

# Agenda for this lecture:

- Sessions and Cookies
- Shopping cart examples
- Web Page Layout

# Maintaining State

- HTTP is a stateless protocol
- Each request is completely independent of the previous
- Most applications need to understand the current state of the application
  - The pervasive "shopping cart"
  - Personalization
  - Workflow

# How can you maintain state?

- State must either be completely part of the request, or maintained on the server
  - Cookies - simple data
  - Sessions - data for a current session (across pages)
  - Databases - persisting data across sessions

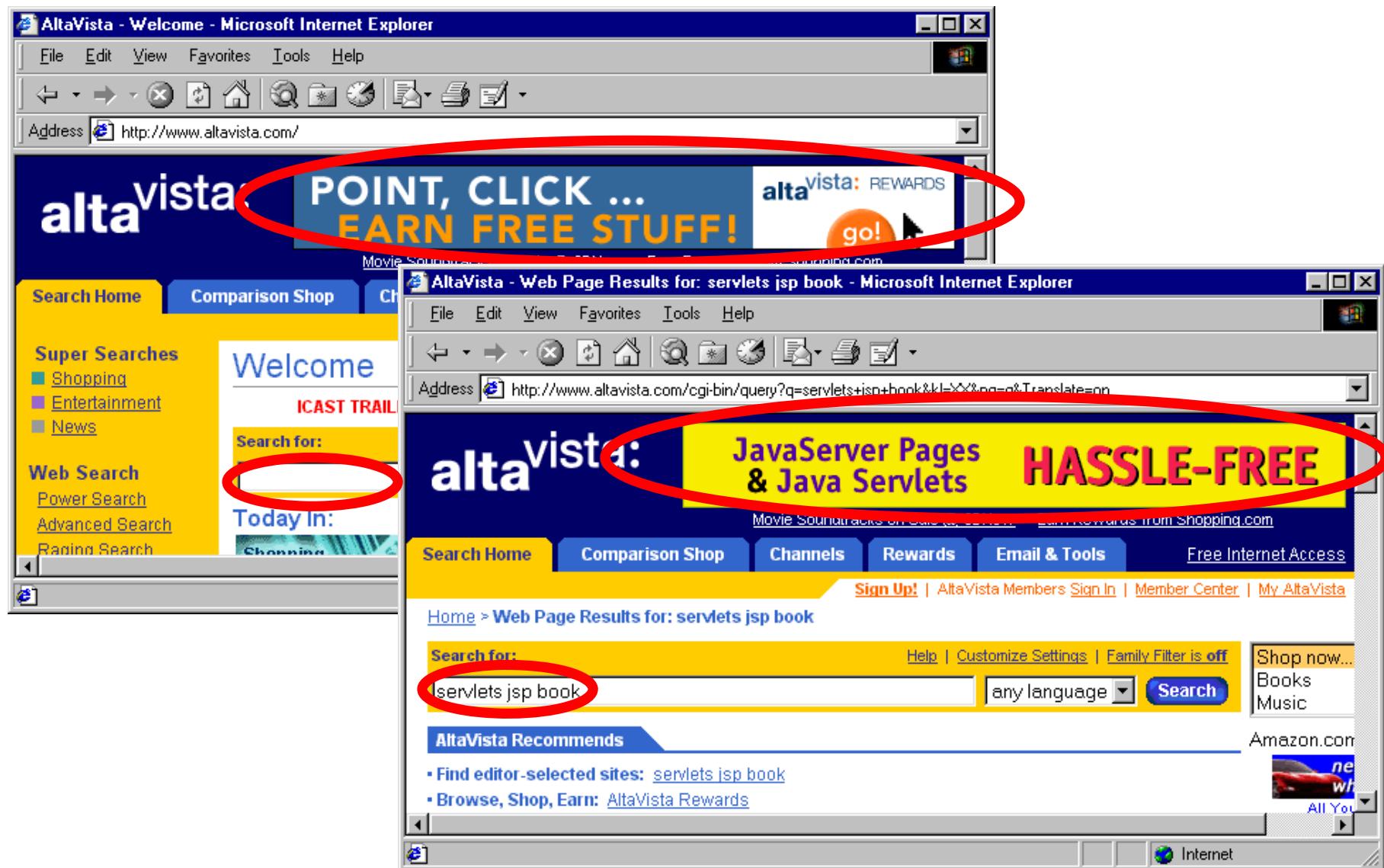
# Cookies

- Defined in RFC 2109
- **A cookie is basically a small piece of information stored by a web browser on the local client.**
- Browsers are expected to only support 20 cookies per Web server, 300 total, and can be limited to 4kb each
- Cookies are returned to the server as HTTP request headers

# Cookie uses

- Cookies are used on the web for a number of purposes
  - Identifying a user during an e-commerce session
  - Tracking Sessions
  - Focusing advertising
  - Tracking usage
  - Personalization
- If used properly, not a security risk

# Cookies and Focused Advertising



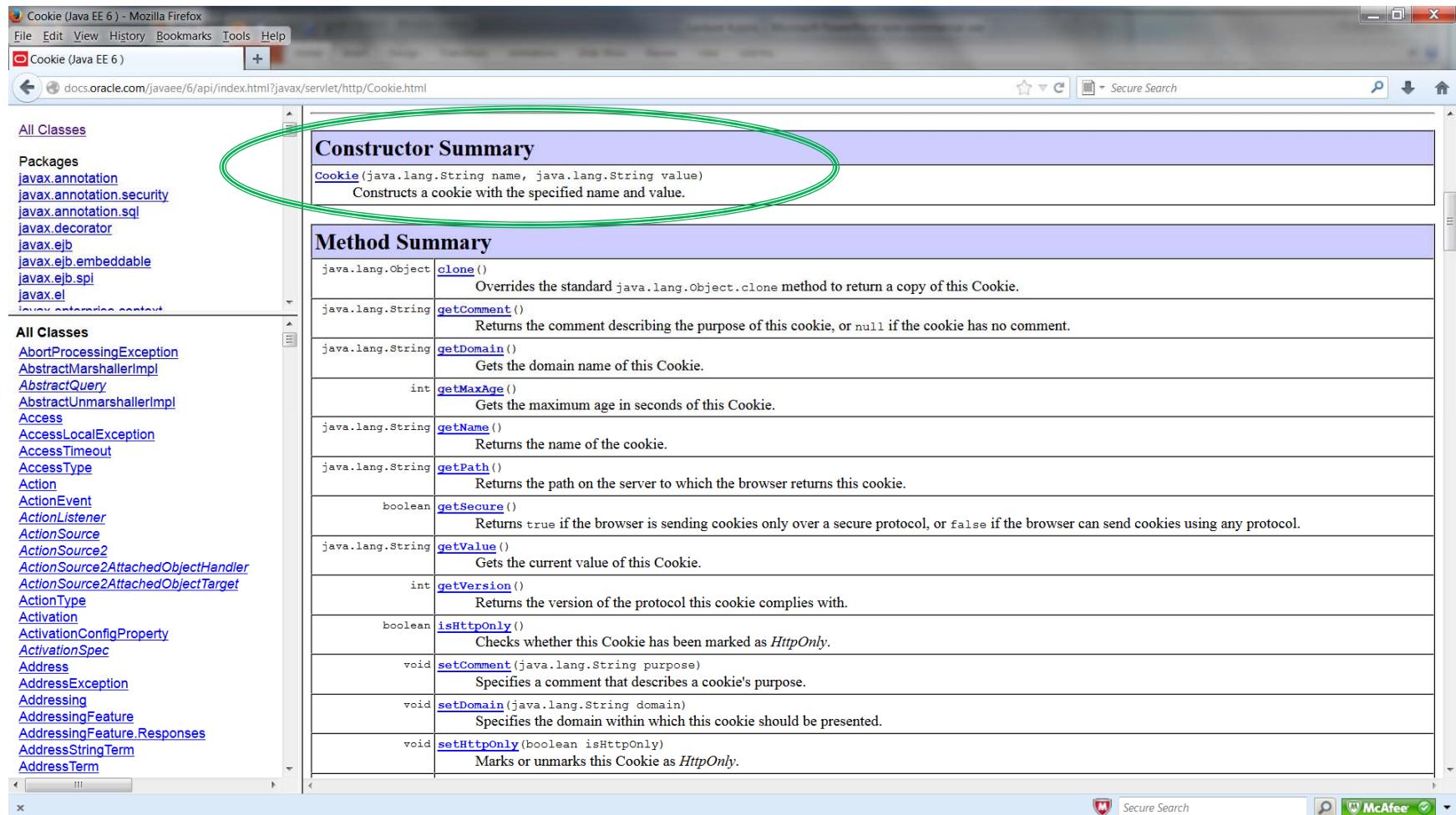
# Problems with Cookies

- **Some may argue that the problem with the cookies is privacy NOT security!**
  - A user can turn them off
  - They can violate privacy
  - If used improperly, a security risk. For example, Poorly designed sites store sensitive information like credit card numbers directly in cookie
- **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

# Cookie API

The screenshot shows a Mozilla Firefox window displaying the Java EE 6 API documentation for the `Cookie` class. The URL in the address bar is `docs.oracle.com/javaee/6/api/index.html?javax/servlet/http/Cookie.html`. The page title is "Cookie (Java EE 6)". The left sidebar contains a navigation tree for the Java EE 6 API, with sections for `javax.security.auth.message.module`, `javax.security.jacc`, `javax.servlet`, `javax.servlet.annotation`, `javax.servlet.descriptor`, `javax.servlet.http`, `javax.servlet.jsp`, `javax.servlet.jsp.el`, `javax.servlet.jsp.jstl.core`, `javax.servlet.jsp.jstl.fmt`, `javax.servlet.jsp.jstl.sql`, and `javax.servlet.jsp.jstl.tlv`. Below this, under `javax.servlet.http`, there are sections for `Interfaces` (including `HttpServletRequest`, `HttpServletResponse`, `HttpSession`, `HttpSessionActivationListener`, `HttpSessionAttributeListener`, `HttpSessionBindingListener`, `HttpSessionContext`, `HttpSessionListener`, `Part`) and `Classes` (including `Cookie`, `HttpServlet`, `HttpServletRequestWrapper`, `HttpServletResponseWrapper`, `HttpSessionBindingEvent`, `HttpSessionEvent`, and `HttpUtils`). The main content area shows the `Class Cookie` documentation, which extends `java.lang.Object` and implements `java.io.Serializable` and `java.lang.Cloneable`. It describes the purpose of a cookie as a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server for session management. It also notes that cookies affect caching and supports both Version 0 and Version 1 cookie specifications. The `Author:` section lists "Various".

# Cookie API



The screenshot shows a Mozilla Firefox browser window displaying the Java EE 6 API documentation for the `Cookie` class. The URL in the address bar is `docs.oracle.com/javaee/6/api/index.html?javax/servlet/http/Cookie.html`.

The page is organized into several sections:

- Left Sidebar:** Contains links to "All Classes" and "Packages". Under "All Classes", there is a long list of Java EE components, including `AbortProcessingException`, `AbstractMarshallerImpl`, `AbstractQuery`, `AbstractUnmarshallerImpl`, `Access`, `AccessLocalException`, `AccessTimeout`, `AccessType`, `Action`, `ActionEvent`, `ActionListener`, `ActionSource`, `ActionSource2`, `ActionSourceAttachedObjectHandler`, `ActionSource2AttachedObjectTarget`, `ActionType`, `Activation`, `ActivationConfigProperty`, `ActivationSpec`, `Address`, `AddressException`, `Addressing`, `AddressingFeature`, `AddressingFeature.Responses`, `AddressStringTerm`, and `AddressTerm`.
- Constructor Summary:** This section is highlighted with a green oval. It contains the constructor for the `Cookie` class:

```
Cookie(java.lang.String name, java.lang.String value)
Constructs a cookie with the specified name and value.
```
- Method Summary:** This section lists various methods of the `Cookie` class, each with its return type and a brief description. The methods are listed in the following order:
  - `java.lang.Object clone()`: Overrides the standard `java.lang.Object.clone` method to return a copy of this `Cookie`.
  - `java.lang.String getComment()`: Returns the comment describing the purpose of this `Cookie`, or `null` if the `cookie` has no comment.
  - `java.lang.String getDomain()`: Gets the domain name of this `Cookie`.
  - `int getMaxAge()`: Gets the maximum age in seconds of this `Cookie`.
  - `java.lang.String getName()`: Returns the name of the `Cookie`.
  - `java.lang.String getPath()`: Returns the path on the server to which the browser returns this `Cookie`.
  - `boolean getSecure()`: Returns `true` if the browser is sending `cookies` only over a secure protocol, or `false` if the browser can send `cookies` using any protocol.
  - `java.lang.String getValue()`: Gets the current value of this `Cookie`.
  - `int getVersion()`: Returns the version of the protocol this `Cookie` complies with.
  - `boolean isHttpOnly()`: Checks whether this `Cookie` has been marked as `HttpOnly`.
  - `void setComment(java.lang.String purpose)`: Specifies a comment that describes a `Cookie`'s purpose.
  - `void setDomain(java.lang.String domain)`: Specifies the domain within which this `Cookie` should be presented.
  - `void setHttpOnly(boolean isHttpOnly)`: Marks or unmarks this `Cookie` as `HttpOnly`.

# Cookie API

The screenshot shows the Java EE 6 API documentation for the `Cookie` class in Mozilla Firefox. The URL in the address bar is `docs.oracle.com/javaee/6/api/index.html?javax/servlet/http/Cookie.html`. The page lists various methods of the `Cookie` class, many of which are highlighted with green ovals. These highlighted methods are:

- `getDomain()` - Gets the domain name of this Cookie.
- `getMaxAge()` - Gets the maximum age in seconds of this Cookie.
- `getName()` - Returns the name of the cookie.
- `getPath()` - Returns the path on the server to which the browser returns this cookie.
- `getSecure()` - Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol.
- `getValue()` - Gets the current value of this Cookie.
- `getVersion()` - Returns the version of the protocol this cookie complies with.
- `isHttpOnly()` - Checks whether this Cookie has been marked as *HttpOnly*.
- `setComment(java.lang.String purpose)` - Specifies a comment that describes a cookie's purpose.
- `setDomain(java.lang.String domain)` - Specifies the domain within which this cookie should be presented.
- `setHttpOnly(boolean isHttpOnly)` - Marks or unmarks this Cookie as *HttpOnly*.
- `setMaxAge(int expiry)` - Sets the maximum age in seconds for this Cookie.
- `setPath(java.lang.String uri)` - Specifies a path for the cookie to which the client should return the cookie.
- `setSecure(boolean flag)` - Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.
- `setValue(java.lang.String newValue)` - Assigns a new value to this Cookie.
- `setVersion(int v)` - Sets the version of the cookie protocol that this Cookie complies with.

