Cookies – Then and Now

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

• Now – Cookies with Perl
  – Create cookie with print() BEFORE header
    • Sent to browser
  – Browser always send 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 read via JavaScript and vice versa
Creating cookie with Perl

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

use CGI qw( :standard );
$name = param( "name" );
$height = param( "height" );
$color = param( "color" );
.expires = gmtime( time() + 86400 );
print( "Set-Cookie: Name=$name; expires=$expires; path=\n" );
print( "Set-Cookie: Height=$height; expires=$expires; path=\n" );
print( "Set-Cookie: Color=$color; expires=$expires; path=\n" );
print( header() );
print( start_html( ) );
print h1("A cookie was stored!  Name:  $name");
print( end_html() );

Read Cookies With Perl

use CGI qw( :standard );

print( header() );
print( start_html( ) );
$name   = cookie( "Name" );
$height = cookie( "Height" );
$color  = cookie( "Color" );
if ($name) {
   print h1("A cookie was found!");
   print h2("Name: $name");
   print h2("Height: $height");
   print h2("Color: $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=m078987&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?

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