Cookies Example

Hello, Paul.
Click here if you are not Paul

Welcome to cookies!
JavaScript: Using Cookies

- **Cookie**
  - Data stored on _____________ to maintain information about client during and between browser sessions
  - A string: identifier=value pairs separated by ;
  - Can be accessed through document.cookie property
  - Set expiration date using expires keyword
  - Use escape function to convert non-alphanumeric characters to hexadecimal escape sequences
  - unescape function converts hexadecimal escape sequences back to English characters

**Storing Cookies – Simple Version**

```javascript
document.writeln("<br/>Cookie is: "+document.cookie);

document.cookie = "name=" + escape("J Smith");
document.writeln("<br/>Cookie is: "+document.cookie);

document.cookie = "rank=" + escape("Captain");
document.writeln("<br/>Cookie is: "+document.cookie);
```
Reading Cookies – Simple Version

myCookies = document.cookie;

cookieElements = myCookies.split("=");

document.writeln("<br/>Identifier stored is: "+ cookieElements[0] +
"<br/>Value stored is: " + cookieElements[1]);

Simple Cookie Example

// reset the document's cookie if wrong person
function wrongPerson() {
  // reset the cookie
  document.cookie = "name=null;" + " expires=Thu, 01-Jan-95 00:00:01 GMT";

  // after removing the cookie reload the page to get a new name
  location.reload();
}

// determine whether there is a cookie
if (document.cookie) {
  var cookie = document.cookie;
  var cookieTokens = cookie.split("=");

  // set name to the part of the cookie that follows the = sign
  name = cookieTokens[1];
  name = unescape(name);
}
else {
  // if there was no cookie then ask the user to input a name
  name = window.prompt("Please enter your name", "Paul");
  document.cookie = "name=" + escape(name);
}

document.writeln("<h1>Hello, " + name + ". </h1>");
document.writeln("<p><a href='javascript:wrongPerson()'>" +
"Click here if you are not " + name + "</a></p>");
Exercise #1: JS:
Ask user for favorite quote using a window prompt.
Save quote in a cookie identified by “favQuote”.
Display quote on the page.

Storing Cookies – More Realistic

- By default, cookies expire when session ends
- Set “expires” attribute to make stick around longer

```javascript
function createCookie(identifier, value, days) {
  if (days) {
    var date = new Date();
    date.setTime(date.getTime() + (days * 24 * 60 * 60 * 1000));
    var expires = "; expires=\"+date.toGMTString();
  }
  else
    var expires = "";
  document.cookie = identifier + "\"=\"+escape(value)+\"\";}

function eraseCookie(identifier) {
  createCookie(identifier, "", -1);
}
```
(modified from http://www.quirksmode.org/js/cookies.html)
Reading Cookies – More Realistic

// Return the 'value' of the cookie with identifier 'desiredId'
// returns null if no match found.
function readCookie(desiredId) {
    // First split the pairs apart on '; '
    var pairs = document.cookie.split('; ');

    // Now split each pair on '='. Check if have a match
    for (var i=0; i < pairs.length; i++) {
        var aPair = pairs[i];

        // split into desired parts and check for match
        var cookieTokens = aPair.split("=");
        var id = cookieTokens[0];
        var value = cookieTokens[1];

        if (id == desiredId) {
            // found desired cookie -- return value
            return unescape(value);
        }
    }
    return null; // no match
}

Exercise #2: JS: Read the value of cookie identified by “favQuote” and display it in a pop-up msg if it exists, otherwise display “no quotes”
Exercise #3: Re-write Simple Cookie Example using the helper functions

Cookies – Java Script and Perl

- Cookies with JavaScript
  - Create cookie
    - document.cookie = “color=red”;
  - Read cookie (from JavaScript)
    - Read and parse document.cookie
    - Use readCookie() function to help with this
  - Where are cookies stored??

- Cookies with Perl
  - Create cookie with print() BEFORE printing header()
    - Sent to browser
  - Browser always sends appropriate cookies back to server with request
  - Read cookie
    - Access $ENV{ “HTTP_COOKIE” } (book does this)
    - Or use cookie() function helper (easier!)
  - Where are cookies stored??

- Cookies created with Perl can be read via JavaScript and vice versa
Create Cookies with Perl

(Assume this file invoked from a HTML form with fields name, height, and color)

```perl
#!/usr/bin/perl
use strict;
use CGI qw( :standard );
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

my $name = param( "name" );
my $height = param( "height" );
my $color = param( "color" );

my $expires = gmtime( time() + 86400 );

print "Set-Cookie: Name=$name; expires=$expires; 
";
print "Set-Cookie: Height=$height; expires=$expires; 
";
print "Set-Cookie: Color=$color; expires=$expires; 
";

print header(); print start_html( );

print h1("3 cookies were stored!  Name:  $name, Height: $height, 
   Color: $color");
print end_html();
```

Read Cookies With Perl

```perl
#!/usr/bin/perl
use strict;
use CGI qw( :standard );
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

print header(); print start_html( );

my $name   = cookie( "Name" );
my $height = cookie( "Height" );
my $color  = cookie( "Color" );

if ($name || $height || $color) {
   print h1("A cookie was found!");
   print h2("Name: $name");
   print h2("Height: $height");
   print h2("Color: $color");
}
else{
   print h1("Could not find cookies for Name, Height, or Color");
}

print( end_html() );
```
Uses for Cookies

• Most common:
  – User logs in using secure page (https)
  – Server checks password. If good, creates cookie
    • E.g. “login=m168987&auth=356af12cd124552”
  – User redirected to other pages. These pages don’t ask for
    password – instead just check that have valid login cookie

  – Why do we need the auth field?

Exercise #4: Perl: a) Create a cookie with identifier “favQuote”
and content “DTT/FSA”
b) change your program to store the quote provided by user
(not hardcoded) through CGI – param name “quote”
Remember

- Relevant cookies always sent by browser to the server

- Can create with JavaScript and read with Perl

- Or create with Perl and read with JavaScript