HTML – Part 2

Making nested lists

```
……................
<ul>
<li>
<h3>USA</h3>
<ol>
    <li>Tigger</li>
    <li>Tiger</li>
    <li>Max</li>
    <li>Smokey</li>
    <li>Sam</li>
</ol>
</li>
<li>
<h3>Australia</h3>
<ol>
        <li>Oscar</li>
        <li>Max</li>
        <li>Tiger</li>
        <li>Sam</li>
        <li>Misty</li>
</ol>
</li>
</ul>
```

Building Tables -- data that fits best in a tabular format

- <table></table> -- used to indent and space your code carefully so you can see the structure of the table in the code.
- By default (in most browsers, anyway), tables don’t show their borders.
- If you want to see basic table borders, you can turn on the table’s border attribute.
- <table border = "1">

Tags in Tables

HTML Tags	Descriptions
<table>	Define a table
<tr>	Define a table row
<td>	Define a table cell
<th>	Define a table header cell
<thead>	Define a group of a table header
<tbody>	Define a group of a table body
<tfooter>	Define a group of a table footer

Adding first row

- Each row is indicated by a <tr></tr> pair.
- Inside the <tr></tr> set, you need some table data.
- The first row often consists of *table headers.* These special cells are formatted differently to indicate that they're labels, rather than data.
- the table headers between the <th> and </th> elements. The contents appear in the order they're defined.

```
<tr>
        <th> Slno </th>
        <th> Regno </th>
        <th>  Name </th>
</tr>
```

Making data rows

- data rows are just like the heading row, except they use <td></td> pairs

```
<tr>
    <td> 1</td>
    <td> 19CSE901 </td>
    <td> Ram </td>
</tr>
```

Example

```
<h1>A Basic Table</h1>
<h2>HTML Super Heroes</h2>
<table border = "1">
<tr>
    <th>Hero</th>
    <th>Power</th>
    <th>Nemesis</th>
</tr>
<tr>
    <td>The HTMLator</td>
    <td>Standards compliance</td>
    <td>Sloppy Code Boy</td>
</tr>
<tr>
    <td>Captain CSS</td>
    <td>Super-layout</td>
    <td>Lord Deprecated</td>
</tr>
<tr>
    <td>Browser Woman</td>
    <td>Mega-Compatibility</td>
    <td>Ugly Code Monster</td>
</tr>
</table>
```

Spanning rows and columns

- To make cells larger than the default is two special attributes: rowspan and colspan.

```
<tr>
    <th>Morning</th>
    <td colspan = "2">Design traps</td>
    <td colspan = "3">Improve Hideout</td>
</tr>
```

HTML Tables

- Tables represent tabular data
  - A table consists of one or several rows
  - Each row has one or more columns
- Tables comprised of several core tags:

  `<table></table>`: begin / end the table

  `<tr></tr>`: create a table row

  `<td></td>`: create tabular data (cell)
- Tables should not be used for layout. Use CSS floats and

Styling your table data

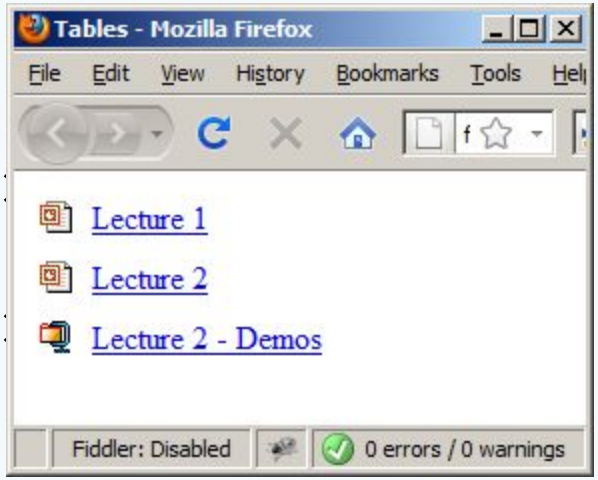
- Table Border:
- <table style="border: solid 1px #aaa999;">
- Birder for cells
- <th style="border: solid 1px #aaa999;">No.</th>
- <td style="border: solid 1px #aaa999;">1</td>

Or

```
<style>
    table {
      border: solid 1px #aaa999;
    }
    table {
          background-color: aqua;    --- Back color of table is changed
     }
     table tr th {   background-color: #808000; } – Header color is changed
    table tr th {
      border: solid 1px #aaa999; --  Border color changed
    }
    table tr td {
      border: solid 1px #aaa999;
    }
  </style>
```

Simple HTML Tables – Example (2)

```
<table cellspacing="0" cellpadding="5">
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture1.ppt">Lecture 1</a></td>
  </tr>
  <tr>
    <td><img src="ppt.gif"></td>
    <td><a href="lecture2.ppt">Lecture 2</a></td>
  </tr>
  <tr>
    <td><img src="zip.gif"></td
    <td><a href="lecture2-demos
      Lecture 2 - Demos</a></td
  </tr>
</table>
```