# Cookie API

- `public Cookie(java.lang.String name, java.lang.String value)`
  - Creates a Cookie
- `public void setMaxAge(int expiry)`
  - Set to age in seconds when Cookie expires
  - 0 deletes it
  - negative value deletes it when browser exits
- getters/setters for Name, Value, Path, Domain, etc.

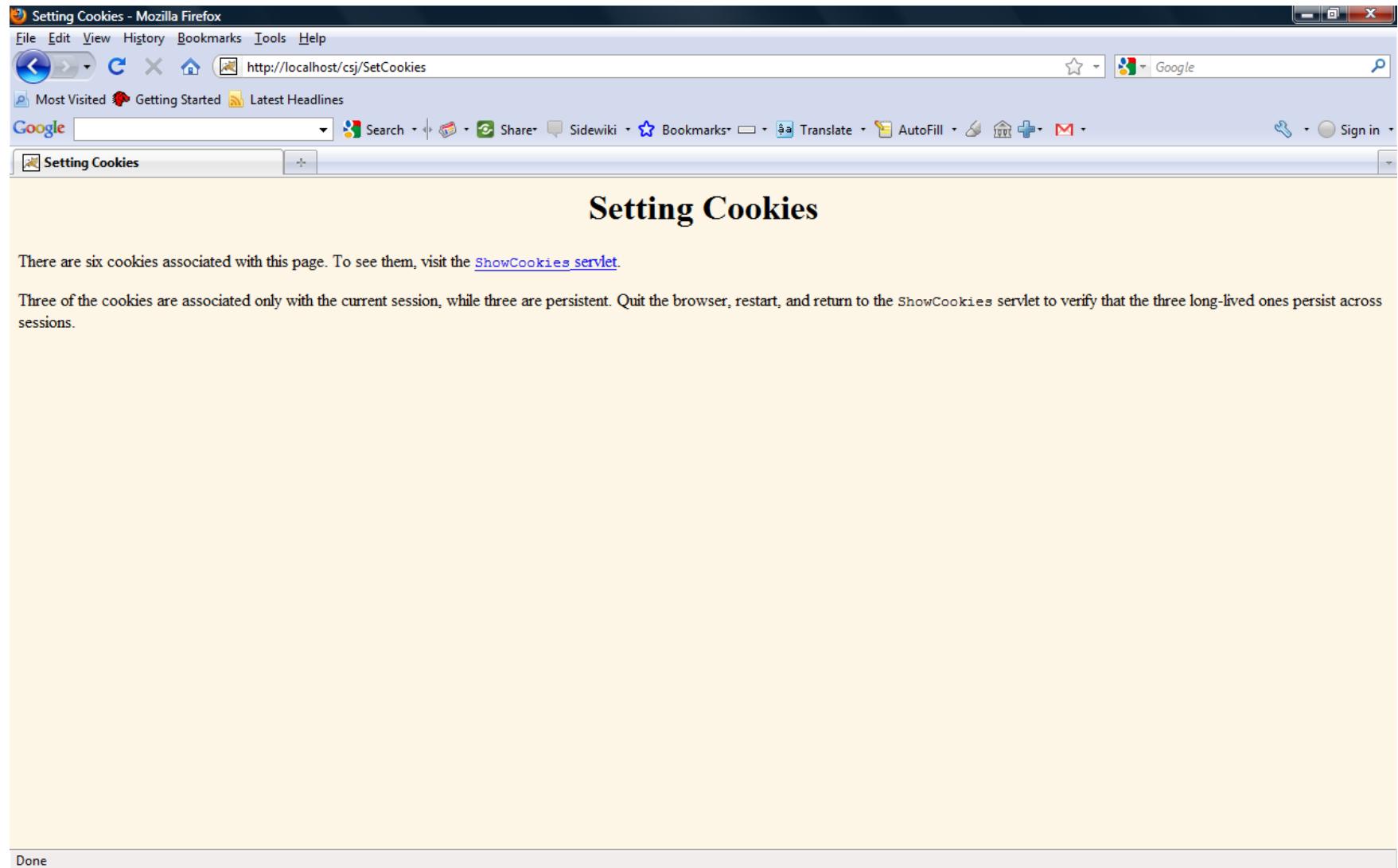
# Cookie API (cont.)

- Cookies are returned by the browser to the server if the Domain and Path match the server
- Getting Cookies from the request
  - `public Cookie[] getCookies()`
- Adding Cookies to the response
  - `public void addCookie(Cookie cookie)`

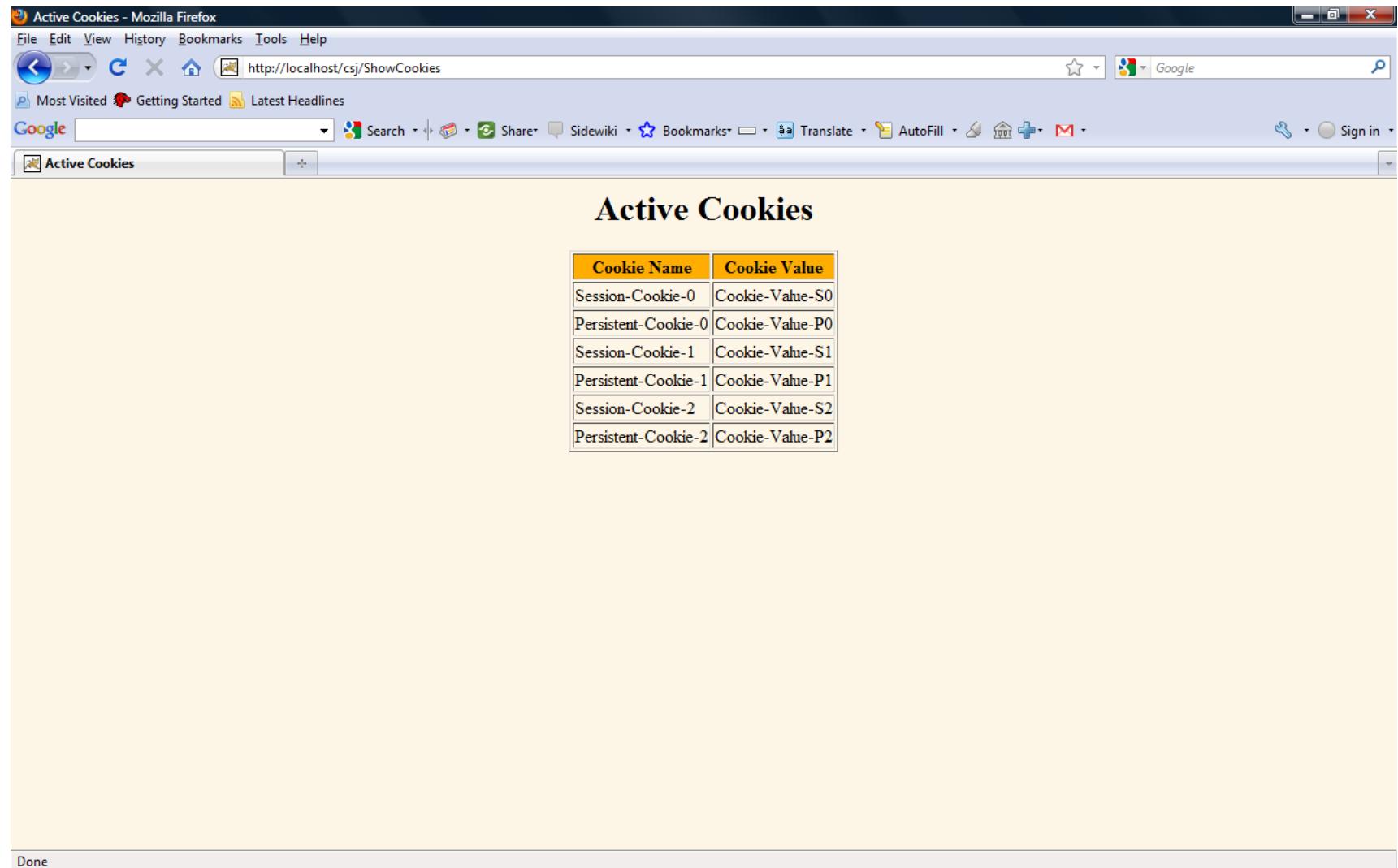
# How to run and test session cookie and persistent cookie?

- From the Browser type:
  - <http://localhost/csj/SetCookies>
    - You will see 3 session cookies and 3 persistent cookie

# How to run and test session cookie and persistent cookie?



# How to run and test session cookie and persistent cookie?



A screenshot of a Mozilla Firefox browser window titled "Active Cookies - Mozilla Firefox". The address bar shows the URL <http://localhost/csj>ShowCookies>. The main content area displays a table titled "Active Cookies" with the following data:

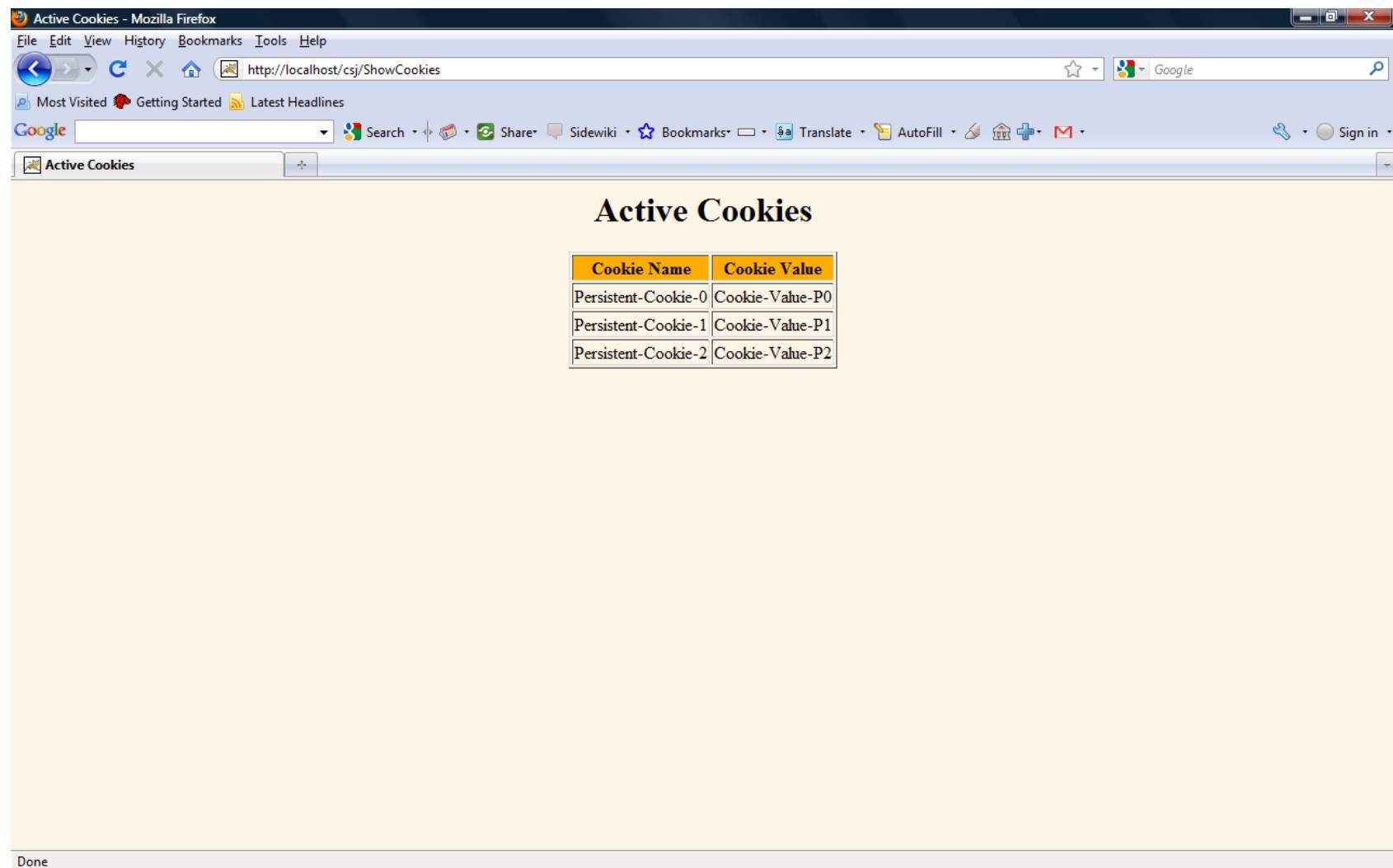
Cookie Name	Cookie Value
Session-Cookie-0	Cookie-Value-S0
Persistent-Cookie-0	Cookie-Value-P0
Session-Cookie-1	Cookie-Value-S1
Persistent-Cookie-1	Cookie-Value-P1
Session-Cookie-2	Cookie-Value-S2
Persistent-Cookie-2	Cookie-Value-P2

The Firefox interface includes standard toolbar icons, a search bar, and a sidebar with links like "Most Visited", "Getting Started", and "Latest Headlines". The status bar at the bottom left shows "Done".

# How to run and test session cookie and persistent cookie?

- Now, kill the browser and start it again by accessing directly the **ShowCookies** without calling **SetCookies**:
  - <http://localhost/csj>ShowCookies>
    - You will see that there are 3 cookies that are still alive.

# How to run and test session cookie and persistent cookie?



A screenshot of a Mozilla Firefox browser window titled "Active Cookies - Mozilla Firefox". The address bar shows the URL "http://localhost/csj>ShowCookies". The main content area displays a table titled "Active Cookies" with three rows of data. The table has two columns: "Cookie Name" and "Cookie Value". The data is as follows:

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

At the bottom left of the browser window, there is a "Done" button.

