

Introducing Modern JavaScript

Modern JavaScript Programming

Doc. v. 0.1 - 25/03/09



Wildan Maulana | [wildan \[at\] tobethink.com](mailto:wildan@tobethink.com)

Object-Oriented JavaScript

```
// The constructor for our 'Lecture'  
// Takes two strings, name and teacher  
function Lecture( name, teacher ) {  
    // Save them as local properties of the object  
    this.name = name;  
    this.teacher = teacher;  
}  
  
// A method of the Lecture class, used to generate  
// a string that can be used to display Lecture information  
Lecture.prototype.display = function(){  
    return this.teacher + " is teaching " + this.name;  
};
```

```
// A Schedule constructor that takes in an  
// array of lectures  
function Schedule( lectures ) {  
    this.lectures = lectures;  
}
```

```
// A method for constructing a string representing  
// a Schedule of Lectures  
Schedule.prototype.display = function(){  
    var str = "";  
  
    // Go through each of the lectures, building up  
    // a string of information  
    for ( var i = 0; i < this.lectures.length; i++ )  
        str += this.lectures[i].display() + " ";  
  
    return str;  
};
```



Using the Class

```
// Create a new Schedule object and save it in
// the variable 'mySchedule'
var mySchedule = new Schedule([
    // Create an array of the Lecture objects, which
    // are passed in as the only argument to the Lecture object
    new Lecture( "Gym", "Mr. Smith" ),
    new Lecture( "Math", "Mrs. Jones" ),
    new Lecture( "English", "TBD" )
]);

// Display the Schedule information as a pop-up alert
alert( mySchedule.display() );
```



Testing Your Code

- Use Firebug plug-in for Firefox



Packaging for Distribution

- Use namespace to avoid conflict with other library
 - For example : public interface library developed by Yahoo

```
// Add a mouseover event listener to the element that has an
// ID of 'body'
YAHOO.util.Event.addListener('body','mouseover',function(){

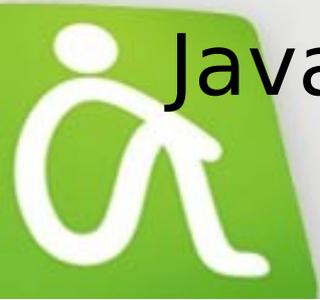
    // and change the background color of the element to red
    this.style.backgroundColor = 'red';

});
```



Unobtrusive DOM Scripting

- Writing unobtrusive code implies a complete separation of your HTML content: the data coming from the server, and the JavaScript code used to make it all dynamic
- Writing modern, unobtrusive code consists of two aspects : the Document Object Model (DOM), and JavaScript events



The Document Object Model

- The DOM was constructed to provide an intuitive way for developers to navigate an XML hierarchy (remember : valid HTML is simply a subset of XML)
- Next : Using DOM to Locate and Manipulate Different DOM Elements



```
<html>
<head>
  <title>Introduction to the DOM</title>
  <script>
    // We can't manipulate the DOM until the document
    // is fully loaded
    window.onload = function(){

      // Find all the <li> elements in the document
      var li = document.getElementsByTagName('li');

      // and add a red border around all of them
      for ( var j = 0; j < li.length; j++ ) {
        li[j].style.border = '1px solid #000';
      }

      // Locate the element with an ID of 'everywhere'
      var every = document.getElementById( "everywhere" );

      // and remove it from the document
      every.parentNode.removeChild( every );

    };
  </script>
</head>
<body>
  <h1>Introduction to the DOM</h1>
  <p class="test">There are a number of reasons why the DOM is awesome, here are some:</p>
  <ul>
    <li id="everywhere">It can be found everywhere.</li>
    <li class="test">It's easy to use.</li>
    <li class="test">It can help you to find what you want, really quickly.</li>
  </ul>
</body>
</html>
```

Events

- Events are the glue that holds together all user interaction within an application
- JavaScript events are complex and diverse



```
<html>
<head>
  <title>Introduction to the DOM</title>
  <script>
    // We can't manipulate the DOM until the document
    // is fully loaded
    window.onload = function(){

      // Find all the <li> elements, to attach the event handlers to them
      var li = document.getElementsByTagName('li');
      for ( var i = 0; i < li.length; i++ ) {

        // Attach a mouseover event handler to the <li> element,
        // which changes the <li>s background to blue.
        li[i].onmouseover = function() {
          this.style.backgroundColor = 'blue';
        };

        // Attach a mouseout event handler to the <li> element
        // which changes the <li>s background back to its default white
        li[i].onmouseout = function() {
          this.style.backgroundColor = 'white';
        };

      }

    };
  </script>
</head>
<body>
  <h1>Introduction to the DOM</h1>
  <p class='test'>There are a number of reasons why the DOM is awesome, here are some:</p>
  <ul>
    <li id='everywhere'>It can be found everywhere.</li>
    <li class='test'>It's easy to use.</li>
    <li class='test'>It can help you to find what you want, really quickly.</li>
  </ul>
</body>
</html>
```

JavaScript and CSS

Cascading style sheets (CSS) serve as the standard for laying out simple, unobtrusive web pages that still afford you (the developer) the greatest amount of power while providing your users with the least amount of compatibility issues.