HTML Tables

- Table rows split into three semantic sections: header, body and footer
  - `<thead>` denotes table header and contains `<th>` elements, instead of `<td>` elements
  - `<tbody>` denotes collection of table rows that contain the very data
  - `<tfoot>` denotes table footer but comes BEFORE the `<tbody>` tag
  - `<colgroup>` and `<col>` define columns (most often used to set column widths)

Complete HTML Table: Example

```
<table>
<colgroup>
  <col style="width:100px" /><col />
</colgroup>
<thead>
  <tr><th>Column 1</th><th>Column 2</th></tr>
</thead>
<tfoot>
  <tr><td>Footer 1</td></tr>
</tfoot>
<tbody>
  <tr><td>Cell 1.1</td><td>Cell 1.2</td></tr>
  <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
</tbody>
```

columns

header

th

footer

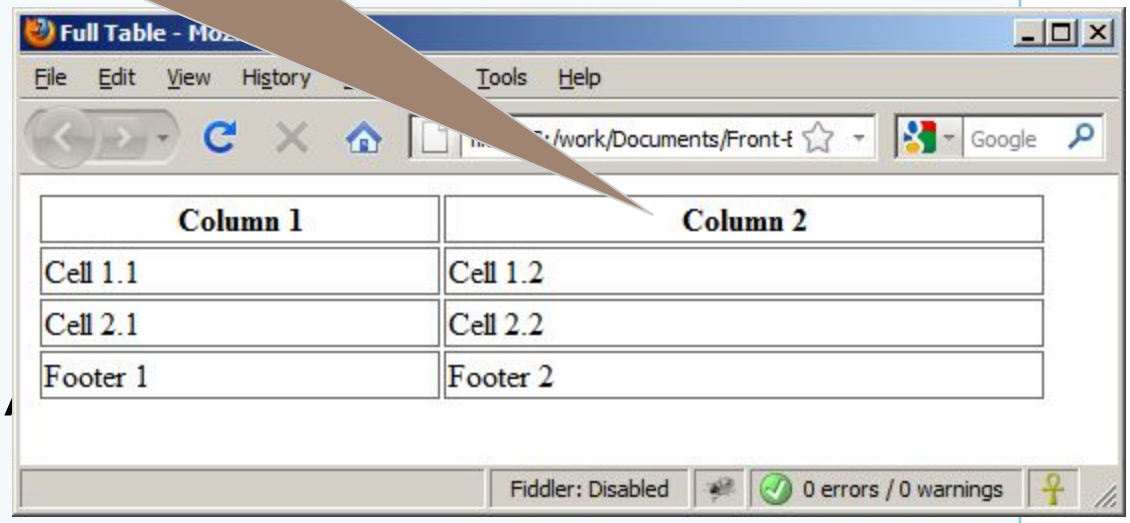
Last comes the body (data)

Complete HTML Table

By default, header text is bold and centered.

table-full.html

```
<table>
<colgroup>
  <col style="width
</colgroup>
<thead>
  <tr><th>Column 1</
</thead>
<tfoot>
  <tr><td>Foot 1</td><td>Footer 2</td></tr>
</tfoot>
<tbody>
  <tr><td>Ce d></tr>
  <tr><td>Cell 2.1</td><td>Cell 2.2</td></tr>
```



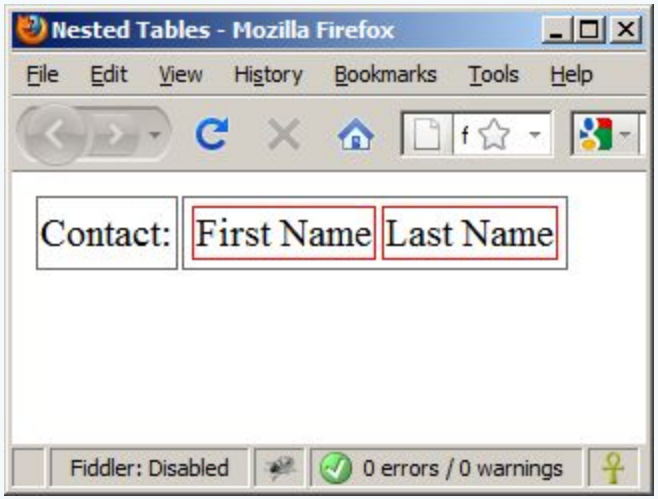
Although the footer is before the data in the code, it is displayed last

Nested Tables

- Table data "cells" (`<td>`) can contain nested tables (tables within tables):

nested-tables.html

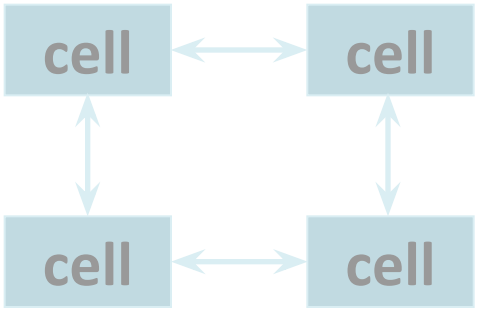
```
<table>
  <tr>
    <td>Contact:</td>
    <td>
      <table>
        <tr>
          <td>First Name</td>
          <td>Last Name</td>
        </tr>
      </table>
    </td>
  </tr>
```



