JavaScript

JavaScript is a programming language meant to be used on the web. It works very well in conjunction with HTML and CSS to allow us to have interactivity in our web pages.

JavaScript is an interpreted language, meaning it does not need to be compiled. Rather, the browser reads the code and translates it for the computer just before it gets executed.

Simple Calculator

JavaScript can be used to fill `form` elements on an HTML page. Find the Simple Calculator page on the Javascript example page, which is linked from the assignment page. The actual Javascript code for this page is displayed below.

```
function calculate(opCode)
{
  with(document.calc){
    if(opCode == "+")
      result.value = eval(op1.value) + eval(op2.value);
    else if(opCode == "-")
      result.value = eval(op1.value) - eval(op2.value);
    else if(opCode == "*")
      result.value = eval(op1.value) * eval(op2.value);
    else if(opCode == "/")
      result.value = eval(op1.value) / eval(op2.value);
  }
}
```

Note that this code is surrounded by the HTML `script` tag. Also notice that the script is checking what the opcode is, then using it to evaluate the two operators.

This script reads two values from the HTML `form` and writes to another. The `form` element used in this example is shown below.

```
<form name = "calc">
<input name = "op1" type = "text"> <br>
<input name = "op2" type = "text"> <br>
<input name = "btnAdd" type = "button" OnClick = "calculate('+')" value = "ADD">
<input name = "btnSubtract" type = "button" OnClick = "calculate('-')" value =
    "SUBT">
<input name = "btnMultiply" type = "button" OnClick = "calculate('*')" value =
    "MULT">
<input name = "btnDivide" type = "button" OnClick = "calculate('/')" value = "DIV">
<br> <input name = "result" type = "text">
</form>
```

Looking at this adds a little context to the Javascript code as well. We can notice, for example, that the line in the script `with(document.calc)` is how the script finds the correct form to fill, as the form is given the name attribute "calc".

Looking at the code for the script, and form, try to understand how these two work together. Also to view the whole HTML file for this page right-click on it, then select <u>V</u>iew Page Source.

Adding JavaScript to Your Page

Now that we have seen the basics for Javascript, it is time to add some to your page. Complete the following for this lab, or if you have another idea of how you wish to use Javascript ask an instructor and do it instead.

- Look through the examples on the Javascript page

- Find an example that you want to use on your webpage, making sure you have an idea of how you will modify it

- Create a new HTML file for your Javascript example, i.e.
  `gedit NameOfYourJavascriptPage.html &`

- Copy the example into your new file (or create one from scratch, if you are comfortable with JavaScript)

- Save your file

- Link this new page from your home page

- Test the Javascript you added

- Link to your home page from your new Javascript page

- Modify the example, adding something meaningful to the example (remember to make small changes, test them, then another small change)