Session - Web Session

April 01, 2019

Agenda - Handling Cookies

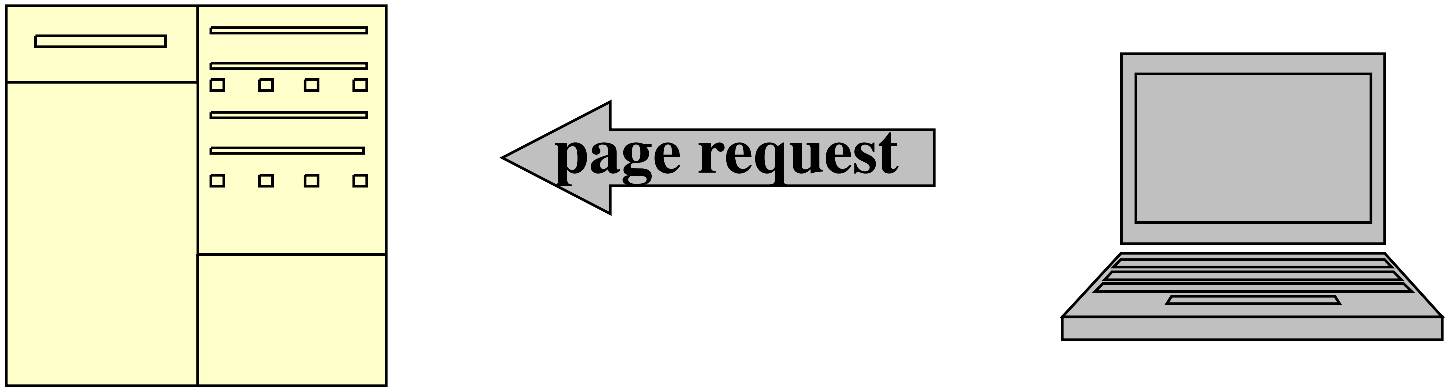
- What is a Cookie
- The potential of cookies
- The problems with cookies
- Sending cookies to browser
- Reading cookies from browser
- Simple cookie-handling servlets
- Cookie utilities
- Methods in the Cookie API
- A customized search engine front end

What is a Cookie?

- **A cookie is a small text file that is stored on a user's computer.**
- **Each cookie on the user's computer is connected to a particular domain.**
- **Each cookie be used to store up to 4kB of data.**
- **A maximum of 20 cookies can be stored on a user's PC per domain.**

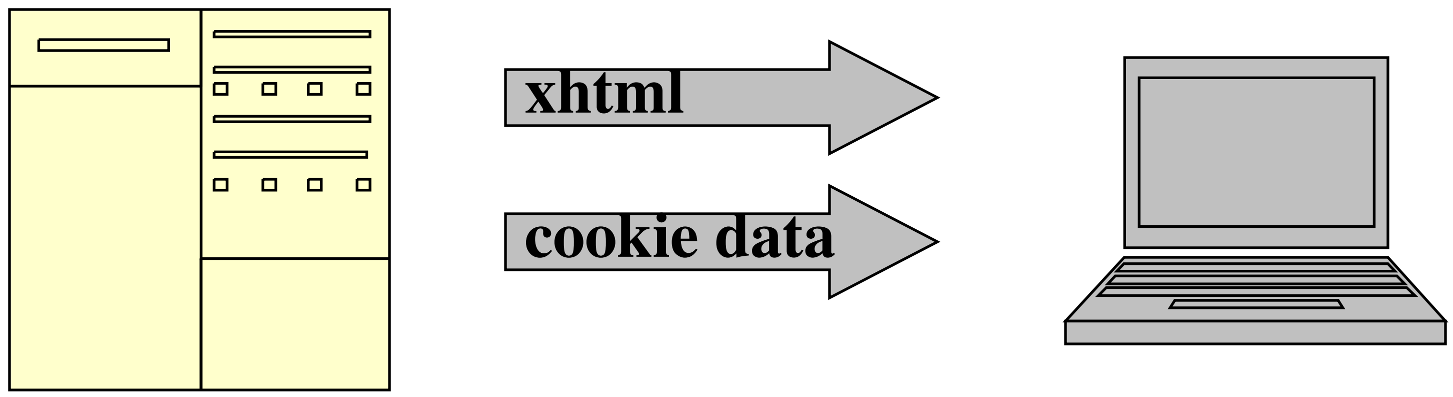
Example (1)

1. **User sends a request for page at http://www.cplusplus.com for the *first* time.**



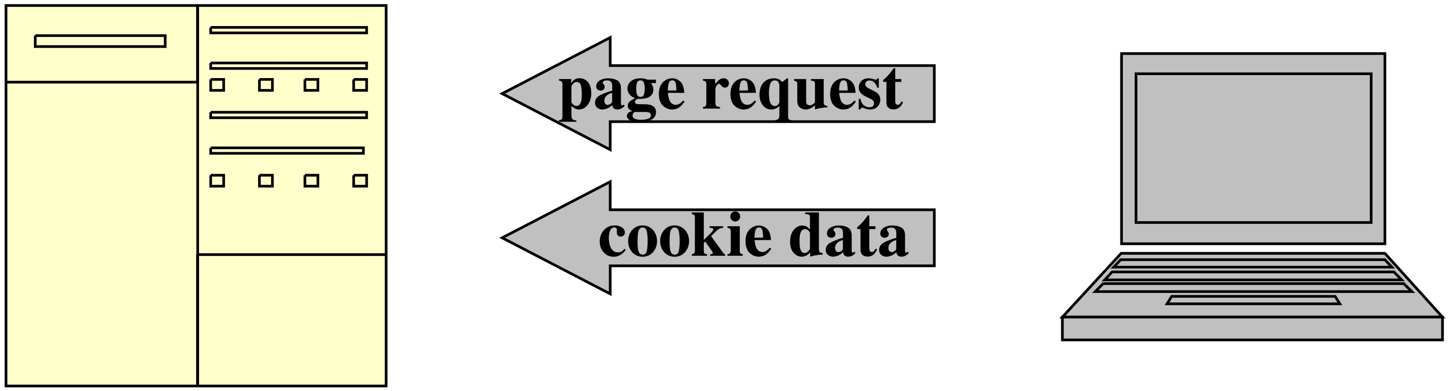
Example (2)

2. Server sends back the page xhtml to the browser AND stores some data in a cookie on the user's PC.



Example (1)

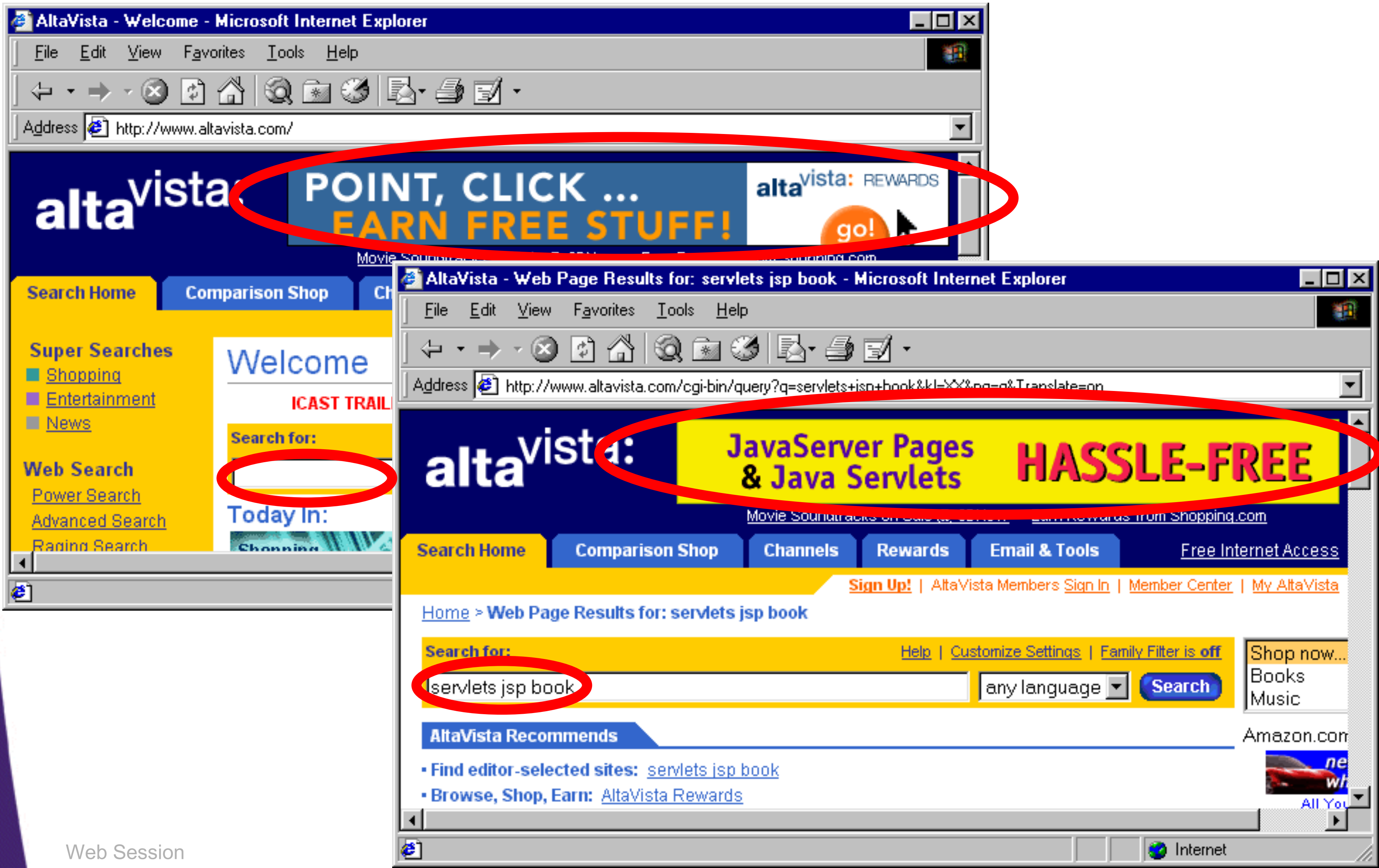
3. At the next page request for domain http://www.cplusplus.com, all cookie data associated with this domain is sent too.