Cell Spacing and Padding

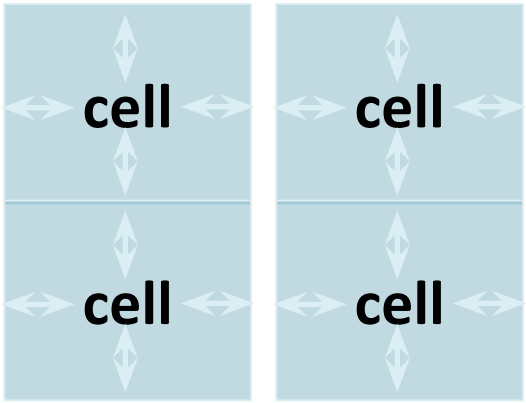
- Tables have two important attributes:

- **`cellspacing`**



- **Defines the empty space between cells**

- **`cellpadding`**

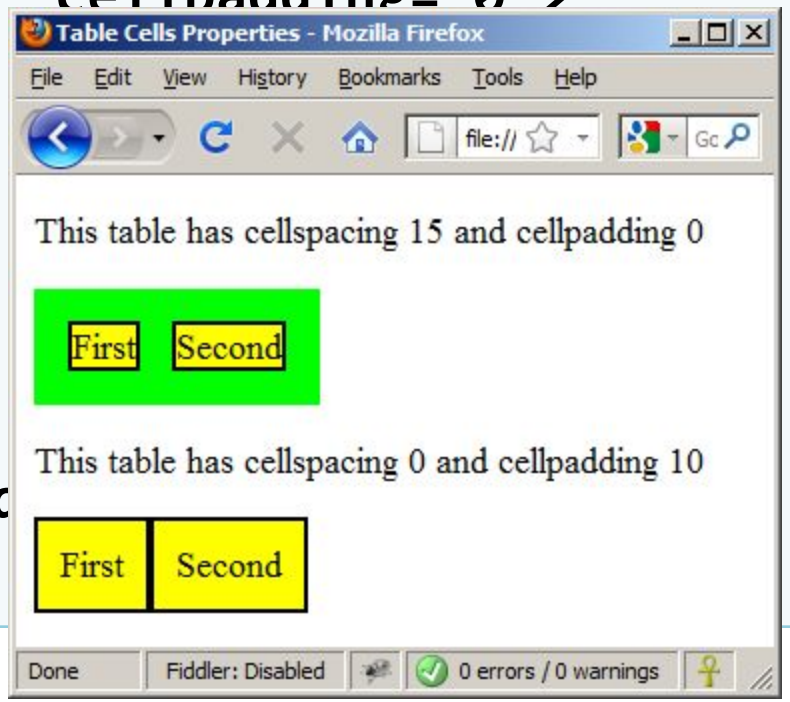


- **Defines the empty space around the cell content**

Cell Spacing and Padding – Example (2)

