Pros and Cons of JavaScript

- Pros:
  - Allows more dynamic HTML pages, even complete web applications
- Cons:
  - Requires a JavaScript-enabled browser
  - Requires a client who trusts the server enough to run the code the server provides
- JavaScript has some protection in place but can still cause security problems for clients
  - Remember JavaScript was invented in 1995 and web-browsers have changed a lot since then

Using JavaScript in your HTML

- JavaScript can be inserted into documents by using the SCRIPT tag

```
<html>
<head>
<title>Hello World in JavaScript</title>
</head>
<body>
  <script type="text/javascript">
    document.write("Hello World!");
  </script>
</body>
</html>
```

Where to Put your Scripts

- You can have any number of scripts
- Scripts can be placed in the HEAD or in the BODY
  - In the HEAD, scripts are run before the page is displayed
  - In the BODY, scripts are run as the page is displayed
- In the HEAD is the right place to define functions and variables that are used by scripts within the BODY

Using JavaScript in your HTML

```
<html>
<head>
<title>Hello World in JavaScript</title>
<script type="text/javascript">
  function helloWorld() {
    document.write("Hello World!");
  }
</script>
</head>
<body>
  <script type="text/javascript">
    helloWorld();
  </script>
</body>
</html>
```

External Scripts

- Scripts can also be loaded from an external file
- This is useful if you have a complicated script or set of subroutines that are used in several different documents

```
<script src="myscript.js"></script>
```

JavaScript Variables

- JavaScript has variables that you can declare with the optional `var` keyword
- Variables declared within a function are local to that function
- Variables declared outside of any function are global variables

```
var myname = "Pat Morin";
```

JavaScript Operators and Constructs

- JavaScript has most of the operators we're used to from C/Java
  - Arithmetic (+, – , *, /, %)
  - Assignment (=, +=, –=, *= /=, %=, ++, ––)
  - Logical (&&, ||, !)
  - Comparison (<, >, <=, >=, ==)
- Note: + also does string concatentation
- Constructs:
  - `if`, `else`, `while`, `for`, `switch`, `case`

Simple User Interaction

- There are three built-in methods of doing simple user interaction
  - `alert(msg)` alerts the user that something has happened
  - `confirm(msg)` asks the user to confirm (or cancel) something
  - `prompt(msg, default)` asks the user to enter some text

```
alert("There's a monster on the wing!");

confirm("Are you sure you want to do that?");

prompt("Enter you name", "Adam");
```

JavaScript Functions

- JavaScript lets you define functions using the `function` keyword
- Functions can return values using the `return` keyword

```
function showConfirm() {
  confirm("Are you sure you want to do that?");
}
```

JavaScript Arrays

- JavaScript has arrays that are indexed starting at 0
- Special version of `for` works with arrays

```
<script type="text/javascript">
  var colors = new Array();
  colors[0] = "red"; colors[1] = "green";
  colors[2] = "blue"; colors[3] = "orange";
  colors[4] = "magenta"; colors[5] = "cyan";
  for (var i in colors) {
    document.write("<div style=\"background-color:"
                   + colors[i] + ";\">"
                   + colors[i] + "</div>\n");
  }
</script>
```

JavaScript Events

- JavaScript can be made to respond to user events
- Common Events:
  - `onload` and `onunload` : when a page is first visited or left
  - `onfocus`, `onblur`, `onchange` : events pertaining to form elements
  - `onsubmit` : when a form is submitted
  - `onmouseover`, `onmouseout` : for "menu effects"
- A complete list of event types is available here
  - http://www.w3schools.com/jsref/jsref_events.asp

Exception Handling

- JavaScript also has `try`, `catch`, and `throw` keywords for handling JavaScript errors

```
try {
  runSomeCode();
} catch(err) {
  var txt="There was an error on this page.\n\n"
          + "Error description: "
          + err.description + "\n\n"
  alert(txt)
}
```