The Potential of Cookies

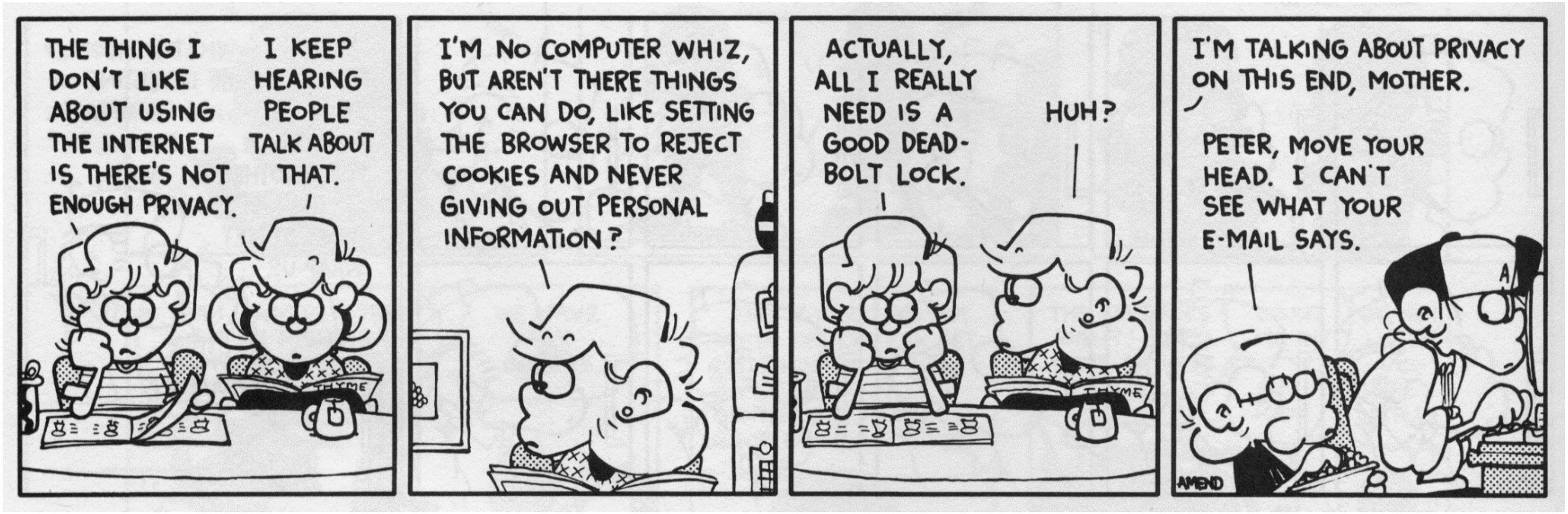
- **Idea**
  - Servlet sends a simple name and value to client.
  - Client returns same name and value when it connects to same site (or same domain, depending on cookie settings).
- **Typical Uses of Cookies**
  - Identifying a user during an e-commerce session
    - Servlets have a higher-level API for this task
  - Avoiding username and password
  - Customizing a site
  - Focusing advertising

Cookies and Focused Advertising



AltaVista - Welcome - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://www.altavista.com/
altavista:
POINT, CLICK ...
EARN FREE STUFF!
altavista: REWARDS
go!
Search Home
Comparison Shop
Super Searches
Shopping
Entertainment
News
Web Search
Power Search
Advanced Search
Welcome
Search for:
Today In:
AltaVista - Web Page Results for: servlets jsp book - Microsoft Internet Explorer
Address http://www.altavista.com/cgi-bin/query?q=servlets+jsp+book&kl=XX&pg=q&Translate=on
JavaServer Pages & Java Servlets
HASSLE-FREE
Search Home
Comparison Shop
Channels
Rewards
Email & Tools
Free Internet Access
Sign Up! | AltaVista Members Sign In | Member Center | My AltaVista
Home > Web Page Results for: servlets jsp book
Search for:
Help | Customize Settings | Family Filter is off
servlets jsp book
any language
Search
Shop now...
Books
Music
AltaVista Recommends
Find editor-selected sites: servlets jsp book
Browse, Shop, Earn: AltaVista Rewards
Amazon.com
Internet

Cookies and Privacy


THE THING I DON'T LIKE ABOUT USING THE INTERNET IS THERE'S NOT ENOUGH PRIVACY.
I KEEP HEARING PEOPLE TALK ABOUT THAT.
I'M NO COMPUTER WHIZ, BUT AREN'T THERE THINGS YOU CAN DO, LIKE SETTING THE BROWSER TO REJECT COOKIES AND NEVER GIVING OUT PERSONAL INFORMATION?
ACTUALLY, ALL I REALLY NEED IS A GOOD DEAD-BOLT LOCK.
HUH?
I'M TALKING ABOUT PRIVACY ON THIS END, MOTHER.
PETER, MOVE YOUR HEAD. I CAN'T SEE WHAT YOUR E-MAIL SAYS.
THYME
AMEND

FoxTrot © 1998 Bill Amend. Reprinted with permission of Universal Press Syndicate. All rights reserved.

Some Problems with Cookies

- **The problem is privacy, not security.**
  - Servers can remember your previous actions
  - If you give out personal information, servers can link that information to your previous actions
  - Servers can share cookie information through use of a cooperating third party like doubleclick.net
  - Poorly designed sites store sensitive information like credit card numbers directly in cookie
  - JavaScript bugs let hostile sites steal cookies (old browsers)
- **Moral for servlet authors**
  - If cookies are not critical to your task, avoid servlets that totally fail when cookies are disabled
  - Don't put sensitive info in cookies

Sending Cookies to Browser

- **Standard approach:**

```
Cookie c = new Cookie("name", "value");
c.setMaxAge(...); // Means cookie persists on disk
// Set other attributes.
response.addCookie(c);
```

- **Simplified approach:**
  - Use LongLivedCookie class:

```
public class LongLivedCookie extends Cookie {
  public static final int SECONDS_PER_YEAR =
    60*60*24*365;

  public LongLivedCookie(String name, String value) {
    super(name, value);
    setMaxAge(SECONDS_PER_YEAR);
  }
}
```