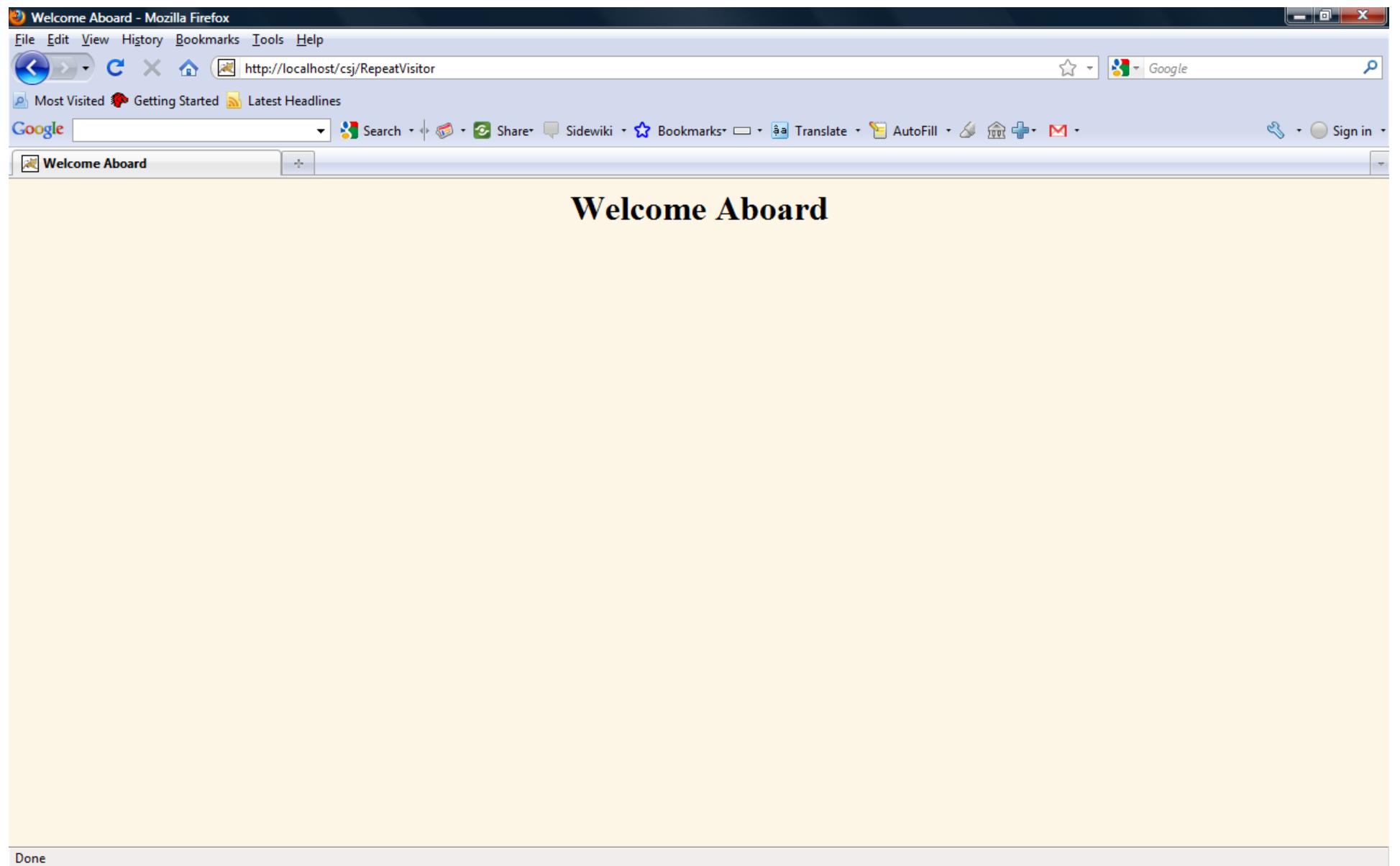
# SetCookies and ShowCookies

- Lets review the source code for
  - SetCookies.java
  - ShowCookies.java

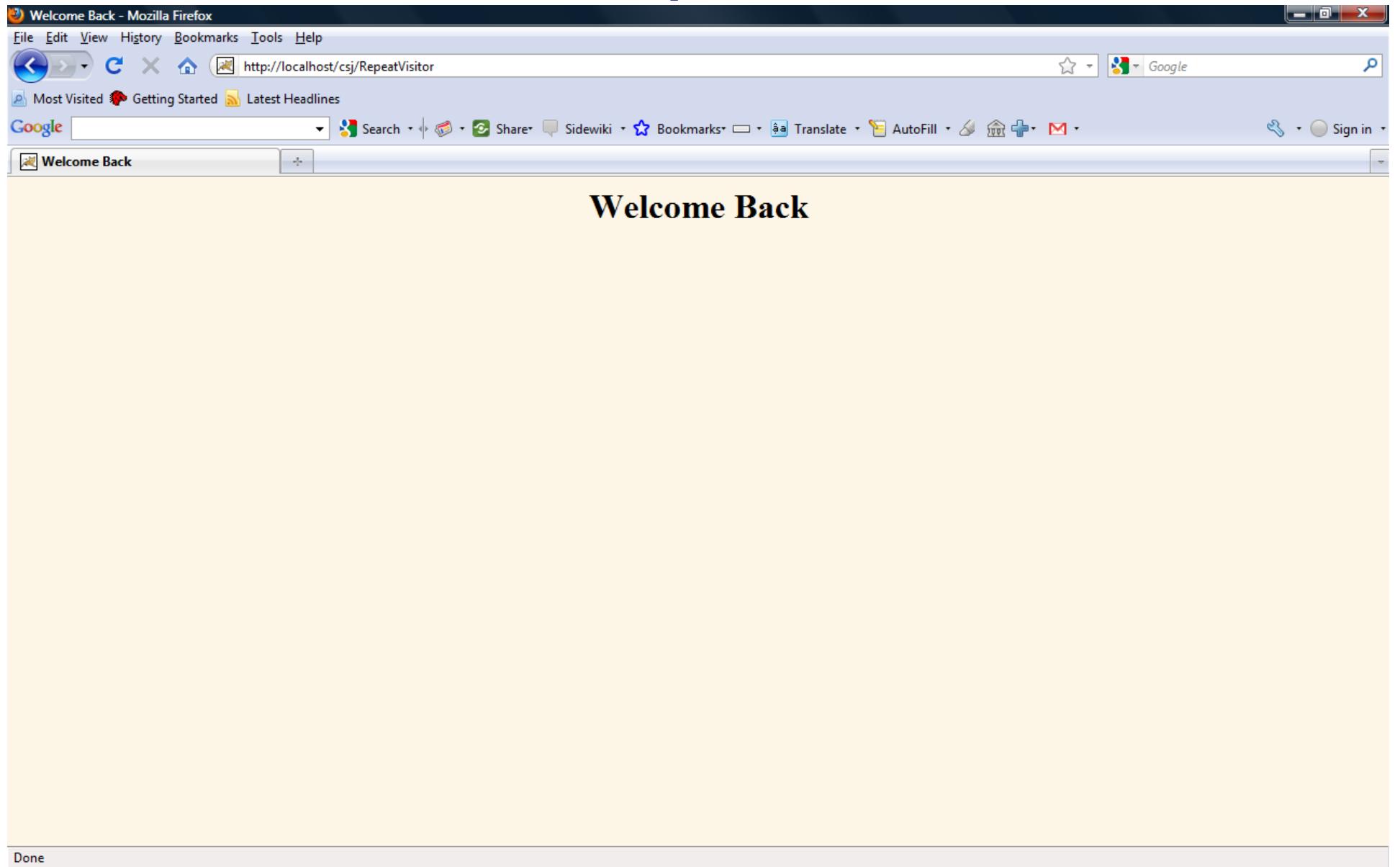
# Cookie for Repeat Visitor

- Lets see an example for a servlet that says "Welcome aboard" to first-time visitors and "Welcome back" to repeat visitors
  - <http://localhost/csj/RepeatVisitor>
  - Refresh the browser after the first visit

# Cookie for Repeat Visitor



# Cookie for Repeat Visitor



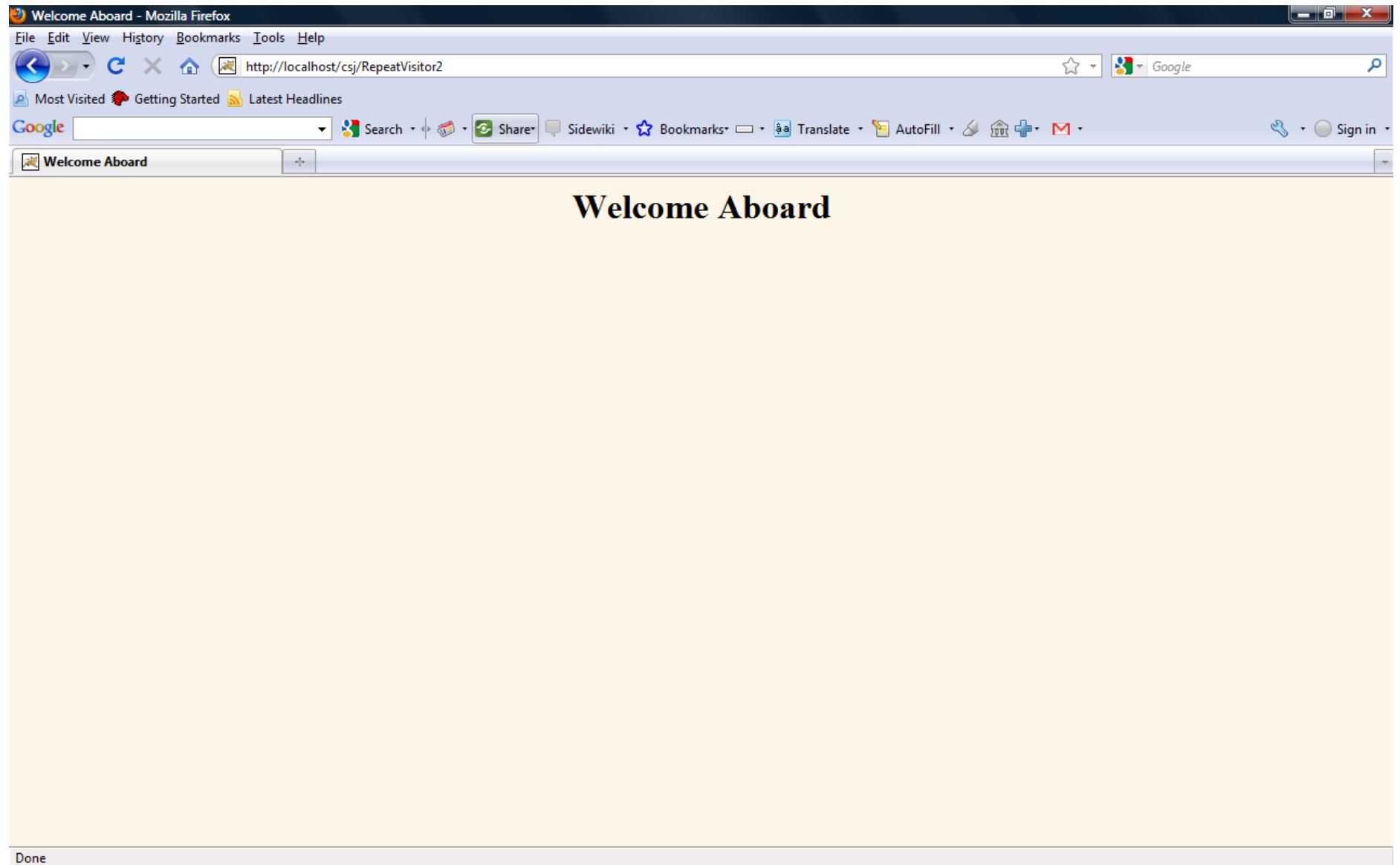
# RepeatVisitor

- Lets review the source code for
  - RepeatVisitor.java

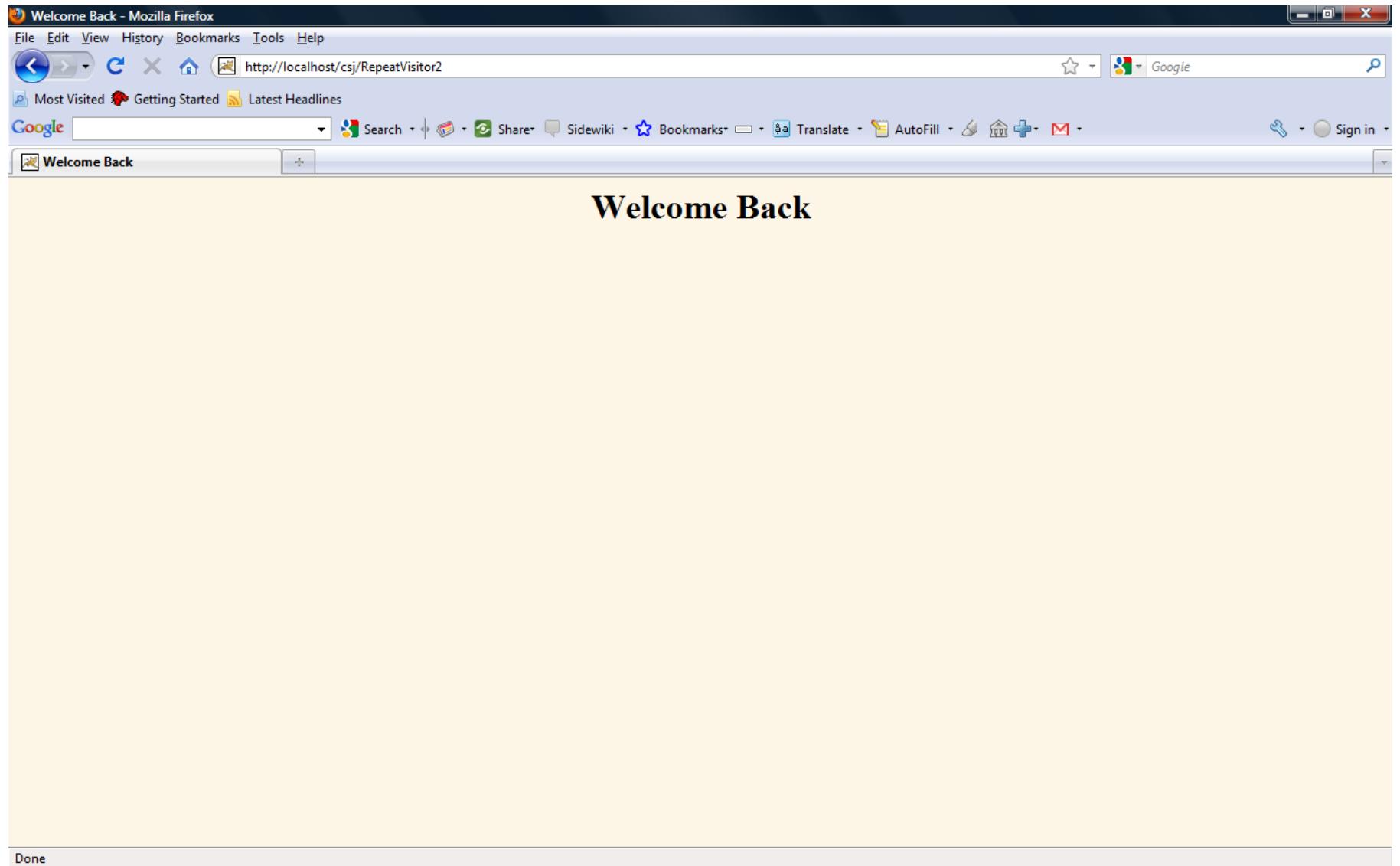
# Repeat Visitor 2

- Better code for the previous RepeatVisitor example as follows:
  - RepeatVisitor2.java servlet uses the CookieUtilities.java and LongLivedCookie.java classes to simplify the code

