1. What is a web page and HTML? (Lesson 1-2)

1.1 What is a web page?

When a web browser (e.g. Internet Explorer or Netscape) displays a web page, it reads from a plain text file, download from the web site or server (e.g. www.ibm.com), consists of code written with a language called Hypertext Markup Language (HTML).

A web browser displays information on your computer by interpreting the HTML that is used to build the web pages on the Web. Web pages usually displayed graphics, sound and multimedia files, as well as links to other pages, files that can be downloaded, and other Internet resources.

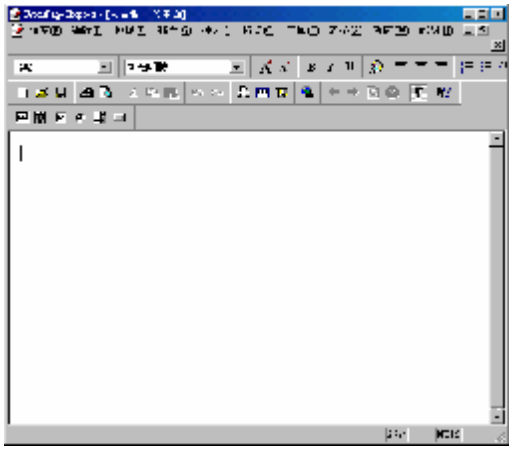
1.2 HTML Documents

What an HTML Document Is?

HTML documents are plain-text (also known as ASCII) files that can be created using any text editor (e.g. Notepad on a Windows machine). You can also use word-processing software if you remember to save your document as "text only with line breaks".

HTML Editors

Some WYSIWYG editors are also available (e.g. Adobe PageMill, Microsoft Frontpage, Dreamweaver). You may wish to try one of them after you learn some of the basics of HTML tagging. WYSIWYG is an acronym for "what you see is what you get"; it means that you design your HTML document visually, as if you were using a word processor, instead of writing the markup tags in a plain-text file and imagining what the resulting page will look like.



Getting Your Files on a Server

If you have access to a Web server, you can post (or upload) your pages to the server so that everybody on the Internet can access and read your pages. Some Internet service provider (ISP) also provides storage for you to post your pages.

On the other hand, some web sites (like www.geocities.com or www.tripod.com) also

provides “free” web hosting for your pages. You may not need to pay for them, however, in return, advertisement will pop out every time your pages are accessed.

Tags Explained

To denote the various elements in an HTML document, you use tags. HTML tags consist of a left angle bracket (<), a tag name, and a right angle bracket (>). Tags are usually paired (e.g., <H1> and </H1>) to start and end the tag instruction. The end tag looks just like the start tag except a slash (/) precedes the text within the brackets. HTML tags are listed below. The following is the content of a simple web page.

```
<html>

<head>
<title>My web page</title>
</head>

<body bgcolor="#FFFFFF">
<H1>Introduction</H1>
</body>
</html>
```

Some elements may include an attribute, which is additional information that is included inside the start tag. For example, you can specify the alignment of images (top, middle, or bottom) by including the appropriate attribute with the image source HTML code. Tags that have optional attributes are noted below.

NOTE: HTML is not case sensitive. <title> is equivalent to <TITLE> or <TiTlE>. There are a few exceptions noted in Escape Sequences below.

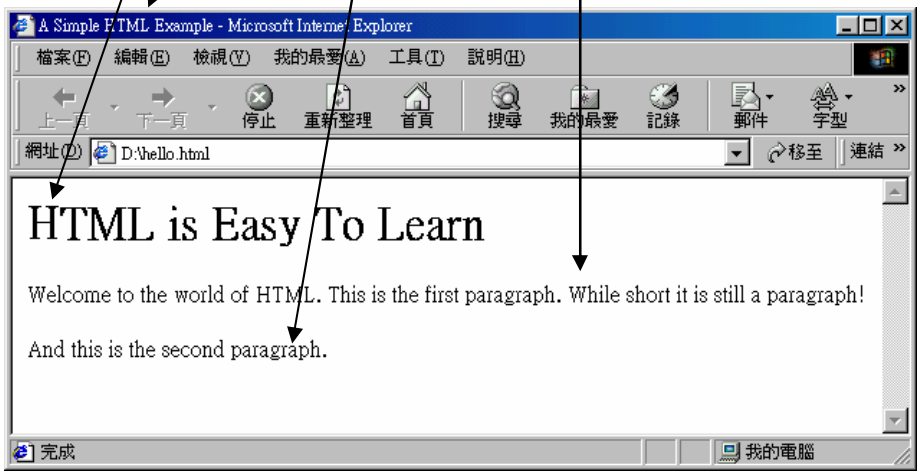
Not all tags are supported by all World Wide Web browsers. If a browser does not support a tag, it will simply ignore it. Any text placed between a pair of unknown tags will still be displayed, however.