Reading Cookies from Browser

- **Standard approach:**

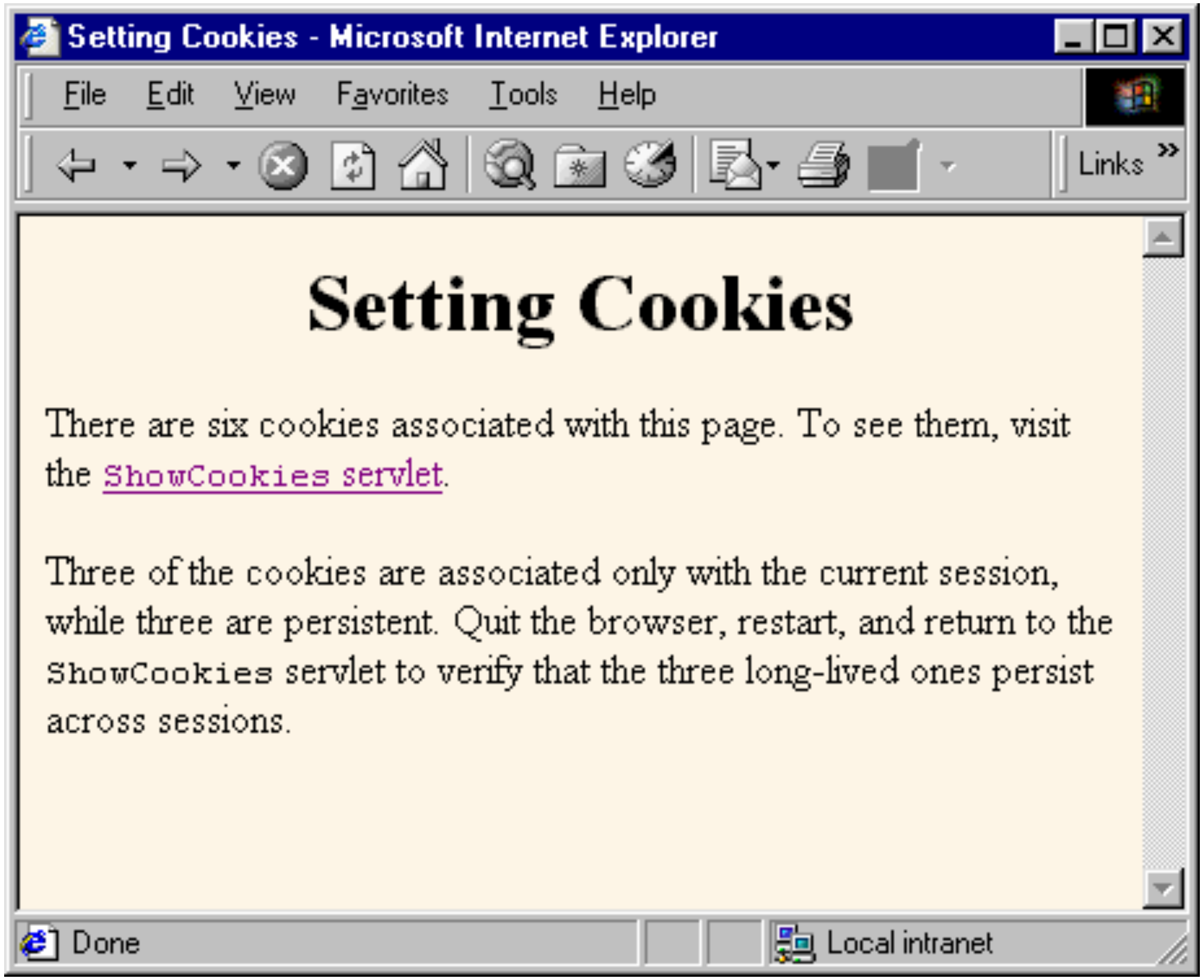
```
Cookie[] cookies = request.getCookies();
if (cookies != null) {
  for(int i=0; i<cookies.length; i++) {
    Cookie c = cookies[i];
    if (c.getName().equals("someName")) {
      doSomethingWith(c);
      break;
    }
  }
}
```

- **Simplified approach:**
  – Extract cookie or cookie value from cookie array by using ServletUtilities.getCookieValue or ServletUtilities.getCookie

Simple Cookie-Setting Servlet

```
public class SetCookies extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    for(int i=0; i<3; i++) {
      Cookie cookie = new Cookie("Session-Cookie-" + i,
                                 "Cookie-Value-S" + i);
      response.addCookie(cookie);
      cookie = new Cookie("Persistent-Cookie-" + i,
                          "Cookie-Value-P" + i);
      cookie.setMaxAge(3600);
      response.addCookie(cookie);
    }
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println(...);
```

Result of Cookie-Setting Servlet


Setting Cookies - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Done
Local intranet

Setting Cookies

There are six cookies associated with this page. To see them, visit the `ShowCookies` servlet.

Three of the cookies are associated only with the current session, while three are persistent. Quit the browser, restart, and return to the `ShowCookies` servlet to verify that the three long-lived ones persist across sessions.

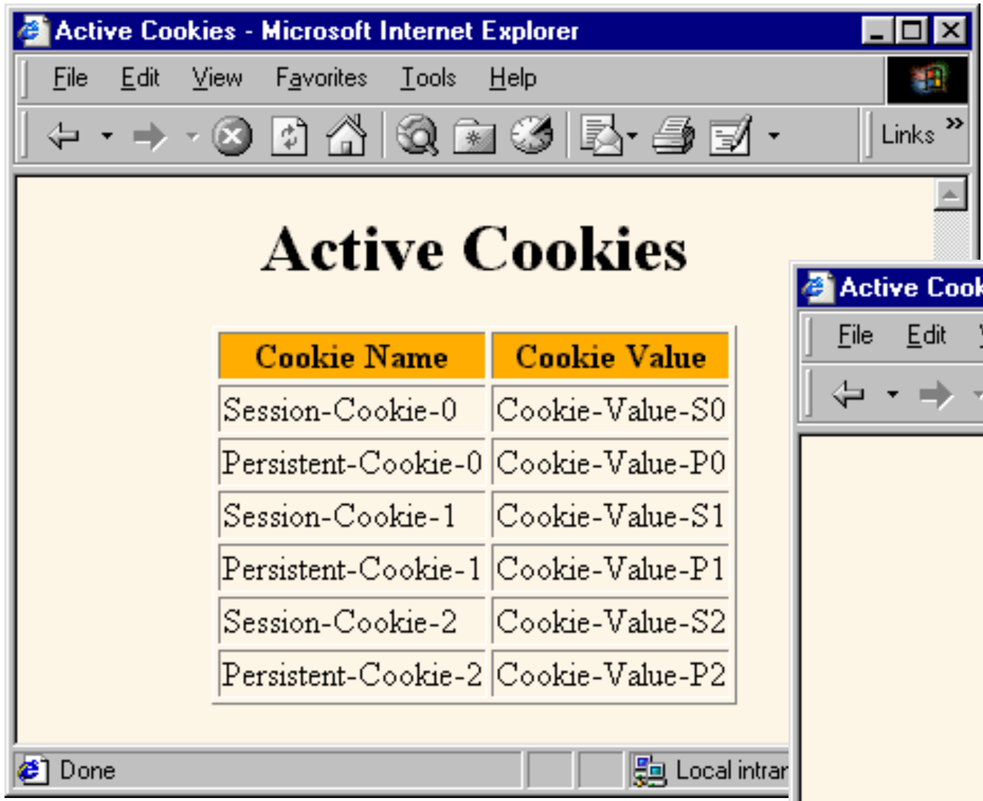
Simple Cookie-Viewing Servlet

```
public class ShowCookies extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    String title = "Active Cookies";
    out.println(ServletUtilities.headWithTitle(title) +
                "<BODY BGCOLOR=\"#FDF5E6\">\n" +
                "<H1 ALIGN=\"CENTER\">" + title +
                "</H1>\n" +
                "<TABLE BORDER=1 ALIGN=\"CENTER\">\n" +
                "<TR BGCOLOR=\"#FFAD00\">\n" +
                "  <TH>Cookie Name\n" +
                "  <TH>Cookie Value");
```

Simple Cookie-Viewing Servlet (Continued)