# Repeat Visitor 2



# Repeat Visitor 2



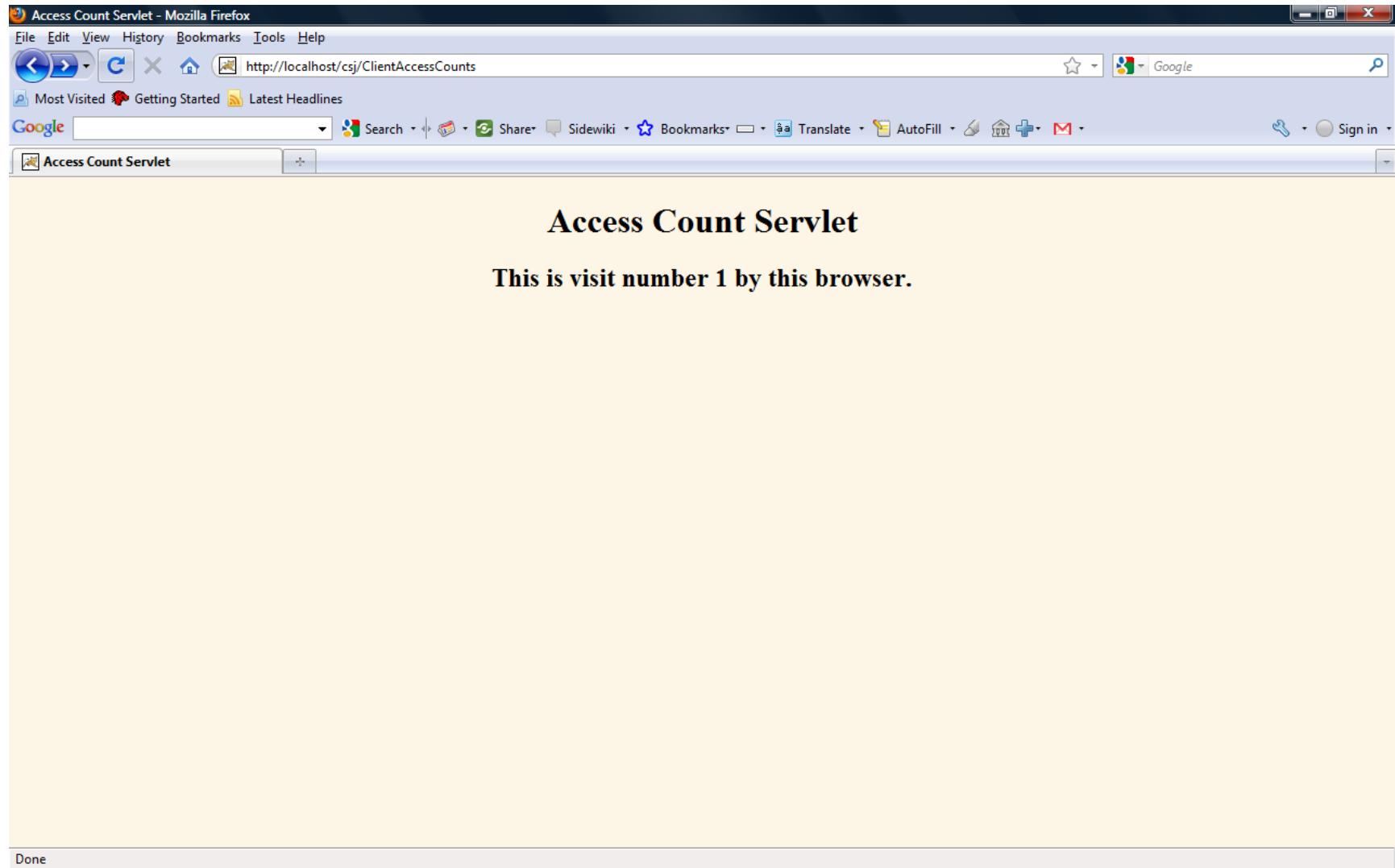
# RepeatVisitor2

- Lets review the source code for
  - RepeatVisitor2.java
  - CookieUtilities.java
  - LongLivedCookie.java

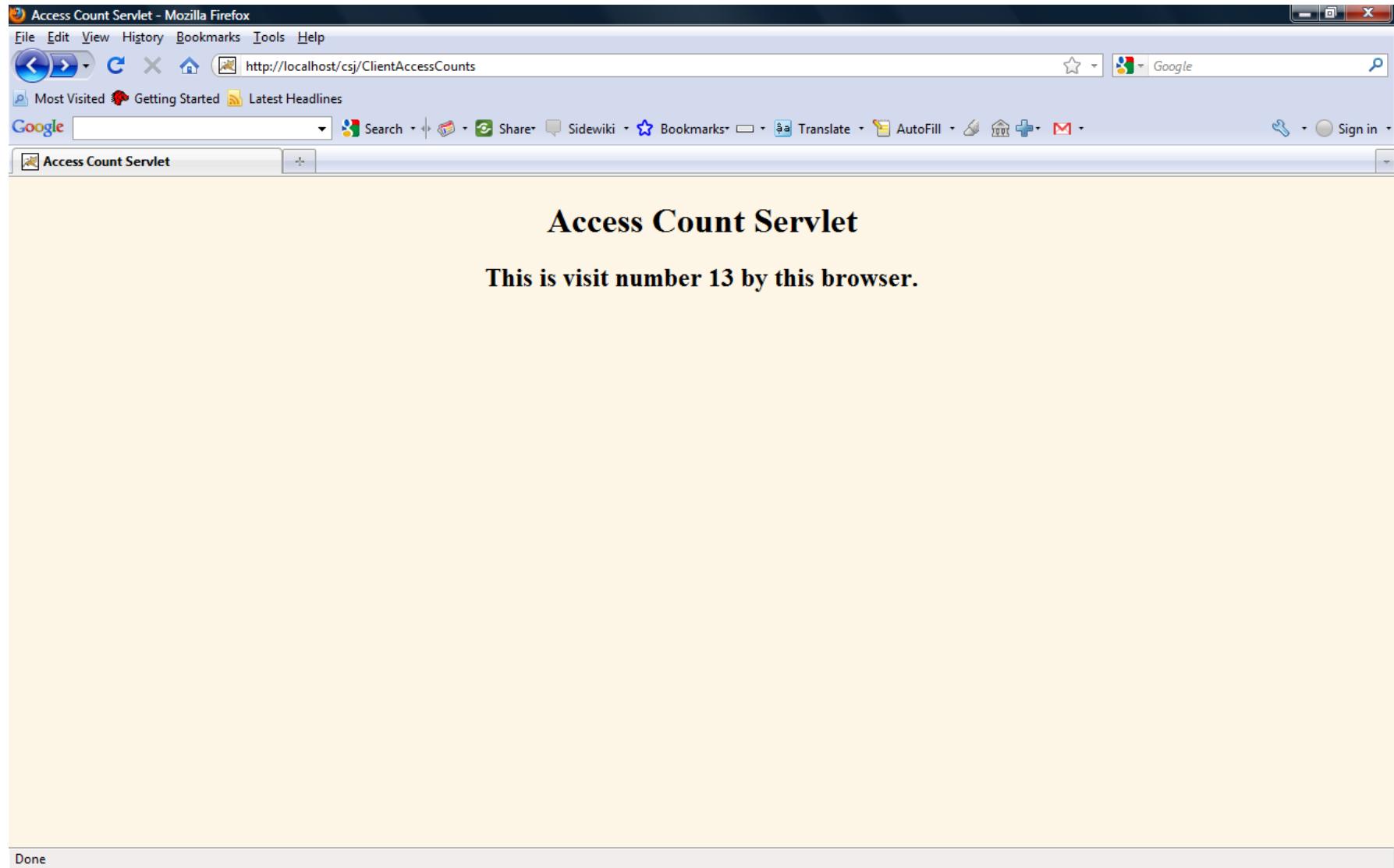
# How to repeatedly change cookie values?

- To replace a previous cookie value, send the same cookie name with a different cookie value.
- If you actually use the incoming Cookie objects, don't forget to do response.addCookie; merely calling setValue is not sufficient. You also need to reapply any relevant cookie attributes by calling setMaxAge, setPath, etc
- The following example illustrates how to repeatedly change cookie values
  - [ClientAccessCounts.java](#). Servlet that prints per-client access counts.

# How to repeatedly change cookie values?



# How to repeatedly change cookie values?



# Session Tracking

- Why session tracking?
  - A **cookie** has a name and a **single value**. It is **stored on the client side**
  - 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?

# Session Tracking

- In web applications, the concept of a session is used to put consecutive requests and responses in context
- Implicit in a *session* are the following concepts:
  - start and end
  - may be terminated by either party
  - state information is present
- The concept of *session tracking* allows stateful information to be **stored on the server** and associated with a specific client

# Session Tracking API

- The Java Servlets API includes support for session tracking
- The interface `HttpSession` encapsulates the session information
- The session is stored on the server
- Tracking is performed using Cookies or URL-rewriting and is transparent to the servlet

# Using the Session

- The scope of an HttpSession is the entire webapp
- The session persists until it expires on the server, or it is expired in code
- Programming issues
  - All session objects have to be Serializable (implement `java.io.Serializable`)
  - The HttpSession can be accessed from more than one thread at a time

# Session API

The screenshot shows a Mozilla Firefox browser window with the title "HttpSession (Java EE 6) - Mozilla Firefox". The address bar contains the URL "docs.oracle.com/javee/6/api/index.html?javax/servlet/http/HttpSession.html". The page content is the JavaEE 6 HttpSession API documentation.

**Overview Package Class Tree Deprecated Index Help**

**javaservlet.http**

**Interface HttpSession**

**public interface HttpSession**

Provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.

The servlet container uses this interface to create a session between an HTTP client and an HTTP server. The session persists for a specified time period, across more than one connection or page request from the user. A session usually corresponds to one user, who may visit a site many times. The server can maintain a session in many ways such as using cookies or rewriting URLs.

This interface allows servlets to

- View and manipulate information about a session, such as the session identifier, creation time, and last accessed time
- Bind objects to sessions, allowing user information to persist across multiple user connections

When an application stores an object in or removes an object from a session, the session checks whether the object implements `HttpSessionBindingListener`. If it does, the servlet notifies the object that it has been bound to or unbound from the session. Notifications are sent after the binding methods complete. For session that are invalidated or expire, notifications are sent after the session has been invalidated or expired.

When container migrates a session between VMs in a distributed container setting, all session attributes implementing the `HttpSessionActivationListener` interface are notified.

A servlet should be able to handle cases in which the client does not choose to join a session, such as when cookies are intentionally turned off. Until the client joins the session, `isNew` returns `true`. If the client chooses not to join the session, `getSession` will return a different session on each request, and `isNew` will always return `true`.

Session information is scoped only to the current web application (`ServletContext`), so information stored in one context will not be directly visible in another.

**Author:**  
Various

**See Also:**  
`HttpSessionBindingListener`, `HttpSessionContext`

Secure Search McAfee

# Session API

The screenshot shows a Mozilla Firefox browser window displaying the Java EE 6 HttpSession API documentation on Oracle's website. The title bar reads "HttpSession (Java EE 6) - Mozilla Firefox". The address bar shows the URL "docs.oracle.com/javaee/6/api/index.html?javax/servlet/http/HttpSession.html". The page content is titled "Method Summary" and lists various methods of the HttpSession interface, each with its return type, name, and a brief description. The left sidebar contains links to "All Classes" and "Packages", including javax.annotation, javax.annotation.security, javax.annotation.sql, javax.decorator, javax.ejb, javax.ejb.embeddable, javax.ejb.spi, javax.el, and javax.enterprise.context.

