EBU6501 – Middleware

Week 3, Day 4: Advanced JavaScript Programming with JSON

Gokop Goteng & Ethan Lau

Queen Mary
University of London

Original version created by: Dr. Gokop Goteng

Lecture Aim and Outcome

- Aim
  - **The aim of the session is to enable students to be proficient in advanced JavaScript programming**

- **Lecture Outcomes/Objectives:**
  - **The objectives/outcomes of the session are:**
    - **To train students to be creative programmers**
    - **To enable students to use their JavaScript programming skills to develop real-life software applications**
    - **Students should know when to use scripting programming languages**

Lecture Outline

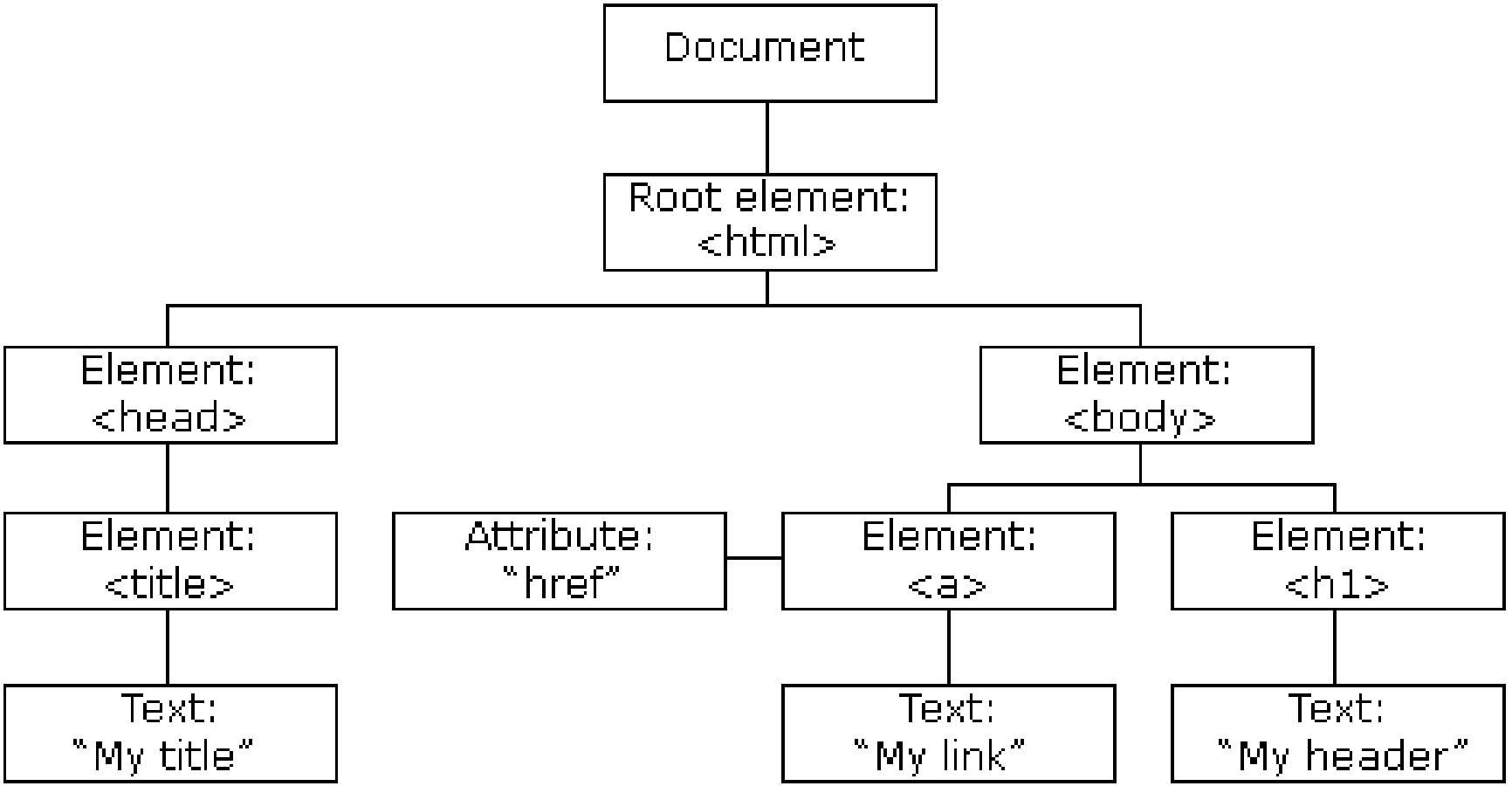
- Revision on JavaScript Programming
  - Discussion on pre-lecture online quiz and work
- Object-Oriented JavaScript Programming
- JS Functions
- Input and Output Commands in JS
- Cookies
- Session Cookies
- jQuery
- JSON
- Scenario of where to use JSON in Real-Life

