
A REVIEW



YOU MUST KNOW
THE FOLLOWING



The Basic HTML Setup

You are **REQUIRED TO KNOW** this

Basic HTML Page Components

- `<html>`
- `<head>`
- `<title> </title>`
- `</head>`
- `<body>`
- `</body>`
- `</html>`

Found in Every Web Document

Explains to web browser the type of document being viewed

Types of Web Documents

Can be static or dynamic

Static Web Pages

- Text and graphics content are specifically named and placed in areas of web page
- Must always know the specific name and location of included files
- Usually saved with .htm or .html file extension

Dynamic Web Pages

- Content normally generated from a database
- Text and graphics usually determined by what is stored in database
- Fields of database used to place information in page
- Files saved based upon scripting language used to create page

Basic Scripting Languages Used

- Active Server Pages (ASP) - .asp
- Coldfusion - .cfm
- Java Server Pages - .jsp
- PHp - .php
- All used with some form of database

Using Tables



Why Use This Method

- Assists in placing information in screen for viewing
- Better sizing of text and graphics areas
- Concise page layouts

What is a Table?

- Series of rows and columns
- Each column/row is a separate cell that can have it's own particular formatting

How Many Rows and Columns?

- Limited only by the amount of information being shown
- Should be based on the way you wish information to be presented in screen

Basic Table Coding Format

`<table>` (opening portion of tag)
`<tr>` (defines start of row)
`<td>` (defines start of column or cell)
`</td>` (defines end of column or cell)
`</tr>` (defines end of row)
`</table>` (closing tag or end of table)

Building The Table

- The container
- `<table> </table>`
- The row
- `<tr> </tr>`
- The cell
- `<td> </td>`

Making the Code Better

- Change the way the table looks by changing it's attributes
- Tables can have width, height, alignment, and border specified
- Individual cells can have same types of coding amplifications made to them
- Individual cells can also have both vertical and horizontal alignments made

Basic Alignment Attributes

`<table BORDER="0">` (table with no border)
`<tr>` (defines start of row)
`<td VALIGN="top">` (all content starts at top of cell)
`</td>` (defines end of column or cell)
`</tr>` (defines end of row)
`</table>` (closing tag or end of table)

Other Cell Attributes

- VALIGN – controls vertical alignment of cell (top, bottom, middle)
- ALIGN – controls horizontal alignment of cell (left, center, right)
- COLSPAN – controls number of merged columns
- ROWSPAN – controls number of merged rows

More Attributes

- CELLSPACING – controls amount of space between table cells
- CELLPADDING – sets amount of space between contents of cell and cell wall

The Embedded Table

A Table Within A Table

Embedding The Table

- Tables can have other tables placed inside of them
- Adds to the configuration of your page layout
- Requires tighter coding of page, but allows for more complex layouts

The Table Code

```
<table>
<tr>
<td valign="top">
<table>
<tr>
<td>TABLE 2</td>
</tr>
</table>
</td>
</tr>
</table>
```

Embedding Same As Making

- THERE IS NO DIFFERENCE IN THE CODING STRUCTURES
- Only difference is in placing tables properly
- Embedded tables is standard method used in web design to control page layout

Using Forms in Your Web Page

How to Collect Information in Your Design

Basic Reason for Using Forms

To collect information

Some Uses for Forms

- Collect personal information about site visitors
- Collect data for online reports and statistics
- Collect purchasing information for eCommerce
- Collect any other needed information

Types of Fields in Forms

- Text Box
- Text Area Box
- Input Buttons
- Radio Buttons
- Checkbox
- Dropdown Box
- Submit Button

Basic Form Tags

- `<form>` defines a form for user input
- `<input>` defines input field
- `<textarea>` defines multi-line input area
- `<select>` defines selectable list
- `<option>` defines option in select box

The First Step in Making Forms

- You must tell the browser that your page includes a form
- Command to start form
- `<form method="post" action="do something">`
- State method of dealing with information
- Tells what to do after form is submitted

Form Methods

- Two different methods for processing forms
- POST used primarily to send information somewhere
- GET used primarily to obtain and show information
- POST can be used for MOST form operations

Form Actions

- Every form has an action
- Can send information to another web page
- Can send information to a page that enters information into database
- Can send information to an email address
- Action="mailto:someone@somewhere.com"

