What is JavaScript?

• JavaScript is not Java!
• JavaScript is a programming language used to make your web pages more interactive.
• JavaScript is a scripting language.
• JavaScript is a client-side language (not a server-side language like PHP).
What JavaScript can NOT do ...

• JavaScript does not allow programs to client's files.
• JavaScript does not write files on the server.
• JavaScript cannot close a window that it has not opened.
• JavaScript cannot read information from a page that is on another server.
What JavaScript CAN do ...

• JavaScript is used to enhance the user interface.
• JavaScript can be used to validate a user’s data entry on a form.
• JavaScript can be used to customize web pages based on user’s selections or actions.
• JavaScript is typically used with forms, for serving cookies, building pages “on the fly” etc.
object-oriented vs. object-based

- JavaScript offers both object-oriented and object-based functionality.
Working with objects and methods

• Every object (such as a button) has properties.
• Every object can have methods which are things it can do such as:
  - buttons... `click()`
  - windows... `open()`
  - text can be... `selected()`
JavaScript also offers object-oriented functionality

• For example, when working with strings:
  - strings are a primitive data type to be used for example as a string literal:
    var strLiteralName = "howdy"
  - strings can also be declared as objects:
    var strObjectName = new String("howdy");
objects and sub-objects

• Use the dot syntax to distinguish objects and sub-objects:
  - window.title
  - form.checkbox

• Use the dot syntax to connote objects and methods:
  - document.write()
  - forms.elements.radio.click()
Document Object Model (DOM)

• The objects that make up a page (or document) can be represented as a tree structure (which you can see for example in Firebug).

• JavaScript considers each item in the tree to be an object.

• Representing this model as a tree diagram is called the DOM (Document Object Model).
AJAX

• AJAX is a combination of the following technologies:
  - XHTML
  - CSS
  - the DOM using JavaScript
  - XML (for data transfers between the server & client)
    - XMLHttpRequest to retrieve data from the server

• We will look at examples later in the semester.
A Word About Accessibility

• Be careful not to use JavaScript for menus, navigation and other ways that restrict who can use your site.

• The `<noscript>` ... `</noscript>` tag is useful in providing alternatives to your code. You can test the `<noscript>` ... `</noscript>` content by temporarily disabling JavaScript in your browser.
JavaScript working environment

• How to edit your code:
  - You can use a standard text editor (TextEdit on the Mac or Notepad in Windows)
  - You can use any standard code editor (e.g. BBedit)
  - You can use the Dreamweaver "code view"

... or any similar tool.
There are many JavaScript debugging tools available for free through the web and there are also debugging tools for $$$.

We will use Firebug in class in order to standardize how we work together and because this will accommodate both Mac and PC users.

We will also use the clumsy but time-honoured practice of displaying variables with an `alert` ...
Calling JavaScript functions

• JavaScript can be placed ...
  - in the <body> ... </body> of a web page
  - in the <head> ... </head> of a web page
  - in an external file with the suffix of .js

• Current “best practices” uses the method of putting all JavaScript in separate script files.

• The syntax is:
  
  <script type="text/javascript" src="javaScriptFile.js">
  </script>
However, when JavaScript is embedded...

• One can place it within HTML comments to shield it from confusing old browsers:

    <!-- [JavaScript goes here ] -->

• Another consideration is to shield the embedded JavaScript from XML interpretation using:

    <script .... >
    //<![CDATA[
    ..... [JavaScript goes here ]
    //]]></script>
JavaScript and .js files

• Using a .js file is analogous to placing all style rules in an external style sheet. The advantages are similar.

• For that reason, we will occasionally embed our JavaScript code in the head of the document in class for ease of testing and demonstration.
JavaScript and CSS

• JavaScript can also be used to create and modify styles by creating rules that are executed by JavaScript.
• You can reference a style sheet and also JavaScript functions within the same web page.
• The goal is to know when to use which!
JavaScript & CSS

• For now, we will use embedded styles and external stylesheets with our JavaScript so you can see how they are used.

• However we will only use JavaScript to create or modify styles for pages that are generated "on the fly".