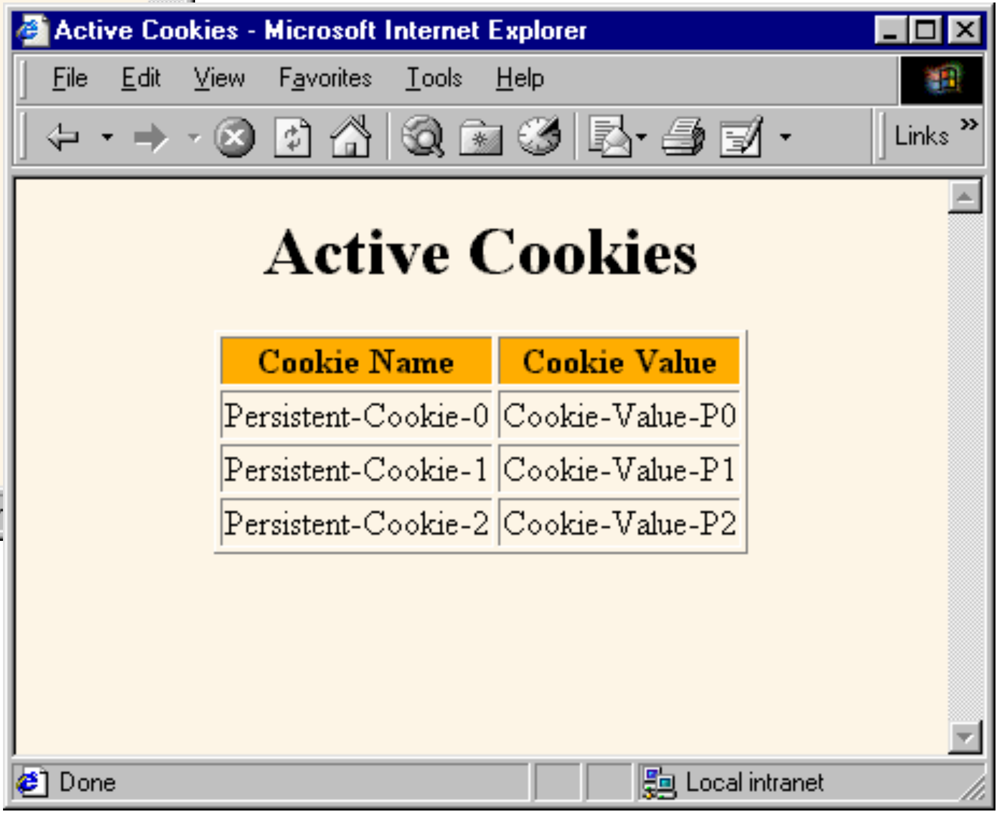
```
    Cookie[] cookies = request.getCookies();
    if (cookies != null) {
      Cookie cookie;
      for(int i=0; i<cookies.length; i++) {
        cookie = cookies[i];
        out.println("<TR>\n" +
                    "  <TD>" + cookie.getName() + "\n" +
                    "  <TD>" + cookie.getValue());
      }
    }
    out.println("</TABLE></BODY></HTML>");
  }
}
```

Result of Cookie-Viewer (Before & After Restarting Browser)


Active Cookies - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Done
Local intran

Active Cookies

Cookie Name	Cookie Value
Session-Cookie-0	Cookie-Value-S0
Persistent-Cookie-0	Cookie-Value-P0
Session-Cookie-1	Cookie-Value-S1
Persistent-Cookie-1	Cookie-Value-P1
Session-Cookie-2	Cookie-Value-S2
Persistent-Cookie-2	Cookie-Value-P2


Active Cookies - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Links
Done
Local intranet

Active Cookies

Cookie Name	Cookie Value
Persistent-Cookie-0	Cookie-Value-P0
Persistent-Cookie-1	Cookie-Value-P1
Persistent-Cookie-2	Cookie-Value-P2

Cookie Utilities

- **Problem**
  - getCookies returns an array of cookies
  - You almost always only care about one particular cookie
- **Solution**
  - Static methods to
    - Extract a cookie value given a cookie name (default value if no match)
    - Extract a Cookie object given a cookie name (null if no match)

ServletUtilities.getCookieValue

```
public static String getCookieValue(Cookie[] cookies,
                                    String cookieName,
                                    String defaultVal) {
  if (cookies != null) {
    for(int i=0; i<cookies.length; i++) {
      Cookie cookie = cookies[i];
      if (cookieName.equals(cookie.getName()))
        return(cookie.getValue());
    }
  }
  return(defaultVal);
}
```

ServletUtilities.getCookie

```
public static Cookie getCookie(Cookie[] cookies,
                               String cookieName) {
  if (cookies != null) {
    for(int i=0; i<cookies.length; i++) {
      Cookie cookie = cookies[i];
      if (cookieName.equals(cookie.getName()))
        return(cookie);
    }
  }
  return(null);
}
```

Methods in the Cookie API

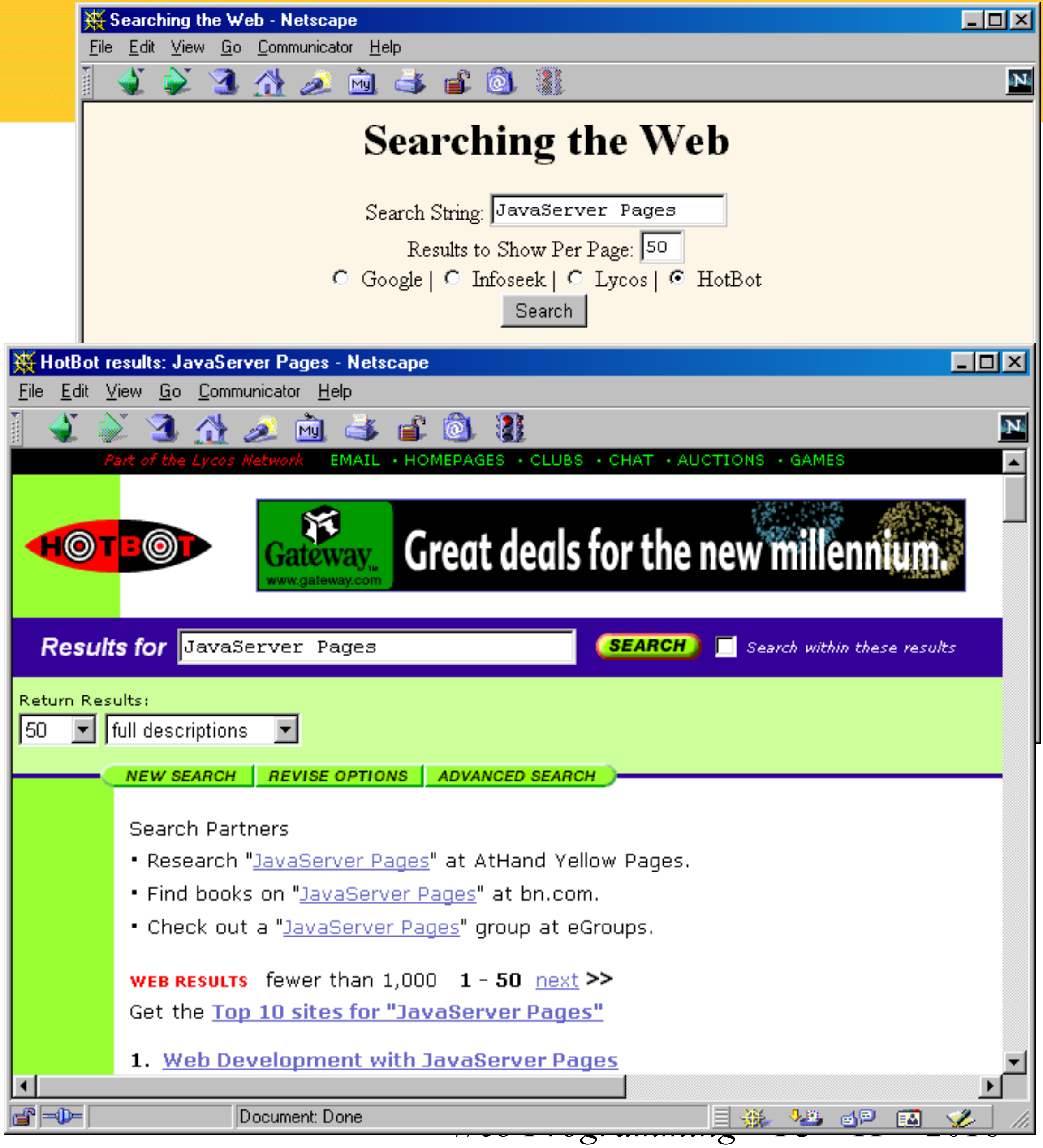
- **getDomain/setDomain**
  - Lets you specify domain to which cookie applies. Current host must be part of domain specified.
- **getMaxAge/setMaxAge**
  - Gets/sets the cookie expiration time (in seconds). If you fail to set this, cookie applies to current browsing session only. See LongLivedCookie helper class given earlier.
- **getName**
  - Gets the cookie name. There is no setName method; you supply name to constructor. For incoming cookie array, you use getName to find the cookie of interest.

Methods in the Cookie API (Continued)

- **getPath/setPath**
  - Gets/sets the path to which cookie applies. If unspecified, cookie applies to URLs that are within or below directory containing current page.
- **getSecure/setSecure**
  - Gets/sets flag indicating whether cookie should apply only to SSL connections or to all connections.
- **getValue/setValue**
  - Gets/sets value associated with cookie. For new cookies, you supply value to constructor, not to setValue. For incoming cookie array, you use getName to find the cookie of interest, then call getValue on the result. If you set the value of an incoming cookie, you still have to send it back out with response.addCookie.

A Customized Search Engine Interface

- Front end remembers settings for search engine, search string, and hits per page
  - Front end *uses* cookies
  - Back end *sets* cookies
  - In real life, don't really show previous queries!