Form Elements

- Make up the basic fields that are shown in the form
- Allow for data entry or selection

Most Used Form Input Elements

- Text Field
- Text Area
- Radio Button
- Checkbox
- Select Box
- Password Box
- Hidden Fields
- Submit Button

Common Features For All Form Tags

- All elements must have a name
- Most have some form of value
- Most have variables for size
- All collect some form of information
- All require viewer to do something
- Makes form interactive

The Form Syntax

Standard Form Syntax

- Simple form with 1 input field and submit button
- `<form method='post' action='script'>`
- `<input type='text' name='fieldname' size='size in pixels'>`
- `<input type='submit' value='Submit'>`
- `</form>`

Using Form Fields

How do They Really Work

Text Field

- Basic box that allows for one line of text
- DOES NOT HAVE A CLOSING TAG
- `<input type="text" name="name" size="99">`
- Must show type, name, size

Information that can be collected

- Person's name, address, phone, city, state, zipcode
- Credit card numbers
- Dates and time

Text Area

- Larger box that allows viewer to write something
- Uses more lines than simple text field
- **REQUIRES A CLOSING TAG**
- `<textarea name="name" rows="25" cols="45">What you write</textarea>`

Radio Button

- Places small circle on page
- When circle is chosen it becomes darker than others
- Can be used in sets
- All can have the same name
- Only one can be used at any time, regardless of how many you have

Making Radio Buttons

- `<input type="radio" name="category heading" value="assigned to specific button">`
- DOES NOT HAVE CLOSING TAG
- EACH BUTTON USED MUST HAVE DIFFERENT VALUE

Checkbox

- Basically same as radio button with three distinct differences
- Places a small square on page
- Marked with a check when selected
- Can check as many as you like

Making Checkbox

- `<input type="checkbox" name="category" value="what it is worth">`
- DOES NOT HAVE CLOSING TAG
- ALL BOXES MUST HAVE DIFFERENT VALUE

Select Box

- Also called Pop-up or Drop-down box
- Allows viewer to make selections from an established list of items
- Can be set to select only one or multiple items from list
- HAS TAG WITHIN TAG
- HAS CLOSING TAG

How to Use Select Box

- `<select name="name" size="1">`
- `<option value="value">Text of Value</option>`
- `</select>`
- Value of option tag is usually text shown to determine selection
- Specific value can be pre-selected to be shown first
- `<option value="value" selected>`

Password Box

- Only used for purposes of entering hidden passwords
- Configured the same as simple text field
- `<input type="password" name="name" size="10">`
- When text is entered stars or periods appear in box
- DOES NOT HAVE CLOSING TAG

Hidden Fields

- Configured same as text box
- Hides information from viewers but has distinctive value in form
- `<input type="hidden" name="name" value="value">`
- DOES NOT HAVE CLOSING TAG
- Can be placed anywhere in form, but usually after starting form tag or immediately before closing tag of form

Submit Buttons

- Used to tell the form to do its work
- Two types of submits
- RESET empties the form and lets viewer start over
- SUBMIT sends information to its destination
- `<input type="submit" value="caption on button">`
- DOES NOT HAVE CLOSING TAG

Ending Your Form

- Must make sure you properly close the form
- `</form>`
- Can have many forms on same page
- Forms can have specific names assigned
- `<form name="maillist" method="post" action="something to do">`

Configuring Forms With Tables

- Form should envelope table
- Table configured based on how you want form to be viewed
- `<form method="post" action="something">`
- `<table width="100%" border="0">`
- `<tr><td></td><td></td></tr>`
- `</table>`
- `</form>`

Special Form Tags

- Provide for extended form configurations beyond normal usage
- `<label>` defines a label to a control
- `<fieldset>` defines a fieldset
- `<legend>` defines caption for fieldset
- `<optgroup>` defines an option group

Defining Fieldsets

- Places border around certain groupings of information in form
- Must have following tags
- <fieldset> defines the fieldset
- <legend> defines caption for set
- BOTH HAVE CLOSING TAGS

Using Fieldset Tag

- <form method="post" action="">
- <fieldset>
- <legend>
- Health information:
- </legend>
- Height <input type="text" size="3">
- Weight <input type="text" size="3">
- </fieldset>
- </form>

Defining Option Groups

- Used within select tags
- Used for defining specific option groupings
- BOTH OPTION AND OPTGROUP TAGS HAVE CLOSINGS

