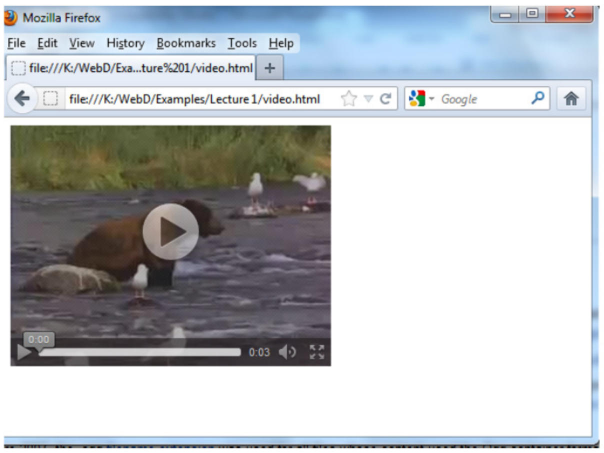
## Including video in HTML5

- Video information is coded into digital streams by ***video codecs***, and the most common codecs include MPEG-4 (**`.mp4`**), Flash Video (**`.flv`**), Ogg (**`.ogg`** or **`.ogv`**) WebM (**`.webm`**) and Audi Video Interleave (**`.avi`**).

## video.html

```
<!DOCTYPE html>
<html>
  <body>

    <video width="320" height="240"
controls="controls">
      <source src="movie.mp4" type="video/mp4" />
      <source src="movie.ogg" type="video/ogg" />
      Your browser does not support the video tag.
    </video>

  </body>
</html>
```

## Running **video.html**


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///K:/WebD/Exa...ture%201/video.html
file:///K:/WebD/Examples/Lecture 1/video.html
Google
0:00
0:03

## Organization Elements

- HTML5 includes new elements that make it easier to organize displayed information.
- These are intended to encapsulate information for organizational purposes; these tags can also be used with Cascading Style Sheets to provide consistent uniform formatting.
- These include:
  - **`<header>`**
  - **`<hgroup>`**
  - **`<footer>`**

## `<header>`

- If a header consists of just on ephrase, it could be a <h1> element. But frequently headers contain more information than can fit in a single element.
- Example:

```
<header>
  <h1> The Podunk Press </h1>
  <h2> "All the news that we can
          fit" </h2>
</header>
```

## `<hgroup>`

- **`<hgroup>`** can be used to contain a larger block of information that precedes the body of a document, such as a table of contents.

```
<hgroup>
  <header>
    <h1> The Podunk Press </h1>
    <h2> "All the news that we can
            fit" </h2>
  </header>
 -- Table of contents -
 </hgroup>
```

## `<footer>`

- The **`<footer>`** element is intended for encapsulating footer information in a document, such as author and copyright information:

```
<footer>
&copy; The Podunk Press, 2012
<br />
Editor In Chief, Squeak Martin
</footer>
```

## organized.html

```
<!DOCTYPE html>
<!-- organized.html
     An example to illustrate organization elements
     of HTML5
     -->
<html lang="en">
  <head>
    <title> Organizational elements </title>
    <meta charset = "utf-8" />
  </head>
```

```
  <body>
    <hgroup>
      <header>
        <h1> The Podunk Press </h1>
        <h2> "All the new that we can fit" </h2>
      </header>

      <ol>
       <li> Local news </li>
       <li> National news </li>
       <li> Sports </li>
       <li> Entertainment </li>
      </ol>
    </hgroup>
    <p>
      -- Put the paper's content here --
    </p>

    <footer>
      &copy; The Podunk Press, 2012
      <br />
      Editor in Chief: Squeak Martin
    </footer>
  </body>
</html>
```

## Syntax differences between HTML and XHTML

- Although XHTML is based on HTML, and they are quite similar, there are several differences:
  - Case sensitivity
  - Closing tags
  - Quoted attribute values
  - Explicit attribute values
  - id and name attributes
  - Element nesting

## Case Sensitivity

- In HTML, **`<BODY>`**, **`<Body>`** and **`<body>`** are all valid and interchangeable.
- In XHTML, only lower case is valid for tag name and attributes.

## Closing tags

- Certain closing tags may be left out in HTML if their presence can be inferred by the browser.

```
<p>One paragraph.
<p>Another paragraph implies the other
  one is closed.
```

- In XHTML, closing tags are required. In tags that do not include contents, it is sufficient to include the slash (`/`) at the end of the tag.

```
This text goes <br /> on another line.
```

## Quoted attribute values

- In HTML, attribute values do not always need quotations marks around; it is necessary when the value include special characters.
- In XHTML, all attribute values need quotation marks around them.

## Explicit attribute values

- In HTML, some attributes are implicit, i.e., if you do not give a value, there is a default value that will be used.

```
<table border>
```

- In XHTML, an value MUST be assigned explicitly to the attribute

```
<table border = "border">
```

- This also applies to **`checked`**, **`multiple`** and **`selected`**.

## id and name attributes

- Some HTML tags use the **`name`** attribute; this was deprecated for some elements in HTML 4.0.
- **`name`** was replaced for virtually all tags in HTML 4.0 by **`id`**.
- XHTML greatly discourages the use **`name`** and encourages the use of **`id`**.

## Element nesting

- HTML may have rules against nesting, but they are not always enforced.
- Examples

```
<a href=""> Some text <a href="">more text<a/>
<b><i>Bold and italics</b></i>
```

- XHTML enforces the rules regarding nesting.