

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



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



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



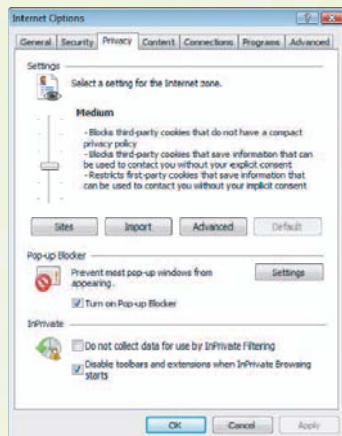
Cookies and Web Browsers

- 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

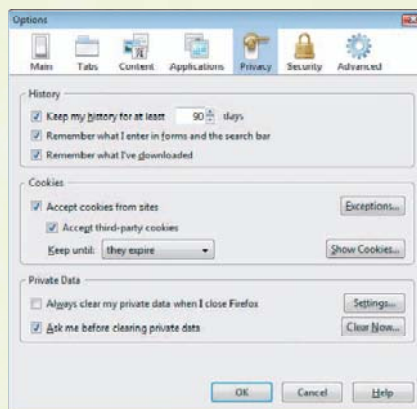


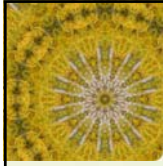
Cookies and Web Browsers

Internet Explorer



Firefox





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	escape()	encodeURIComponent()	encodeURIComponent()
blank space	%20	%20	%20
!	%21	!	!
@	@	@	%40
#	%23	#	%23
\$	%24	\$	%24
%	%25	%25	%25
^	%5E	%5E	%5E
&	%26	&	%26
*	*	*	*
(%28	((
)	%29))
-	-	-	-
_	_	_	_
=	%3D	=	%3D
+	+	+	%2B
:	%3A	:	%3A
;	%3B	;	%3B



Escape Character Codes and Functions (continued)

Character	escape()	encodeURIComponent()	encodeURIComponent()
.	.	.	.
*	%22	%22	%22
'	%27	'	'
\	%5C	%5C	%5C
/	/	/	%2F
?	%3F	?	%3F
<	%3C	%3C	%3C
>	%3E	%3E	%3E
~	%7E	~	~
[%5B	%5B	%5B
]	%5D	%5D	%5D
{	%7B	%7B	%7B
}	%7D	%7D	%7D
`	%60	%60	%60



Writing a Cookie Name and Value

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

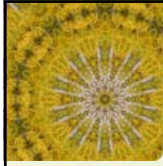
```
/* Add new code below */  
  
function writeCookie(cname, cvalue) {  
    if (cname && cvalue != "") {  
        var cstring = cname + "=" + escape(cvalue);  
        document.cookie = cstring;  
    }  
}
```

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



Setting the Cookie Expiration Date

- Set expiration dates
 - With a date object that can be converted using `toGMTString()` method, or
 - Relative to current date

```
function writecookie(cname, cvalue, expdate) {  
    if (cname && cvalue != "") {  
        var cstring = cname + "=" + escape(cvalue);  
        if (expdate) cstring += "; expires=" + expdate.toGMTString();  
        document.cookie = cstring;  
    }  
}
```

Diagram annotations:

- A red box labeled "date object" points to the `expdate` parameter in the function signature.
- A red box labeled "converts the date object to a GMT date string" points to the `expdate.toGMTString()` method call.



Setting the Cookie Path

- Cookies are associated with folders and directories, not a single document; cookies created in one Web page can be accessible to other pages in the Web site
- To ensure that a cookie is readable from within any folder on the Web server, set the path to the root folder of the Web server
- No need to set the cookie path if all Web pages are placed within a single folder on the server; cookie path is a concern only if pages are stored in different directories



Setting the Cookie Domain

- Cookies are restricted to work within a particular Web domain and cannot be accessed across domains
- For a site that employs multiple domains and subdomains, add the text string `;domain=domain-name`
 - The cookie will be available to any page within the domain or its subdomains
- **First-party cookies** versus **third-party cookies (tracking cookies)**



