Crash Course on

Web Development

Dulan S. Dias
www.dulandias.com

July 2019

Introduction to HTML

Tags

- The essence of HTML programming is tags
- A tag is a keyword enclosed by angle brackets (Example: <I>)
- There are opening and closing tags for many but not all tags; The affected text is between the two tags

What is HTML?

- HTML, otherwise known as HyperText Markup Language, is the language used to create Web pages
- Using HTML, you can create a Web page with text, graphics, sound, and video

More Tags...

- The opening and closing tags use the same command except the closing tag contains and additional forward slash /
- For example, the expression `<B> Warning </B>` would cause the word **‘Warning’** to appear in bold face on a Web page

Nested Tags

- Whenever you have HTML tags within other HTML tags, you must close the nearest tag first
- Example:

```
<H1> <I> The Nation </I> </H1>
```

Structure of a Web Page

- All Web pages share a common structure
- All Web pages should contain a pair of <HTML>, <HEAD>, <TITLE>, and <BODY> tags

```
<HTML>
<HEAD>
<TITLE> Example </TITLE>
</HEAD>
<BODY>
     This is where you would include
     the text and images on your Web
     page.
</BODY>
</HTML>
```

The <TITLE> Tag

- Choose the title of your Web page carefully; The title of a Web page determines its ranking in certain search engines
- The title will also appear on Favorite lists, History lists, and Bookmark lists to identify your page

Text Formatting

- Manipulating text in HTML can be tricky; Oftentimes, what you see is NOT what you get
- For instance, special HTML tags are needed to create paragraphs, move to the next line, and create headings

Text Formatting Tags

<B> **Bold Face** </B>

<I> *Italics* </I>

<U> Underline </U>

<P> New Paragraph </P>

<BR> Next Line

Headings

- Web pages are typically organized into sections with headings; To create a heading use the expression <Hn>….</Hn> where n is a number between 1 and 7
- In this case, the 1 corresponds to the largest size heading while the 7 corresponds to the smallest size

Aligning Text

- The ALIGN attribute can be inserted in the <P> and <Hn> tags to right justify, center, or left justify the text
- For example, <H1 ALIGN=CENTER> The New York Times </H1> would create a centered heading of the largest size

Comment Statements

- Comment statements are notes in the HTML code that explain the important features of the code
- The comments do not appear on the Web page itself but are a useful reference to the author of the page and other programmers
- To create a comment statement use the <!-- …. --> tags

The Infamous Blink Tag

- It is possible to make text blink using the <BLINK> … </BLINK> tag
- However, it is best to use this feature at most sparingly or not at all; What seems like a good idea to a Web designer can become very annoying to a Web user
- The <BLINK> tag is not supported by Internet Explorer

Page Formatting

- To define the background color, use the BGCOLOR attribute in the <BODY> tag
- To define the text color, use the TEXT attribute in the <BODY> tag
- To define the size of the text, type <BASEFONT SIZE=n>

Example

```
<HTML>
<HEAD>
<TITLE> Example </TITLE>
</HEAD>
<BODY BGCOLOR=“black”  TEXT=“white”>
<BASEFONT SIZE=7>
     This is where you would include the text and images on your Web page.
</BODY>
</HTML>
```

Inserting Images

- Type <IMG SRC = “image.ext”>, where image.ext indicates the location of the image file
- The WIDTH=n and HEIGHT=n attributes can be used to adjust the size of an image
- The attribute BORDER=n can be used to add a border in pixels thick around the image

Alternate Text

- Some browsers don’t support images. In this case, the ALT attribute can be used to create text that appears instead of the image.
- Example:

  <IMG SRC=“satellite.jpg” ALT = “Picture of satellite”>

Links

- A link lets you move from one page to another, play movies and sound, send email, download files, and more….
- A link has three parts: a **destination**, a **label**, and a **target**
- To create a link type

```
<A HREF=“page.html”> label </A>
```

Anatomy of a Link

<A HREF=“page.html”> label </A>

- In the above link, “page.html” is the destination. The destination specifies the address of the Web page or file the user will access when he/she clicks on the link.
- The label is the text that will appear underlined or highlighted on the page

Example: Links

- To create a link to CNN, I would type:

