Javascript

Making Websites Dynamic

JavaScript

Making The Web Dynamic

JavaScript: Why Bother?

- Low Response Time
  - Validation of Forms before submission
- Richer "Application" Experience
  - Google Maps

Competing Technologies

- Other Plugins:
  - Java Applets: Not as well supported
  - Shockwave: Largely replaced by Flash
  - Flash: Best Competitor
- All run within a subsection of a page!
- None as well supported due to need for a plugin to handle the content

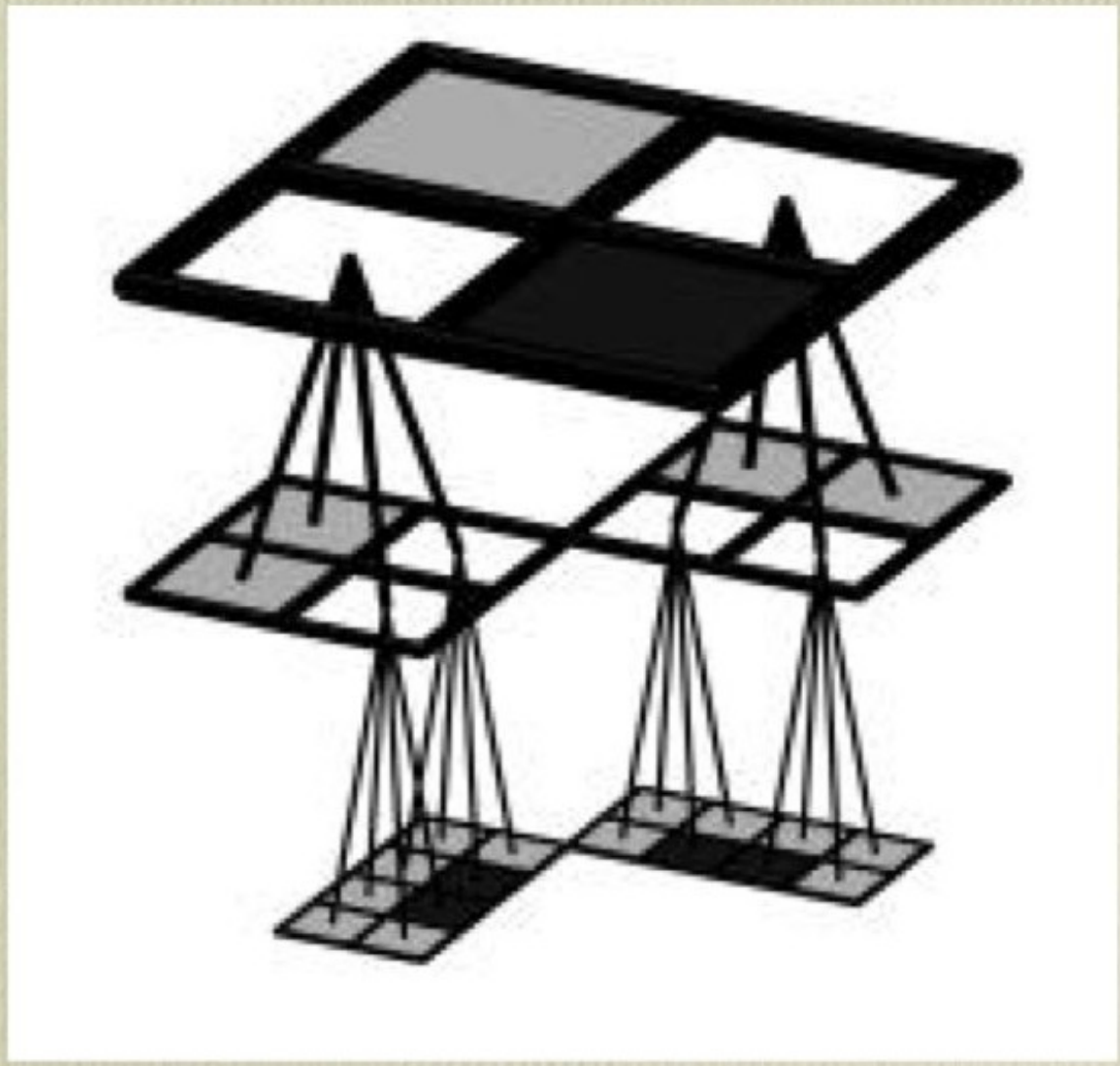
JavaScript: History

- Developed by Brendan Eich as part of Netscape 2.0 in 1995
- Adopted by Internet Explorer and all other browsers since 1996
- "Javascript" name intended to confuse with Sun's Java technology
- Officially named ECMAScript
- Standardized by ECMA International