The Minimal HTML Document

Every HTML document should contain certain standard HTML tags. Each document consists of head and body text. The head contains the title, and the body contains the actual text that is made up of paragraphs, lists, and other elements. Browsers expect specific information because they are programmed according to HTML and SGML specifications.

Required elements are shown in this sample bare-bones document:
The image shows the result of the above HTML when view by a web browser.

```
<html>
<head>
<TITLE>A Simple HTML Example</TITLE>
</head>
<body>
<H1>HTML is Easy To Learn</H1>
<P>Welcome to the world of HTML.
This is the first paragraph. While short it is
still a paragraph!</P>
<P>And this is the second paragraph.</P>
</body>
</html>
```



The required elements are the <html>, <head>, <title>, and <body> tags (and their corresponding end tags).

1.3 Markup Tags

HTML (<HTML> </HTML>)

This element tells your browser that the file contains HTML-coded information. The file extension .html also indicates this an HTML document and must be used. (If you are restricted to 8.3 filenames (e.g., LeeHome.htm, use only .htm for your extension.)

HEAD (<HEAD> </HEAD>)

The head element identifies the first part of your HTML-coded document that contains the title. The title is shown as part of your browser's window (see below).

TITLE (<TITLE> </TITLE>)
The title element contains your document title and identifies its content in a global context. The title is typically displayed in the title bar at the top of the browser window, but not inside the window itself. The title is also what is displayed on someone's hotlist or bookmark list, so choose something descriptive, unique, and relatively short. A title is also used to identify your page for search engines (such as HotBot or Infoseek).

BODY (<BODY> </BODY>)
The second--and largest--part of your HTML document is the body, which contains the content of your document (displayed within the text area of your browser window). The tags explained below are used within the body of your HTML document.

Headings (<H1> </H1>, <H2> </H2>,)
HTML has six levels of headings, numbered 1 through 6, with 1 being the largest. Headings are typically displayed in larger and/or bolder fonts than normal body text. The first heading in each document should be tagged <H1>.

Do not skip levels of headings in your document. For example, don't start with a level-one heading (<H1>) and then next use a level-three (<H3>) heading.

Paragraphs (<P> </P>)
In the above example, shown in the Minimal HTML Document section, the first paragraph is coded as

```
<P>Welcome to the world of HTML.
This is the first paragraph.
While short it is
still a paragraph!</P>
```

In the source file there is a line break between the sentences. A Web browser ignores this line break and starts a new paragraph only when it encounters another <P> tag. For example, the following would produce identical output as the first bare-bones HTML example:

```
<H1>Level-one heading</H1>
<P>Welcome to the world of HTML. This is the
first paragraph. While short it is still a
paragraph! </P> <P>And this is the second paragraph.</P>
```

Using the <P> and </P> as a paragraph container means that you can center a

paragraph by including the ALIGN=alignment attribute in your source file.

```
<P ALIGN=CENTER>
This is a centered paragraph.
</P>
```

When the above HTML code is read by the browser, the following is displayed:

This is a centered paragraph.

Assignment

1. Use a notepad to type in the HTML code in P.3 and save it with the name "mypage.htm".
2. Open the typed html file with a browser (e.g. Internet Explorer or Netscape)
3. What is the result if the following lines are added inside the <body> </body> of a web page:

```
<p><a href="http://www.ibm.com">go to ibm</a></p>
<p><a href="http://www.cnn.com" target="_blank">go to cnn</a></p>
```

2. Learning Javascript (Lesson 3-4)

2.1 What is Javascript?

Put simply, javascript is a specialized programming language. It is a programming language whereby you can insert scripts (mini programs) into a web page using a simple text editor. It's actually capable of quite a bit.

2.2 Writing and Testing Your First Script

(Please refer to http://www.javascriptmall.com/learn/lesson2.htm)

What is JavaScript?

I have already shown you that JavaScript can be used to display a specific greeting based on the time of day and can be used to generate a simple clock.

It can also display today's date and time in many different formats:

March 9, 2002

Saturday, March 9, 2002

9.3.02

2002 年 3 月 9 日 PM 06:57:48

There are only -798 days until the year 2000.