  <A HREF=“http://www.cnn.com”>CNN</A>

- To create a link to MIT, I would type:

  <A HREF=“http://www.mit.edu”>MIT</A>

Changing the Color of Links

- The LINK, VLINK, and ALINK attributes can be inserted in the <BODY> tag to define the color of a link
  - LINK defines the color of links that have not been visited
  - VLINK defines the color of links that have already been visited
  - ALINK defines the color of a link when a user clicks on it

Using Links to Send Email

- To create a link to an email address, type <A HREF=“mailto:email_address”> Label</A>
- For example, to create a link to send email to myself, I would type: <A HREF=“mailto: ktdunn@mit.edu”>email Katie Dunn</A>

Anchors

- Anchors enable a user to jump to a specific place on a Web site
- Two steps are necessary to create an anchor. First you must create the anchor itself. Then you must create a link to the anchor from another point in the document.

Anchors

- To create the anchor itself, type <A NAME=“anchor name”>label</A> at the point in the Web page where you want the user to jump to
- To create the link, type <A HREF=“#anchor name”>label</A> at the point in the text where you want the link to appear

Example: Anchor

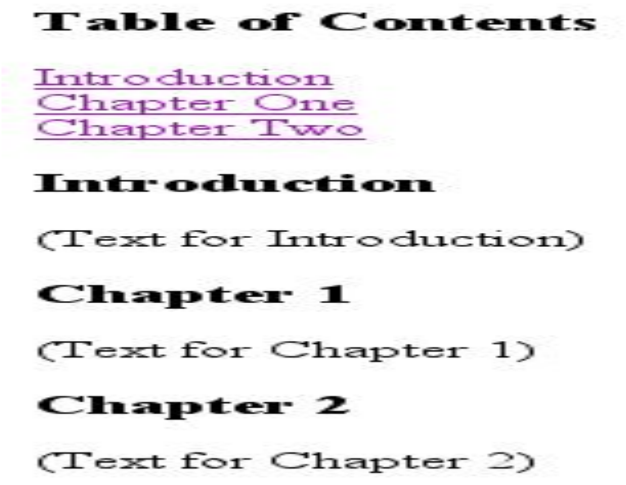


Table of Contents

Introduction
Chapter One
Chapter Two

Introduction

(Text for Introduction)

Chapter 1

(Text for Chapter 1)

Chapter 2

(Text for Chapter 2)

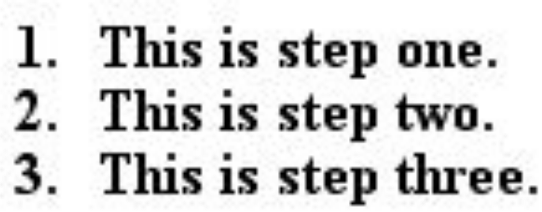
```
<A HREF="#chap2">Chapter Two</A><BR>

<A NAME="chap2">Chapter 2 </A>
```

Ordered Lists

- Ordered lists are a list of numbered items.
- To create an ordered list, type:

```
<OL>
    <LI> This is step one.
    <LI> This is step two.
    <LI> This is step three.
</OL>
```

Here’s how it would look on the Web:



1. This is step one.
2. This is step two.
3. This is step three.

More Ordered Lists….

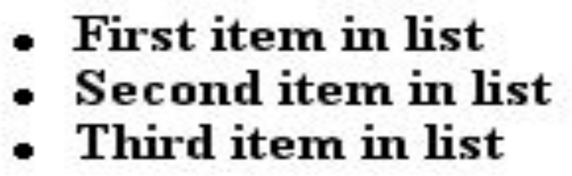
- The TYPE=x attribute allows you to change the the kind of symbol that appears in the list.
  - A is for capital letters
  - a is for lowercase letters
  - I is for capital roman numerals
  - i is for lowercase roman numerals

Unordered Lists

- An unordered list is a list of bulleted items
- To create an unordered list, type:

```
<UL>
    <LI> First item in list
    <LI> Second item in list
    <LI> Third item in list
</UL>
```

Here's how it would look on the Web:



- First item in list
- Second item in list
- Third item in list

More Unordered Lists...