Searching the Web - Netscape

Searching the Web

Search String: JavaServer Pages

Results to Show Per Page: 50

( ) Google | ( ) Infoseek | ( ) Lycos | (●) HotBot

Search

HotBot results: JavaServer Pages - Netscape

Part of the Lycos Network EMAIL • HOMEPAGES • CLUBS • CHAT • AUCTIONS • GAMES

Results for JavaServer Pages SEARCH ☐ Search within these results

Return Results: 50 full descriptions

NEW SEARCH | REVISE OPTIONS | ADVANCED SEARCH

Search Partners

- Research "JavaServer Pages" at AtHand Yellow Pages.
- Find books on "JavaServer Pages" at bn.com.
- Check out a "JavaServer Pages" group at eGroups.

WEB RESULTS fewer than 1,000 1 - 50 next >>

Get the Top 10 sites for "JavaServer Pages"

1. Web Development with JavaServer Pages

Document: Done

File Edit View Go Communicator Help
HOTBOT
Gateway www.gateway.com
Great deals for the new millennium.

Front End to SearchEngines Servlet

```
public class SearchEnginesFrontEnd extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    Cookie[] cookies = request.getCookies();
    String searchString =
      ServletUtilities.getCookieValue(cookies,
                                      "searchString",
                                      "Java Programming");
    String numResults =
      ServletUtilities.getCookieValue(cookies,
                                      "numResults",
                                      "10");
    String searchEngine =
      ServletUtilities.getCookieValue(cookies,
                                      "searchEngine",
                                      "google");
```

Front End to SearchEngines Servlet (Continued)

```
...
out.println
  (...
   "<FORM ACTION=\"/servlet/" +
     "coreservlets.CustomizedSearchEngines\">\n" +
   "<CENTER>\n" +
   "Search String:\n" +
   "<INPUT TYPE=\"TEXT\" NAME=\"searchString\"\n" +
   "       VALUE=\"" + searchString + "\"><BR>\n" +
   "Results to Show Per Page:\n" +
   "<INPUT TYPE=\"TEXT\" NAME=\"numResults\"\n" +
   "       VALUE=" + numResults + " SIZE=3><BR>\n" +
   "<INPUT TYPE=\"RADIO\" NAME=\"searchEngine\"\n" +
   "       VALUE=\"google\"" +
   checked("google", searchEngine) + ">\n" +
   ...);
```

Customized SearchEngines Servlet (Back End)

```
public class CustomizedSearchEngines extends HttpServlet {
  public void doGet(HttpServletRequest request,
                    HttpServletResponse response)
      throws ServletException, IOException {
    String searchString =
      request.getParameter("searchString");
    if ((searchString == null) ||
        (searchString.length() == 0)) {
      reportProblem(response, "Missing search string.");
      return;
    }
    Cookie searchStringCookie =
      new LongLivedCookie("searchString", searchString);
    response.addCookie(searchStringCookie);
    ...
  }
}
```

Summary

- **Cookies involve name/value pairs sent from server to browser and returned when the same page, site, or domain is visited later**
- **Let you**
  - Track sessions (use higher-level API)
  - Permit users to avoid logging in at low-security sites
  - Customize sites for different users
  - Focus content or advertising
- **Setting cookies**
  - Call Cookie constructor, set age, call response.addCookie
- **Reading cookies**
  - Call request.getCookies, check for null, look through array for matching name, use associated value

Agenda - Session Tracking

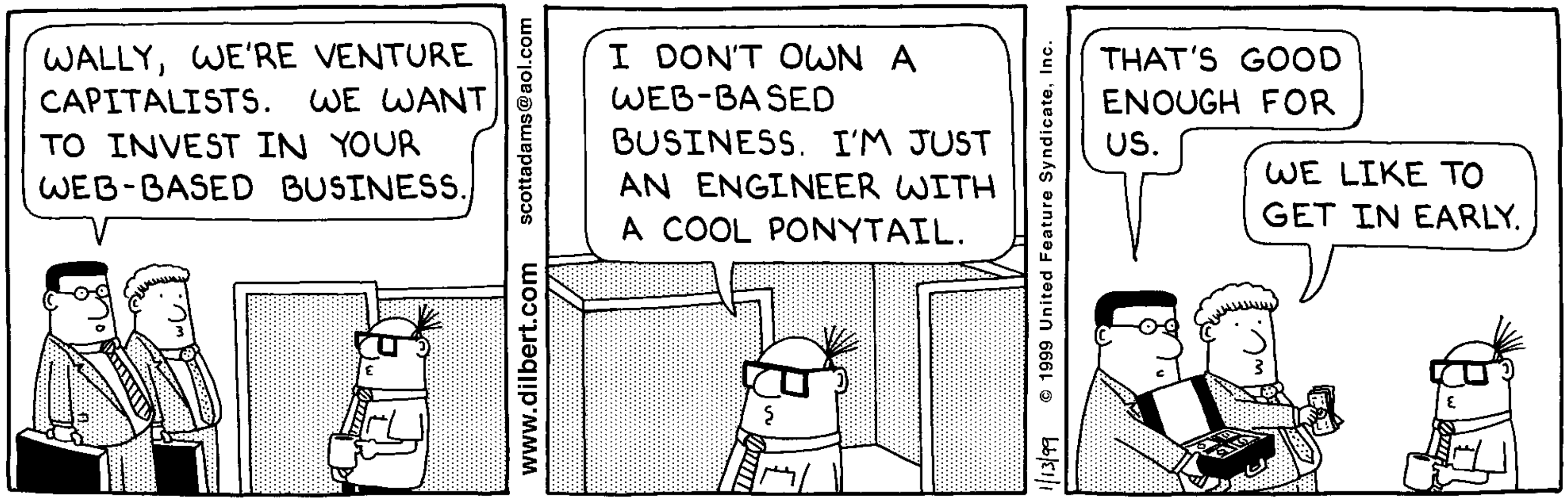
- The purpose of session tracking
- Rolling your own session tracking
- The session tracking API
- Per-client access counts
- Shopping carts and online stores

What is a session?

- **A Session refers to all the request that a single client makes to a server.**
- **A session is specific to the user and for each user a new session is created to track all the request from that user.**
- **Every user has a separate session and separate session variable is associated with that session.**

Session Tracking and E-Commerce

- **Why session tracking?**
  - When clients at on-line store add item to their shopping cart, how does server know what's already in cart?
  - When clients decide to proceed to checkout, how can server determine which previously created cart is theirs?


WALLY, WE'RE VENTURE CAPITALISTS. WE WANT TO INVEST IN YOUR WEB-BASED BUSINESS.
scottadams@aol.com
www.dilbert.com
I DON'T OWN A WEB-BASED BUSINESS. I'M JUST AN ENGINEER WITH A COOL PONYTAIL.
© 1999 United Feature Syndicate, Inc.
1/13/99
THAT'S GOOD ENOUGH FOR US.
WE LIKE TO GET IN EARLY.

Dilbert used with permission of United Syndicates Inc.

Rolling Your Own Session Tracking: Cookies

- **Idea: associate cookie with data on server**

```
String sessionID = makeUniqueString();
Hashtable sessionInfo = new Hashtable();
Hashtable globalTable = findTableStoringSessions();
globalTable.put(sessionID, sessionInfo);
Cookie sessionCookie =
  new Cookie("JSESSIONID", sessionID);
sessionCookie.setPath("/");
response.addCookie(sessionCookie);
```

- **Still to be done:**
  - Extracting cookie that stores session identifier
  - Setting appropriate expiration time for cookie
  - Associating the hash tables with each request
  - Generating the unique session identifiers

Rolling Your Own Session Tracking: URL-Rewriting

- **Idea**
  - Client appends some extra data on the end of each URL that identifies the session
  - Server associates that identifier with data it has stored about that session
  - E.g., http://host/path/file.html;jsessionid=1234
- **Advantage**
  - Works even if cookies are disabled or unsupported
- **Disadvantages**
  - Lots of tedious processing
  - Must encode all URLs that refer to your own site
  - Links from other sites and bookmarks can fail

Rolling Your Own Session Tracking: Hidden Form Fields

- **Idea:**

```
<INPUT TYPE="HIDDEN" NAME="session" VALUE="...">
```

- **Advantage**
  - Works even if cookies are disabled or unsupported
- **Disadvantages**
  - Lots of tedious processing
  - All pages must be the result of form submissions

The Session Tracking API

- **Session objects live on the server**
- **Automatically associated with client via cookies or URL-rewriting**
  - Use request.getSession(true) to get either existing or new session
    - Behind the scenes, the system looks at cookie or URL extra info and sees if it matches the key to some previously stored session object. If so, it returns that object. If not, it creates a new one, assigns a cookie or URL info as its key, and returns that new session object.
- **Hashtable-like mechanism lets you store arbitrary objects inside session**
  - setAttribute (putValue in 2.1) stores values
  - getAttribute (getValue in 2.1) retrieves values

Accessing Session Data

```
HttpSession session = request.getSession(true);
ShoppingCart cart =
   (ShoppingCart)session.getAttribute("shoppingCart");
if (cart == null) { // No cart already in session
   cart = new ShoppingCart();
   session.setAttribute("shoppingCart", cart);
}
doSomethingWith(cart);
```

HttpSession Methods

- **getAttribute (getValue in old servlet spec 2.1)**
  - Extracts a previously stored value from a session object. Returns null if no value is associated with given name.
- **setAttribute (putValue in ver. 2.1)**
  - Associates a value with a name. Monitor changes: values implement HttpSessionBindingListener.
- **removeAttribute (removeValue in ver. 2.1)**
  - Removes values associated with name.
- **getAttributeNames (getValueNames in 2.1)**
  - Returns names of all attributes in the session.
- **getId**
  - Returns the unique identifier.