Method Summary	
java.lang.Object	<a href="#">getAttribute(java.lang.String name)</a> Returns the object bound with the specified name in this session, or <code>null</code> if no object is bound under the name.
java.util.Enumeration<java.lang.String>	<a href="#">getAttributeNames()</a> Returns an Enumeration of String objects containing the names of all the objects bound to this session.
long	<a href="#">getCreationTime()</a> Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
java.lang.String	<a href="#">getId()</a> Returns a string containing the unique identifier assigned to this session.
long	<a href="#">getLastAccessedTime()</a> Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container received the request.
int	<a href="#">getMaxInactiveInterval()</a> Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
ServletContext	<a href="#">getServletContext()</a> Returns the ServletContext to which this session belongs.
HttpSessionContext	<a href="#">getSessionContext()</a> <i>Deprecated. As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.</i>
java.lang.Object	<a href="#">getValue(java.lang.String name)</a> <i>Deprecated. As of Version 2.2, this method is replaced by <a href="#">getAttribute(java.lang.String)</a>.</i>
java.lang.String[]	<a href="#">getValueNames()</a> <i>Deprecated. As of Version 2.2, this method is replaced by <a href="#">getAttributeNames()</a>.</i>
void	<a href="#">invalidate()</a> Invalidates this session then unbinds any objects bound to it.
boolean	<a href="#">isNew()</a> Returns <code>true</code> if the client does not yet know about the session or if the client chooses not to join the session.
void	<a href="#">putValue(java.lang.String name, java.lang.Object value)</a> <i>Deprecated. As of Version 2.2, this method is replaced by <a href="#">setAttribute(java.lang.String, java.lang.Object)</a>.</i>
void	<a href="#">removeAttribute(java.lang.String name)</a> Removes the object bound with the specified name from this session.
void	<a href="#">removeValue(java.lang.String name)</a> <i>Deprecated. As of Version 2.2, this method is replaced by <a href="#">removeAttribute(java.lang.String)</a>.</i>

# Session API

The screenshot shows a Mozilla Firefox browser window displaying the HttpSession API documentation from Oracle's Java EE 6 API index. The title bar reads "HttpSession (Java EE 6) - Mozilla Firefox". The left sidebar contains navigation links for "All Classes" and "Packages", listing various Java annotations and components. The main content area is a table detailing the methods of the HttpSession interface.

		Returns an Enumeration of String objects containing the names of all the objects bound to this session.
	long <a href="#">getCreationTime()</a>	Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.
	java.lang.String <a href="#">getId()</a>	Returns a string containing the unique identifier assigned to this session.
	long <a href="#">getLastAccessedTime()</a>	Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container received the request.
	int <a href="#">getMaxInactiveInterval()</a>	Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
	ServletContext <a href="#">getServletContext()</a>	Returns the ServletContext to which this session belongs.
	HttpSessionContext <a href="#">getSessionContext()</a>	<b>Deprecated.</b> As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.
	java.lang.Object <a href="#">getValue(java.lang.String name)</a>	<b>Deprecated.</b> As of Version 2.2, this method is replaced by <a href="#">getAttribute(java.lang.String)</a> .
	java.lang.String[] <a href="#">getValueNames()</a>	<b>Deprecated.</b> As of Version 2.2, this method is replaced by <a href="#">getAttributeNames()</a> .
	void <a href="#">invalidate()</a>	Invalidates this session then unbinds any objects bound to it.
	boolean <a href="#">isNew()</a>	Returns true if the client does not yet know about the session or if the client chooses not to join the session.
	void <a href="#">putValue(java.lang.String name, java.lang.Object value)</a>	<b>Deprecated.</b> As of Version 2.2, this method is replaced by <a href="#">setAttribute(java.lang.String, java.lang.Object)</a> .
	void <a href="#">removeAttribute(java.lang.String name)</a>	Removes the object bound with the specified name from this session.
	void <a href="#">removeValue(java.lang.String name)</a>	<b>Deprecated.</b> As of Version 2.2, this method is replaced by <a href="#">removeAttribute(java.lang.String)</a> .
	void <a href="#">setAttribute(java.lang.String name, java.lang.Object value)</a>	Binds an object to this session, using the name specified.
	void <a href="#">setMaxInactiveInterval(int interval)</a>	Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.

# Session API

- In the `HttpServletRequest` class:
  - `HttpSession getSession()`
  - `HttpSession getSession( boolean create)`
  - Returns the current `HttpSession` associated with this request, or If there is no current session and `create` is `true`, returns a new session.
  - Default: `create` is `true`
- `String getRequestedSessionId()`
  - Returns the session ID.

# Session API (cont)

- The `HttpSession` class:

```
void setAttribute( String name, Object value)
```

- Binds an object to the specified name in this session.

```
Object getAttribute( String name)
```

- Returns the object bound to the specified name in this session, or
- Returns `null` if no object is bound to the name.

```
void removeAttribute( String name)
```

- Removes the binding with the specified name and the object from this session.
- Can do the same by calling `setAttribute(name, null);`

# Session API (cont)

`String getId()`

- Returns the session ID.

`Enumeration getAttributeNames()`

- Returns an Enumeration of string objects containing the names of all the objects bound to this session.

`long getCreationTime()`

- Returns the time when this session was created in ms since the epoch

`long getLastAccessedTime()`

- Returns the last time the client sent a request associated with this session.

# Session API (cont)

```
int getMaxInactiveInterval()
```

- Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.
- Tomcat default: 30 minutes

```
void setMaxInactiveInterval( int interval )
```

- Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.
- A negative time indicates the session should never expire.

# Session API (cont)

`void invalidate()`

- Invalidates this session and unbinds any objects in this session.

`boolean isNew()`

- Returns `true`
  - if the client does not yet know about the session, or
  - if the client chooses not to join the session (i.e. cookies are disabled).

# The ShowSession.java Run

[A screenshot of a Mozilla Firefox browser window titled "Session Tracking Example - Mozilla Firefox". The address bar shows the URL <http://localhost/csj>ShowSession>. The page content displays a welcome message "Welcome, Newcomer" and a table titled "Information on Your Session:".

\*\*Welcome, Newcomer\*\*

\*\*Information on Your Session:\*\*

Info Type	Value
ID	CFBB771944FD0E98ED7447ED72CE91A7
Creation Time	Tue Oct 12 20:27:06 CDT 2010
Time of Last Access	Tue Oct 12 20:27:06 CDT 2010
Number of Previous Accesses	0

A green callout bubble points to the "Number of Previous Accesses" value of 0, indicating it's the user's first visit. The browser interface includes a menu bar, toolbars, and a status bar at the bottom.

Done

1st Visit](http://localhost/csj>ShowSession</a></u></p></div><div data-bbox=)

# ShowSession.java

<http://localhost/csj>ShowSession>

The screenshot shows a Mozilla Firefox browser window titled "Session Tracking Example - Mozilla Firefox". The address bar contains the URL "http://localhost/csj>ShowSession". The main content area displays a "Welcome Back" message and a table titled "Information on Your Session:". The table has two columns: "Info Type" and "Value". The rows show session details: ID (CFBB771944FD0E98ED7447ED72CE91A7), Creation Time (Tue Oct 12 20:27:06 CDT 2010), Time of Last Access (Tue Oct 12 20:28:19 CDT 2010), and Number of Previous Accesses (12). A large green arrow points from the "Number of Previous Accesses" cell to a green callout bubble containing the text "12th Visit".

Info Type	Value
ID	CFBB771944FD0E98ED7447ED72CE91A7
Creation Time	Tue Oct 12 20:27:06 CDT 2010
Time of Last Access	Tue Oct 12 20:28:19 CDT 2010
Number of Previous Accesses	12

12th Visit

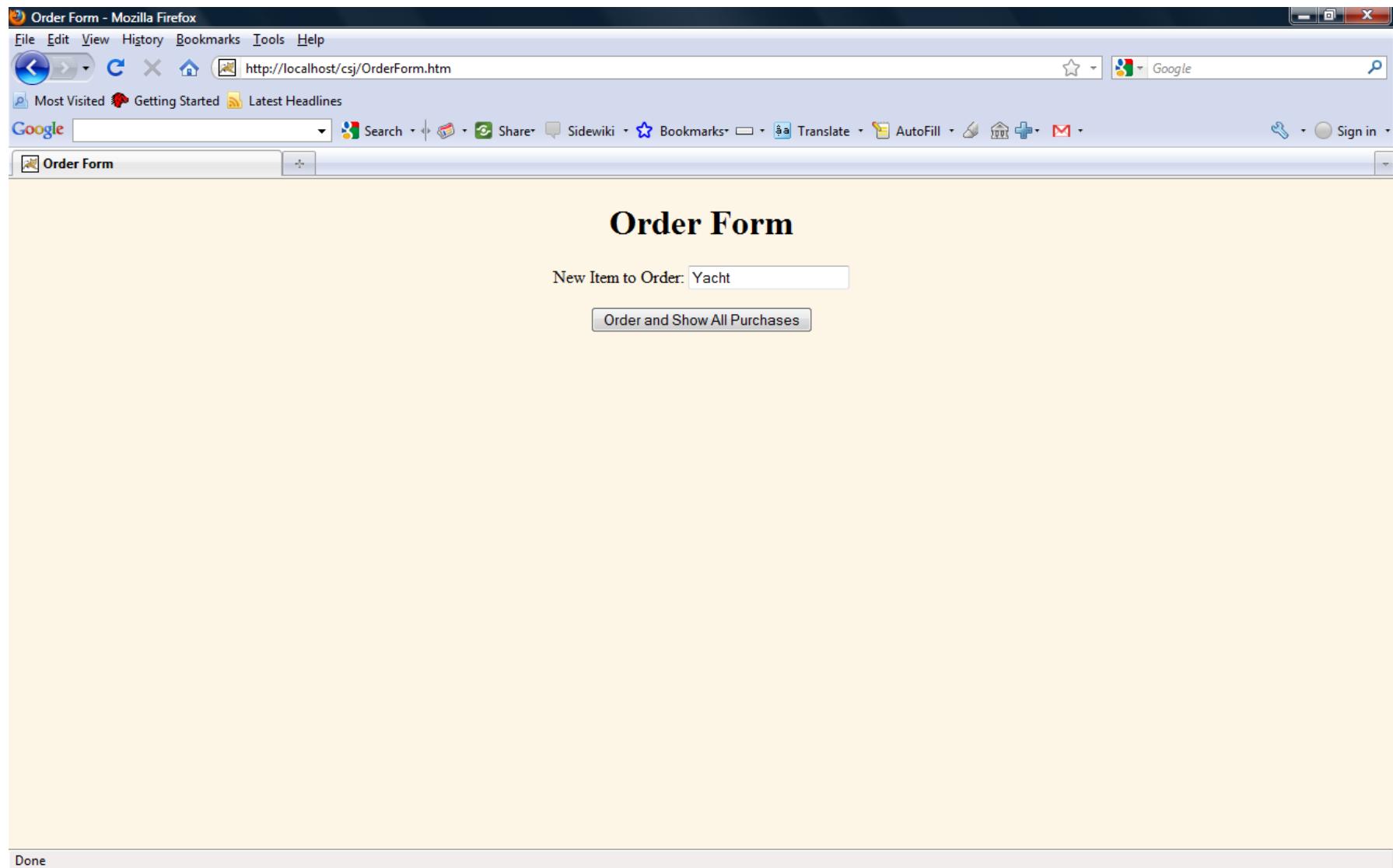
## Accumulating a List of User Data

- Mutable data structures (*mutable* data structure such as an array, List, Map) are often used to maintain a set of data associated with the user.
- In the next example, we present a simplified example in which we maintain a basic list of items that each user has purchased.

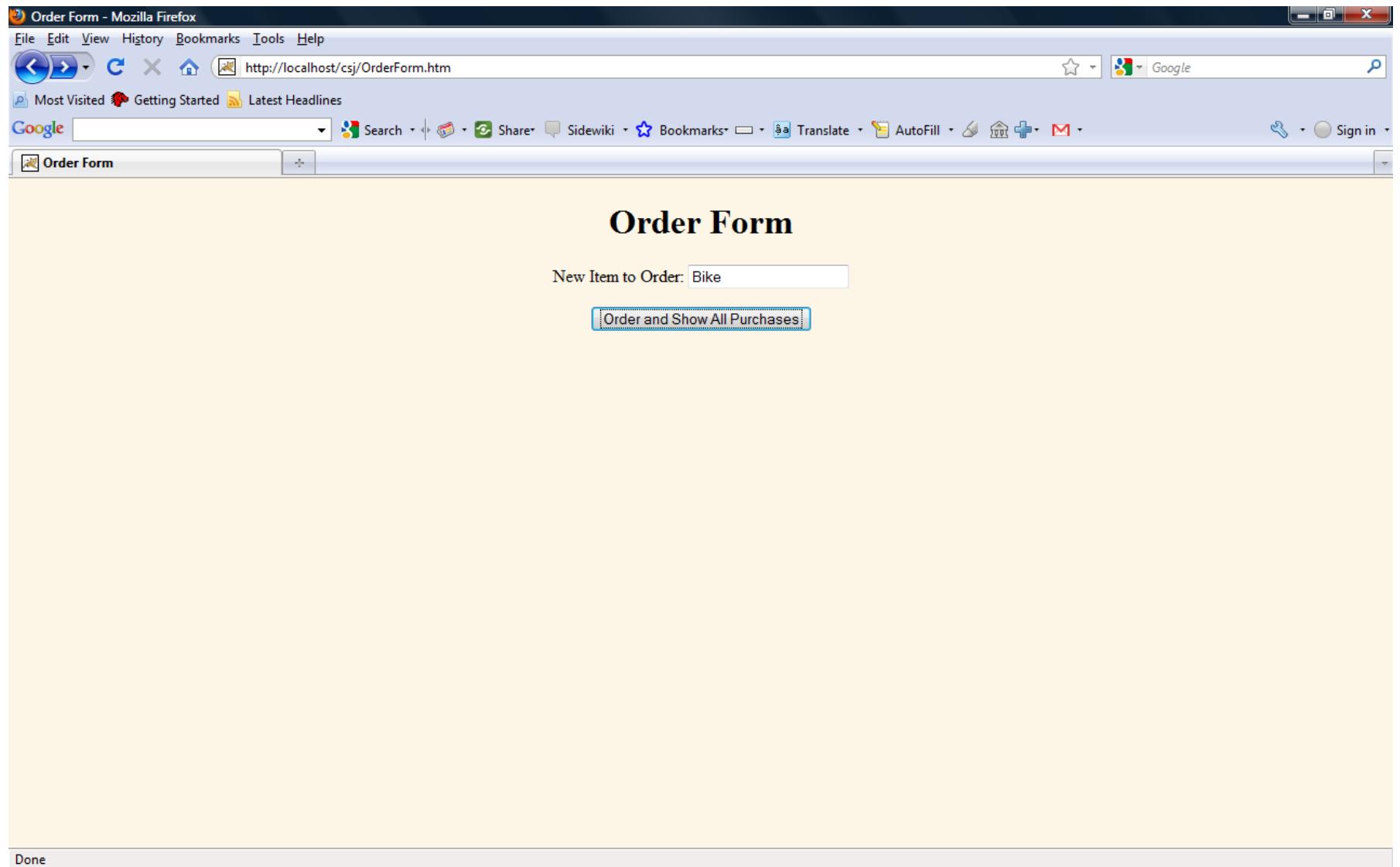
# Accumulating a List of User Data

- ShowItems.java. Servlet that displays a list of items being ordered. Accumulates them in an ArrayList with no attempt at detecting repeated items. Uses OrderForm.html to collect data.
- OrderForm.html. Front end to the ShowItems servlet.