- The TYPE=shape attribute allows you to change the type of bullet that appears
  - *circle* corresponds to an empty round bullet
  - *square* corresponds to a square bullet
  - *disc* corresponds to a solid round bullet; this is the default value

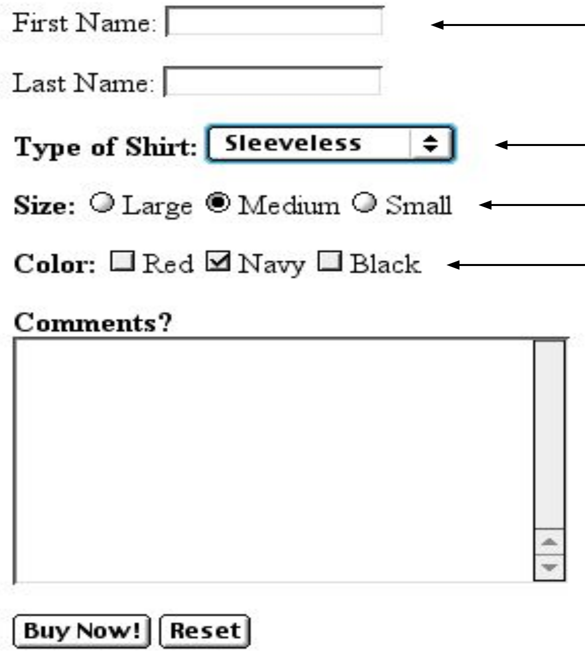
Forms

- What are forms?
  - An HTML form is an area of the document that allows users to enter information into fields.
  - A form may be used to collect personal information, opinions in polls, user preferences and other kinds of information.

Forms

- There are two basic components of a Web form: the shell, the part that the user fills out, and the script which processes the information
- HTML tags are used to create the form shell. Using HTML you can create text boxes, radio buttons, checkboxes, drop-down menus, and more...

Example: Form



First Name:

Last Name:

Type of Shirt: Sleeveless

Size: ○ Large ● Medium ○ Small

Color: □ Red ☑ Navy □ Black

Comments?

Text Box

Drop-down Menu

Radio Buttons

Checkboxes

Text Area

Reset Button

Submit Button

The Form Shell

- A form shell has three important parts:
  - the <FORM> tag, which includes the address of the script which will process the form
  - the form elements, like text boxes and radio buttons
  - the submit button which triggers the script to send the entered information to the server

Creating the Shell

- To create a form shell, type <FORM METHOD=POST ACTION="script_url"> where "script_url" is the address of the script
- Create the form elements
- End with a closing </FORM> tag

Creating Text Boxes

- To create a text box, type <INPUT TYPE=“text” NAME=“name” VALUE=“value” SIZE=n MAXLENGTH=n>
- The NAME, VALUE, SIZE, and MAXLENGTH attributes are optional

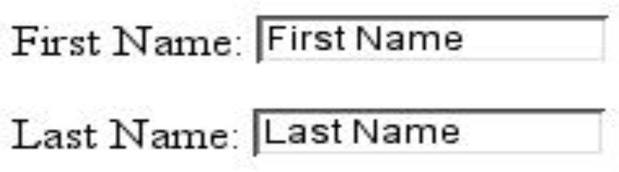
Text Box Attributes

- The NAME attribute is used to identify the text box to the processing script
- The VALUE attribute is used to specify the text that will initially appear in the text box
- The SIZE attribute is used to define the size of the box in characters
- The MAXLENGTH attribute is used to define the maximum number of characters that can be typed in the box

Example: Text Box

```
First Name: <INPUT
TYPE="text"
NAME="FirstName"
VALUE="First Name"
SIZE=20>
<BR><BR>

Last Name: <INPUT
TYPE="text"
NAME="LastName"
VALUE="Last Name"
SIZE=20
```

- Here’s how it would look on the Web:



First Name: First Name

Last Name: Last Name

Creating Larger Text Areas

- To create larger text areas, type <TEXTAREA NAME=“name” ROWS=n1 COLS=n2 WRAP> Default Text </TEXTAREA>, where n1 is the height of the text box in rows and n2 is the width of the text box in characters
- The WRAP attribute causes the cursor to move automatically to the next line as the user types