Revision on JavaScript Programming

- What is a JavaScript?
- What is DOM?
- Write a simple “Hello World” JavaScript program

Revision on JavaScript Programming

Describe this Diagram



Object-Oriented JavaScript Programming

- Remember how to define objects:
  - Example
    - var myUsers = {“myFirstName”: “Wang”, “mySurName”:”Lu”};
- You can also create objects as in other high level languages in advanced JavaScript programming
  - Example:

```
var userObject = new Object();
userObject.lastLoginTime = new Date();
alert(userObject.lastLoginTime);
```

Object-Oriented JavaScript Programming

- You can use the dot (.) or [] operators to achieve the same result
  - Th following examples will print the same results:

```
var userObject = new Object();
userObject.lastLoginTime = new Date();
alert(userObject.lastLoginTime);
```

Or

```
var userObject = {}; // equivalent to new Object()
userObject["lastLoginTime"] = new Date();
alert(userObject["lastLoginTime"]);
```

Or

```
var userObject = { "lastLoginTime": new Date() };
alert(userObject.lastLoginTime);
```

JS Functions

- Simple JS function:

Format:

```
function functionName(para1, para2) {
   return functionValue;
}
```

Examples:

```
function func(x) {
    alert(x);
}

func(" My First Function");
```

Or

```
var func = function(x) {
    alert(x);
};

func("My Second Function");
```

JS Functions

- Function as a constructor:

```
var func = new Function("x", "alert(x);");
func(“My Third Function");
```

;

Input and Output Commands in JS

- Input commands or methods:
  - Prompt
  - stdIn
- Output commands or methods
  - alert
  - stdOut
- Example:

```
var strInput = WScript.StdIn.ReadAll();
WScript.StdOut.Write(strInput)
```

Cookies

- Cookies are **functionalities in web-browsers** that enable **data and information** about web activities to be **saved** for **future reference**
- It improves performance and saves time
- There are 3 types of cookies
  - **Session cookies**
    - Created on the same browser that a user is using
    - Created to last for that session
  - **First party cookies**
    - Created on the same browser that a user is using
  - **Third party cookies**
    - Can be created by only a third party browser and not the current browser being used by the user

jQuery

- jQuery is an JavaScript **cross-platform library** and APIs
- It is used to **simply client-side programming** with **HTML**
- It is the most used JS library on web applications
- It is used for
  - Manipulating documents
  - Selecting DOM
  - Handling events
    - Clicking, Downloading, uploading, saving, deleting, etc

jQuery

- jQuery consists of the following
  - DOM element selections
  - DOM manipulation
  - Events
  - AJAX (Asynchronous JavaScript and XML)
  - JSON parsing

jQuery

- jQuery **library** and **APIs** can be included in JS using the syntax:

```
<script src="jquery.js"></script>
```

jQuery Syntax

- Basic syntax is: **$(*selector*).*action*()**
- Where:
  - A $ sign to define/access jQuery
  - A (*selector*) to "query (or find)" HTML elements
  - A jQuery *action*() to be performed on the element(s)

jQuery

Some syntax:

$(this).hide() - hides the current element.

$("p").show() - show all <p> elements.

$("p").fadein() - fade effects for all <p> elements.

Example:

https://www.w3schools.com/jquery/tryit.asp?filename=tryjquery_eff_fadeout_fadein

Some jQuery APIs

- **add()**
  - Create a new jQuery object with elements added to the set of matched elements.
- **.clearQueue()**
  - Remove from the queue all items that have not yet been run.
- **.click()**
  - Bind an event handler to the “click” JavaScript event, or trigger that event on an element.
- **.clone()**
  - Create a deep copy of the set of matched elements.