It can put scrolling text on your page:

It can put a drop down menu on your page:

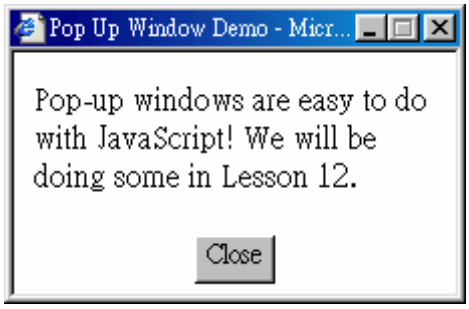
It can get basic information about your visitor and his Browser:

```
I see you have a computer with windows installed.
Browser: Microsoft Internet Explorer
JavaScript is enabled!
You are currently in an area whose time is 8 hours ahead of GMT.
```

It can do some cool things with images, such as a rollover:

It can do pop-up windows:

Click here to view a JavaScrdipt pop-up window



It is fast:

JavaScript is 100% text, just like HTML is. Therefore it loads quickly from the server to your computer. JavaScript runs on your computer and does not have to download any additional applications to do it's job.

It can be mixed with HTML:

You use the HTML tag combination <SCRIPT> </SCRIPT> to identify JavaScript. This makes it HTML friendly.

It can be annoying to your visitors:

- if you use too much.
- are more interested in doing cute things rather than adding functionality to your page
- take away some of the browser features that your visitor relies on, example using the status bar at the bottom of your browser for something that keeps your visitor from seeing normal messages.

It is Object Oriented:

JavaScript is composed of what is called objects that have properties and methods. The window that you are now using is an object that has properties such as height and width. Methods do things for us.

For instance, I used a JavaScript method to open the pop-up window in the demo above. The pop-up window contains a close button. Pressing this button causes another JavaScript method to run and close the window.

It is called jscript by Microsoft:

Microsoft Internet Explorer browsers have what is called jscript. The results from a browser that has jscript is basically the same as one that has JavaScript.

Our First Script

Let's start our journey into the learning of JavaScript by typing in the following HTML code. We will add to it and eventually wind up with a working clock similar to the one above. Of course it will take us a several lessons to get there because we first need to learn the basics.

```
<HTML>
<HEAD>
<TITLE>Our First Script</TITLE>
</HEAD>
<BODY>
```

```
<CENTER>
<H1>Our First Script</H1>
</CENTER>
<P>Today is
</BODY>
</HTML>
```

Save this code as script1.html. It looks something like this when we load it into our browser by using File/Open on the menu.

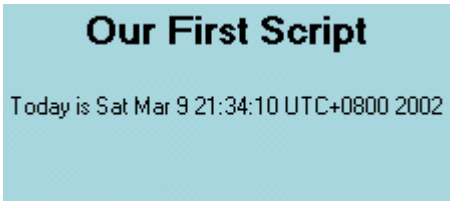
Not very exciting! Lets add some JavaScript to our code. Add the following script after the line <P>Today is

```
<SCRIPT Language="JavaScript">
var today = new Date()
document.write(today)
</SCRIPT>
```

Your html code should now look like this.

```
<HTML>
<HEAD>
<TITLE>Our First Script</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Our First Script</H1>
</CENTER>
<P>Today is
<SCRIPT Language="JavaScript">
var today = new Date()
document.write(today)
</SCRIPT>
</BODY>
</HTML>
```

Save this script and reload it into you browser. It should render something like this:



The <SCRIPT> tags

The HTML tags that are used to identify JavaScript are <SCRIPT language="JavaScript"> and </SCRIPT>.

Most browsers in use today have some version of JavaScript built in. However, there are still a few in use that don't understand JavaScript. These browsers will ignore the <SCRIPT> tag but will display the text between the tags.
The complete template for JavaScript is therefore:

```
<SCRIPT language="JavaScript"><!--

  (The JavaScript code)

//--></SCRIPT>
```

Our completed first Script

Ok, add the two necessary lines to our first script to hide it from older browsers. Our completed html code should now look that shown below. Make sure that you reload the script in the browser to prove that it still works the same.

```
<HTML>
<HEAD>
<TITLE>Our First Script</TITLE>
</HEAD>
<BODY>
<CENTER>
<H1>Our First Script</H1>
</CENTER>
<P>Today is
<SCRIPT Language="JavaScript"><!--
var today = new Date()
document.write(today)
//--> </SCRIPT>
</BODY>
</HTML>
```

Assignment

1. Surf the web and try to determine where JavaScript was used. Note pages that used JavaScript in an effective way.
2. Make a very simple web page using a series of document.write() methods to generate everything that you will see on the screen. This means that all of your BODY section will be a series of document.write() methods. Include at least one image in your web page. You might even want to include the date as we learned how to do in our first script. Realize that this is not the best way to do this simple web page but it does prepare us to do bigger things soon.