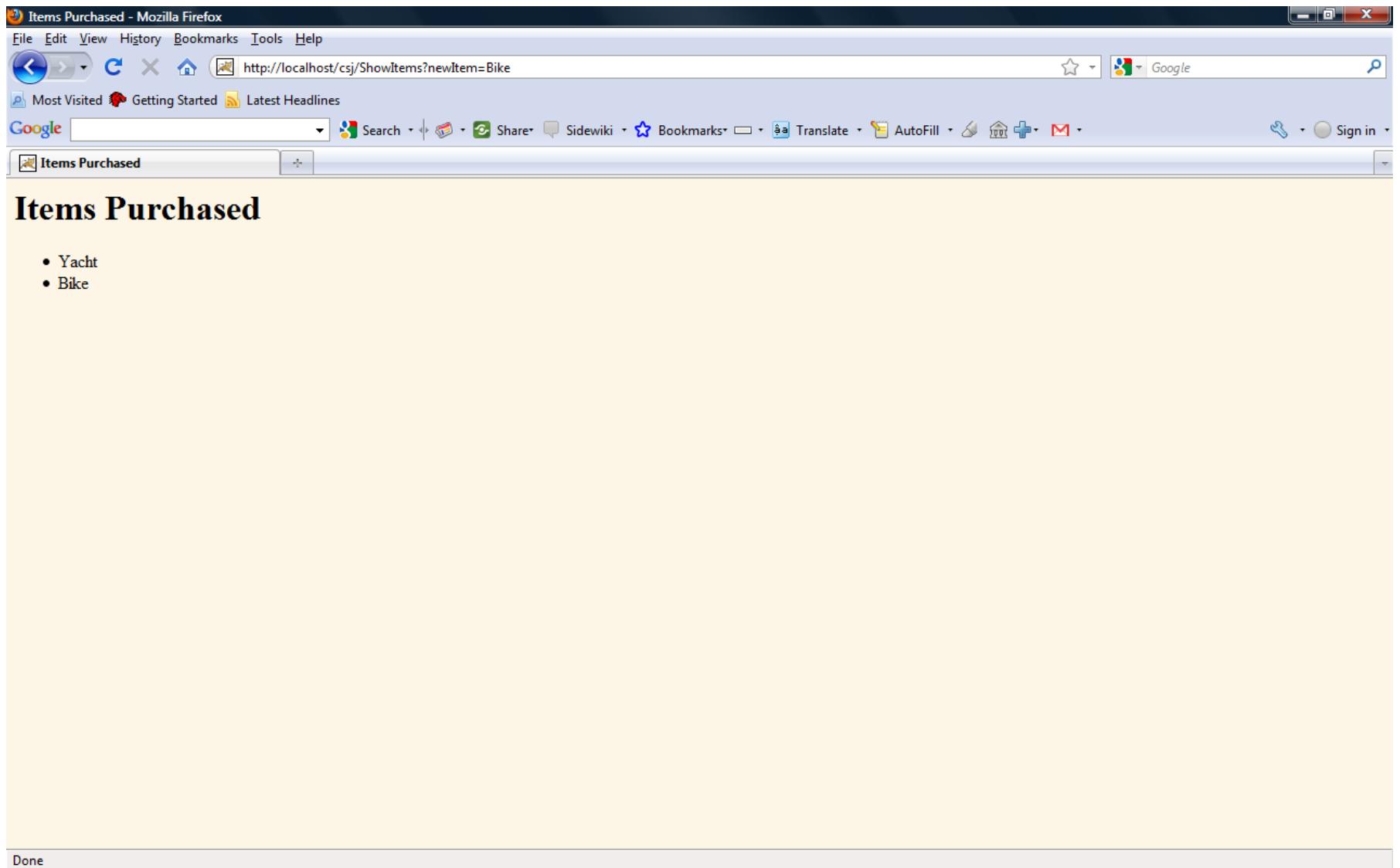
# Accumulating a List of User Data



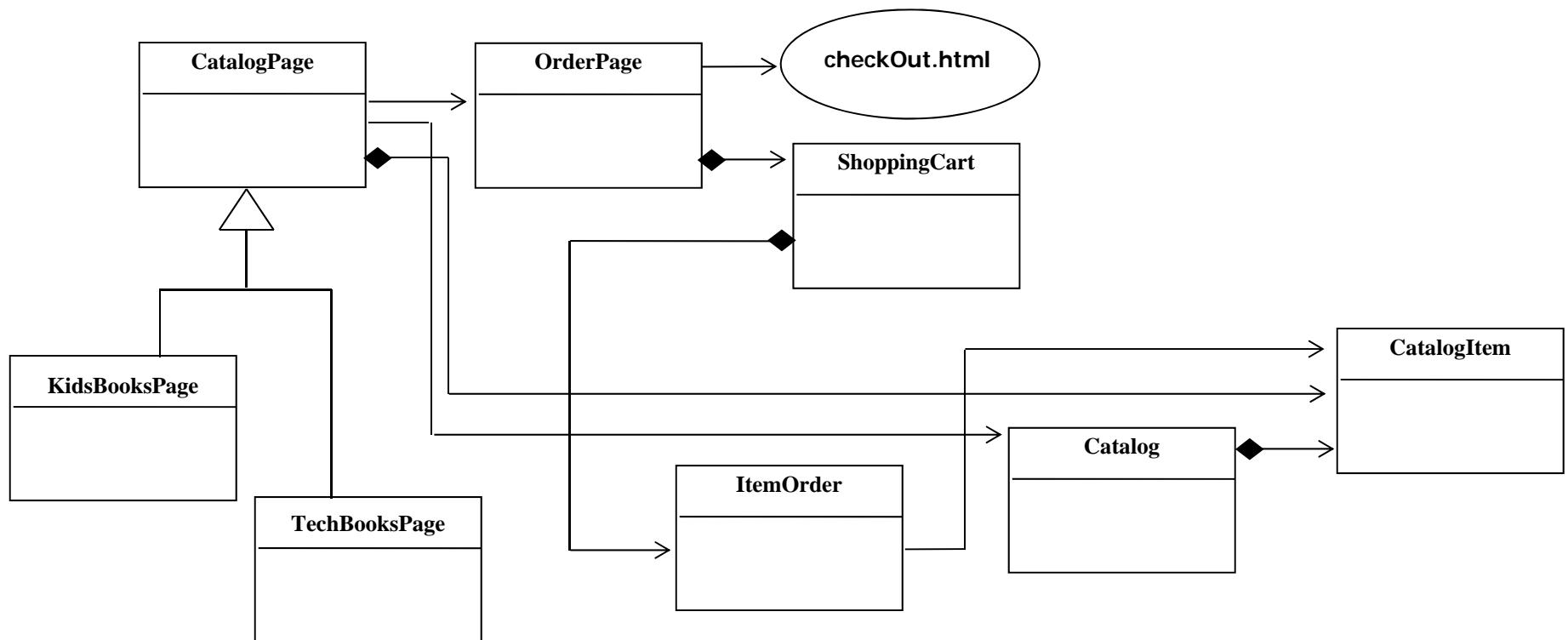
# Accumulating a List of User Data



# Accumulating a List of User Data



# The Shopping Cart Example



# The Shopping Cart Example

- For this example we will use **encodeURL** method from **HttpServletResponse**, why? We want to include the sessionID for session tracking between the servlets

The screenshot shows a Mozilla Firefox browser window displaying the Java EE 5 SDK documentation for the `encodeURL` method of `HttpServletResponse`. The URL in the address bar is `docs.oracle.com/javaee/5/api/index.html?javax/servlet/http/HttpServletResponse.html`.

The left sidebar contains a navigation tree with categories like `javax.resource.spi.security`, `javax.resource.spi.work`, `javax.security.jacc`, `javax.servlet`, `javax.servlet.http`, `javax.servlet.jsp`, `javax.servlet.jsp.el`, `javax.servlet.jsp.tagext`, `javax.servlet.http`, `Interfaces`, `HttpServletRequest`, `HttpServletResponse`, `HttpSession`, `HttpSessionActivationListener`, `HttpSessionAttributeListener`, `HttpSessionBindingListener`, `HttpSessionContext`, and `HttpSessionListener`. Below these are `Classes` sections for `Cookie`, `HttpServlet`, `HttpServletRequestWrapper`, `HttpServletResponseWrapper`, `HttpSessionBindingEvent`, `HttpSessionEvent`, and `HttpUtils`.

The main content area shows the `encodeURL` method documentation. It includes parameters (`name` - the header name), returns (`true` if the named response header has already been set; `false` otherwise), and a detailed description: "Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged. The implementation of this method includes the logic to determine whether the session ID needs to be encoded in the URL. For example, if the browser supports cookies, or session tracking is turned off, URL encoding is unnecessary." A callout box highlights a note: "For robust session tracking, all URLs emitted by a servlet should be run through this method. Otherwise, URL rewriting cannot be used with browsers which do not support cookies." The parameters (`url` - the url to be encoded) and returns (`the encoded URL if encoding is needed; the unchanged URL otherwise.`) are also listed.

# The Shopping Cart Example

- [CatalogPage.java](#). Base class for pages showing catalog entries.  
Servlets that extend this class must specify the catalog entries that they are selling and the page title.
- [KidsBooksPage.java](#). A specialization of the [CatalogPage](#) servlet that displays a page selling three famous kids-book series. Orders are sent to the [OrderPage](#) servlet.

# The Shopping Cart Example

- TechBooksPage.java. A specialization of the CatalogPage servlet that displays a page selling two famous computer books. Orders are sent to the OrderPage servlet.
- OrderPage.java. Servlet that records new item orders (if any) and then displays all items in the shopping cart.
- Checkout.html. Page that handles the final checkout.
- ShoppingCart.java. A shopping cart data structure used to track orders. The OrderPage servlet associates one of these carts with each user session.

# The Shopping Cart Example

- CatalogItem.java. Describes a catalog item for an online store.  
Elements of this class are *indirectly* stored in the ShoppingCart (after they are embedded within an ItemOrder).
- ItemOrder.java. Class that associates a catalog item with a specific order by keeping track of the number ordered and the total price.  
Elements of this class are *directly* stored in the ShoppingCart (after each is populated with a CatalogItem).
- Catalog.java. A catalog that lists the items available in inventory.

# The Shopping Cart

<http://localhost/csj/KidsBooksPage>

A screenshot of a Mozilla Firefox browser window. The title bar reads "All-Time Best Children's Fantasy Books - Mozilla Firefox". The address bar shows the URL "http://localhost/csj/KidsBooksPage". The toolbar includes standard buttons for back, forward, search, and refresh, along with links for "Most Visited", "Getting Started", and "Latest Headlines". The menu bar has options like File, Edit, View, History, Bookmarks, Tools, and Help. The main content area displays three book entries with "Add to Shopping Cart" buttons.

**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*.

---

***The Prydain Series* by Lloyd Alexander (\$19.95)**

Humble pig-keeper Taran joins mighty Lord Gwydion in his battle against Arawn the Lord of Annuvon. Joined by his loyal friends the beautiful princess Eilonwy, wannabe bard Ffleddur 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*.

---

***The Harry Potter Series* by J.K. Rowling (\$59.95)**

The first five 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*, *Harry Potter and the Prisoner of Azkaban*, *Harry Potter and the Goblet of Fire*, and *Harry Potter and the Order of the Phoenix*.

---

Done

# The Shopping Cart

<http://localhost/csj/TechBooksPage>

All-Time Best Computer Books - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/csj/TechBooksPage

Most Visited Getting Started Latest Headlines

Google Search Share Sidewiki Bookmarks Translate AutoFill

All-Time Best Computer Books

## All-Time Best Computer Books

---

***Core Servlets and JavaServer Pages 2nd Edition (Volume 1) by Marty Hall and Larry Brown (\$39.95)***

The definitive reference on servlets and JSP from Prentice Hall and Sun Microsystems Press.

Nominated for the Nobel Prize in Literature.

---

***Core Web Programming, 2nd Edition by Marty Hall and Larry Brown (\$49.99)***

One stop shopping for the Web programmer. Topics include

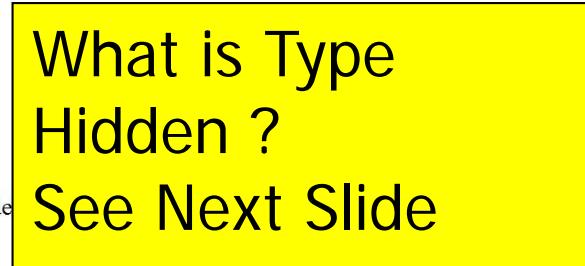
- Thorough coverage of Java 2; including Threads, Networking, Swing, Java 2D, RMI, JDBC, and Collections
- A fast introduction to HTML 4.01, including frames, style sheets, and layers.
- A fast introduction to HTTP 1.1, servlets, and JavaServer Pages.
- A quick overview of JavaScript 1.2

---

Done

# The Shopping Cart Example

- **Handling the Orders:**
  - KidsBooksPage and TechBooksPage are of supertype Servlet CatalogPage



```
CatalogPage - Notepad
File Edit Format View Help
<HEAD><TITLE>" + title + "</TITLE></HEAD>\n" +
"<BODY BGCOLOR="#FDF5E6">\n" +
"<H1 ALIGN="CENTER">" + title + "</H1>");

CatalogItem item;
for(int i=0; i<items.length; i++) {
    out.println("<HR>");
    item = items[i];
    // Show error message if subclass lists item ID
    // that's not in the catalog.
    if(item == null) {
        out.println("<FONT COLOR="RED">" +
                    "Unknown item ID " + itemIDs[i] +
                    "</FONT>");
    } else {
        out.println();
        String formURL =
            "/csj/OrderPage";
        // Pass URLs that reference own site through encode
        formURL = response.encodeURL(formURL);
        out.println
            ("<FORM ACTION=\"" + formURL + "\">\n" +
             "<INPUT TYPE="HIDDEN" NAME="itemID" " +
             " VALUE=\"" + item.getItemId() + "\">\n" +
             "<H2>" + item.getShortDescription() +
             " (" + item.getCost() + ")</H2>\n" +
             item.getLongDescription() + "\n" +
             "<P>\n<CENTER>\n" +
             "<INPUT TYPE="SUBMIT" " +
             "VALUE="Add to Shopping Cart">\n" +
             "</CENTER>\n<P>\n</FORM>");

    }
}
```

Ln 106, Col 3

What is Type  
Hidden ?  
See Next Slide

# The Shopping Cart

## Input Type = Hidden

HTML input type Attribute - Mozilla Firefox

File Edit View History Bookmarks Tools Help

HTML input type Attribute

www.w3schools.com/tags/att\_input\_type\_hidden.asp

Input type: hidden

Example

Define a hidden field (not visible to a user). A hidden field often store a default value, or can have its value changed by a JavaScript:

```
<input type="hidden" name="country" value="Norway">
```

Try it yourself »

A screenshot of a Mozilla Firefox browser window displaying the W3Schools "HTML input type Attribute" page. The "Input type: hidden" section is highlighted with a yellow circle. A specific sentence within the "Example" text is highlighted with a yellow box: "A hidden field often store a default value, or can have its value changed by a JavaScript:". Below the example is a code snippet: <input type="hidden" name="country" value="Norway">. A green button labeled "Try it yourself »" is visible.

