Getting to know JavaScript

Pete Alston

CSCI 4448/5448 - Object Oriented Analysis & Design
• In the beginning ...
  – Static vs. Dynamic Content
• Issues with Client Side Scripting
• What is JavaScript?
• Syntax and the Document Object Model
• Moving forward with JavaScript
  – AJAX
• Libraries and Frameworks
• JavaScript 2.0 – A New Beginning?
• Resources
In the beginning ...

- HTML allowed people to display content on the World Wide Web (WWW)
  - Fairly straight-forward tags and easy to learn
- As the WWW has evolved, demand for greater control over the ‘look and feel’ of a page
  - WWW is a dynamic medium and HTML is not able to handle this
  - Web pages need to interact with their users and vice-versa
- Need to make use of a *scripting language* to handle interactivity
  - Ability to render/produce HTML to be displayed in the browser
  - Users want *dynamic* web pages; not *static*
Static vs. Dynamic Content – Why does it matter?

- Pre-determined vs. Interactive
- Persistent Layout vs. Personalisation
- Message vs. Service
- HTML/CSS vs. PHP, ASP, Ruby
- Implementation & Maintenance
Issues with Client Side Scripting

- Performance
- Dependency
- Incomplete Scripts
- Identification
- Flexibility
  - Data
  - Layout
- Language
What is JavaScript?

- JavaScript is **not** Java
  - Although there are some similarities regarding syntax and functionality
    - Cat.paws.back.left
    - Car.start(), document.write()
- Java is a fully functional object-oriented programming language
  - JavaScript is designed to run within a Web browser and interact with HTML and DOM (Document Object Model)
- JavaScript can be placed anywhere on a Web page
  - Just need to be enclosed in a `<script>` tag
What is JavaScript?

- A *client side* scripting language which can provide interactivity within Web pages
- Developed by Netscape in 1995
  - Has evolved a lot since its creation
  - Depending on the browser, it has also developed in different directions
- It is an *object based* language
  - It uses ‘prototypes’ to simulate inheritance, which means you can define classes and use them at the same time
- Client side means it works on your machine
  - Your Web browser processes the scripts, not the server
Alternatives to JavaScript

- ECMAScript
  - JavaScript
  - ActionScript
- VBScript/JScript
- E4X (ECMAScript for XML)
- LiveScript
- TCL/TK
- XUL/XSLT
ECMAScript – What’s that?

• Standardised scripting language
  – Widely used on the Web
• Developed out of a need for standards
  – JavaScript & JScript (Microsoft version of JavaScript) were supposed to be the same, but they are not
• Number of dialects or implementations of ECMAScript are available today
  – ActionScript
  – Objective-J
  – WMLScript
ECMAScript Dialects - Syntax

- **JavaScript**
  - `document.write('Hello World!');`
- **ActionScript**
  - `var greet:TextField = new TextField();
    greet.text = "Hello World";
    this.addChild(greet);`
- **JScript**
  - `document.write(“Hello World”);`
Syntax

- JavaScript offers similar functionality to Java
  - Variables
  - Operators, Comparisons & Conditions
  - Control Structures
    - if, else, for, switch, etc.
  - Functions
    - You can stop the browser executing code at runtime by putting it in a function
    - You then make a call to the function to execute the code

- In many ways, it looks very similar to Java
  - But it isn’t!
Syntax – Example (1)

```html
<html><head></head>
<body>
<script type="text/javascript">
    document.write("<h1>Hello World</h1>"神通);
</script>
</body>
</html>
```

Accessing object & method
Syntax – Example (2)

```html
<html><head>
<script type="text/javascript">
function myfunc(value) {
    alert(value);
}
</script></head>
<body><form>
<input type="button" onclick="myfunction('Howdy')" value="Ok"></form>
</body>
</html>
```
if (condition)
    {
        some code
    }
else
    {
        some code
    }

var colorChoice = prompt(“What is your favourite color?”);

if (colorChoice==“Blue”) {
    { document.write(“That’s the right answer!”); }
} else {
    { document.write(“Sorry, that’s not correct!”); }
}
Syntax – Example (4)