HTML Revisited

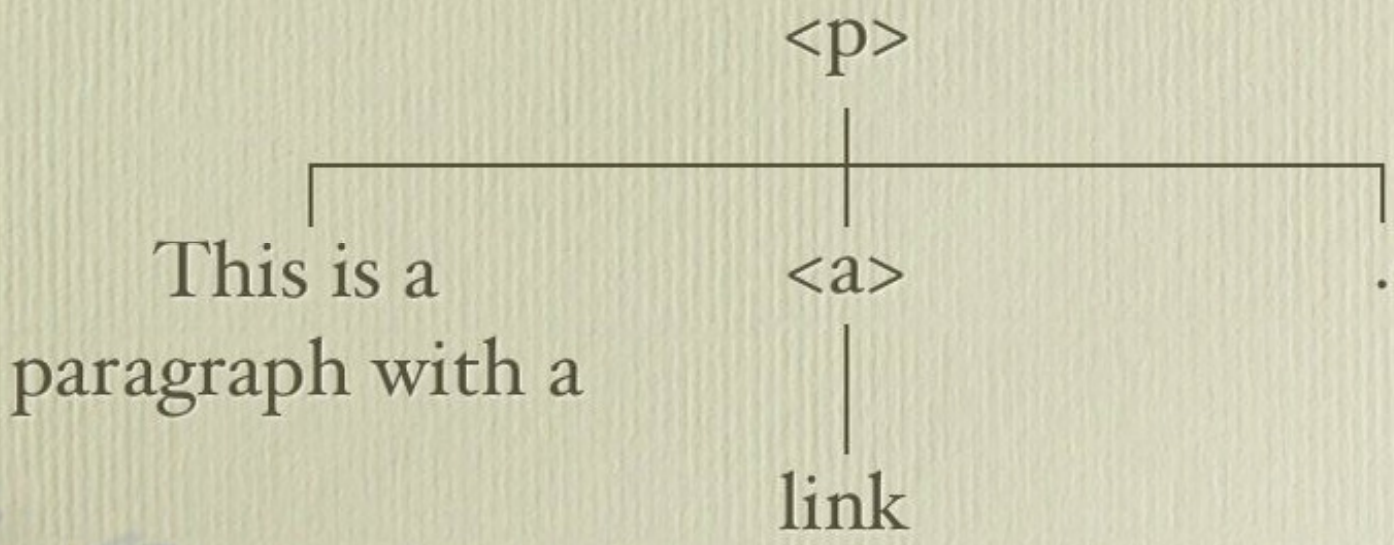
- HTML documents are hierarchical

Tree Data Structures



HTML Elements

```
<p>This is a paragraph with a
<a href="link.html">link</a>.</p>
```