Ultimately, dynamic HTML is about exploring what can be achieved when JavaScript and CSS interact with each other and how you can best use that combination to create impressive results.

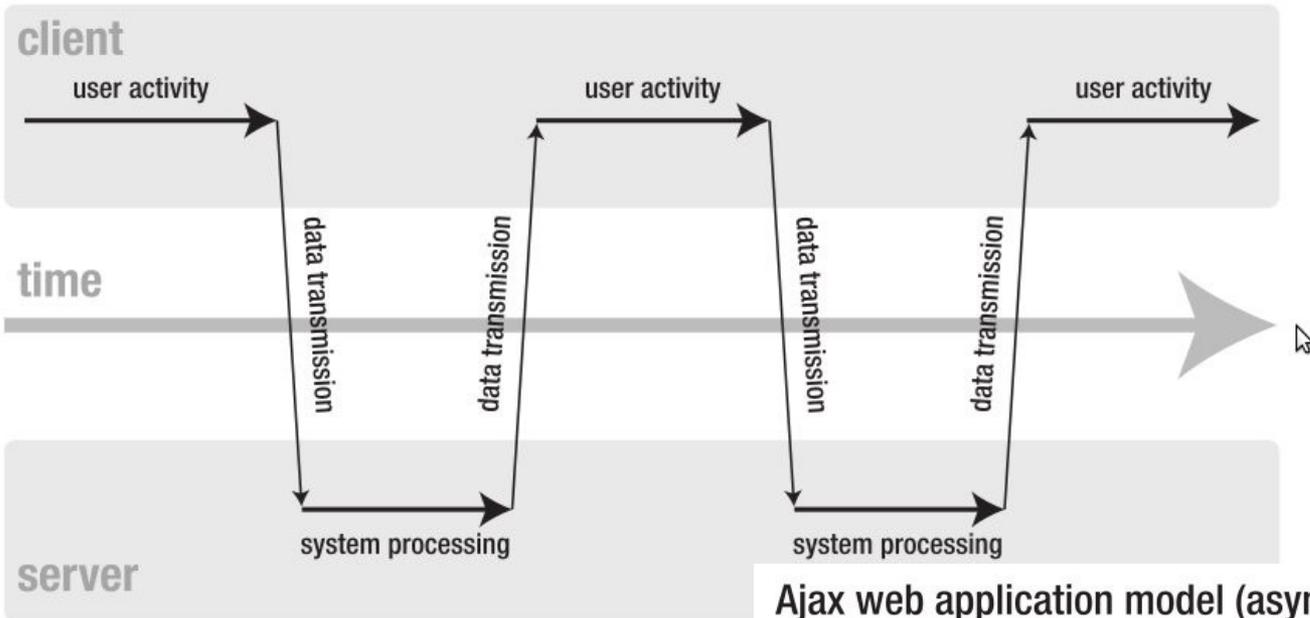


Ajax

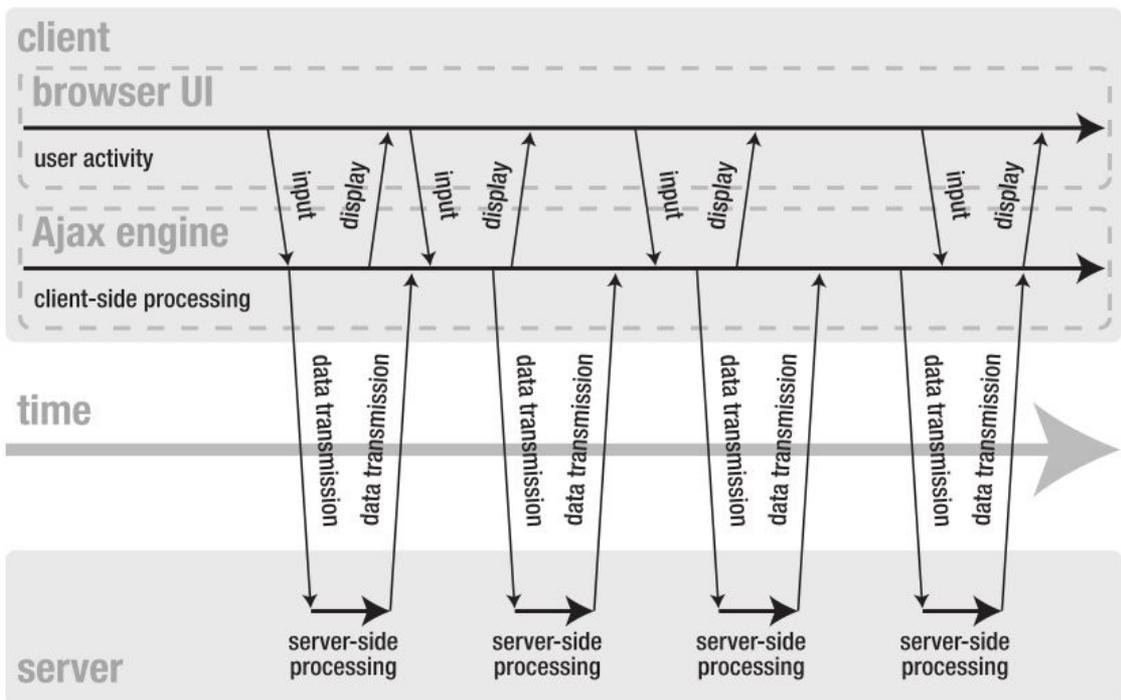
- Ajax, or Asynchronous JavaScript and XML, is a term coined in the article “Ajax: A New Approach to Web Applications” (<http://www.adaptivepath.com/publications/essays/archives/000385.php>) by Jesse James Garrett
- The term Ajax encompasses hundreds of permutations for data communication, but all center around a central premise: that additional requests are made from the client to the server even after the page has completely loaded



classic web application model (synchronous)



Ajax web application model (asynchronous)



Browser Support

- <http://developer.yahoo.com/yui/articles/gbs/gbs.html>

	Win 98	Win 2000	Win XP	Mac 10.0	Mac 10.2	Mac 10.3	Mac 10.3.x	Mac 10.4
IE 7.0	n/a	n/a	A-grade	n/a	n/a	n/a	n/a	n/a
IE 6.0	A-grade	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a
IE 5.5	A-grade	A-grade	n/a	n/a	n/a	n/a	n/a	n/a
IE 5.0	C-grade	C-grade	n/a	C-grade	C-grade	C-grade	C-grade	C-grade
Netscape 8.0	X-grade	X-grade	A-grade	n/a	n/a	n/a	n/a	n/a
Firefox 1.5	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Firefox 1.0.7	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade	A-grade
Mozilla 1.7.12	X-grade	X-grade	A-grade	X-grade	X-grade	X-grade	X-grade	X-grade