Example: Text Area

```
<B>Comments?</B>
<BR>
<TEXTAREA NAME="Comments" ROWS=10 COLS=50 WRAP>
</TEXTAREA>
```

Creating Radio Buttons

- To create a radio button, type <INPUT TYPE="radio" NAME="name" VALUE="data">Label, where "data" is the text that will be sent to the server if the button is checked and "Label" is the text that identifies the button to the user

Example: Radio Buttons

```
<B> Size: </B>
 <INPUT TYPE="radio" NAME="Size"
  VALUE="Large">Large
 <INPUT TYPE="radio" NAME="Size"
  VALUE="Medium">Medium
 <INPUT TYPE="radio" NAME="Size"
   VALUE="Small">Small
```

Creating Checkboxes

- To create a checkbox, type <INPUT TYPE=“checkbox” NAME=“name” VALUE=“value”>Label
- If you give a group of radio buttons or checkboxes the same name, the user will only be able to select one button or box at a time

Example: Checkboxes

```
    <B> Color: </B>
    <INPUT TYPE="checkbox" NAME="Color"
VALUE="Red">Red
    <INPUT TYPE="checkbox" NAME="Color"
       VALUE="Navy">Navy
     <INPUT TYPE="checkbox" NAME="Color"
       VALUE="Black">Black
```

Creating Drop-down Menus

- To create a drop-down menu, type <SELECT NAME=“name” SIZE=n MULTIPLE>
- Then type <OPTION VALUE= “value”>Label
- In this case the SIZE attribute specifies the height of the menu in lines and MULTIPLE allows users to select more than one menu option

Example: Drop-down Menu

```
<B>WHICH IS FAVOURITE FRUIT:</B>
<SELECT>
<OPTION VALUE="MANGOES">MANGOES
<OPTION VALUE="PAPAYA">PAPAYA
<OPTION VALUE="GUAVA">GUAVA
<OPTION VALUE="BANANA"> BANANA
<OPTION VALUE="PINEAPPLE">PINEAPPLE
</SELECT>
```

Creating a Submit Button

- To create a submit button, type <INPUT TYPE="“submit”>
- If you would like the button to say something other than submit, use the VALUE attribute
- For example, <INPUT TYPE=“submit” VALUE=“Buy Now!”> would create a button that says “Buy Now!”

Creating a Reset Button

- To create a reset button, type <INPUT TYPE=“reset”>
- The VALUE attribute can be used in the same way to change the text that appears on the button

Tables

- Tables can be used to display rows and columns of data, create multi-column text, captions for images, and sidebars
- The <TABLE> tag is used to create a table; the <TR> tag defines the beginning of a row while the <TD> tag defines the beginning of a cell

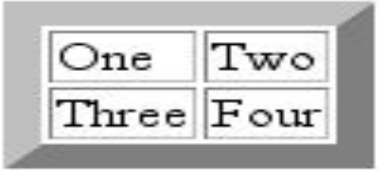
Adding a Border

- The BORDER=n attribute allows you to add a border n pixels thick around the table
- To make a solid border color, use the BORDERCOLOR=“color” attribute
- To make a shaded colored border, use BODERCOLORDARK=“color” and BORDERCOLORLIGHT=“color”

Creating Simple Table

```
<TABLE BORDER=10>
    <TR>
        <TD>One</TD>
        <TD>Two</TD>
    </TR>
    <TR>
        <TD>Three</TD>
        <TD>Four</TD>
    </TR>
</TABLE>
```

- Here’s how it would look on the Web:



One	Two
Three	Four

Adjusting the Width

- When a Web browser displays a table, it often adds extra space. To eliminate this space use the WIDTH =n attribute in the <TABLE> and <TD> tags
- Keep in mind - a cell cannot be smaller than its contents, and if you make a table wider than the browser window, users will not be able to see parts of it.

