CP 476
Internet Computing

Instructor: Masoomeh Rudafshani

Agenda

- HTML History
- HTML Syntax and Rules

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
- A table is specified as the content of a <table> tag
- In HTML5, tables do not have lines between the rows or between the columns
  - We can add those with Cascading Style Sheets, as will be discussed in Chapter 3
- Tables are given titles with the <caption> tag, which can immediately follow <table>

Tables

- Each row of a table is specified as the content of a <tr> tag
- The row 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
- Example
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/table.html
- 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
- If the rows have labels and there is a spanning column label, the upper left corner must be made larger, using rowspan
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/cell_span.html

Tables

- Table Sections are header, body, and footer, which are the following HTML elements:
  - **thead, tbody, and tfoot**
- If a document has multiple tbody elements, they are separated by thicker horizontal lines

Tables

- Uses 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

- A form is the usual way information is gotten from a browser user to a server
- HTML has tags to create a collection of objects that implement this information gathering
  - The objects are called widgets or controls or components
- When the Submit button of a form is clicked, the form's values are sent to the server for processing
- 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 that is to be called when the Submit button is clicked
  - XHTML requires it; HTML does not
- If the form has no action, the value of action is the empty string

Forms

- The **method** attribute of **<form>** specifies one of the two possible techniques of transferring the form data to the server, get and post
  - The default is get
  - get and post are discussed later in the course
- Widgets
  - Many are created with the **<input>** tag
  - The type attribute of <input> specifies the kind of widget being created

Forms: Widgets

- **Text**
  - 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
  - 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 = "12" />
```

  - Widgets should be placed in label elements

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

</label>
```

Forms: Widgets

- **Password** – just like text except asterisks are displayed, rather than the input characters
- **Checkboxes** - to collect multiple choice input
  - 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"
    - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/checkbox.html

Form: Widgets

- 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
  - If no button in a radio button group is ‘pressed’, the browser often ‘presses’ the first one
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/radio.html

Forms: Widgets

- The <select> tag
  - There are two kinds of menus, those that behave like checkboxes and those that behave like radio buttons (the default)
    - 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 item of a menu is specified with an **<option>** tag, whose pure text content (no tags) is the value of the item
  - An <option> tag can include the selected attribute, which when assigned "selected" specifies that the item is preselected
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/menu.html

Forms: Widgets

- Text areas - created with **<textarea>**
  - 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
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/textarea.html

Forms: Widgets

- Action buttons
  - Both are created with <input>

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

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

  - Submit has two actions:
    - Encode the data of the form
    - 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
- https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/popcorn.html

Forms

https://www.w3schools.com/code/tryit.asp?filename=FZ9YKHO1BPAQ

1) One input is missing
2) Add an option of blue to the menu

The **audio** element

- Prior to HTML5, a plug-in was required to play sound while a document was being displayed
- Audio encoding algorithms are called audio codecs – e.g., MP3, Vorbis
- Coded audio data is stored in containers—e.g., Ogg, MP3, and Wav
  - file name extension indicates the container, not the audio code
- Vorbis code is stored in Ogg containers
- MP3 code is stored in MP3 containers
- Wav code is stored in Wav containers

The <audio> element

- General syntax

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

- Browser chooses the first audio file it can play and skips the content
  - if none, it displays the text content
- Different browsers have different audio capabilities
- The controls attribute, 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:
  - https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/audio.html

The <video> element

- Prior to HTML5, there was no standard way to play video clips while a document was being displayed
- Video codecs are stored in containers
- Video codecs:
  - H.264 (MPEG-4 AVC) – can be stored in MPEG-4
  - Theora – can be stored in any container
  - VP8—can be stored in any container
- Different browsers support different codecs

The <video> element

- The **width** and height **attributes** set the screen size
- 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
- https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/testvideo.html

