External Libraries

Chapter 10

1. Introduction

2. Prototype
   - Accessing Styles
   - DOM Traversal
   - Forms

3. Scriptaculous

- There are many libraries available to use that make JavaScript programming easier and add new features.
- The same code doesn't always work the same on each browser.
- Most of the common features you want to implement have probably already been implemented.
- http://www.jsdb.io/ is a useful website for finding libraries.
- jQuery, Prototype and MooTools are 3 popular libraries.

- To be consistent with the book, we will look at Prototype and Scriptaculous.
- Prototype is implemented in a large library. Simply download the js file from the website and load it to your page:
  <script src="prototype.js" type="text/javascript"></script>
- The API is located at http://api.prototypejs.org/

- Prototype adds many features to JavaScript.
- Methods for Numbers can be found at http://api.prototypejs.org/language/Number/
- Arrays become much easier to deal with. The API can be found at http://api.prototypejs.org/language/Array/
  Several methods you should be aware of: clone, compact, first, flatten, last, uniq, without.
- String manipulation becomes easier. The API is at http://api.prototypejs.org/language/String/
- There are several methods available to all objects. API: http://api.prototypejs.org/language/Object/

- **document.getElementById** is a very common method call. Prototype shorthands this by using the $ function.

```
//the old way
document.getElementById("myParagraph").innerHTML = "New text
    here";
//using Prototype
$("myParagraph").innerHTML = "New text here";
```

- Several methods within Prototype only work if you use the $ function. All of the following methods are called with $("someId").methodName(...)
- **scrollTo** - scrolls page until the element appears at the top of the page.
- **remove** - causes the element to remove itself from the page.
- **identify** - gives an element an id if it doesn't have one and returns the element's id.
- Many of these methods return the current element allowing you to chain method calls in one line:

```
$("myID").cleanWhiteSpace().show().scrollTo();
```

- Many more methods at http://api.prototypejs.org/dom/Element/

- Reading styles for specific elements is surprisingly difficult and not standard.
- Prototype has a **getStyle** method which returns the element's current value for that property. For example:

```
var size = $("someId").getStyle("font-size");
$("someId").style.fontSize = size - 2;
```

- However, the above is not unobtrusive. We should use className's.
- Prototype has the following methods for classes: addClassName, hasClassName, removeClassName and classNames.

- Traversing the DOM tree can be difficult because of attribute and text nodes.
- Prototype provides methods that target only element nondes.
- **ancestors** - an array of objects that are ancestors of the current node.
- **siblings** - an array of objects that are siblings of the current node (elements only).
- **childElements** - an array of objects that are children of the current node (elements only).
- Other notable methods include: previousSiblings, nextSiblings, descendants.

- Some browsers do not support querySelector.
- The **$$** function is essentially equivalent to document.querySelectorAll. For example, this returns an array of paragraphs with the className of temp:

```
var getParas = $$("p.temp");
for(var i = 0; i < getParas.length; i++){
    getParas[i].show();
}
```

- A common pattern is to get a DOM object for a form and ready or modify it's value.
- $F accepts a form control's id and returns the value.

```
var text = $F("someTextBox"); //get text from text box
```

- Other Prototype methods useful for form manipulation include: activate, clear, disable, enable, present, setValue.
- http://api.prototypejs.org/dom/Form/Element/ contains a complete list.

- Small visual enhancements are difficult to code: animation, fading in/out, highlighting, drag-and-drop, auto-complete, in-place editors, . . .
- Scriptaculous is built upon the Prototype library.
- Must include Prototype before Scriptaculous.
- Can be found at http://script.aculo.us/
- It is relatively new and not entirely polished. You may encounter bugs.

Example

```
<script src="prototype.js" type="text/javascript"></script>
<script src="scriptabulous.js" type="text/javascript"></script>
<script src="yourScript.js" type="text/javascript"></script>
```

- Always access elements using $ or $$.
- Calling a method is rather simple:

```
$("strawberry").shake();
```

- Check out http://madrobby.github.io/scriptaculous/demos/ for demos on what can be done.