

Tutorial 9

Storing Data with Cookies

Objectives

- Learn about client-side cookies
- Set a cookie's value
- Set a cookie's expiration date
- Set a cookie's path, domain, and secure flag
- Read data from a cookie
- Write cookie values to a data form

2

New Perspectives on
JavaScript and AJAX, 2nd Edition

Objectives (cont'd)

- Delete a cookie
- Write a multivalued cookie
- Read a multivalued cookie
- Create a cookie to track the last visit date
- Test for browser support of cookies

3

New Perspectives on
JavaScript and AJAX, 2nd Edition

Introducing Cookies

- **Cookies** are text files containing data the Web browser can retrieve and display
- Overcome limitations of HTTP communication protocol, which is a **stateless protocol**
- **Server-side cookies** vs. **client-side cookies**
 - Client-side cookies contain textual data only -- cannot contain a computer virus
 - Browsers place limits on size and number of client-side cookies

4

New Perspectives on
JavaScript and AJAX, 2nd Edition

Cookies and Web Browsers

- Each browser handles and stores cookies in a different way
- A cookie generated while viewing a Web site under one browser is *not* available if user switches to a different browser

5

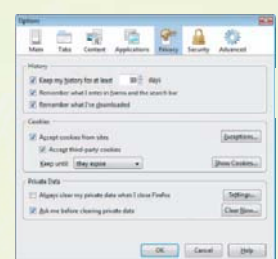
New Perspectives on
JavaScript and AJAX, 2nd Edition

Cookies and Web Browsers

Internet Explorer



Firefox



6

New Perspectives on
JavaScript and AJAX, 2nd Edition

Structure of a Cookie

- `name=value;expires=date;path=directory;domain=domain-name;secure`
- Only the `name=value` pair is required

Writing a Cookie Name and Value

- Created using the `document.cookie` property
 - `document.cookie = cookie_string`
- Three methods to convert non-alphanumeric characters to escape codes, which replace blank spaces separating words
 - `escape()`
 - `encodeURIComponent()`
 - `encodeURIComponent()`

Escape Character Codes and Functions

Character	<code>escape()</code>	<code>encodeURIComponent()</code>	<code>encodeURIComponent()</code>
Blank space	%20	%20	%20
!	%21	!	!
@	%40	@	%40
#	%23	#	%23
\$	%24	\$	%24
%	%25	%25	%25
^	%5E	%5E	%5E
&	%26	&	%26
*	%2A	%2A	%2A
(%28	((
)	%29))
+	%2B	+	+
,	%2C	,	,
-	%2D	-	-
.	%2E	.	.
:	%3A	:	%3A
;	%3B	;	%3B

Escape Character Codes and Functions (continued)

Character	<code>escape()</code>	<code>encodeURIComponent()</code>	<code>encodeURIComponent()</code>
.	%2E	%2E	%2E
/	%2F	%2F	%2F
:	%3A	%3A	%3A
;	%3B	%3B	%3B
<	%3C	%3C	%3C
=	%3D	%3D	%3D
>	%3E	%3E	%3E
?	%3F	%3F	%3F
@	%40	%40	%40
[%5B	%5B	%5B
\	%5C	%5C	%5C
]	%5D	%5D	%5D
^	%5E	%5E	%5E
_	%5F	%5F	%5F
`	%60	%60	%60
{	%7B	%7B	%7B
	%7C	%7C	%7C
}	%7D	%7D	%7D
~	%7E	%7E	%7E

Writing a Cookie Name and Value

- The `writeCookie()` function
 - Writes a cookie given the cookie name and value

```

/* Add new code below */
function writeCookie(cname, cvalue) {
    if (cname && cvalue != "") {
        var cstring = cname + "=" + escape(cvalue);
        document.cookie = cstring;
    }
}
    
```

replaces all non-alphanumeric characters with escape codes

Setting the Cookie Expiration Date

- **Session cookies** expire when current Web session is completed
 - Omit expiration date
- **Persistent cookies** exist until a specified expiration date
 - `expires=wday, dd-mmm-yyyy hh:mm:ss GMT`

-
- The diagram illustrates the process of converting a date object to a GMT date string. It features a code block for a JavaScript function and two explanatory annotations with arrows.
- ```
function writecookie(cname, cvalue, expires) {
 if (cname && cvalue != "") {
 var cstring = cname + "=" + escape(cvalue);
 if (expires) cstring += "; expires=" + expires.toGMTString();
 }
 document.cookie = cstring;
}
```
- Annotations:**
- An arrow points from the `expires` parameter in the function signature to a box labeled **date object**.
  - An arrow points from the `expires.toGMTString()` call in the code to a box labeled **converts the date object to a GMT date string**.