Iframe

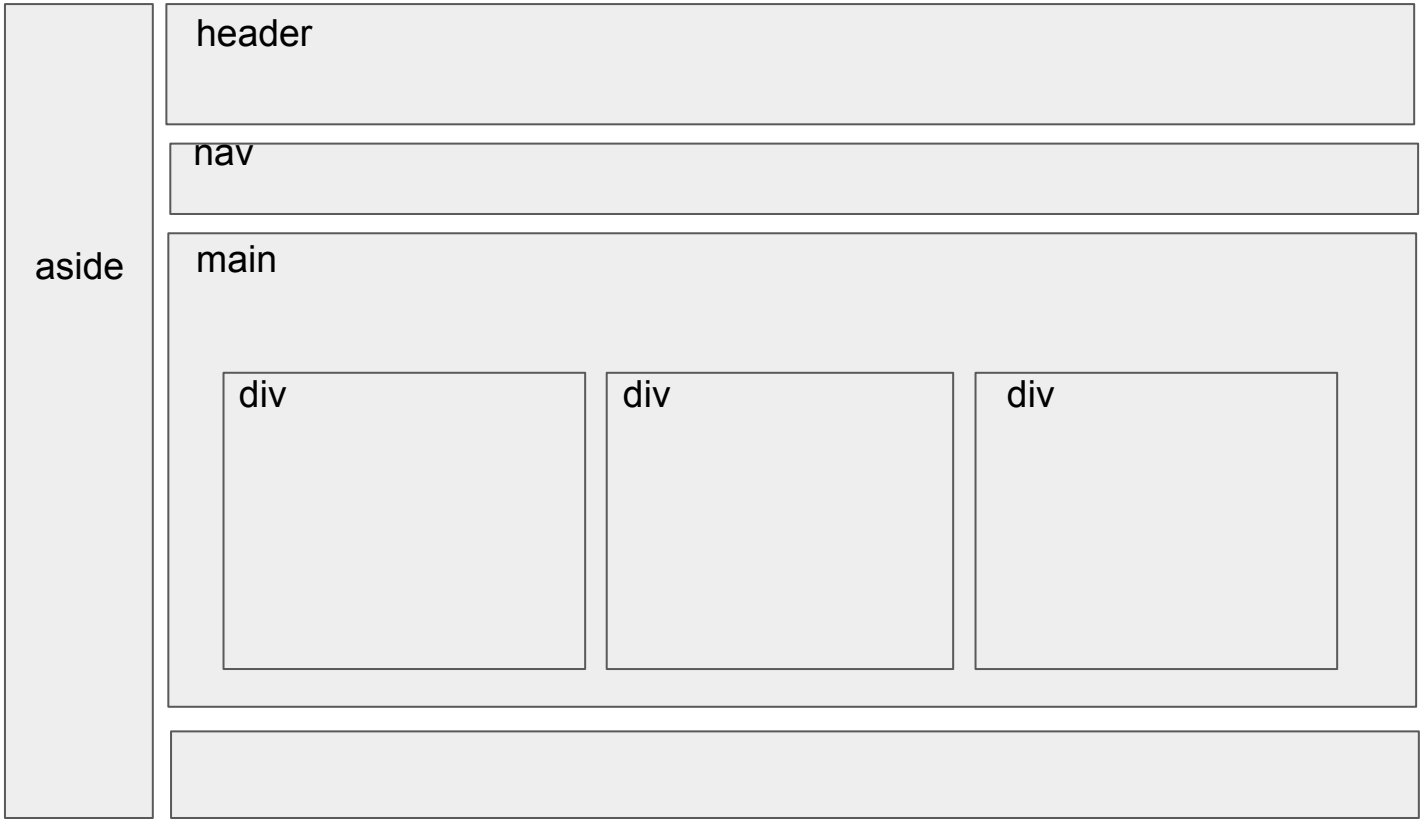
Div Element

- Defines a division or section in an HTML document.
-
- Can contain other HTML elements, including other div elements
  - To style them