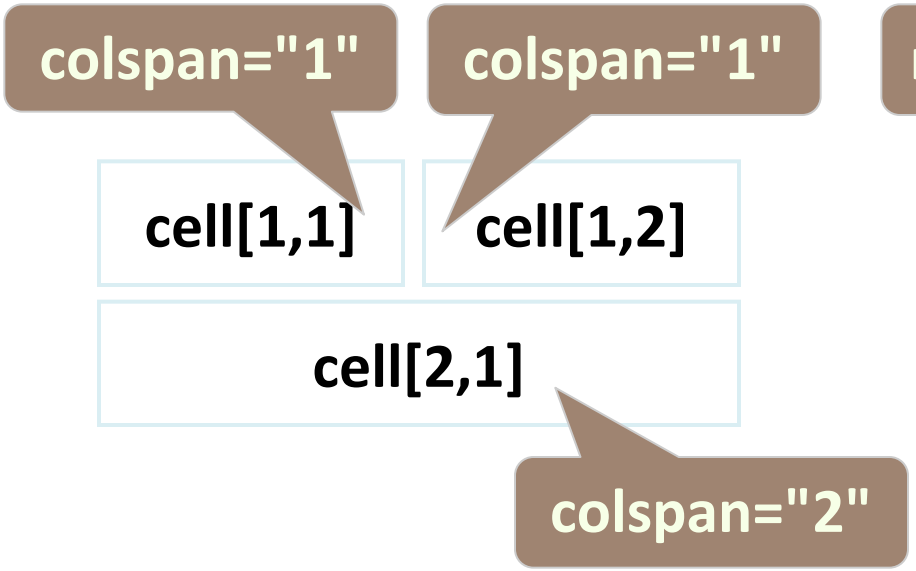
table-cells.html

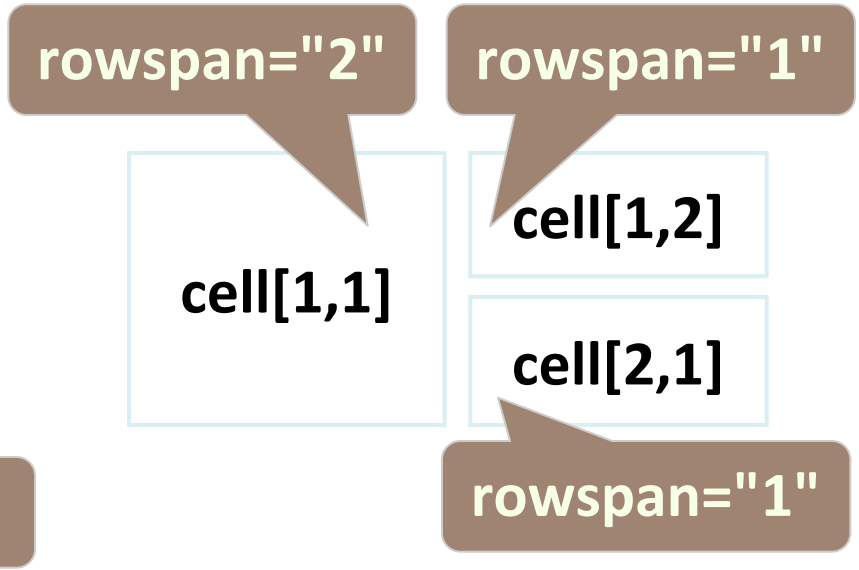
```
<html>
  <head><title>Table Cells</title></head>
  <body>
    <table cellspacing="15" cellpadding="0">
      <tr><td>First</td>
      <td>Second</td></tr>
    </table>
    <br/>
    <table cellspacing="0"
      <tr><td>First</td><td
    </table>
  </body>
</html>
```



Column and Row Span

- Table cells have two important attributes:
  - **`colspan`**
  - **`rowspan`**



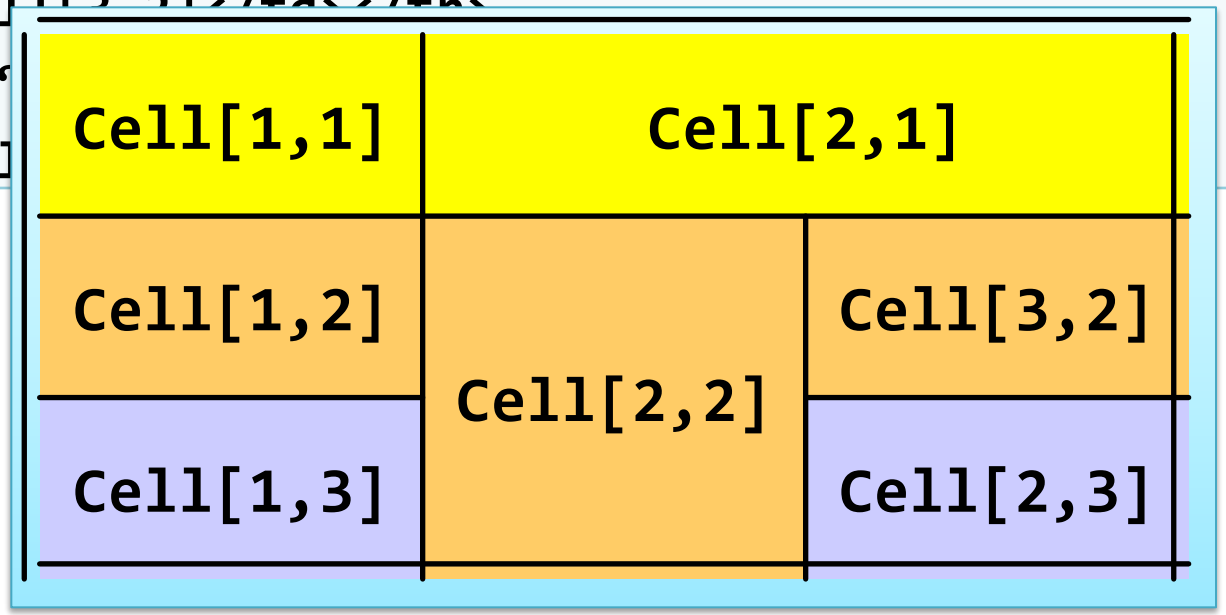


- **Defines how many columns**
- **Defines how many rows the**

Column and Row Span –Example (2)

table-colspan-rowspan.html

```
<table cellspacing="0">
    <tr class="1"><td>Cell[1,1]</td>
        <td colspan="2">Cell[2,1]</td></tr>
    <tr class="2"><td>Cell[1,2]</td>
        <td rowspan="2">Cell[2,2]</td>
        <td>Cell[3,2]</td></tr>
    <tr class="
        <td>Cel
</table>
```