- 14

- 15

- 16

17

contents of

cookie names and values found in the current folder

## Retrieving a Cookie

- Use the `split()` method to split a cookie into an array:
  - `var cookiesArray = document.cookie.split("; ");`

document.cookie

```
firstName=Kevin; lastName=Geary; email=Kevin@geary.com; state=TX; age=77; country=USA; phone=409-420-1234; email=Kevin@geary.com; phone=409-420-1234; number=1234
```

| cookieArray[0] | firstName=Kevin       |
|----------------|-----------------------|
| cookieArray[1] | lastName=Geary        |
| cookieArray[2] | email=Kevin@geary.com |
| cookieArray[3] | state=TX              |
| cookieArray[4] | age=77                |
| cookieArray[5] | country=USA           |
| cookieArray[6] | email=Kevin@geary.com |
| cookieArray[7] | phone=409-420-1234    |
| cookieArray[8] | number=1234           |

19

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Retrieving a Cookie

- Use `retrieveCookie()` function to retrieve cookie values created in previous sessions and write them to pages on a Web site

```
function retrieveCookie(cName) {
 if (document.cookie) {
 var cookiesArray = document.cookie.split("; ");
 for (var i = 0; i < cookiesArray.length; i++) {
 if (cookiesArray[i].split("=")[0] == cName) {
 return unescape(cookiesArray[i].split("=")[1]);
 }
 }
 }
}
```

splits document.cookie into an array and searches through to locate the cookie named cName

20

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Deleting a Cookie

- Set the cookie's expiration date to a past date, or
- Delete cookies within Options or Preferences dialog box of most browsers

```
function deleteCookie(cName) {
 if (document.cookie) {
 var cookiesArray = document.cookie.split("; ");
 for (var i = 0; i < cookiesArray.length; i++) {
 if (cookiesArray[i].split("=")[0] == cName) {
 document.cookie = cName + "=expiredThu, 01-Jan-1970 00:00:00 GMT";
 }
 }
 }
}
```

sets the expiration date to a past date to delete the cookie

21

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Storing Multiple Values in a Cookie

- Using **subkeys**
  - To store multiple values in a cookie, format the name/value pair as

`cName=field1=value1&field2=value2&field3=value3`

22

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Storing Multiple Values in a Cookie

- Writing a multivalued cookie
  - Create a function named `writeMCookie()` that writes a field name and value into a multivalued cookie

```
function writeMCookie(cName, fName, fValue, expires, path, domain, secure) {
 if (cName && fName && fValue != "") {
 // create the subkey
 var subkey = fName + "=" + escape(fValue);
 document.cookie = cName + "=" + subkey + (expires ? "; expires=" + expires : "") + (path ? "; path=" + path : "") + (domain ? "; domain=" + domain : "") + (secure ? "; secure" : "");
 }
}
```

23

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Storing Multiple Values in a Cookie

- Retrieving data from a multivalued cookie
  - Add a `retrieveMCookie()` function to retrieve the value of a specified field within a multivalued cookie
    - First the function retrieves the entire cookie value
    - Then the function retrieves the value of a specific subkey within that value

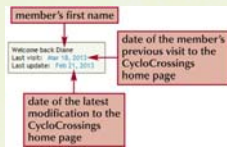
```
function retrieveMCookie(cName, fName) {
 if (document.cookie) {
 // retrieve the cookie value
 // retrieve the field value within the cookie
 }
}
```

24

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Creating a Welcome Back Message

- Use lastVisit cookie to store the date on which the user last visited the site
- Use document.lastModified property to access the date that the document was last modified



25

New Perspectives on  
JavaScript and AJAX, 2nd Edition

## Testing for Cookie Support

- To test whether cookies are enabled by a current Web browser:
  - Use navigator.cookieEnabled object property
- To test whether cookies are enabled for older legacy browsers:
  - Write a function that creates/retrieves a temporary session cookie and returns a Boolean value indicating whether the operation was successful/unsuccessful

26

New Perspectives on  
JavaScript and AJAX, 2nd Edition