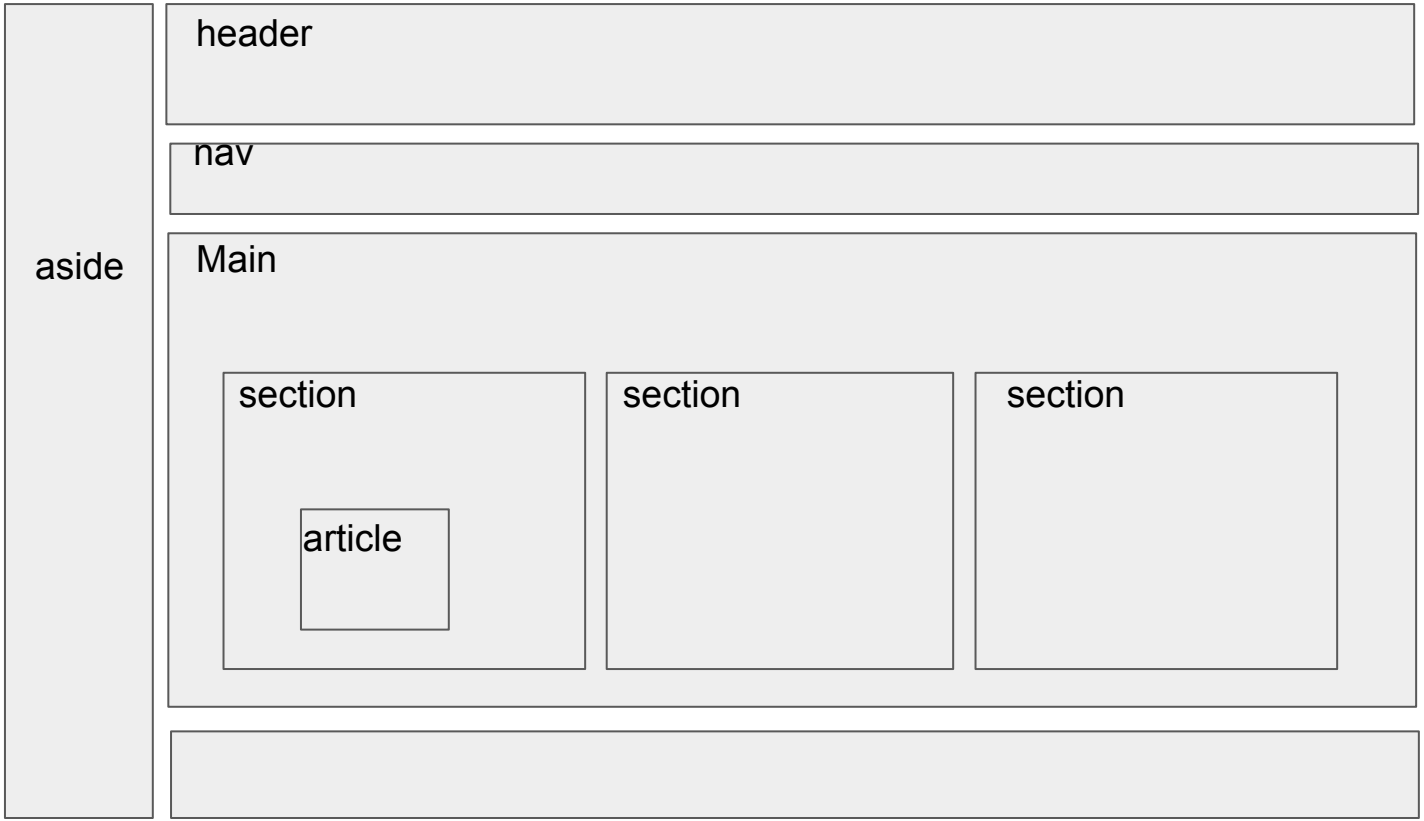
Organization Elements

- Header Elements
  - hgroup – a container for header information
- The nav Element – navigation sections (list of links)
- The section Element – a container for sections
- The article Element – a container for self-contained part of a document
  - Forum post, Blog post, News story, Comment
- The aside Element: a container for content aside from the page content
  - The aside content should be related to the surrounding content
- Footer Elements
  - footer – a container for footer information
- https://hopper.wlu.ca/~mrudafshani/cp476/examples/textBookExamples/organized.html

HTML5 organizational elements



HTML5 Organization Elements



The **time** Element

- For putting a time stamp on a document
- It has two parts: text and machine-readable part
  - datetime attribute (optional)
    - the 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
    - is given as the content of time
    - The two parts need not specify the same date
- Deficiencies:
  - Dates prior to the Christian era are not possible
  - No approximations

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

Chrome Developer Tools

Inspect a page