## Input type: image

### Example

Define an image as a submit button:

```
<input type="image" src="img_submit.gif" alt="Submit">
```

Try it yourself »

## Input type: month

hidden

Highlight All Match

Secure Search

McAfee

This part of the image shows the bottom of the browser window, specifically the address bar and the status bar. The address bar contains the word "hidden". The status bar includes links for "Highlight All" and "Match", a "Secure Search" button, and the McAfee security logo.

# The Shopping Cart Example

- **Handling the Orders:**

- OrderPage.java is a servlet that handles the orders coming from the various catalog pages. It uses session tracking to associate a shopping cart with each user.
- Since each user has a separate session, it is unlikely that multiple threads will be accessing the same shopping cart simultaneously.
- However, a few circumstances in which concurrent access could occur, such as when a single user has multiple browser windows open and sends updates from more than one in quick succession.

# The Shopping Cart Example

- **Handling the Orders:**

- So, what is the remedy?
  - Your code should synchronize access based upon the session object.
  - This synchronization prevents other threads that use the same session from accessing the data concurrently, while still allowing simultaneous requests from different users to proceed.

# The Shopping Cart Example

- See below how to use **Synchronized** on the **session** object

```
public class OrderPage extends HttpServlet {  
    public void doGet(HttpServletRequest request,  
                      HttpServletResponse response)  
        throws ServletException, IOException {  
        HttpSession session = request.getSession();  
        ShoppingCart cart;  
        synchronized(session) {  
            cart = (ShoppingCart)session.getAttribute("shoppingCart");  
            // New visitors get a fresh shopping cart.  
            // Previous visitors keep using their existing cart.  
            if (cart == null) {  
                cart = new ShoppingCart();  
                session.setAttribute("shoppingCart", cart);  
            }  
        }
```

# The Shopping Cart

<http://localhost/csj/OrderPage?itemID=hall001&numItems=2>

The screenshot shows a Mozilla Firefox browser window with the title "Status of Your Order". The address bar displays the URL: "http://localhost/csj/OrderPage?itemID=hall001&numItems=2". The main content area is titled "Status of Your Order" and contains a table of four items. Each item row includes an "Update Order" button and a numeric input field for quantity.

Item ID	Description	Unit Cost	Number	Total Cost
lewis001	<i>The Chronicles of Narnia</i> by C.S. Lewis	\$19.95	1	<input type="button" value="Update Order"/> \$19.95
rowling001	<i>The Harry Potter Series</i> by J.K. Rowling	\$59.95	1	<input type="button" value="Update Order"/> \$59.95
hall001	<i>Core Servlets and JavaServer Pages 2nd Edition</i> (Volume 1) by Marty Hall and Larry Brown	\$39.95	2	<input type="button" value="Update Order"/> \$79.90
hall002	<i>Core Web Programming, 2nd Edition</i> by Marty Hall and Larry Brown	\$49.99	1	<input type="button" value="Update Order"/> \$49.99

Done

# The Shopping Cart

<http://localhost/csj/Checkout.htm>

A screenshot of a Mozilla Firefox browser window titled "Checking Out - Mozilla Firefox". The address bar shows the URL <http://localhost/csj/Checkout.htm>. The page content is centered under the heading "Checking Out". It contains the following text:

We are sorry, but our electronic credit-card-processing system is currently out of order. Please send a check to:

Marty Hall  
coreservlets.com, Inc.  
6 MeadowSweet Ct., Suite B1  
Reisterstown, MD 21136-6020  
410-429-5535  
hall@coreservlets.com

Since we have not yet calculated shipping charges, please sign the check but do not fill in the amount. We will generously do that for you.

At the bottom left of the browser window, there is a "Done" button.

# Webpage Layout

The screenshot shows a Java Documentation page titled "A Visual Guide to Layout Managers". The page is part of "The Java™ Tutorials". The left sidebar contains a navigation tree for layout managers, including "Laying Out Components Within a Container", "A Visual Guide to Layout Managers", and various sub-sections like "Using Layout Managers", "How Layout Management Works", and "How to Use Various Layout Managers". The main content area features a heading "A Visual Guide to Layout Managers" and a list of layout manager classes: BorderLayout, BoxLayout, CardLayout, FlowLayout, GridBagLayout, GridLayout, GroupLayout, and SpringLayout. Below this, a note discusses the use of these managers and links to examples. A note at the bottom explains the choice between GroupLayout and GridBagLayout. A footer at the bottom of the page includes a link to "Working With Layouts in JavaFX".

File Edit View History Bookmarks Tools Help

A Visual Guide to Layout M... +

docs.oracle.com/javase/tutorial/uiswing/layout/visual.html

Secure Search

ORACLE® Java™ Documentation

The Java™ Tutorials

Laying Out Components  
Within a Container  
A Visual Guide to Layout Managers  
Using Layout Managers  
How Layout Management Works  
How to Use Various Layout Managers  
How to Use BorderLayout  
How to Use BoxLayout  
How to Use CardLayout  
How to Use FlowLayout  
How to Use GridBagLayout  
How to Use GridLayout  
How to Use GroupLayout  
How to Use GroupLayout Example  
How to Use SpringLayout  
Creating a Custom Layout Manager  
Doing Without a Layout Manager (Absolute Positioning)  
Solving Common Layout Problems

« Previous • Trail • Next »

Home Page > Creating a GUI With JFC/Swing > Laying Out Components Within a Container

**A Visual Guide to Layout Managers**

Several AWT and Swing classes provide layout managers for general use:

- BorderLayout
- BoxLayout
- CardLayout
- FlowLayout
- GridBagLayout
- GridLayout
- GroupLayout
- SpringLayout

This section shows example GUIs that use these layout managers, and tells you where to find the how-to page for each layout manager. You can find links for running the examples in the how-to pages and in the [example index](#).

---

**Note:** This lesson covers writing layout code by hand, which can be challenging. If you are not interested in learning all the details of layout management, you might prefer to use the GroupLayout layout manager combined with a builder tool to lay out your GUI. One such builder tool is the [NetBeans IDE](#). Otherwise, if you want to code by hand and do not want to use GroupLayout, then GridBagLayout is recommended as the next most flexible and powerful layout manager.

---

If you are interested in using JavaFX to create your GUI, see [Working With Layouts in JavaFX](#).

**BorderLayout**

# Webpage Layout

Screenshot of a web browser displaying a Java Swing application example. The browser's menu bar includes File, Edit, View, History, Bookmarks, Tools, and Help. The title bar shows "A Visual Guide to Layout M...". The address bar shows "docs.oracle.com/javase/tutorial/uiswing/layout/visual.html#border". The page content discusses BorderLayout and shows a screenshot of a window titled "BorderLayoutDemo" with four buttons arranged in a grid: "Button 1 (PAGE\_START)" at the top, "Button 2 (CENTER)" in the middle, "5 (LINE\_END)" on the right, and "Long-Named Button 4 (PAGE\_END)" at the bottom.

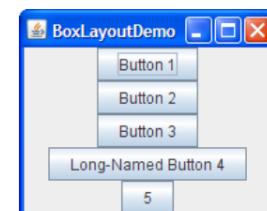
If you are interested in using JavaFX to create your GUI, see [Working With Layouts in JavaFX](#).

**BorderLayout**



Every content pane is initialized to use a `BorderLayout`. (As [Using Top-Level Containers](#) explains, the content pane is the main container in all frames, applets, and dialogs.) A `BorderLayout` places components in up to five areas: top, bottom, left, right, and center. All extra space is placed in the center area. Tool bars that are created using `JToolBar` must be created within a `BorderLayout` container, if you want to be able to drag and drop the bars away from their starting positions. For further details, see [How to Use BorderLayout](#).

## BoxLayout



The `BoxLayout` class puts components in a single row or column. It respects the components' requested maximum sizes and also lets you align components. For further details, see [How to Use BoxLayout](#).

## CardLayout



# Webpage Layout

File Edit View History Bookmarks Tools Help

A Visual Guide to Layout Managers

docs.oracle.com/javase/tutorial/uiswing/layout/visual.html#border

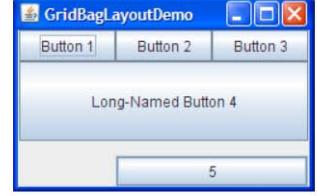
Secure Search

FlowLayout



FlowLayout is the default layout manager for every JPanel. It simply lays out components in a single row, starting a new row if its container is not sufficiently wide. Both panels in CardLayoutDemo, shown previously, use FlowLayout. For further details, see [How to Use FlowLayout](#).

GridBagLayout



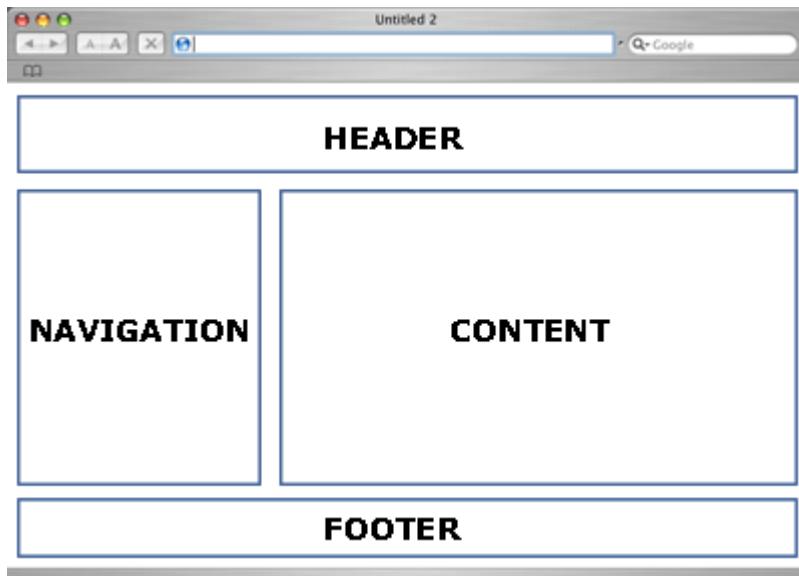
GridBagLayout is a sophisticated, flexible layout manager. It aligns components by placing them within a grid of cells, allowing components to span more than one cell. The rows in the grid can have different heights, and grid columns can have different widths. For further details, see [How to Use GridBagLayout](#).

GridLayout

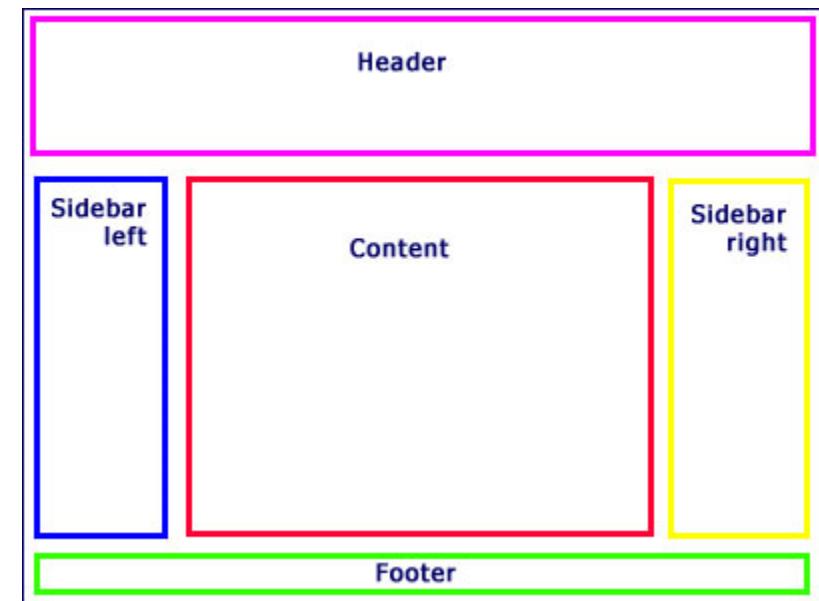


GridLayout simply makes a bunch of components equal in size and displays them in the requested number of rows and columns. For further details, see [How to Use GridLayout](#).

# Webpage Layout



1



2

# Webpage Layout as a Table

The screenshot shows a web-based code editor interface titled "Tryit Editor v2.1". The title bar includes standard menu options: File, Edit, View, History, Bookmarks, Tools, and Help. Below the title bar is a toolbar with icons for file operations like Open, Save, and Print, along with a "Secure Search" button.