SIMPLE EXAMPLE

```
<table>
<tr>
<th>Name</th>
<th>Favorite Color</th>
</tr>
<tr>
<td>Bob</td>
<td>Yellow</td>
</tr>
 <tr>
<td>Michelle</td>
 <td>Purple</td>
</tr>
</table>
```

Sample Code without Borders

```
<table>
  <tr>
    <td>Item A1</td>
    <td>Item A2</td>
    <td>Item A3</td>
  </tr>
  <tr>
    <td>Item B1</td>
    <td>Item B2</td>
    <td>(this is B3)</td>
  </tr>
  <tr>
    <td>Item C1</td>
    <td>-C2-</td>
    <td>*C3*</td>
  </tr>
  <tr>
    <td>Item number D1</td>
    <td>Item D2</td>
    <td>Item D3</td>
  </tr>
</table>
```

Item A1	Item A2	Item A3
Item B1	Item B2	(this is B3)
Item C1	-C2-	*C3*
Item number D1	Item D2	Item D3

Do the following to this code

1. Add borders
2. Add cell padding
3. Add header

Table Creation-Exercise

- Create a table as shown below using table tags

Slno	Course Code	Title	Credits
1	19CSE101	CP	3
2	19CSE103	UID	3
3	19CSE102	DS	2
4	19MAT100	DM	4

A complex table

Invoice #123456789			14 January 2025
Pay to: Acme Billing Co. 123 Main St. Cityville, NA 12345		**Customer:** John Smith 321 Willow Way Southeast Northwestershire, MA 54321	
Name / Description	**Qty.**	**@**	**Cost**
Paperclips	1000	0.01	10.00
Staples (box)	100	1.00	100.00
Subtotal			110.00
Tax		8%	8.80

Graphical User Interface Design

GUI / UI Design

HTML - Forms

formDemo.html

localhost/haio/book_1/chap_7/formDemo.html

Form Demo

Text input

Text box text here

Password ••••••

Text Area Your text here

Selecting elements

Select List one

Check boxes Green Eggs Ham

Radio buttons small medium large

Buttons

standard button input button Reset Submit

Why Forms?

- Allows user to interact with data
- Forms are used to create (rather primitive) GUIs on Web pages
  - Usually the purpose is to ask the user for information
  - The information is then sent back to the server
- You can create forms with ordinary HTML, but to make them *do* something, you need a programming language – We shall look at Java Script for this later

What is a form?

- A form is an area that can contain form elements
  - The syntax is: <form ***parameters***> ***...form elements...*** </form>
  - Form elements include: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc
    - Other kinds of tags can be mixed in with the form elements
  - A form usually contains a Submit button to send the information in he form elements to the server
  - The form's ***parameters*** tell JavaScript how to send the information to the server
  - Forms can be used for other things, such as a GUI for simple programs

Forms and JavaScript

- The JavaScript language can be used to make pages that "do something"
  - You *can* use JavaScript to write complete programs, but...
  - Usually you just use snippets of JavaScript here and there throughout your Web page
  - JavaScript code snippets can be attached to various form elements
    - For example, you might want to check that a zipcode field contains a 5-digit integer before you send that information to the server
- Microsoft calls its version of JavaScript "active scripting"
- Forms can be used without JavaScript, and JavaScript can be used without forms, but they work well together
- JavaScript for forms is covered later

Syntax

<form>

.

form elements

.

</form>

- An HTML form contains **form elements**.
- Form elements are different types of input elements, like text fields, checkboxes, radio buttons, submit buttons, and more.

The <form> tag

- The <form ***arguments***> … </form> tag encloses form elements (and probably other elements as well)
- The arguments to form tell what to do with the user input
  - action="***url***" (required)
    - Specifies where to send the data when the Submit button is clicked
  - method="get"(default)
    - Form data is sent as a URL with ?form_data info appended to the end
    - Can be used *only* if data is all ASCII and not more than 100 characters
  - method="post"
    - Form data is sent in the body of the URL request
    - Cannot be bookmarked by most browsers
  - target="***target***"
    - Tells where to open the page sent as a result of the request
    - ***target***= _blank means open in a new window
    - ***target***= _top means use the same window

Forms

- The FORM element is used to create a data input form.
- A region using forms is enclosed within the <FORM> </FORM> tags.
- A document can have several forms, but the forms should not be embedded.
- The FORM element has three attributes:
  - ACTION, METHOD, and ENCTYPE.