3. Values, Variables, and Operators (Lesson 5-6)

(Please refer to http://www.javascriptmall.com/learn/lesson4.htm)

3.1 Values

A value in JavaScript is a piece of information that can be a string, number, Boolean, Null, Object, or a Function as defined in the following table:

Value Types

Type	Description	Example
string	a series of characters inside quote marks	"JavaScript is cool"
number	any numeric value (not inside quotes)	4.55
Boolean	either true or false	true
Null	Empty, devoid of any value	null
Object	properties and methods of an object	
Function	value returned by a function	

3.2 Variables

A variable is a place holder in your computer's memory for information that can change. This place in memory is given a name in JavaScript so that the you can easily refer to it. You have to declaring a variable as follows:

```
var myVariable
```

Once the variable is declared, you can set it to any one of the values shown in the above table. Here is an example:

```
myVariable = 5
```

Later on in the program you can change its value to something different:

```
myVariable = 33
```

You can even change its type by simply assigning it a different type of value. For instance our variable was originally assigned a number. We can make it a string later in the program by simply assigning it one as follows:

```
myVariable = "this is a string"
```

You can initialize a variable and assign it a value at the same time as follows:

```
var myVariable = "JavaScript is cool"
```

You can even leave out the var keyword as follows:

```
myVariable = "this is not too cool"
```

Lab Time

Lets try some of these variables in our Lab. Type in the following code in the area that is provided in the BODY of our lab. Remember that the <SCRIPT> tags are already provided for you. All you have to do is type in the two lines shown in the lighter blue.

```
<SCRIPT Language="JavaScript">
<!-- hide from old browsers
myVariable = "JavaScript is Cool"
document.write(myVariable)
//-->
</SCRIPT>
```

You should see the words 'JavaScript is Cool' in the top frame when you press the Test Button.
Modify the code in our lab by replacing the string "JavaScript is Cool" with each of the following values, the number 4.5, the Boolean true, and the Null value null. Make sure you press Test for each change to view the results in the upper frame.

```
<SCRIPT Language="JavaScript">
<!-- hide from old browsers
var myVariable = "This is a String"  // string
document.write(myVariable)
document.write("<BR>")

myVariable = 45.55  // number
document.write(myVariable)
document.write("<BR>")

myVariable = true  // Boolean
document.write(myVariable)
document.write("<BR>")

var myVariable = null  // Null
document.write(myVariable)
document.write("<BR>")

myVariable = new Date  // object
document.write(myVariable)
document.write("<BR>")
//-->
</SCRIPT>
```

3.3 Operators

Operators are symbols that are used with variables to allow us to perform certain functions, such as adding, subtracting and etc. The table below lists the operators available in JavaScript and describes its function

Operators

operator	what it does
x + y (numeric)	Adds x and y
x + y (string)	Concatenates x and y
x - y	Subtracts x from y
x * y	Multiplies x b y
x / y	Divides x by y
x % y	Remainder of x / y , modulus
x++, ++x	x = x + 1
x--, --x	x = x - 1
-x	Reverses the sign of x

Lab Time

Declare and initialize two variables, x and y, with the values 12 and 27, respectively. Declare a third value z and set it equal to the sum of these two variables. Now use document.write(z) to display the results. When you press Test, you should of course get 39 displayed.

```
39
```

Now that you have that working we are going to add to it. Under the first two variables, declare and initialize two more variables xStr and yStr with the values "
This is " and "fun!" respectively. Declare and set a third variable zStr equal to xStr + yStr. Now type document.write(zStr) under the line document.write(z). When you press Test you should get the following:

```
39
This is fun!
```

Lets change the line document.write(z) to
document.write("The sum of " + x + " + " + y + " = " + z)
Now when you press Test, you should get the following:

```
The sum of 12 + 27 = 39
This is fun!
```

Try changing the values for the variables x and y to something different and pressing test to see the results. Doing this should have given you some feel for the advantage of using variables.

Assignment

Write javascript to add up the product of 1,2,3,4,5,6 and displayed the result in a web page.

4. If Statement, Boolean, Functions (Lesson 7-8)

(Please refer to http://www.javascriptmall.com/learn/lesson5.htm)

This lesson introduces you to two essential constructs of the JavaScript language, the if statement and the function.

The if statement is used to make decisions in JavaScript.

The function allows you to repeat specific JavaScript statements anytime you want by calling the same statements without writing new ones. For example, if I want to add two numbers at several different locations in my program, I can just write the code one time and designate it as a function. Then I can call that function anytime I want to add two numbers.