Setting Cookie Security

- Define whether or not the cookie can be sent only to a Secure Sockets Layer (SSL) using the HTTPS transfer protocol
- To force the browser to employ a secure channel:
 - Add the text string `;secure` to the cookie definition
- If the secure keyword is omitted from the cookie definition, the browser will transfer the cookie using the HTTP protocol

Setting the Cookie Path, Domain, and Security Code

```
function writeCookie(cName, cValue, expDate, cPath, cDomain, cSecure) {  
    if (cName && cValue != "") {  
        var cString = cName + "=" + escape(cValue);  
        if (expDate) cString += ";expires=" + expDate.toGMTString();  
        if (cPath) cString += ";path=" + cPath;  
        if (cDomain) cString += ";domain=" + cDomain;  
        if (cSecure) cString += ";secure";  
        document.cookie = cString;  
    }  
}
```

path associated with the cookie

Boolean value indicating whether to transmit the cookie over a secure connection

domain associated with the cookie

17

*New Perspectives on
JavaScript and AJAX, 2nd Edition*

Storing Form Data in Cookies

document.cookie

```
firstName=Kevin; lastName=Geary; street1=100%  
20North%20Lane; city=Redmond; state=38; zip=97756;  
country=United%20States;  
email=kgeary@cyclocrossings.com; phone=428541429%  
20555%20-4200706; member1=true
```

contents of document.cookie

Delete	Cookie Name	Cookie Value
<input type="checkbox"/>	firstName	Kevin
<input type="checkbox"/>	lastName	Geary
<input type="checkbox"/>	street1	100%20North%20Lane
<input type="checkbox"/>	city	Redmond
<input type="checkbox"/>	state	38
<input type="checkbox"/>	zip	97756
<input type="checkbox"/>	country	United%20States
<input type="checkbox"/>	email	kgeary@cyclocrossings.com
<input type="checkbox"/>	phone	%28541429%20555%20-4200706
<input type="checkbox"/>	member1	true

cookie names and values found in the current folder

18

*New Perspectives on
JavaScript and AJAX, 2nd Edition*

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; street1=1304208North20Lane; city=Redmond; state=38; zip=97756; country=United%20States; email=kgeary@cyclocrossings.com; phone=428541829820555429-4200706; member1=true	
var cookiesArray = document.cookie.split("; ");	
cookiesArray[0]	firstName=Kevin
cookiesArray[1]	lastName=Geary
cookiesArray[2]	street1=1304208North20Lane
cookiesArray[3]	city=Redmond
cookiesArray[4]	state=38
cookiesArray[5]	zip=97756
cookiesArray[6]	country=United%20States
cookiesArray[7]	email=kgeary@cyclocrossings.com
cookiesArray[8]	phone=428541829820555429-4200706
cookiesArray[9]	member1=true

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



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 + "=expires=Thu, 01-Jan-1970 00:00:01 GMT";  
            }  
        }  
    }  
}
```

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

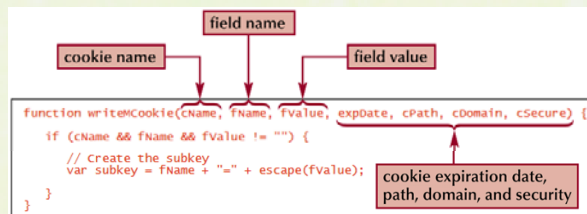


Storing Multiple Values in a Cookie

- Using **subkeys**
 - To store multiple values in a cookie, format the name/value pair as
`cName=field1=value1&field2=value2&field3=value3`

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

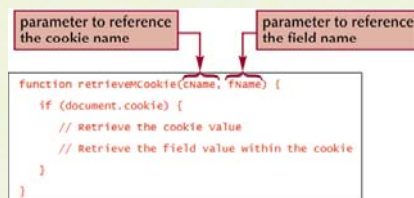


23

*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



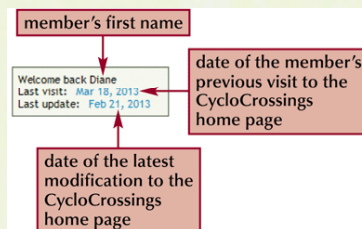
24

*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



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