Forms

- **METHOD:**
  - **Specifies the way in which the data from the user are encoded.**
  - **The default METHOD is GET, although the POST method is preferred.**
  - **GET: The CGI program receives the encoded form input in the QUERY_STRING variable, which follows the "?" in the URL that calls the script.**
  - **POST: The CGI script or program receives the encoded form input in its standard input stream. The CONTENT_LENGTH must be used.**

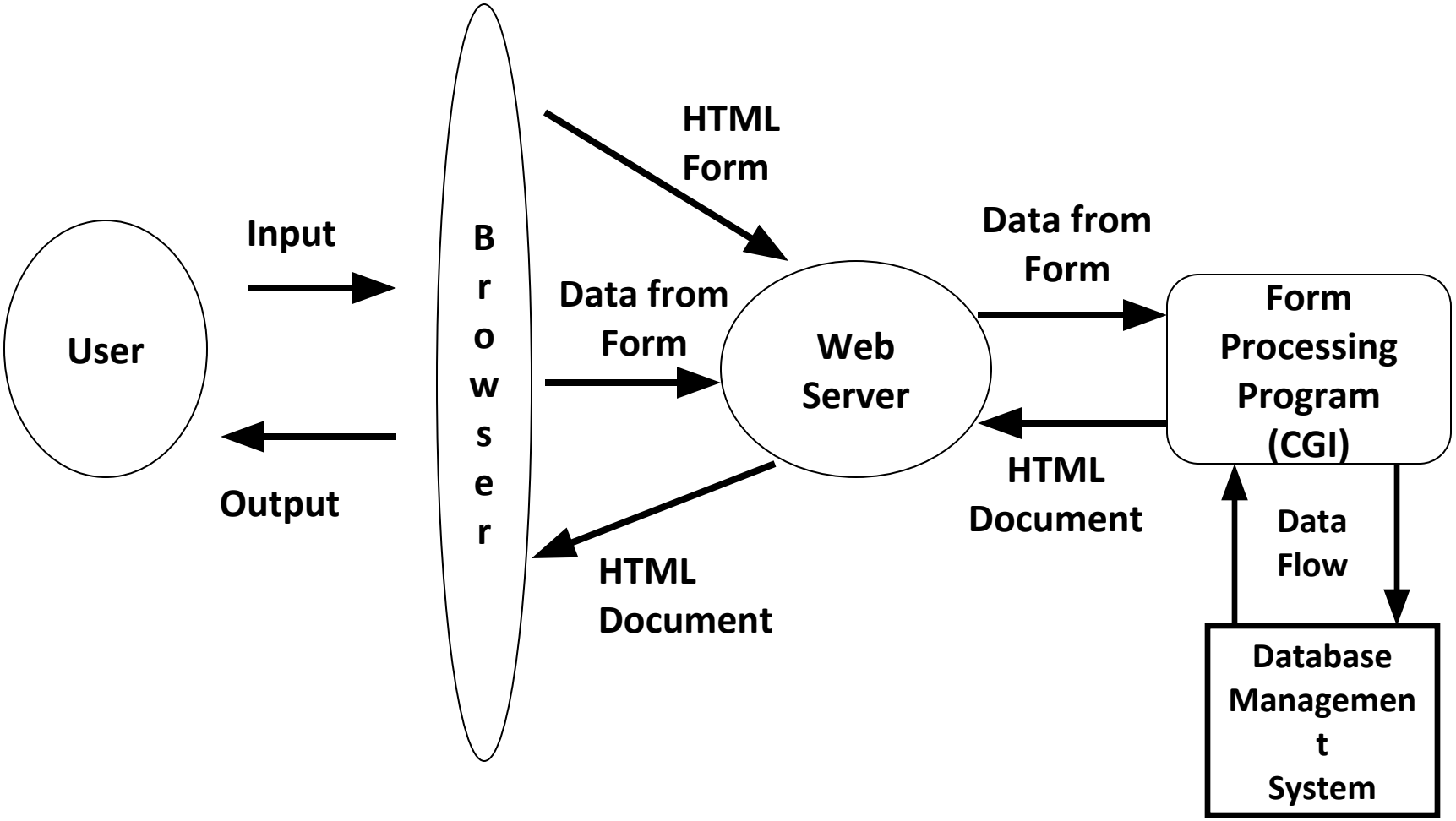
Forms

- **ACTION:**
  - **Specifies the destination URL to which the form should be submitted, once it has been completed by the user.**
  - **If no URL is specified, the URL of the current document containing the form is used.**
  - **MAILTO Action: The data from the form is mailed to the specified E-mail address. Use the POST method.**

Forms

- **ENCTYPE:**
  - **Tell the browser how the data from a form should be encoded when it is returned to the server.**
  - **The default is "application/x-www-form-urlencoded" that converts spaces to "+" and uses "&" to delineated different data fields.**

Form Processing



Flow of Information for Forms

Form Elements

- A form: A container for form elements. Although the form element itself isn't usually a visible part of the page (like the body tag)
- Text boxes: These standard form elements allow the user to type text into a one-line element.
- Password boxes: These boxes are like text boxes, except they automatically obscure the text to discourage snooping.
- Text areas: These multi-line text boxes accommodate more text than the other types of text boxes. You can specify the size of the text area the user can type into.
- Select lists: These list boxes give the user a number of options. The user can select one element from the list. You can specify the number of rows to show or make the list drop down when activated.