Carleton UNIVERSITY
Canada's Capital University

Comments in JavaScript

- Comments in JavaScript are delimited with // and /* */ as in Java and C++

JavaScript Objects

- JavaScript is object-oriented and uses the same method-calling syntax as Java
- We have already seen this with the `document` object

```
document.write("Hello World!");
```

Carleton UNIVERSITY
Canada's Capital University

Built-In JavaScript Objects

- Some basic objects are built-in to JavaScript
  - String
  - Date
  - Array
  - Boolean
  - Math

JavaScript Strings

- A String object is created every time you use a string literal (just like in Java)
- Have many of the same methods as in Java
  - charAt, concat, indexOf, lastIndexOf, match, replace, search, slice, split, substr, substring, toLowerCase, toUpperCase, valueOf
- There are also some HTML specific methods
  - big, blink, bold, fixed, fontcolor, fontsize, italics, link, small, strike, sub, sup
- Don't use the HTML methods (use CSS instead)
  - This is the worst kind of visual formatting

JavaScript Dates

- The Date class makes working with dates easier
- A new date is initialized with the current date
- Dates can be compared and incremented

```
var myDate = new Date();
myDate.setFullYear(2007,2,14);

var today = new Date();
var nextWeek = today + 7;

if (nextWeek > today) {
  alert("You have less than one week left");
}
```

Carleton UNIVERSITY
Canada's Capital University

JavaScript Arrays and Booleans

- We have already seen the Array class
- The Boolean class encapsulates a boolean value

The JavaScript Math Class

- The Math class encapsulates many commonly-used mathematical entities and formulas
- These are all class methods
  - abs, acos, asin, atan, atan2, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan
- These are all class methods
  - E, LN2, LN10, LOG2E, LOG10E, PI, SQRT1_2, SQRT2

```
if (Math.cos(Math.PI) != 0) {
  alert("Something is wrong with Math.cos");
}
```

JavaScript and the DOM

- The Document Object Model (DOM) is a specification that determines a mapping between programming language objects and the elements of an HTML document
- Not specific to JavaScript

HTML DOM Objects

- Environment objects
  - Window, Navigator, Screen, History, Location, Document
- HTML objects
  - Anchor, Area, Base, Body, Button, Event, Form, Frame, Frameset, Iframe, Image, Checkbox, FileUpload, Hidden, Password, Radio, Reset, Submit, Text, Link, Meta, Object, Option, Select, Style, Table, TableCell, TableRow, TextArea

HTML DOM: Document

- The Document object represents an HTML document and can be used to access all documents in a page
- A Document contains several collections
  - anchors, forms, images, links
- Has several properties
  - body, cookie, domain, lastModified, referrer, title, URL
- Has several useful methods
  - getElementById, getElementsByName, getElementsByTagName, write, writeln, open, close

HTML DOM: Document

- An instance of Document is already created for you, called document

```
function changeF() {
  var cText = document.getElementById("c");
  var fText = document.getElementById("f");
  ...
}

...

<input type="text" id="c" onchange="changeC()">C
<input type="text" id="f" onchange="changeF()">F
```

Carleton UNIVERSITY
Canada's Capital University

HTML DOM: Form Elements

- One of the most common uses of JavaScript is for form validation
- Several HTML DOM classes encapsulate form elements
  - Form, Button, Checkbox, Hidden, Password, Text, Radio, Reset, Submit, Textarea
- **Warning:** Using JavaScript is not a substitute for validating form data in CGI scripts