Opera 8.5	X-grade	X-grade	A-grade	C-grade	C-grade	C-grade	X-grade	X-grade
Safari 1.0	n/a	n/a	n/a	X-grade	n/a	n/a	n/a	n/a
Safari 1.1	n/a	n/a	n/a	X-grade	X-grade	n/a	n/a	n/a
Safari 1.2	n/a	n/a	n/a	X-grade	X-grade	X-grade	n/a	n/a
Safari 1.3	n/a	n/a	n/a	n/a	n/a	X-grade	A-grade	n/a
Safari 2.0	n/a	n/a	n/a	n/a	n/a	n/a	n/a	A-grade



Consistent Development Based - Features

- Core JavaScript 1.5: The most current, accepted version of JavaScript. It has all the features necessary to support fully functional object-oriented JavaScript. Internet Explorer 5.0 doesn't support full 1.5, which is the primary reason developers don't like to support it.
- XML Document Object Model (DOM) 2: The standard for traversing HTML and XML documents. This is absolutely essential for writing fast applications.
- XMLHttpRequest: The backbone of Ajax—a simple layer for initiating remote HTTP requests. All browsers support this object by default, except for Internet Explorer 5.5–6.0; however, they each support initiating a comparable object using ActiveX.
- CSS: An essential requirement for designing web pages. This may seem like an odd requirement, but having CSS is essential for web application developers. Since every modern browser supports CSS, it generally boils down to discrepancies in presentation that cause the most problems. This is the primary reason Internet Explorer for Mac is less frequently supported.





Q&A



Reference

- Pro JavaScript Techniques, *John Resig*, Apress