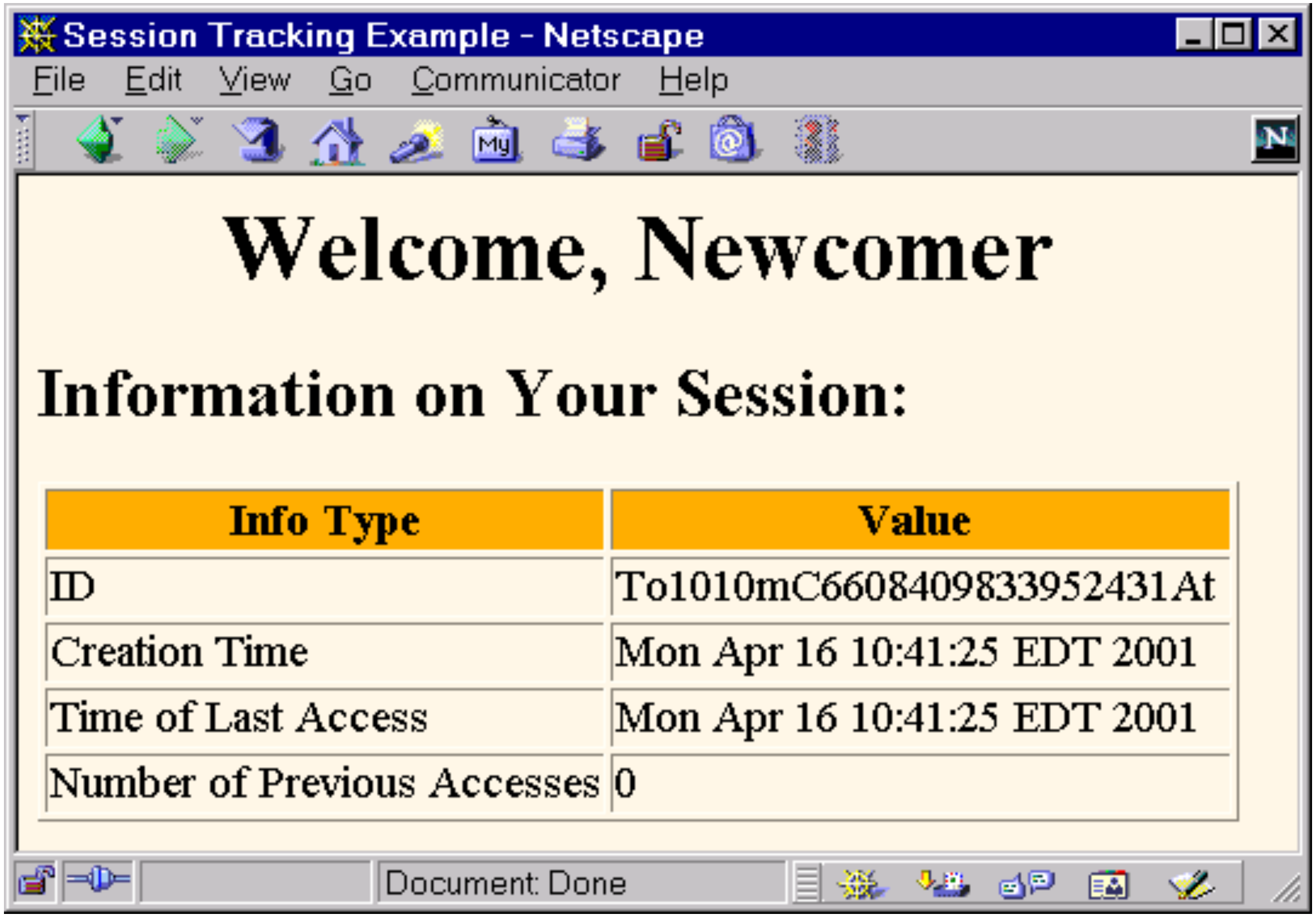
HttpSession Methods (Continued)

- **isNew**
  - Determines if session is new to *client* (not to *page*)
- **getCreationTime**
  - Returns time at which session was first created
- **getLastAccessedTime**
  - Returns time at which session was last sent from client
- **getMaxInactiveInterval, setMaxInactiveInterval**
  - Gets or sets the amount of time session should go without access before being invalidated
- **invalidate**
  - Invalidates the session and unbinds all objects associated with it