Form Elements

- Check boxes: These non-text boxes can be checked or not. Check boxes act independently — more than one can be selected at a time (unlike radio buttons).
- Radio buttons: Usually found in a group of options, only one radio button in a group can be selected at a time. Selecting one radio button deselects the others in its group.
- Buttons: These elements let the user begin some kind of process. The Input button is used in JavaScript coding whereas the Submit buttons are used for server-side programming The Reset button is special because it automatically resets all the form elements to their default configurations.

The <input> tag

- **The <input> Element**
  - The <input> element is the most important form element.
  - The <input> element can be displayed in several ways, depending on the **type** attribute.
- Most, but not all, form elements use the input tag, with a type="…" argument to tell which kind of element it is
  - type can be text, checkbox, radio, password, hidden, submit, reset, button, file, or image
- Other common input tag arguments include:
  - name: the name of the element
  - id: a unique identifier for the element
  - value: the "value" of the element; used in different ways for different values of type
  - readonly: the value cannot be changed
  - disabled: the user can't do anything with this element
  - Other arguments are defined for the input tag but have meaning only for certain values of type

TYPE Attribute

- **TEXT type:**
  - Specifies a single line text entry field.
  - Can be used with the MAXLENGTH and SIZE attributes (MAXLENGTH >= SIZE)

```
<P><B> First Name:</B> <INPUT NAME="fname"
  TYPE = text MAXLENGTH=30 SIZE =30></P>
<P><B> Last Name:</B> <INPUT NAME="lname"
  TYPE = text MAXLENGTH=30 SIZE =30></P>
```

Text input

A text field:

```
<input type="text" name="textfield" value="with an initial value" />
```

A text field: with an initial value

A multi-line text field

```
<textarea name="textarea" cols="24" rows="2">Hello</textarea>
```

A multi-line text field Hello

A password field:

```
<input type="password" name="textfield3" value="secret" />
```

A password field: ••••••

- Note that two of these use the input tag, but one uses textarea

Buttons

- A submit button:
  <input type="submit" name="Submit" value="Submit" />
- A reset button:
  <input type="reset" name="Submit2" value="Reset" />
- A plain button:
  <input type="button" name="Submit3" value="Push Me" />

A submit button: Submit

A reset button: Reset

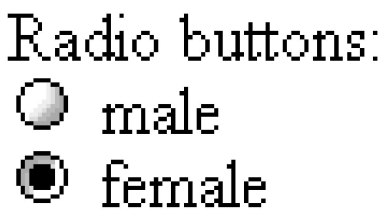
A plain button: Push Me

- submit: send data
- reset: restore all form elements to their initial state
- button: take some action as specified by JavaScript

- Note that the type is input, not “button”

Radio buttons

```
Radio buttons:<br>
    <input type="radio" name="radiobutton" value="myValue1" />
    male<br>
    <input type="radio" name="radiobutton" value="myValue2”
            checked="checked" />female
```



- If two or more radio buttons have the same name, the user can only select one of them at a time
  - This is how you make a radio button “group”
- If you ask for the value of that name, you will get the value specified for the selected radio button
- As with checkboxes, radio buttons do not contain any text

Labels

- In many cases, the labels for controls are not part of the control
  - <input type="radio" name="gender" value="m" />male
  - In this case, clicking on the word “male” has no effect
- A label tag will bind the text to the control
  - <label><input type="radio" name="gender" value="m" />male</label>
  - Clicking on the word “male” now clicks the radio button
- w3schools says that you should use the for attribute:
  - <label for="lname">Last Name:</label>
    <input type="text" name="lastname" id="lname" />
  - In my testing (Firefox and Opera), this isn’t necessary, but it may be for some browsers
- Labels also help page readers read the page correctly
- Some browsers may render labels differently

Checkboxes

- A checkbox:

```
<input type="checkbox" name="checkbox"
        value="checkbox" checked="checked">
```

A checkbox: ☑

- type: "checkbox"
- name: used to reference this form element from JavaScript
- value: value to be returned when element is checked
- Note that there is *no text* associated with the checkbox
  - Unless you use a label tag, only clicking on the box itself has any effect

Drop-down menu or list

- A menu or list:

```
<select name="select">
    <option value="red">red</option>
    <option value="green">green</option>
    <option value="BLUE">blue</option>
</select>
```

A menu or list: red

- Additional arguments:
  - size: the number of items visible in the list (default is "1")
  - multiple
    - if set to "true" (or just about anything else), any number of items may be selected
    - if omitted, only one item may be selected
    - if set to "false", behavior depends on the particular browser

Hidden fields

- <input type="hidden" name="hiddenField" value="nyah">
  <-- right there, don't you see it?

A hidden field: <-- right there, don't you see it?

- What good is this?
  - All input fields are sent back to the server, including hidden fields
  - This is a way to include information that the user doesn't need to see (or that you don't want her to see)
  - The value of a hidden field can be set programmatically (by JavaScript) before the form is submitted

Forms - Example

```
<html>
<head>
<title>Get Identity</title>
</head>
<body>
<p><b>Who are you?</b></p>
<form method="post" action="">
  <p>Name:
    <input type="text" name="textfield">
  </p>
  <p>Gender:
    <label><input type="radio" name="gende
    <label><input type="radio" name="gende
  </p>
  </form>
</body>
</html>
```

Who are you?