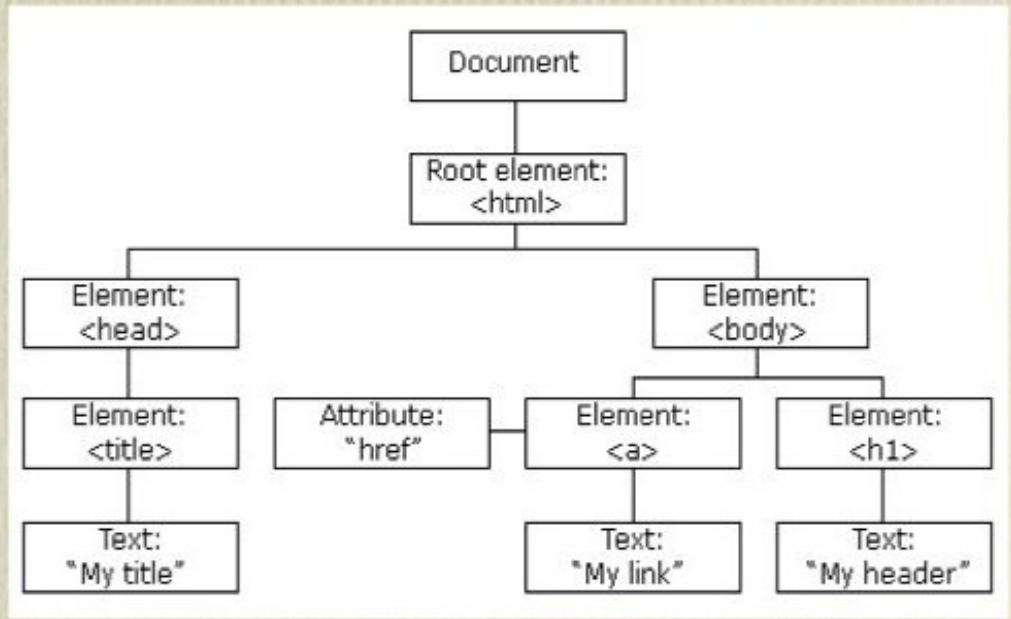
DOM:
Document Object Model

- A Tree Representation of the HTML
- As parsed and understood by browser
- Can be read and modified using JavaScript

DOM: Example

```
<html>
   <head>
      <title>My title</title>
   </head>

   <body>
      <a href="#1">My link</a>
      <h1>My header</h1>
   </body>
</html>
```



Drawing DOM Trees

- <img src="1.jpg">welcome to Ghana!
- <table>
  <tr><td>Today</td><td>1-2</td></tr>
  <tr><td>Tomorrow</td><td>3-4</td></tr>
  </table>
- Paragraph 1 Paragraph 2

JavaScript in HTML

- <script type="text/javascript">
  // <!--
- // -->
  </script>

JavaScript in XHTML

- <script type="text/javascript">
  //<![CDATA[
- //]]>
  </script>

JavaScript: External

- <link type="text/javascript" src="hello.js"></link>
- This is the best way!
- Not always possible

JavaScript: Hello World

- Enter the following in the address bar of your favorite browser:

  javascript:alert('Hello World!');

  - Called Bookmarklets a combination of Bookmark and Applet
  - Uses the javascript: URL Scheme

JavaScript: Syntax

- Inspired by C, C++ and Java
  - var a = 3;
  - // This is a line comment
  - /* This is also a comment */
  - document.body.style.backgroundColor="blue";

Case Sensitive!

JavaScript: Statements

- Statements are commands to the browser
- document.write("Hello World!");
- Statements end with either ; or a newline
- It is better to always use a ;
  - Allows multiple statements on one line

JavaScript: Code

- Code is simply a series of statements
- They are executed in order
- <script type="text/javascript">
  document.write("<h1>This is a heading</h1>");
  document.write(" This is a paragraph. ");
  document.write(" This is another paragraph. ");
  </script>

JavaScript: Event Oriented

- Code is attached to an event
- An event is fired when something happens
  - mouse over an element (onMouseOver)
  - mouse no longer over an element (onMouseOut)
  - document finishes loading (onLoad)
  - form gets submitted (onSubmit)
  - form element changes (onChange)
  - etc...

JavaScript: Example

```
<html>
  <body>
    <h1 onClick="alert('Hello World!');">
      CLICK HERE
    </h1>
  </body>
</html>
```

JavaScript: Variables

- Names to hold values
- Can be thought of as “containers”
- Like in Algebra

- var number;
  number = 3;
- = is assignment
  placing a value
  in a container

JavaScript: Variables

- JavaScript is case sensitive
- Thus, y and Y are two different variables
- Variable names begin with a letter or _
- You don't have to declare variables
  var x = 3; is the same as x = 3;

JavaScript: Variables

- In JavaScript:
  Variables are not "Typed"
    - Not true for all programing languages
- Values are typed
- 3 is a number
- "text" is a string
- x = 3; // x holds a number
  x = "text"; // x holds a string

JavaScript: Operators

- Arithmetic:
  - Addition: +
  - Subtraction: -
  - Multiplication: *
  - Division: /
  - Modulus: %
  - Increment: ++
  - Decrement: --
- Assignment:
  - Basic: =
  - Addition: +=
  - Subtraction: -=
  - Multiplication: *=
  - Division: /=
  - Modulus: %=

JavaScript: Strings

- Strings are treated special with respect to addition

JavaScript: Blocks

- Statements can be grouped into blocks
- {
  alert(“Hello World!”);
  alert(“How is it going?);
  }

JavaScript: Example 2

```
<html>
<head>
   <title>Original Title</title>
</head>
<body>
   <h1 onClick="document.title = 'One';">
      Click To Change Title To One
   </h1>
   <h1 onClick="document.title = 'Two';">
      Click To Change Title To Two
   </h1>
</body>
</html>
```