Using Optgroup Tag

```
■ <select>
  <optgroup label="Swedish Cars">
    <option value ="volvo">Volvo</option>
    <option value ="saab">Saab</option>
  </optgroup>
  <optgroup label="German Cars">
    <option value
  ="mercedes">Mercedes</option>
    <option value ="audi">Audi</option>
  </optgroup>
</select>
```

Cascading Style Sheets

Doing More With Less

What is CSS

- **CSS** stands for **Cascading Style Sheets**
- Styles define **how to display** HTML elements
- Styles are normally stored in **Style Sheets**
- Styles were added to HTML 4.0 **to solve a problem**
- **External Style Sheets** can save you a lot of work
- External Style Sheets are stored in **CSS files**
- Multiple style definitions will **cascade** into one

Styles Solve Common Problem

- Helps solve compatibility issues between different browsers
- Can save extra work
- All major browsers support style sheets
- Internet Explorer
- Firefox
- Netscape
- Opera
- Macintosh

Cascading Order

- From lowest to highest
- Browser default
- External style sheet
- Internal style sheet (inside the <head> tag)
- Inline style (inside an HTML element)
- Inline style (inside an HTML element) has the highest priority

Syntax

- CSS syntax made up of 3 separate parts
- Selector {property: value}
- Normally the html tag or element to define
- Property is attribute to be changed
- Each property can have value
- Property and value separated by colon and surrounded by curly brackets

Values with Multiple Words

- If value is multiple words, put quotes around value
- Sample
- P {font-family "sans serif"}

Defining More Than One Property

- Separate each property with semicolon
- Sample
- P {text-align:center;color:red}
- Best to describe one property on each line
- P
- {text-align:center;
- Color: black;
- Font-family: arial
- }

Grouping

- Can group selectors by separating each with comma
- Sample
- H1, h2, h3, h4, h5, h6
- {
- Color: red
- }
- All header elements in page would be colored red

Class Selector

- Can define different styles for same html element
- P.right {text-align: right}
- P.center {text-align: center}
- Can also omit tag name in selector to define style used by all elements for certain class
- .center {text-align: center}
- DO NOT START a class name with a number

ID Selector

- Can define styles for elements with id
- Style rule will match any element having same id value
- #green {color: green}
- <table id="green">
- All portions of the table will be colored green
- DO NOT START AN ID NAME WITH A NUMBER

Commenting Your Code

- Comments used to explain your code
- Help when you edit source code at a later date
- Comments is ignored by browser
- CSS comments begin with /*
- CSS comments end with */
- Html comments begin with <!--
- Html comments end with -->

Add Styles to Elements with Particular Attributes

- Can apply styles to html elements
- Sample
- Input [type="text"]
- {background-color: blue}
- All text boxes in forms would have background color of blue

How to Insert Style Sheets

- When browser reads style sheet, will format document according to it
- Three ways to insert style sheets
- External style sheet
- Internal style sheet
- Inline styles

External Style Sheet

- Ideal for applying styles to multiple pages
- Change look of entire web site from only one file
- Each page must link to style sheet
- Link must be in document head section

Linking External Style Sheets

- File must have .css extension
- <head>
- <link rel="stylesheet" type="text/css" href="stylesheet.css">
- </head>

Internal Style Sheet

- Used when a single document has unique style
- Defined in head section using <style> tag
- <head>
- <style type="text/css">
- P {margin-left: 20px}
- H1 {color: red}
- </style>
- </head>

Inline Styles

- Should only be used to apply style to single occurrence of element
- Use to change attribute of relevant tag
- Sample
- `<p style="text-align: center; color:blue">` This is a paragraph
- `</p>`
- All text in paragraph would be centered and blue in color

SAMPLE STYLE SHEET



Applies to Body and HyperLinks

```
■ body
■ {
■ background: #F9F9F7 url("../images/a1.gif") repeat-x;
■ font-size: 12px;
■ font-family: "trebuchet ms", helvetica, sans-serif;
■ color: #000000;
■ line-height: 18px;
■ }
■ a
■ {
■ color: #FF7800;
■ text-decoration: underline;
■ }
■ a:hover
■ {
■ text-decoration: none;
■ }
```

Saving the file

All cascading style sheets saved as .css