JSON

- **JSON** -JavaScript Object Notation
- It is a **lightweight data-interchange** format
  - Change data from one format to another
- It is created as a **sub-set** of JavaScript programming language
- It is **language independent**
  - Because of this it is used for data inter-change
- It is built on **two structures**
  - A collection of name/value pairs
    - This is *object*, record, struct, dictionary, hash table, keyed list, or associative array in other languages
  - An ordered list of values
    - This called an *array*, vector, list, or sequence in other languages
- JSON is a simple, **text-based way** to **store** and **transmit structured data**

Advantages of using JSON

- By using a simple syntax, you can easily store anything from a single number through to strings, arrays, and objects using nothing but a string of plain text.
  - You need little programming skill to programme using JSON
- It's **compact**
- It's easy for both computers and people to read and write
- It **maps very easily** onto the data structures used by most programming languages (numbers, strings, booleans, nulls, arrays and associative arrays)
- Nearly all programming languages contain functions or libraries that can read and write JSON structures

Source: Elated Home Articles

Where can we use JSON?

- JSON is most commonly **used in web applications** to send data from the server to the browser.
  - In most cases you transfer JSON data from server to browser using AJAX (Asynchronous JavaScript + XML)
    - AJAX lets your web application exchange data and messages between the browser and the server without having to reload the page
- JSON is also used to **send data** from the **browser** to the **server**
  - But in this case the JSON string must be properly encoded as a GET or POST parameter.
  - This approach is less common, since the data sent in AJAX requests tend to be fairly simple (for example, a product ID).

How to write a JSON String

- A **JSON string** contains either **an array of values or an object** (an associative **array of name/value pairs**).
- An ***array*** is surrounded by square brackets ([]) and contains a comma-separated list of values.
- An ***object*** is surrounded by curly brackets ({})and contains a comma-separated list of name/value pairs.
- A ***name/value pair*** consists of a field name (in double quotes), followed by a colon (:), followed by the field value.

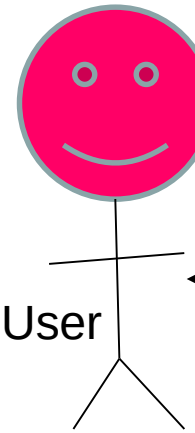
- A ***value*** in an array or object can be:
  - A number (integer or floating point)
  - A string (in double quotes)
  - A boolean (true or false)
  - Another array (surrounded by square brackets, [and])
  - Another object (surrounded by curly brackets, { and })
  - The value null

Scenario of where to use JSON in Real-Life

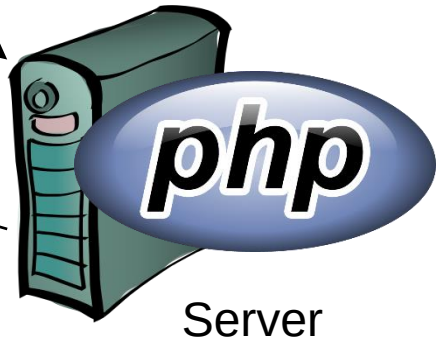
1. A user **clicks** a product in an online store (Client)

Client

2. The JavaScript running in the browser **makes an AJAX request** to a PHP script running on the server, passing it the ID of the clicked product



User



Server

3. The PHP script **retrieves** the product name and other info from the products database, **encodes the data as a JSON string**, **and sends the string back to the browser**

4. The JavaScript running in the browser **decodes the JSON string** and displays the product details in the page for the user.

Simple Example of how to Store Shopping Cart in JSON Format

```
{
  "orderID": 12345,
  "shopperName": "John Smith",
  "shopperEmail": "johnsmith@example.com",
  "contents": [
    {
      "productID": 34,
      "productName": "SuperWidget",
      "quantity": 1
    },
    {
      "productID": 56,
      "productName": "WonderWidget",
      "quantity": 3
    }
  ],
  "orderCompleted": true
}
```

Explanation of the Example

- At the top level, the curly braces ({ and }) are written, which creates an object.
- Inside the object, we have several *name/value pairs*:
  - "orderID": 12345
    - A property with the name "orderId" and the integer value 12345
  - "shopperName": "John Smith"
    - A property with the name "shopperName" and the string value "John Smith"
  - "shopperEmail": "johnsmith@example.com"
    - A property with the name "shopperEmail" and the string value "johnsmith@example.com"
  - "contents": [...]
    - A property with the name "contents", whose value in an array
  - "orderCompleted": true
    - A property with the name "orderCompleted" and the boolean value true
- Inside the "contents" array, we have 2 objects representing individual order lines in the cart. Each object contains 3 properties: productID, productName, and quantity.