Centering a Table

- There are two ways to center a table
  - Type <TABLE ALIGN=CENTER>
  - Enclose the <TABLE> tags in opening and closing <CENTER> tags

Wrapping Text around a Table

- It is possible to wrap text around a table. This technique is often used to keep images and captions together within an article.
- To wrap text around a table, type <TABLE ALIGN = LEFT> to align the table to the left while the text flows to the right.
- Create the table using the <TR>, <TD>, and </TABLE> tags as you normally would

Adding Space around a Table

- To add space around a table, use the HSPACE=n and VSPACE=n attributes in the <TABLE> tag
- Example:

```
<TABLE HSPACE=20 VSPACE=20>
```

Spanning Cells Across Columns

- It is often necessary to span one cell across many columns. For example, you would use this technique to span a headline across the columns of a newspaper article.
- To span a cell across many columns, type <TD COLSPAN=n>, where n is the number of columns to be spanned

Spanning Cells Across Rows

- To span a cell across many rows, type <TD ROWSPAN=n>, where n is the number of rows

Aligning Cell Content

- By default, a cell’s content are aligned horizontally to the left and and vertically in the middle.
- Use VALIGN=direction to change the vertical alignment, where “direction” is top, middle, bottom, or baseline
- Use ALIGN=direction to change the horizontal alignment where “direction” is left, center, or right

Controlling Cell Spacing

- Cell spacing is the space *between* cells while cell padding is the space *around* the contents of a cell
- To control both types of spacing, use the CELLSPACING =n and CELLPADDING=n attributes in the <TABLE> tag

Nesting Tables

- Create the inner table
- Create the outer table and determine which cell of the outer table will hold the inner table
- Test both tables separately to make sure they work
- Copy the inner table into the cell of the outer table
- Don’t nest too many tables. If you find yourself doing that, find an easier way to lay out your Web page

Changing a Cell’s Color

- To change a cell’s color, add the BGCOLOR=“color” attribute to the <TD> tag
- Example:

  <TD BGCOLOR=“blue”>

Dividing Your Table into Column Groups

- You can divide your table into two kinds of column groups: structural and non-structural.
- Structural column groups control where dividing lines are drawn; Non-structural groups do not
- Both let you format an entire column of cells at once

Column Groups

- To create structural column groups, type <COLGROUP SPAN=n> after the <TABLE> tag, where n is the number of columns in the group
- To create non-structural column groups, type <COL SPAN=n>, where n is the number of columns in the group

Dividing Table into Horizontal Sections

- You can also create a horizontal section consisting of one or more rows. This allows you to format the rows all at once
- To create a horizontal section, type <THEAD>, <TBODY>, or <TFOOT> before the first <TR> tag of the section
- Netscape does not support these tags

Controlling Line Breaks

- Unless you specify otherwise a browser will divide the lines in a cell as it sees fit.
- The NOWRAP attribute placed within the <TD> tag forces the browser to keep all the text in a cell on one line
- Example:
  - <TD NOWRAP>Washington, D.C.

Cascading Style Sheets (CSS)

CSS

- Useful for creating one unified look for an entire web site.
- Helps to separate style from content.
- Can be used for creating absolute positioning.

History

- The W3C released the specs for CSS1 in 1996.
- Both browsers quickly implemented the specs.
- BUT both also added their own custom tricks, some of which were based on predictions of the CSS2 standard
- In mid-1998 the CSS2 standard came out.
- I will be presenting information based on CSS1
  - It is supported by 4.0+ versions of both browsers

Defining CSS

- Styles can be defined in three different ways.
  - The style tag

    <style>…</style>
  - An embedded style attribute

    <p style="color:red">
  - An external style sheet

    <link ….>

The style tag

```
<style type=“text/css”>
 <!--
     h1 {color: blue; font-style: italic}
-->
</style>
```

- The style tag must be closed
- For CSS the type is always *text/css*
- The html comment tags are optional allowing for backwards compatibility.
  - Not often used in practice.

Embedded style attribute

<b style=“color:black;
text-decoration:overline,underline;”>text</b>

- Most tags accept style as a valid attribute.
- Best used for one-shot styles or special cases.

An External Style Sheet

<link rel=stylesheet type="text/css"
href="mystyles.css">

- Let's you link an extenal file.
  - Great for using the same styles on many pages
  - The file should have only CSS and no html in it.
- Links should be located in the header.
- You can have multiple link tags in a single document.