A Servlet Showing Per-Client Access Counts

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
    throws ServletException, IOException {
  response.setContentType("text/html");
  PrintWriter out = response.getWriter();
  String title = "Session Tracking Example";
  HttpSession session = request.getSession(true);
  String heading;
  Integer accessCount =
    (Integer)session.getAttribute("accessCount");
  if (accessCount == null) {
    accessCount = new Integer(0);
    heading = "Welcome, Newcomer";
  } else {
    heading = "Welcome Back";
    accessCount = new Integer(accessCount.intValue() + 1);
  }
  session.setAttribute("accessCount", accessCount);
```

First Visit to ShowSession Servlet

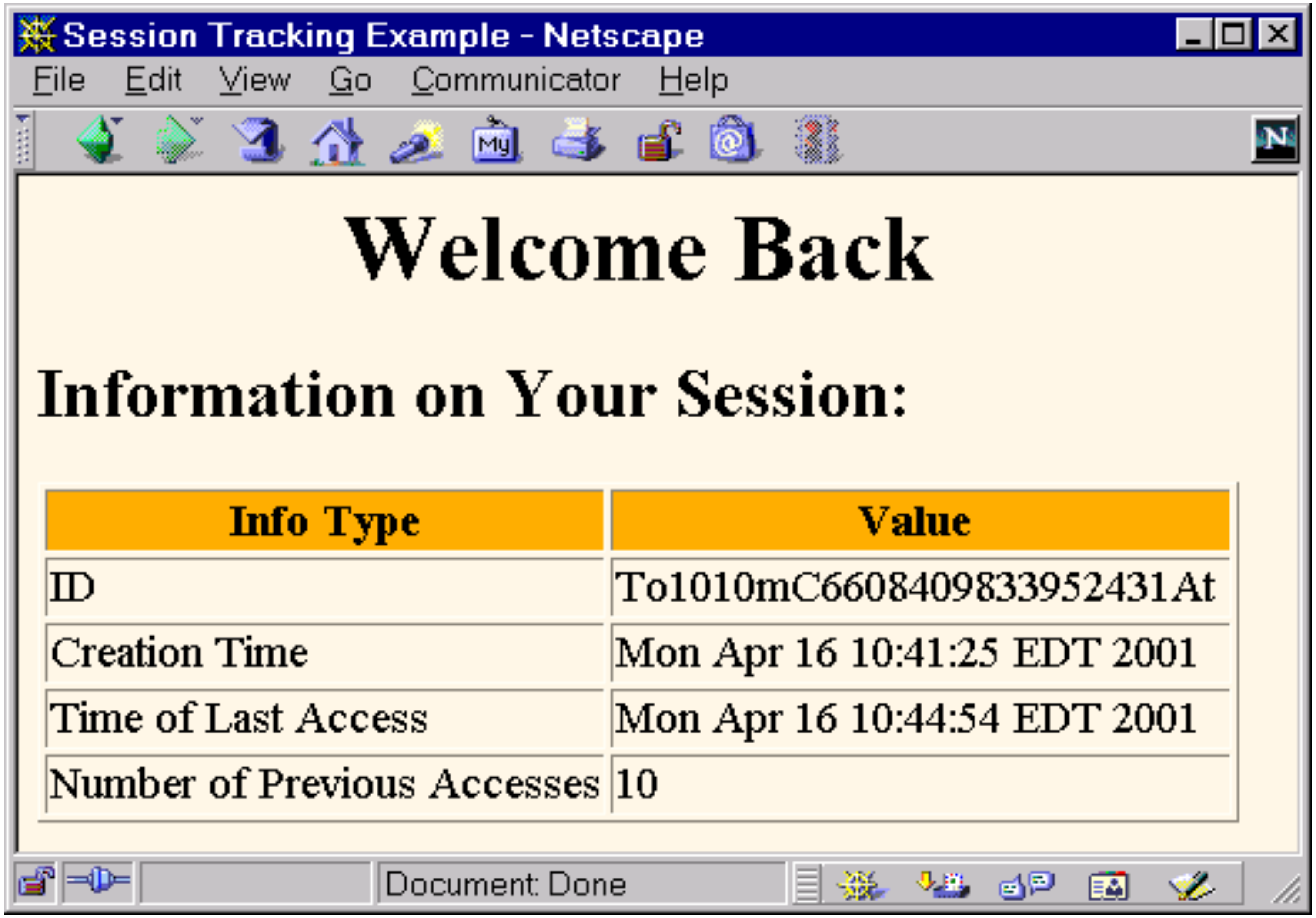

Session Tracking Example - Netscape
File Edit View Go Communicator Help
Document: Done

Welcome, Newcomer

Information on Your Session:

Info Type	Value
ID	To1010mC6608409833952431At
Creation Time	Mon Apr 16 10:41:25 EDT 2001
Time of Last Access	Mon Apr 16 10:41:25 EDT 2001
Number of Previous Accesses	0

Eleventh Visit to ShowSession Servlet

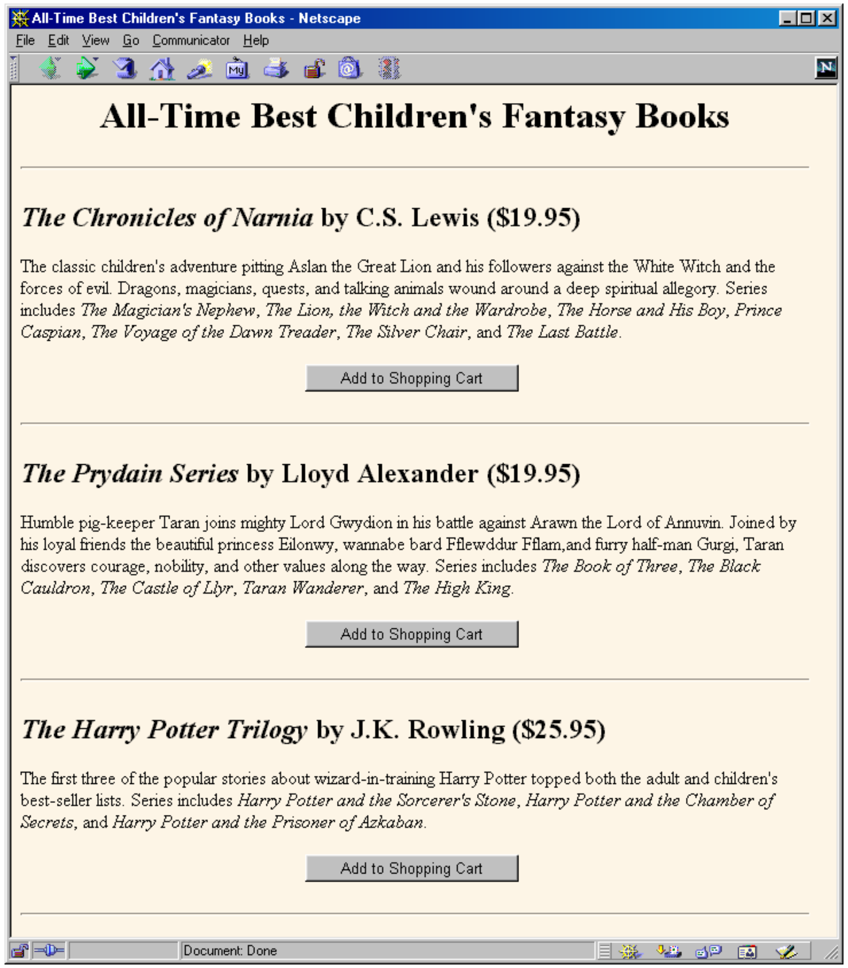

Session Tracking Example - Netscape
File Edit View Go Communicator Help
Document: Done

Welcome Back

Information on Your Session:

Info Type	Value
ID	To1010mC6608409833952431At
Creation Time	Mon Apr 16 10:41:25 EDT 2001
Time of Last Access	Mon Apr 16 10:44:54 EDT 2001
Number of Previous Accesses	10

Session Tracking and Shopping Carts



All-Time Best Children's Fantasy Books - Netscape
File Edit View Go Communicator Help
Document: Done

All-Time Best Children's Fantasy Books

The Chronicles of Narnia by C.S. Lewis ($19.95)

The classic children's adventure pitting Aslan the Great Lion and his followers against the White Witch and the forces of evil. Dragons, magicians, quests, and talking animals wound around a deep spiritual allegory. Series includes *The Magician's Nephew*, *The Lion, the Witch and the Wardrobe*, *The Horse and His Boy*, *Prince Caspian*, *The Voyage of the Dawn Treader*, *The Silver Chair*, and *The Last Battle*.

Add to Shopping Cart

The Prydain Series by Lloyd Alexander ($19.95)

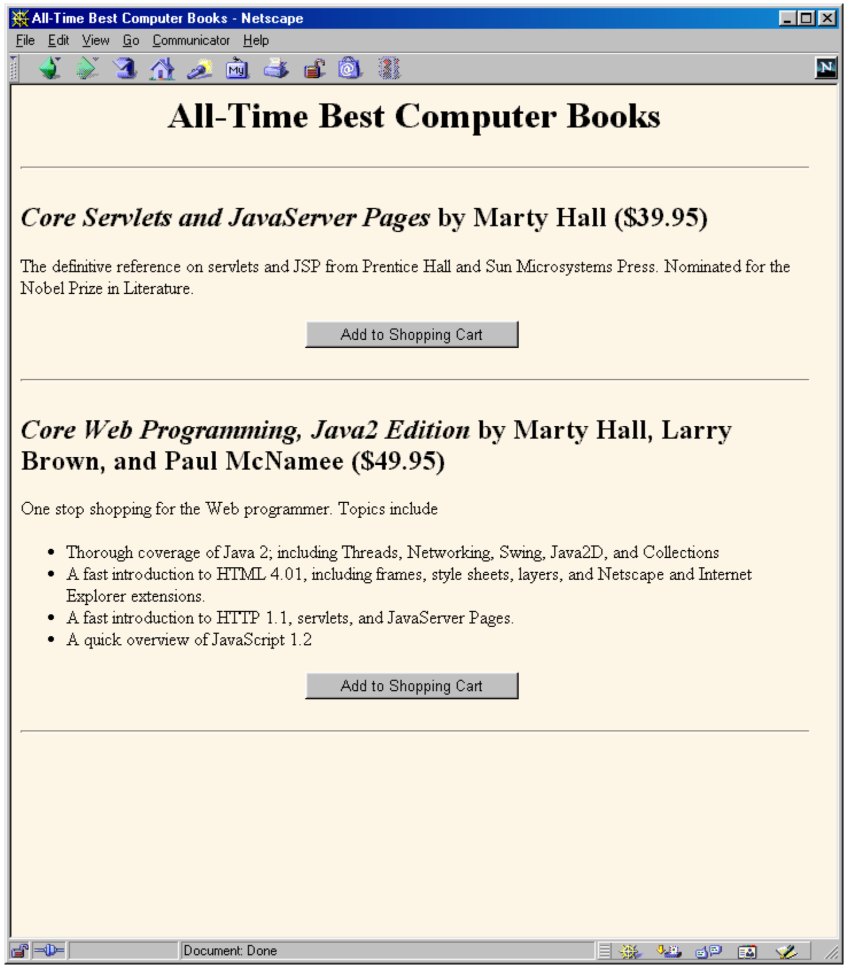
Humble pig-keeper Taran joins mighty Lord Gwydion in his battle against Arawn the Lord of Annuvin. Joined by his loyal friends the beautiful princess Eilonwy, wannabe bard Fflewddur Fflam,and furry half-man Gurgi, Taran discovers courage, nobility, and other values along the way. Series includes *The Book of Three*, *The Black Cauldron*, *The Castle of Llyr*, *Taran Wanderer*, and *The High King*.

Add to Shopping Cart

The Harry Potter Trilogy by J.K. Rowling ($25.95)

The first three of the popular stories about wizard-in-training Harry Potter topped both the adult and children's best-seller lists. Series includes *Harry Potter and the Sorcerer's Stone*, *Harry Potter and the Chamber of Secrets*, and *Harry Potter and the Prisoner of Azkaban*.

Add to Shopping Cart



All-Time Best Computer Books - Netscape