Converting JSON to JavaScript (Example)

```
<script type="text/javascript">
var cart = {
  "orderID": 12345,
  "shopperName": "John Smith",
  "shopperEmail": "johnsmith@example.com",
  "contents": [
    {
      "productID": 34,
      "productName": "SuperWidget",
      "quantity": 1
    },
    {
      "productID": 56,
      "productName": "WonderWidget",
      "quantity": 3
    }
  ],
  "orderCompleted": true
};
</script>
```

Create and Read JSON Strings from JavaScript

- **To create a JSON string,** you start with a variable containing some data, then pass it through a function to turn that data into a JSON string.
- **To read a JSON string,** you start with a JSON string representing some data, then pass it through a function to create a variable containing the data.

Example: Create JSON Strings from JavaScript Variable

- The built-in JavaScript method “JSON.Stringify()” takes a JS variable and output a JSON string
- The JSON string will represent the contents of the variable

```
<script type="text/javascript">

var cart = {
  "orderID": 12345,
  "shopperName": "John Smith",
  "shopperEmail": "johnsmith@example.com",
  "contents": [
    {
      "productID": 34,
      "productName": "SuperWidget",
      "quantity": 1
    },
    {
      "productID": 56,
      "productName": "WonderWidget",
      "quantity": 3
    }
  ],
  "orderCompleted": true
};
```

alert (JSON.stringify(cart));

```
</script>
```

Example: Create JSON Strings from JavaScript Variable

The statement:

alert (JSON.stringify(cart));

Will print:

{"orderID":12345,"shopperName":"John Smith",
"shopperEmail":"johnsmith@example.com",
"contents":[{"productID":34,"productName":"SuperWidget",
"quantity":1},
{"productID":56,"productName":"WonderWidget","quantity":3}],
"orderCompleted":true}

Example: Create JavaScript Variable from JSON String

- The built-in JavaScript method “JSON.parse()” takes a JSON string and return a JS object or array
- The JS object or array contents the JSON string data

```
<script type="text/javascript">

var jsonString = '
{
  "orderID": 12345,
  "shopperName": "John Smith",
  "shopperEmail": "johnsmith@example.com",
  "contents": [
    {
      "productID": 34,
      "productName": "SuperWidget",
      "quantity": 1
    },
    {
      "productID": 56,
      "productName": "WonderWidget",
      "quantity": 3
    }
  ],
  "orderCompleted": true
}
';

var cart = JSON.parse ( jsonString );

alert ( cart.shopperEmail );
alert ( cart.contents[1].productName );
</script>
```

Example: Create JavaScript Variable from JSON String

The statements:

```
var cart = JSON.parse ( jsonString );
 alert ( cart.shopperEmail );
alert ( cart.contents[1].productName );
```

Will print:

```
johnsmith@example.com
WonderWidget
```

Example: Create JavaScript to Display Dates

```
<script>
  var currentDate = new Date(),
      day = currentDate.getDate(),
      month = currentDate.getMonth() + 1,
      year = currentDate.getFullYear();
  document.write(day + "/" + month + "/" + year)
</script>
```

Class Work

- Group Students into 3 Groups
- Group 1: Design a scenario where a user uses JSON string and wants to print out the result as a Javascript variable
- Group 2: Write a Javascript programme that takes in a JSON string and display it as JS variable
- Group 3: Discuss the advantages of using JavaScript and JSON

Summary

- Revision on JavaScript Programming
- Advanced JavaScript Programming
- JSON
- Class Work

Study Materials

- Learning PHP, MySQL, JavaScript, CSS and HTML 5: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon
- A Software Engineer Learns HTML5, JavaScript and jQuery: A Guide to Standards-based Web Applications by Dane Cameron