CSS syntax

- There are two parts to a CSS statement the class selector and the attributes.
  - The selector says who to apply the style to.
  - The attributes say how to format the selected portion

P {margin-left: 5em; margin-right: 5em;}

Selectors

- The selector indicates what elements the style should be applied to.
- By default this is all elements of the group indicated.
  - eg: all the <b>, <i>, <p> tags
- There are 3 kinds of **subgroup selectors**:
  - Class selectors
  - ID selectors
  - Contextual selectors

Class Selector

- Class selectors names should follow the same syntax you would use for a javascript variable name.
  - Allows the potential for scripting
- To create one you simply choose a valid element name and append your class name with a period.
- You can create a general class by simply omitting the document element.

Class selector Example

```
<style>
P {font-size: 14pt; margin-left: 2em; margin- right: 2em}
P.narrow {color:blue; margin-left: 5em; margin-right: 5em
.mygeneric {color: yellow}
 </style>
<p>This paragraph is normal </p>
<p class=“narrow”>This paragraph is narrow and in blue</p>
<span class=“mygeneric”>This would be yellow<span>
```

ID Selector

- ID selectors let you define a rule that applies to only one element in the entire document

```
#special3  {border: 5px ridge}
<p ID=“special3”>This text is special</p>
```

Contextual Selector

- Allows you to apply a pattern only to a particular context.

```
<style type="text/css">
    P {font-size 14pt; color: black;}
    P EM {font-size: 16pt; color red}
</style>
```

  - In this example only text in a emphasized section inside a paragraph will be in red.

More On selectors

- You may select multiple selectors by separating them with a comma

  h1, h2, h3, h4 {color: green}

- There's all kind of wacky stuff in the CSS2 spec you can read up on.

Properties

- Basic syntax:
  - The properties are enclosed in curly braces
  - Properties are seperated from one another by semi-colons
  - Properies are serperated from their values by colons.
    - Each property must have atleast one value
    - Multiple values are separated by commas

Property Values

- There are 5 kinds of property values
  - Keyword properties: underline, visible, ettc.
    - Not case-sensitive
  - Length properties: 1in, 4px, 5cm
  - Percentage values: line-height: 120%
  - url property values: url(service://server.com/pathname)
  - Color propety values: rgb(5,10,230)

Length Property Values

- In units of:
  - Relative:
    - em: height of ‘m’ in the current font
    - ex: height of ‘x’ in the current font
  - Pixels
    - px
  - Absolute
    - in, cm, mm,
    - pt: Points (1/72 of an inch)
    - pc: Picas (twelve points)

Color property values

- Can be specified by
  - Keyword: red, blue, green, black…
  - 3-digit hex: #78C --> #7788CC
  - rgb as
    - Decimal rgb(255,255,255)
    - Percentage rgb(50%,50%,50%)
    - Don’t leave space between rgb and opening parenthesis

Property list

- There are more than I can list, but here’s a list of most that work for both browsers
- background, background-color, background-image, border-color, border-style, border-width, clear, clip, color, display, float, font-family, font-size, font-style, font-weight, height, line-height, margin, padding, position, text-align, text-decoration, visibility
- There are many more.

DIVS

- Divs are a great tool for positioning
- The <div> tag is used for containing other tags or text, but applies no information on its own,
- By embedding a style into you div you can assign it an absolute or relative position

Positioning attributes

- Top and left indicate the offset from the upper left corner of the “positioning context”
- Width and height indicate the size of the div
- Z-index: a non-negative integer value is used for determining stacking precedence. Higher number are on top
- Visibility: either inherit, visible, or hidden

Absolute positioning

```
<div id=“logo” style=“position:absolute;
  left:100px; top:50px”><img
  src=“mylogo.gif”>Look at my logo!<div>
```

Introduction to JavaScript

Events

- Events are triggered by user actions or the browser itself.
- Common User actions:
  - Common events include: onMouseOver, onMouseOut, onClick
    - These 3 events are mostly commonly used with anchor tags
  - Others include onFocus, onBlur
- Document events:
  - onLoad and onUnLoad are examples of events not triggered by the user.