Name:

Gender: Male Female

Forms

- **Form Input: INPUT**
  - **Only used within a FORM element and is denoted by <INPUT>.**
  - **Attributes:**
    - ***NAME*: The name of the particular element.**
    - ***MAXLENGTH*: The maximum number of characters that can be entered by users in a text field.**
    - ***SIZE*: Specifies the size of the field and depends on its type.**
    - ***SRC*: Denote URL for an image.**
    - ***VALUE*: Contain the initial value displayed to users.**
    - ***TYPE*: Defines the type of data used in the field.**
    - ***CHECKED*: Indicates that a checkbox or radio button is selected.**
    - ***DISABLED*: Prevents the field from receiving focus.**
    - ***ALIGN*: Alignment if image is used.**
    - ***READONLY*: Prevents modification of the contents of the field.**

TYPE Attribute

- **SUBMIT and RESET Types:**
  - **SUBMIT: Used to submit the form's content, as specified by the ACTION attribute.**
  - **RESET: Set all fields in the form to their initial values.**

```
<P>INPUT TYPE=SUBMIT>
<INPUT TYPE=RESET>

<P><INPUT TYPE=SUBMIT VALUE = "Place Your Order">
<INPUT TYPE=RESET VALUE = "Start over">
```

TYPE Attribute

- **BUTTON Input Type:**
  - Creates a button whose use can be defined through scripting and onClick event.
  - Use to create a back button.
  - Only useful to browsers that support scripting.

```
<FORM><P><INPUT TYPE="button" VALUE="Back to
Last Document" onClick="history.back(
)"></P></FORM>
```

TEXTAREA

- Let users enter more than one line of text.
- Uses attributes ROWS and COLS to size.
- WRAP Attribute:
  - OFF: No wrapping
  - VIRTUAL: Display wraps but long lines are sent as one line.
  - PHYSICAL: Word wraps and text is sent with wrap points.

More on Forms - More input elements

- *Date* -- Setting the input type to date indicates that you wish the user to enter a date value.
- Tag: <input type="date“ id = "date" />
- Usage: <p> Date: <label>

```
        <input type="date" id="date" />
    </label></p>
```

- *Time* -- input type is to allow the user to enter a time. Time is stored in hh:mm format
- Tag: <input type = "time” id = "time" />
- Usage:<p> Time: <label>

```
 <input type="time" id="time" />
 </label></p>
```

More form tags …

- *Datetime* -- combines date and time into a single element
- *Number* -- allows the input of numerical data. This often consists of a text field followed by some kind of selector (say up and down arrows), or it might change the virtual keypad of a portable device to handle only numeric input.
- Tag: <input type = "number“ id = "number“ max = "10“ min = "0" />
- Usage:
   Number: <label> <input type = "number“ id = "nu mber“ max = "10“ min = "0" /></label>

Some More ….

- *Email* -- *g*enerally looks like a plain text field, but it validates on an e-mail address
- <input type="email” id = "txtEmail" />
- Usage: Email: <label>

<input type = "email“ id = "txtEmail“ /></label>

HTML <audio> Element

- <audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
  Your browser does not support the audio element.
  </audio>
- The controls attribute adds audio controls, like play, pause, and volume.
- The <source> element allows you to specify alternative audio files which the browser may choose from. The browser will use the first recognized format.
- **The text between the <audio> and </audio> tags will only be displayed in browsers that do not support the <audio> element.**

<video> tag

- <video> element specifies a standard way to embed a video in a web page
- <video width="320" height="240" controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
  </video>
- The controls attribute adds video controls, like play, pause, and volume.
- Always include width and height attributes. If height and width are not set, the page might flicker while the video loads.
- The <source> element allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
- **The text between the <video> and </video> tags will only be displayed in browsers that do not support the <video> element.**

Input Types … a glance

- <input type="button">
- <input type="checkbox">
- <input type="color">
- <input type="date">
- <input type="datetime-local">
- <input type="email">
- <input type="file">
- <input type="hidden">
- <input type="image">
- <input type="month">
- <input type="number">
- <input type="password">
- <input type="radio">
- <input type="range">
- <input type="reset">
- <input type="search">
- <input type="submit">
- <input type="tel">
- <input type="text">
- <input type="time">
- <input type="url">
- <input type="week">

HTML Input - value Attribute

- The readonly attribute specifies that the input field is read only
- The disabled attribute specifies that the input field is disabled. -- A disabled input field is unusable and un-clickable, and its value will not be sent when submitting the form
- size attribute specifies the size (in characters) for the input field
- maxlength attribute specifies the maximum allowed length for the input field

HTML 5 Does not end here…

More Tags.. More Tags..

More and More of attributes..