Javascript Cookies

Web Authoring and Design

Benjamin Kenwright

Outline

- What do we mean by Javascript Cookies?
- What do we use Cookies for?
- How do we load and save Cookies using Javascript
- Examples
- Summary
- Review/Discussion

Do you like Cookies?

- Who is this?

Cookie Monster (Sesame Street)

Getting Information About the Document

- Several properties of the document object include information about the current document in general. For e.g. document.title represent the title of the current page, defined by the HTML < title > tag.

Title

```
<script type="text/javascript">
    alert(document.title);
</script>
```

- The above code will alert "The DOM", the Title of the HTML page

Other important information about the document

- **document.URL :** Represents the document's URL.
- **document.lastModified :** The date which the document was last modified.
- **document.cookie :** It enables you to read or set a cookie for the document.
- **document.images :** It returns a collection of images used in the document

What are Cookies ?

- Cookies is the most efficient method of **remembering** and tracking preferences, purchases, commissions, and other information required for better visitor experience or site statistics

How It Works ?

- Your server sends some data to the visitor's browser in the form of a cookie
- The browser may accept the cookie
- If it does, it is stored as a plain text record on the visitor's hard drive
- Now, when the visitor arrives at another page on your site, the browser sends the same cookie to the server for retrieval
- Once retrieved, your server knows/remembers what was stored earlier

Cookies are a plain text data record of 5 variable-length fields

- **Expires** − The date the cookie will expire. If this is blank, the cookie will expire when the visitor quits the browser.
- **Domain** − The domain name of your site.
- **Path** − The path to the directory or web page that set the cookie. This may be blank if you want to retrieve the cookie from any directory or page.
- **Secure** − If this field contains the word "secure", then the cookie may only be retrieved with a secure server. If this field is blank, no such restriction exists.
- **Name=Value** − Cookies are set and retrieved in the form of key-value pairs

Storing Cookies

- The simplest way to create a cookie is to assign a string value to the document.cookie object, which looks like this

```
document.cookie =
  "key1=value1;key2=value2;expires=date";
```

Example

```
<html>
   <head>

      <script type = "text/javascript">
         <!--
            function WriteCookie()
            {
               if( document.myform.customer.value == "" ){
                  alert("Enter some value!");
                  return;
               }
               cookievalue= escape(document.myform.customer.value) + ";";
               document.cookie="name=" + cookievalue;
               document.write ("Setting Cookies : " + "name=" + cookievalue );
            }
         //-->
      </script>

   </head>

   <body>

      <form name="myform" action="">
         Enter name: <input type="text" name="customer"/>
         <input type="button" value="Set Cookie" onclick="WriteCookie();"/>
      </form>

   </body>
</html>
```

Reading Cookies

- Reading a cookie is just as simple as writing one, because the value of the **document.cookie** object is the cookie
- So you can use this string whenever you want to access the cookie
- The document.cookie string will keep a list of **name=value** pairs separated by semicolons, where **name** is the name of a cookie and value is its string value
- You can use strings' **split()** function to break a string into key and values

Example

```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function ReadCookie()
            {
               var allcookies = document.cookie;
               document.write ("All Cookies : " + allcookies );

               // Get all the cookies pairs in an array
               cookiearray = allcookies.split(';');

               // Now take key value pair out of this array
               for(var i=0; i<cookiearray.length; i++){
                  name = cookiearray[i].split('=')[0];
                  value = cookiearray[i].split('=')[1];
                  document.write ("Key is : " + name + " and Value is : " + value);
               }
            }
         //-->
      </script>

   </head>
   <body>

      <form name="myform" action="">
         <p> click the following button and see the result:</p>
         <input type="button" value="Get Cookie" onclick="ReadCookie()"/>
      </form>

   </body>
</html>
```

Setting Cookies Expiry Date

- You can extend the life of a cookie beyond the current browser session by setting an expiration date and saving the expiry date within the cookie.
- This can be done by setting the ‘**expires**’ attribute to a date and time

Example

```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function WriteCookie()
            {
               var now = new Date();
               now.setMonth( now.getMonth() + 1 );
               cookievalue = escape(document.myform.customer.value) + ";"

               document.cookie="name=" + cookievalue;
               document.cookie = "expires=" + now.toUTCString() + ";"
               document.write ("Setting Cookies : " + "name=" + cookievalue );
            }
         //-->
      </script>

   </head>
   <body>

      <form name="myform" action="">
         Enter name: <input type="text" name="customer"/>
         <input type="button" value="Set Cookie" onclick="WriteCookie()"/>
      </form>

   </body>
</html>
```

Deleting a Cookie

- Sometimes you will want to delete a cookie so that subsequent attempts to read the cookie return nothing.
- To do this, you just need to set the expiry date to a time in the past.

Example

```
<html>
   <head>

      <script type="text/javascript">
         <!--
            function WriteCookie()
            {
               var now = new Date();
               now.setMonth( now.getMonth() - 1 );
               cookievalue = escape(document.myform.customer.value) + ";"

               document.cookie="name=" + cookievalue;
               document.cookie = "expires=" + now.toUTCString() + ";"
               document.write("Setting Cookies : " + "name=" + cookievalue );
            }
          //-->
      </script>

   </head>
   <body>

      <form name="myform" action="">
         Enter name: <input type="text" name="customer"/>
         <input type="button" value="Set Cookie" onclick="WriteCookie()"/>
      </form>

   </body>
</html>
```

This Week

- Review Slides
- Read Associated Chapters
- Work through Javascript Examples
  - ▷Update GitHub Account/Webpage
- **Start Early**

Summary

- Overview of Javascript and Cookies
- Hands-On/Practical

Questions/Discussion

Example Cookie Helper Functions

```
<script>
function createCookie(name,value,days) {
    var expires = "";
    if (days) {
        var date = new Date();
        date.setTime(date.getTime() + (days*24*60*60*1000));
        expires = "; expires=" + date.toUTCString();
    }
    document.cookie = name + "=" + value + expires + "; path=/";
}

function readCookie(name) {
    var nameEQ = name + "=";
    var ca = document.cookie.split(';');
    for(var i=0;i < ca.length;i++) {
        var c = ca[i];
        while (c.charAt(0)==' ') c = c.substring(1,c.length);
        if (c.indexOf(nameEQ) == 0) return c.substring(nameEQ.length,c.length);
    }
    return null;
}

function eraseCookie(name) {
    createCookie(name,"",-1);
}

// Now, calling functions

// 1...
createCookie('ppkcookie','testcookie',7);

// 2...
var x = readCookie('ppkcookie')
if (x) {
    // ..do something with x
}
</script>
```