Event example

- the code:

```
<a href="http://www.nowhere.com" onclick="alert('I said don\'t do that!!!');return false;"> Don't click here</a>
```

- **onclick** is an *event*.
  - It is the event that occurs when the user clicks the link.
- **alert** is a *function*.
  - It takes a single *argument* which is a string.

Alert

- **alert** is used to grab the attention of the user
  - Syntax: alert(*string*);
    - Ex: **alert(“You have not entered a valid name.”)**
    - Ex. **alert(“Hello “ + name + “!”);**
- The look of the alert box is defined by the browser and platform.
- Gets the users attention but abusing it can easily be abused

Variables

- Declare Variables with **var**
  - **var i**;
  - Not necessary but a good practice.
- Assign variables with =
  - **var i = 7;**
  - **var name = “Eric”;**
- JavaScript is a non-typed language

Variable assignment

Valid

- **x = 7;**
- **y = x;**
- **x = y + 10;**
- **y = y + 4;**
- **var first = “jon”;**
- **var last = “doe”;**
- **var name = fist + “ “ + last;**

Not Valid

- **7 = x;**

Object example

- The code:

```
<a href="http://web.mit.edu" onMouseOver="window.status='Clear
   here to go to the sp.772 web site.'; return true;"
   onMouseOut="window.status=' '; return true;”> our
   webpage</a>. Watch the status bar
```

- **window** is an object.
- **status** is a property of window
  - Setting the status bar sets the message at the bottom of the browser.

Objects

- Javascript is an object oriented language.
  - Though because it is so ‘loose’ with its typing many don’t think of it as one.
- Objects are variables with properties and fields
  - These properties may be other variables or functions
- The “.” operator is used to access the methods and properties of an object.
  - Eg: eric.age = 24;

new

- Objects are declared using new.
  - **var eric = new Object()**
- Properties are not declared with **var**. You just assign them.
  - **eric.name = “Eric Traub”;**
    eric.age = “24”;
- Will talk about how to assign methods to an object next week.

Script example

- The code:

```
<script language="javascript">
<!--
var now = new Date();
var hour = now.getHours();

if (hour > 5 && hour <= 11) {
   document.write("<b>Good Morning!</b>");
} else if (hour > 11 && hour <= 17) {
   document.write("<b>Good Afternoon!</b>");
} else {
   document.write("<b>Good Evening!</b>");
}
//  -->
</script>
```

Script Example part 2

- <script> is the tag used to indicate that you are writing javascript and not html.
- **if** and **else** are used to make conditional statements.
- **document.write** tells javascript to insert html into the page

The <script> tag

- The main property of <script> is language.
  - Eg. **<script language =“javascript”>**
  - If you don’t specify a language the default is javascript.
  - You can also specify a version number
    - **<script language =“javascript”>**
- Always close the <script> tag, or your code will not work.
- Enclose your javascript code within a html comment block <!-- -->, to make the page compatible with non-javascript browsers.

document.write

- **document.write** will write it’s argument as plain html onto the page.
  - Eg: **document.write(“Hello there “ + name);**

if statement

- **if** is used to execute code only is some condition is true

```
if (age >= 18) {
      adult = true;
}
```

- **else** used with if to execute a second statement if the condition is false

```
if (sex == “male”) {
      title = “Mr”;
} else {
      title = “Ms”;
}
```

else if

- **else if** used for linking several if statement together.

```
if (phd == true) {
        title = “Dr.”;
} else if (sex == “male”) {
        title = “Mr.”;
} else if (sex == “female”) {
        title = “Ms.”;
} else {
        title = “????”;  /* Not male or female! */
}
```

Comparison

- There are several *operators* used for making comparisons between variables
- The main arithmetic comparitors are:
    - == equal
    - < less than
    - < less than or equal to
    - > greater than
    - >= greater than or equal to
    - != not equal to

Logic

- When making comparison you can also use logic functions
  - && for and
  - || for or (that’s two ‘pipes”; shift backslash)
  - ! For not

```
Eg: if (age > 18 && age < 65) {
            price  = “$10”;
      } else {
             price = “$6”;
       }
```

More Logic example

```
if  (customer.age <= 18 &&  movie.rating ==
  “R” && !(customer.parentPresent())) {
        document.write(“No admittance”);
}
```