File Edit View Go Communicator Help
Document: Done

All-Time Best Computer Books

Core Servlets and JavaServer Pages by Marty Hall ($39.95)

The definitive reference on servlets and JSP from Prentice Hall and Sun Microsystems Press. Nominated for the Nobel Prize in Literature.

Add to Shopping Cart

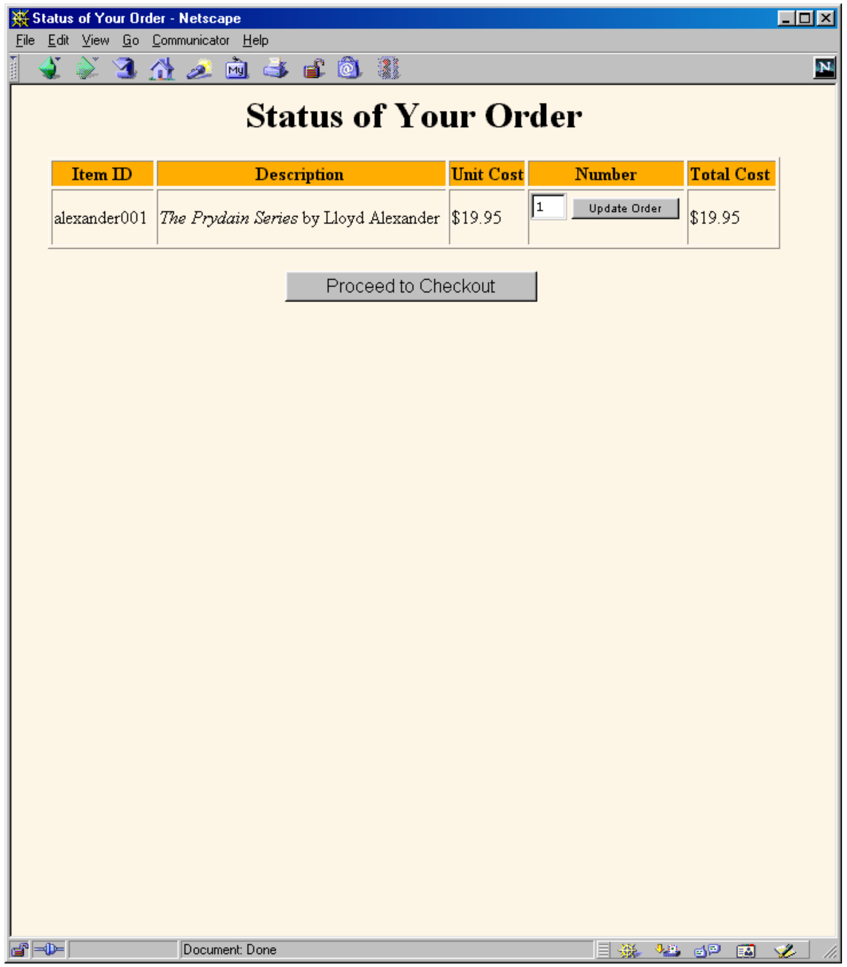
Core Web Programming, Java2 Edition by Marty Hall, Larry Brown, and Paul McNamee ($49.95)

One stop shopping for the Web programmer. Topics include

- Thorough coverage of Java 2; including Threads, Networking, Swing, Java2D, and Collections
- A fast introduction to HTML 4.01, including frames, style sheets, layers, and Netscape and Internet Explorer extensions.
- A fast introduction to HTTP 1.1, servlets, and JavaServer Pages.
- A quick overview of JavaScript 1.2

Add to Shopping Cart

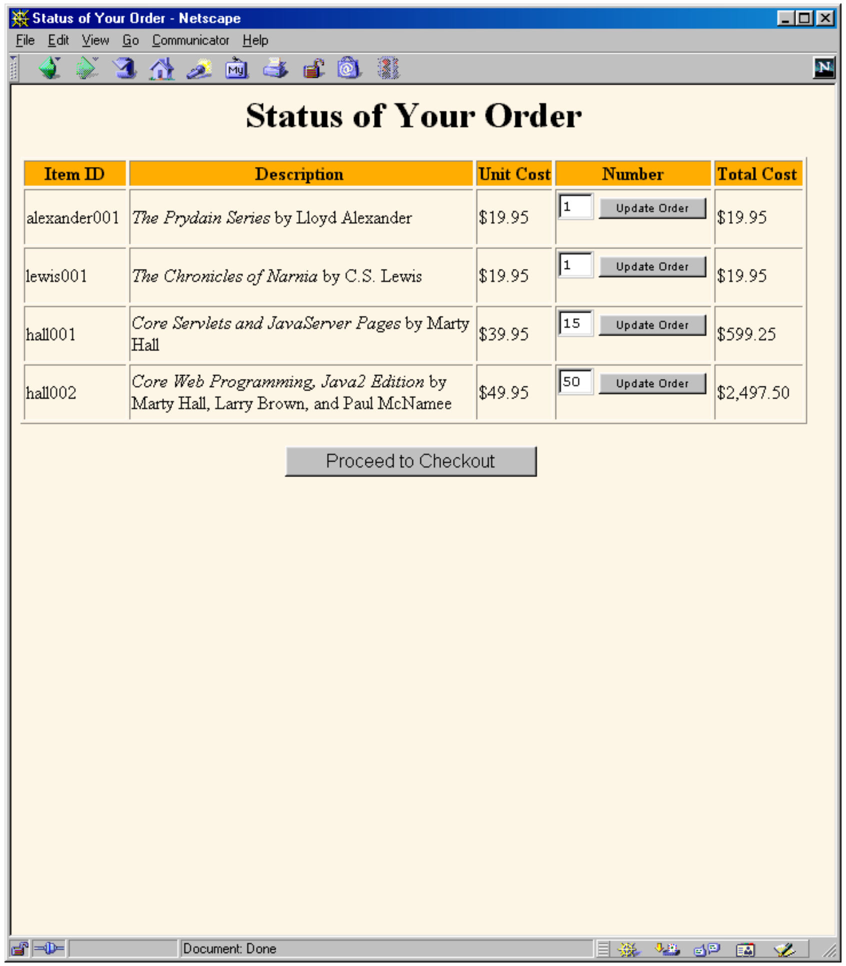
Session Tracking and Shopping Carts (Continued)


Status of Your Order - Netscape
File Edit View Go Communicator Help
Document: Done

Status of Your Order

Item ID	Description	Unit Cost	Number	Total Cost
alexander001	*The Prydain Series* by Lloyd Alexander	$19.95	1 Update Order	$19.95

Proceed to Checkout


Status of Your Order - Netscape
File Edit View Go Communicator Help
Document: Done

Status of Your Order

Item ID	Description	Unit Cost	Number	Total Cost
alexander001	*The Prydain Series* by Lloyd Alexander	$19.95	1 Update Order	$19.95
lewis001	*The Chronicles of Narnia* by C.S. Lewis	$19.95	1 Update Order	$19.95
hall001	*Core Servlets and JavaServer Pages* by Marty Hall	$39.95	15 Update Order	$599.25
hall002	*Core Web Programming, Java2 Edition* by Marty Hall, Larry Brown, and Paul McNamee	$49.95	50 Update Order	$2,497.50

Proceed to Checkout

What Changes if Server Uses URL Rewriting?

- **Session tracking code:**
  - No change
- **Code that generates hypertext links back to same site:**
  - Pass URL through response.encodeURL.
    - If server is using cookies, this returns URL unchanged
    - If server is using URL rewriting, this appends the session info to the URL
    - E.g.:
```
String url = "order-page.html";
url = response.encodeURL(url);
```
- **Code that does sendRedirect to same site:**
  - Pass URL through response.encodeRedirectURL

Summary

- **Although it usually uses cookies behind the scenes, the session tracking API is higher-level and easier to use than the cookie API**
  - If server supports URL-rewriting, *your* code unchanged
- **Session information lives on server**
  - Cookie or extra URL info associates it with a user
- **Obtaining session**
  - request.getSession(true)
- **Associating values with keys**
  - session.setAttribute (or session.putValue)
- **Finding values associated with keys**
  - session.getAttribute (or session.getValue)
    - Always check if this value is null

Homework

- Reading chapter 19: Java Server Page (Core Web Programming, second edition)
- Doing the exercise of chapter 5 (Web State Management) in the Book: "Giáo trình xây dựng ứng dụng web cho thương mại điện tử trên Netbeans".