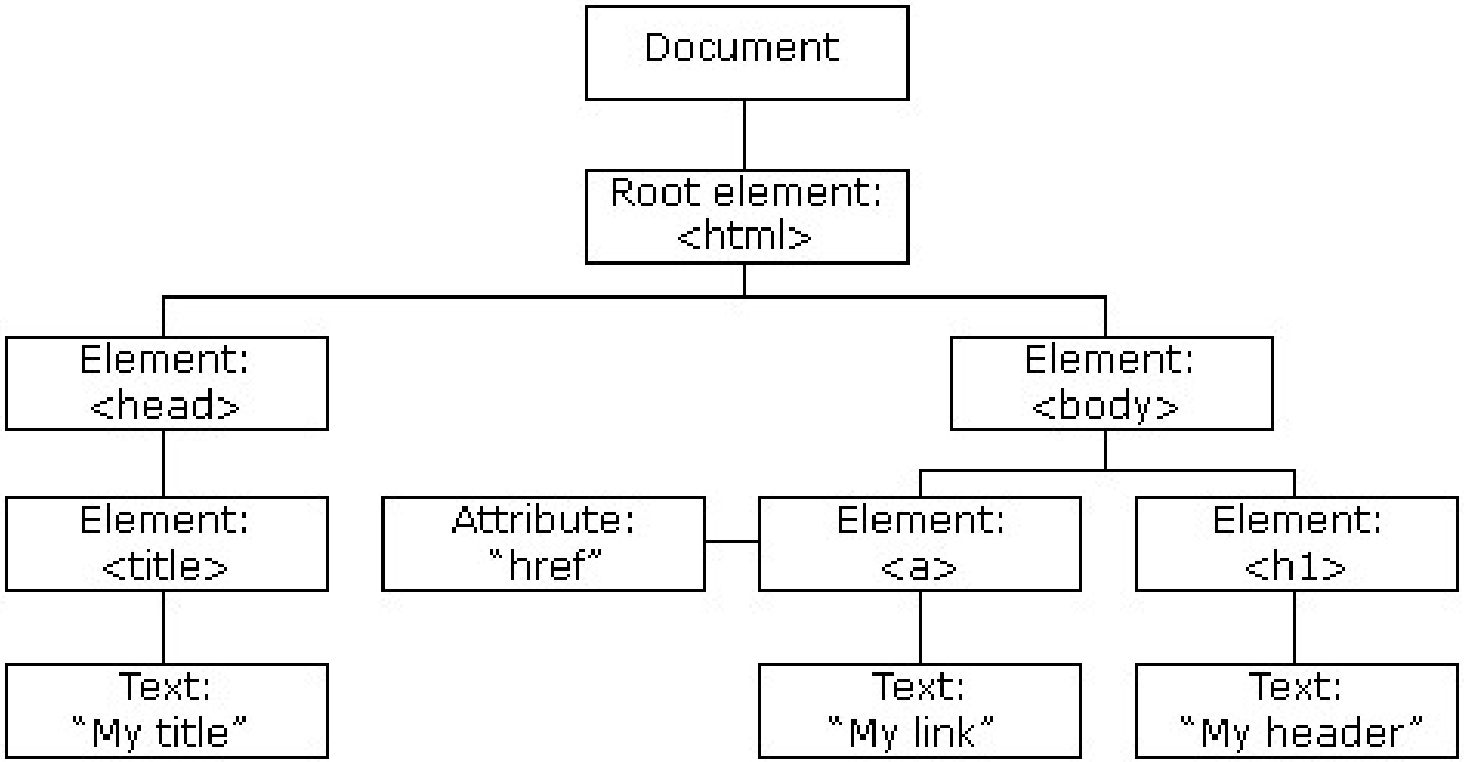
HTML DOM: Text

- A text entry field (`input type="text"`) is encapsulated by a Text object
- Variables
  - value, maxLength, id, size, name, tabindex, readOnly
- Changing these variables has an immediate effect on the displayed data

HTML DOM: The Document Tree

- Accessing elements and changing their properties lets us do simple things like form validation, data transfer etc
- HTML DOM lets us do much more
- We can create, delete, and modify parts of the HTML document
- For this we need to understand the Document Tree

HTML DOM: The Document Tree



Carleton UNIVERSITY
Canada's Capital University

Navigating the Document Tree

- With JavaScript we can navigate the document tree
- document.getElementById(), getElementsByName(), and getElementsByTagName() return nodes in the document tree
- Information can be obtained from
  - nodeName – The tag name
  - nodeValue – The the text of a text node
  - nodeType – The kind of node


Carleton
UNIVERSITY
Canada's Capital University