Web Application Architectures

Module 6: Presentation/User Interface
Lecture 7: JavaScript & JQuery

THE UNIVERSITY of NEW MEXICO

JavaScript

- JavaScript is a lightweight, interpreted programming language that was designed to be embedded within, and provide scripting capabilities to, any application.
- A scripting language is used to write "scripts" that are distinct from the core code of an application. Scripts are often written in a language that is different from that of the core application, and they are typically intended to be created/modified by end users.
- Client-side JavaScript combines the scripting capabilities of the JavaScript interpreter with the document object model (DOM) defined in a web browser, enabling executable content to be distributed over the web. All major browsers include a JavaScript interpreter (engine).

JavaScript

- It is also possible to run JavaScript in other environments.
  Ex. Node.js is a server-side JavaScript application for developing networking applications.

- JavaScript is primarily used to write functions that are embedded in or included from HTML pages delivered by a web server, for the purpose of dynamically modifying the HTML elements on client-side pages through the DOM.

- Because JavaScript code runs locally in the browser, it can quickly respond to user actions, making the web application as a whole more responsive.
  Ex. JavaScript can be used to update a portion of webpage, or to perform client-side validations, prior to submitting form data to the web application.

The JavaScript Language

THE UNIVERSITY of NEW MEXICO

- The syntax of the JavaScript language is highly influenced by C; however JavaScript is actually a multi-paradigm language, supporting the prototype-based, functional, imperative and scripting approaches to programming.

- It's easy to access the JavaScript interpreter in your browser:

  - Put JavaScript code in an HTML file, enclosed within the `<script>` tag, and render the file in your browser.

  - Use the JavaScript console provided as a part of the "developer tools" that come with your browser.

JavaScript Interpreters

THE UNIVERSITY of NEW MEXICO

- Consider the recurrence relation for computing the Fibonacci numbers:

$$F(n) = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F(n-1) + F(n-2), & \text{if } n > 1 \end{cases}$$

- Create a file `fib.html`, containing the following JavaScript code:

```
<script>
  document.write("<h2>Table of Fibonacci Numbers</h2>");
  for (i=0, j=1, k=0, f=0; i<50; i++, f=j+k, j=k, k=f) {
    document.write("Fibonacci (  + i +  ") = " + f);
    document.write("<br>");
}
</script>
```

- To execute this code, in your browser address bar type:

```
file://<path_to_file>/fib.html
```

JavaScript and Browser Security

- Client-side JavaScript opens up the possibility for authors to deliver malicious scripts to the browser.
- Browsers guard against this using two strategies:
  - JavaScript code is run in a sandbox that only allows web-related actions to be performed, not general-purpose programming tasks (no writing to disk, creating files, etc.).
  - JavaScript code is constrained by the same origin policy – scripts from one website do not have access to information such as usernames, passwords, or cookies from other websites.

JavaScript Libraries

THE UNIVERSITY of NEW MEXICO

- Numerous JavaScript libraries, containing pre-written JavaScript code, have been developed for the purpose of making the the development of JavaScript-based web applications easier.
- The most popular JavaScript library is jQuery, and it is now supported by default in Rails.
- An entire ecosystem has built up around jQuery, including companion libraries (e.g., jQuery UI) and plug-ins (both official and unofficial).
- jQuery supports the notion of unobtrusive JavaScript – the separation of *behavior* from document *structure*. With unobtrusive JavaScript, you never embed *any* JavaScript expressions or statements within the body of an HTML document, either as attributes of HTML elements (such as onclick) or in script blocks.

- The jQuery developers placed a high priority on ensuring that it worked consistently across all major browsers.

- The focus in jQuery is on retrieving elements from HTML pages, and performing operation on them.

- Elements are retrieved via selectors, using the same selectors that are in CSS. To collect a group of elements, you pass the selector to the jQuery function as follows:

  ```
  jQuery(<selector>)
  ```

- You will more likely encounter the shortcut version of this function call:

  ```
  $(<selector>)
  ```

  This function returns a JavaScript object containing an array of DOM elements that match the selector.

- The returned elements are actually "wrapped" with additional functions (methods), and these elements are therefore referred to as the wrapped set. Available methods: `api.jquery.com`.
  Ex.

```
$("div.byebye").hide();
```

  This will hide all of the `div` elements in the document that belong to the class `bye-bye`.

- Many of the jQuery methods also return a wrapped set, so it's common to chain methods in jQuery.
  Ex.

```
$("div.byebye").hide().addClass("removed");
```

  This will add a new class, called `removed`, to the hidden div elements.