Conditions

- And = &&
  ```javascript
  if (hour==12 && minute==0) {
    alert("it's noon")
  }
  ```

- Or = ||
  ```javascript
  if (hour==11 || hour==10) {
    alert("it's less than 2 hours till noon")
  }
  ```

- Not = !
  ```javascript
  if (!(hour==11)) {
    alert("it's more than 1 hour till noon")
  }
  ```

```javascript
var hour = 11;
var minute = 0;
```
Figure 12.1 The tree structure, showing nodes is just another way of looking at how a HTML page is organised
Source: Negrino & Smith (2007:283)
DOM – Accessing data

• The DOM proves to be a useful tool when trying to add interactivity to your Web page
  – Use the tree structure to trace the path through the nodes

• How do we get access to that data?
  – Follow the path!

```html
<html><head></head>
<body>
  <form name='frmsample'>
    <input type='text' name='txtname'>
  </form>
</body></html>
```

```
document.frmsample.txtname.value;
```
• You can use the DOM to retrieve data/values contained within a web page

```html
<form name='frmsample'>
  <input type="text" name="txtname">
  <p id="text"></p>
</form>
```

• How do we get access to that data?

```javascript
var uname = document.frmsample.txtname.value;
alert("Text is: " + uname);
document.getElementById('text').innerHTML = uname;
```
Moving forward with JavaScript

- The WWW is **always** changing
  - Advances have given us Google Maps, Flickr, MyYahoo
- These sites function more like desktop applications as opposed to Web sites
  - Faster, more responsive User Interface (UI)
  - Superior interaction and a better User Experience (UE)
- Technology behind these sites was AJAX
  - Asynchronous JavaScript and XML
- It is not a new technology ....
  - Just a new way of using existing technologies
What is AJAX?

• Coined by Jesse James Garret in 2005
• A technique combining long standing Web technologies
  – XHTML and CSS for structure and presentation
  – DOM for displaying and manipulating pages
  – XML for formatting data
  – XMLHttpRequest object to transfer data
    • Similar to Microsoft’s ActiveX Object (the yellow bar in IE?!?)
    • JavaScript to display and interact with the above
• Web sites can offer more functionality by communicating with a Web server through JavaScript
  – Normally, you need to make use of a server side scripting language to communicate with a server (e.g. PHP, Ruby, etc.)
Traditional Web Application

Synchronous page requests and refreshes
AJAX Web Application

Asynchronous page refreshes

JavaScript Today

- As we have seen, it is pretty simple to make web pages dynamic
  - Collecting user data, displaying a message, passing variables
- You could create your own functions
  - But it takes time and effort and you may get it wrong(!)
- What if we could make it simpler
  - Use pre-defined functions and features to ensure consistency across your Web site
  - All you need to do is to make a reference to them
  - In most cases, it reduces the amount of coding
Libraries and Frameworks

- Pre-written functions which can aide in the development of Web apps
  - Serve a number of purposes (Animation, DOM access, AJAX)
  - Allow integration with other languages such as CSS, PHP, Ruby
- Similar thing when coding in ActionScript/Java
  - import flash.display.Sprite;
- Similar to referencing an external CSS
  - <link href="mystyle.css" rel="stylesheet" media="screen" />
- There is more of a demand for dynamic Web apps
  - Less emphasis on simple things, more on AJAX
Libraries vs. Frameworks

- Frameworks are more like APIs and are different to Libraries
  - Enhances everything, not just the Document Object Model
    - Arrays, Strings, Data Types, Classes
  - Most ‘libraries’ state that they are frameworks
    - Some may offer the functionality, but they not be known for it
- Why the fuss?
  - Libraries can be easier to navigate understand
  - Frameworks take the whole of the JavaScript language into account
    - But they will still offer all the ‘goodies’ of a library
What is available