4.1 If Statement

You use the if statement in JavaScript to make decisions. The syntax for it is as follows:

```
if (condition){
  statements
  }
```

The if keyword identifies this as an if statement. The condition in the parenthesis ( ) is evaluated to determine if true, and if so then the statements inside the curly braces { } are executed, otherwise they are skipped and the programs continues with the first line after the if statement.

An optional else statement can be included with the if statement as follows:

```
if (condition){
  statements
  }
else{
  statements
  }
```

In this case, the statements inside of curly brackets { } after the else keyword are executed if the condition of the if statement is false.

if statement Results

condition	returns true if
x == y	x and y are equal
x != y	x and y are not equal
x > y	x is greater than y
x >= y	x is greater or equal to y
x < y	x is less than y
x <= y	x is less than or equal to y

```
<HTML>
<HEAD></HEAD>
<BODY bgcolor="CCFFFF">

<P>Testing if statement<BR>
<SCRIPT language="JavaScript"><!--
var num = 4

if(num == 4){
  document.write("<B>Satisfied!</B><BR>")
  }
//--></SCRIPT>
rest of program starts here

</BODY>
</HTML>
```

Take a close look at the above script. It contains a variable num which is initialized to 4 and an if statement that is true if num = = 4. You will see the following:

Testing if statement
Satisfied!
rest of program starts here

Change the value of the num variable to some number other than 4, for instance try 3 (i.e. change the line var num = 4 to var num = 3). Now the requirements of the if statement will not be satisfied and you will get the following:

Testing if statement
rest of program starts here

Now change the condition and test it with different value of num:

Conditions	**var num =**
num!= 4	3, 4
num > 4	0, 4, 10
num >= 4	0, 4, 10

4.2 Boolean Operators

This is the **and** operator, &&. Here is a rewrite of the script using the and operator.

```
if (grade < 90 && grade >= 80){
  document.write('Grade is a "B"')
  }
```

The and operator, && , allows you to combine two conditions so that both must be true to satisfy the if condition. Another Boolean operator is the **or** operator, ||, which combines two conditions such that the if statement is satisfied if either condition is true. The third boolean operator is the **Not** Operator, !, which makes a condition that returns true, false and vice versa.

Look again at the above script and notice that phrase 'Grade is a "B"' contains both double quotes and single quotes. This is the way you can put quotes inside of quotes in JavaScript.

4.3 Functions

What if you wanted to use our grading script for more than one student? You would have to repeat it for each student. This could get quite lengthy if there were more than a very few students. You can however use a function to contain the script and call it every time your need it. Here is the syntax for a function.

```
function name (parameters){
  statements
  }
```

The function keyword identifies this as a function. The parameters in the parenthesis ( ) provide a means of passing values to the function. The function can return a value by using the return keyword.

You better put your functions in the HEAD section of your document.
The best way to demonstrate a function is with an example. Suppose we want to make a function that we can call anytime to add two numbers together. Here is one way of writing the function.

```
<HTML>
<HEAD>
<SCRIPT language="JavaScript"><!--
function myAdder (num1, num2){
   var total = num1 + num2
   document.write(total)
   }
//--></SCRIPT>
</HEAD>
<BODY>
<SCRIPT language="JavaScript"><!—-

  myAdder(23, 56)

//--></SCRIPT>
</BODY>
</HTML>
```

You then can call the function from anywhere in your document. The output is:

79

Lets try a variation to the function that we wrote in our last lab experiment. This time we will let the function return the value rather than print it to the document. Then, we

will print the returned value to the document from the BODY section. Modify our function in the head section so that it looks like this:

```
function myAdder (num1, num2){
  var total = num1 + num2
  return total
  }
```

Here is the modified code that replaces the script in the BODY section.

```
document.write(myAdder(23, 56))
document.write("<BR>" + myAdder(100, 480))
```

When you test this new version, the results should be

79
580

You will want to use the first method sometime and the second on others. It will all depend on exactly what you are doing.

Assignment

1. Write a script that contains a function that you can pass a mark as a parameter and have it display a passed or failed condition. Demonstrate the functions by printing out the pass/fail status for five students.
2. Make a final modification to the script that we did in problems 1 so that a letter grade is displayed instead of pass or fail. This letter grade should be based on the following:

   A is a grade of 90 or greater
   B is a grade of 80 or greater, but less that 90
   C is a grade of 70 or greater, but less than 80
   F is any grade below 70

3. Study the chapter about "Forms, Events and Dialog Boxes" http://www.javascriptmall.com/learn/lesson6.htm. Write a web page with javascript that enable user to input two numbers and return the total of them.