JavaScript: Comparison & Logic

- == is equal to (value)
- === is exactly equal to (value and type)
- != not equal to
- > greater than
- < less than
- >= greater than or equals
- <= less than or equals

JavaScript: Using Comparison

- Can be used in conditional statements
- var x = 5;
  if (x == 5) alert ("x is 5!");
- Conditional statements take blocks
- if (x == 5) {
      alert("x is 5!");
      document.write("x is 5!");
  }

JavaScript: Logical Operators

- Conditions can be combined with logic
  - && and if (x==5 && y > 6)
  - || or if (x==5 || x==6)
  - ! not if (!(x == 5 || x == 6))

Flow Control

Making Decisions

JavaScript: If Statements

- if (condition) { /* code */ }
- if (condition) { /* code */ }
  else { /* code */ }
- if (condition) { /* code */ }
  else if(condition2){ /* code */ }
  else { /* code */ }

JavaScript: Switch

- Compact form of if ... else if ... else if
- Where all conditions test the same variable

```
if (variable == value) {
    // code block 1
}
else if (variable == value2) {
    // code block 2
}
else {
    // default code
}
```

```
switch (variable) {
case value:
    // code block 1;
    break;
case value2:
    // code block 2;
    break;
default:
    // default code
```

JavaScript: Switch Break!

- Watch out for missing breaks!

```
switch (variable) {
case value1:
    // code block 1;
case value2:
    // code block 2;
    break;
default:
    // default code
```

```
if (variable == value1) {
    // code block 1
}
if (variable == value1 ||
variable == value2) {
    // code block 2
}
else {
    // default code
}
```

JavaScript: Conditional Operator

- The only operator with three terms
- An inline version of an if statement
- (condition) ? value1 : value2
- Example: x = (y==10)? 4 : 6;
  - x == 4 if y == 10 otherwise x == 6

JavaScript: Functions

- Functions give a name to a block
- Functions can take parameters
- function *functionName*(var1, var2, ...)
  {
  // Code Goes Here
  }
- functions can return a value

JavaScript: Functions

```
<html>
 <head>
  <script type="text/javascript">
    function displaymessage()
    {
      alert("Hello World!");
    }
  </script>
 </head>
 <body>
  <form>
    <input type="button" value="Click me!"
                  onClick="displaymessage()" />
  </form>
 </body>
</html>
```

JavaScript: Functions

```
<html>
  <head>
    <script type="text/javascript">
      function multiply(a,b)
      {
        return a*b;
      }
    </script>
  </head>
  <body> 8 * 10 =
    <script type="text/javascript">
      document.write(multiply(8,10));
    </script>
  </body>
</html>
```

JavaScript: Scoping

- Scope is the visibility of a variable
- Functional scoping in JavaScript!
- Destroyed when the function exits
- Variables declared outside functions are "global" and visible everywhere
  - Globals are considered bad

JavaScript: Looping

- while (condition) {
  // code to execute
  }

- do {
  // code to execute
  } while (condition);

JavaScript: Looping

- ```
  var i = 0;
  while (i < 10) {
      // Code to execute
      i++;
  }
  ```
- ```
  for (INIT; CONDITION; UPDATE) {
      // Code to execute
  }
  ```
- ```
  for (i = 0; i < 10; i++) {
      // Code to execute
  }
  ```
  ```

JavaScript: Looping & Arrays

- for (VARIABLE in ARRAY) {
  // Code To Execute
  }
- Arrays are a special type of variable
- Arrays hold multiple values in one name
  with numerical indexes
- var people = new Array();
  people[0] = “Nick”;
  people[1] = “Viola”;
- var people = new Array(“Nick”, “Viola”); // Constructor
- var people = [“Nick”, “Viola”]; // The Same

JavaScript: Error Handling

- try {
  // Code which may have problems
  } catch (err) {
  // Deal with the problem here
  }

- throw err;
  allows you to jump to a catch

JavaScript: Error Handling

```
try {
   var val = prompt("What is 5+3?");
   if (val == 8) {
      alert("That's correct!");
   } else {
      throw val;
   }
} catch (err) {
   alert("Wrong! You entered: " + err);
}
```

JavaScript: Dialogs

- alert("Hello World!");

- var ok=confirm("Okay?");
  if (ok) {
      alert("You pressed OK!");
  } else {
      alert("You pressed Cancel!");
  }

- var name=prompt("Your Name?", "Nick");
  alert("Hi " + name + "!");