- Libraries
  - jQuery (Although it does have functionality to be a framework), Yahoo UI (YUI), MochiKit, Dojo Toolkit, script.aculo.us, Google Web Toolkit (GWT),
- Frameworks
  - Prototype, ExtJS, MooTools, Spry
- What do they offer?
  - DOM Manipulation & Traversal
  - Animation/Effects
  - AJAX & CSS
  - Plug-ins
Examples

- jQuery
  - W3Schools.com/jquery / Animated Robot
- Yahoo UI
- Script.aculo.us
  - http://script.aculo.us / Drag and drop functionality
- MooTools
  - http://mootools.net / Toggle display
- ExtJS
  - http://www.sencha.com/products/js/ / Drag and Drop
Making a choice

• Features
  – What is that you are trying to create? Are any ruled out initially?

• Documentation & Examples
  – Clear instructions, Tutorials, Demos readily available?

• File Size
  – Is the whole of the file needed, or can you just make use with certain aspects? Is there a minified version?

• License
  – Is it free? Or is there a small print detailing a charge for commercial use?

• Ease of use
  – How easy is it to pick up? Libraries are easy to pick up
JavaScript 2.0 –
A New Beginning?

- The next ‘major’ revision of the language
  - Also known as ECMAScript 4
  - Currently under development, now under the code name ‘Harmony’
- Motivated by the need to ‘achieve better support for programming in the large’ (Horwat, n.d.)
  - Does not mean creating large programs, rather programs
    - Written by more than one person
    - Assembled from packages
    - Live in heterogeneous environments
    - That evolve over time
- New functionality makes JavaScript look more like a *programming* language
  - However, it is not meant be a high performance programming language for writing large, complex programs
  - Isn’t that what Java is for?!
- Two key features
  - Optional types and checking
  - Ability to support classes and objects
JavaScript 2.0 – A New Beginning?

- Optional types & type checking
  - More similar to Java when creating variables, e.g.
    - `var username = ‘Pete’` (old)
    - `var username:person = ‘Pete’` (new)

- Ability to create classes
  - No need for the use of functions to *simulate* inheritance
  - Can create classes in the same was as Java (for example)

- Other features include
  - Versioning
  - Conditional compilation
    - Creating code to run in a number of environments
  - Better mappings for data types and interfaces
  - And more ....

- All in all, a much more structured language for the Web 😊
  - But it has not been implemented for use yet!
    - JavaScript 1.8.5 is the latest stable version, but does not support any of the above!
Summary

• JavaScript is a complex language
  – Each section in this presentation could have one presentation to itself!

• JavaScript ≠ Java
  – Although there are some similarities

• It is powerful language when it comes to creating content for the Web
  – Has a number of benefits compared to Adobe Flash
  – But it is not the only language

• Libraries and Frameworks are designed to make it easier
  – You just need to pick the one(s) best for your project

• JavaScript 2.0 is an attempt to make the language more object oriented
  – Should make it easier when trying to create Web applications
  – But it is not ready for implementation yet

• There are a LOTS of resources available to help you learn JavaScript
  – Some are mentioned on the next slide
Three resources I would recommend ...

• W3Schools.com
  – [http://www.w3schools.com/](http://www.w3schools.com/)
    • Resources on JavaScript, AJAX, VBScript and others. One of the best sites I have found to give you an introduction to a particular language

• “JavaScript and AJAX for the Web”
  – Tom Negrino and Dori Smith, 7th edition
    • An excellent book, with plenty of examples you can use to help you learn the language

• “Head First JavaScript”
  – Michael Morrison
    • Part of the ‘Head First’ series, these books are really simple to understand and explain complex issues in a more ‘interesting’ way!
References/Bibliography
Ten Promising JS Frameworks - http://sixrevisions.com/javascript/promising_javascript_frameworks/