The main workspace displays an HTML code snippet. The code uses a table structure to layout the page. It includes a header section, a navigation section, a content section, and a footer section. The code is as follows:

```
<!DOCTYPE html>
<html>
<body>

<table>
  <tr>
    <td colspan="2">
      <div class="header">
        Header
      </div>
    </td>
  </tr>
  <tr>
    <td>
      <div class="navigation">
        Navigation
      </div>
    </td>
    <td>
      <div class="content">
        Content
      </div>
    </td>
  </tr>
  <tr>
    <td colspan="2" style="vertical-align: bottom">
      <div class="footer">
        Footer
      </div>
    </td>
  </tr>
</table>
</body>
</html>
```

Below the code editor is a message: "Edit the code and click "Submit" to see the result." To its right is a "Submit" button. At the top of the editor window, there is an advertisement for Google Adwords: "Advertise locally or globally. Get a \$75 credit when you spend your first \$25" with a "Show me how" button and a Google logo.

The right side of the interface shows the "Result" of the submitted code. The output is a simple layout with three columns: "Header", "Navigation Content", and "Footer".

# Webpage Layout

File Edit View History Bookmarks Tools Help

Best Buy: The Ultimate Tec... www.bestbuy.com Secure Search

Search by Keyword, SKU # or Item # WEEKLY DEALS Stores Orders Help Credit Cards Español

PRODUCTS SERVICES SHOPS & DEALS GIFTS my BEST BUY Sign In or Create an Account

Appliances TV & Home Theater Computers & Tablets Cell Phones Cameras & Camcorders Audio Car Electronics & GPS Video Games, Movies & Music Health, Fitness & Beauty Home & Office Wearable Technology Clearance & More

iPhone 6 Bigger than bigger

Shop Now

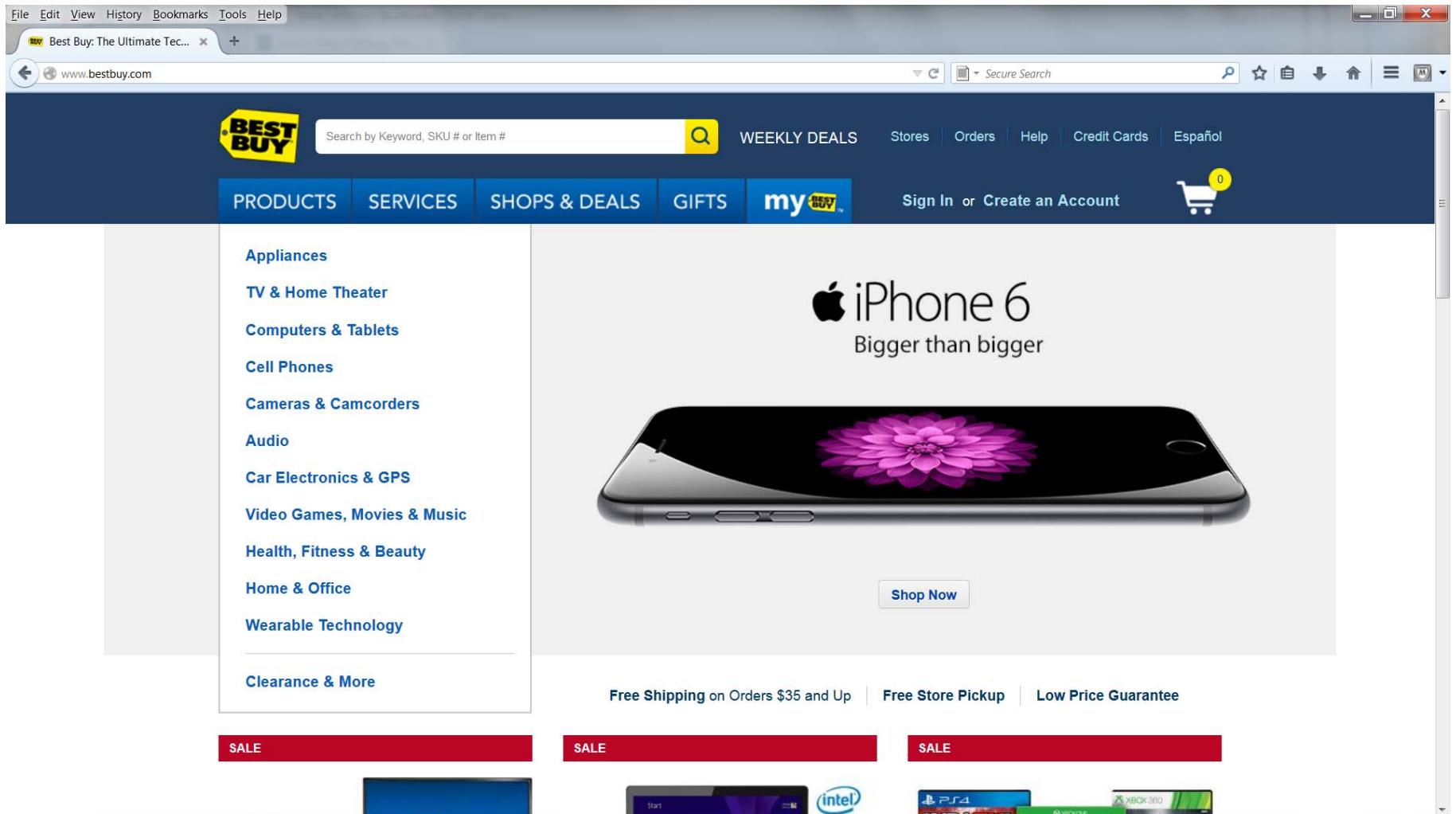
Free Shipping on Orders \$35 and Up | Free Store Pickup | Low Price Guarantee

SALE

SALE

SALE

Start intel PS4 XBOX ONE



# Webpage Layout

File Edit View History Bookmarks Tools Help

Walmart.com: Save money.... +

www.walmart.com

Free shipping on orders of \$50 or more

Trending Now My Store Plainfield All Search Go My Account Hello, Sign In

Value of the Day

- Electronics & Office
- Movies, Music & Books
- Home, Furniture & Patio
- Clothing, Shoes & Jewelry
- Baby & Kids
- Toys & Video Games
- Sports, Fitness & Outdoors
- Auto & Home Improvement
- Photo & Gifts
- Crafts & Party Supplies
- Pharmacy, Health & Beauty
- Grocery, Household & Pets
- Halloween
- See All Departments

**Special Buys** on special tech

Shop Now

Sony 32" LED LCD HDTV just \$199

SONY

RCA 9" Dual Core Tablet ROLLBACK \$69 was \$89.99

T-Mobile Samsung Galaxy S3 just \$299 with Bonus \$50 Walmart Gift Card

While SUPPLIES Last

Walmart Gift Card \$50

Johnson & Johnson

Cleans & Shines the first time, every time

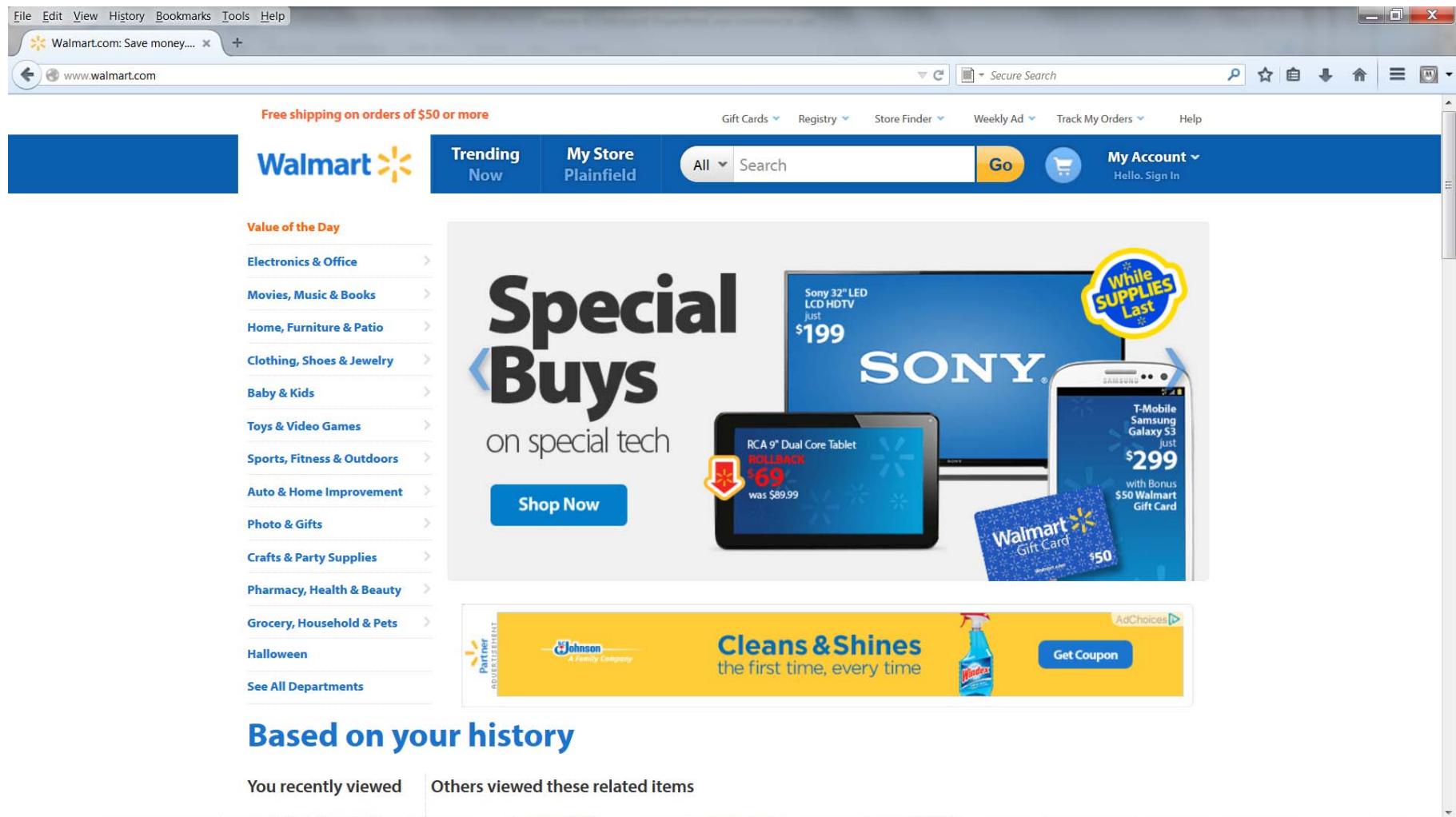
AdChoices

Get Coupon

Based on your history

You recently viewed

Others viewed these related items



# Webpage Layout

The screenshot shows the homepage of the Home Depot website. The layout includes a top navigation bar with links for File, Edit, View, History, Bookmarks, Tools, and Help. The main title "Home Improvement Made..." is displayed in the browser tab. A search bar with the placeholder "What can we help you find?" and a magnifying glass icon is positioned above a sidebar. The sidebar features a "Shop By Department" dropdown menu with categories like Tools, Fasteners, Hardware, Building Materials, Electrical, Janitorial, Appliances (which is currently selected), Bath, Decor, Doors & Windows, Flooring, HVAC, Kitchen, Lighting & Ceiling Fans, Lumber & Composites, Outdoors, Paint, Plumbing, and Storage & Organization. Below the sidebar is a large image of a yellow rake. The main content area contains several product categories: REFRIGERATORS, WASHERS & DRYERS, DISHWASHERS, COOKING APPLIANCES, VACUUM CLEANERS & FLOOR CARE, SMALL APPLIANCES, HEATING, COOLING & AIR QUALITY, RELATED CATEGORIES, and a "Project: How-To" section. Logos for LG, HOTPOINT, Amana, and GE are displayed. A promotional banner for "Lawn & Refuse Bags" is visible on the right side.

File Edit View History Bookmarks Tools Help

Home Improvement Made... x +

www.homedepot.com

Search All ▾ What can we help you find?

Project: How-To ▾ Sign In or Register Your Account ▾

**Shop By Department ▾**

- Tools
- Fasteners
- Hardware
- Building Materials
- Electrical
- Janitorial
- Appliances**
- Bath
- Decor
- Doors & Windows
- Flooring
- HVAC
- Kitchen
- Lighting & Ceiling Fans
- Lumber & Composites
- Outdoors
- Paint
- Plumbing
- Storage & Organization

**REFRIGERATORS** >

- French Door
- Side by Side
- Top Freezer
- Bottom Freezer
- Mini Refrigerators
- Freezers & Ice Makers
- Beverage, Keg & Wine Coolers

**WASHERS & DRYERS** >

- Top Load Washers
- Front Load Washers
- Dryers

**DISHWASHERS** >

**COOKING APPLIANCES** >

- Electric Ranges
- Gas Ranges
- Microwaves
- Cooktops
- Wall Ovens
- Range Hoods

**VACUUM CLEANERS & FLOOR CARE** >

- Coffee, Tea & Espresso Makers
- Popcorn & Entertaining Electrics
- Toasters, Toaster Ovens & Countertop Ovens
- Mixers

**SMALL APPLIANCES** >

- Air Conditioners
- Portable Fans
- Air Purifiers
- Dehumidifiers
- Heaters

**HEATING, COOLING & AIR QUALITY** >

**RELATED CATEGORIES**

- Appliance Special Buys
- Garbage Disposals
- Water Heaters

**Lawn & Refuse Bags**  
Bolsas para césped y desechos

More saving.  
More doing.  
IMPROVE YOUR HOME AND THE ENVIRONMENT

[www.homedepot.com/b/Appliances/N-5yc1vZbv1w?cm\\_sp=d-flyout-Appliances](http://www.homedepot.com/b/Appliances/N-5yc1vZbv1w?cm_sp=d-flyout-Appliances)

# Webpage Layout

File Edit View History Bookmarks Tools Help

Target : Expect More. Pay Less. www.target.com

Secure Search

flip the script. new pharmacy. new perks. \$4 generics? got 'em.  
get started by transferring your prescription >

sign in / account find a store weekly ad gift cards registries TargetLists REDcard > TARGET SUBSCRIPTIONS Learn more

shop all categories all search your cart

clothing, shoes & jewelry >  
baby & kids >  
home, furniture & patio >  
electronics >  
toys & video games >  
movies, music & books >  
sports, fitness & outdoors >  
beauty, health & pharmacy >  
grocery, household & pets >  
gifts, parties & photo >  
clearance & deals >  
Halloween >  
all categories >

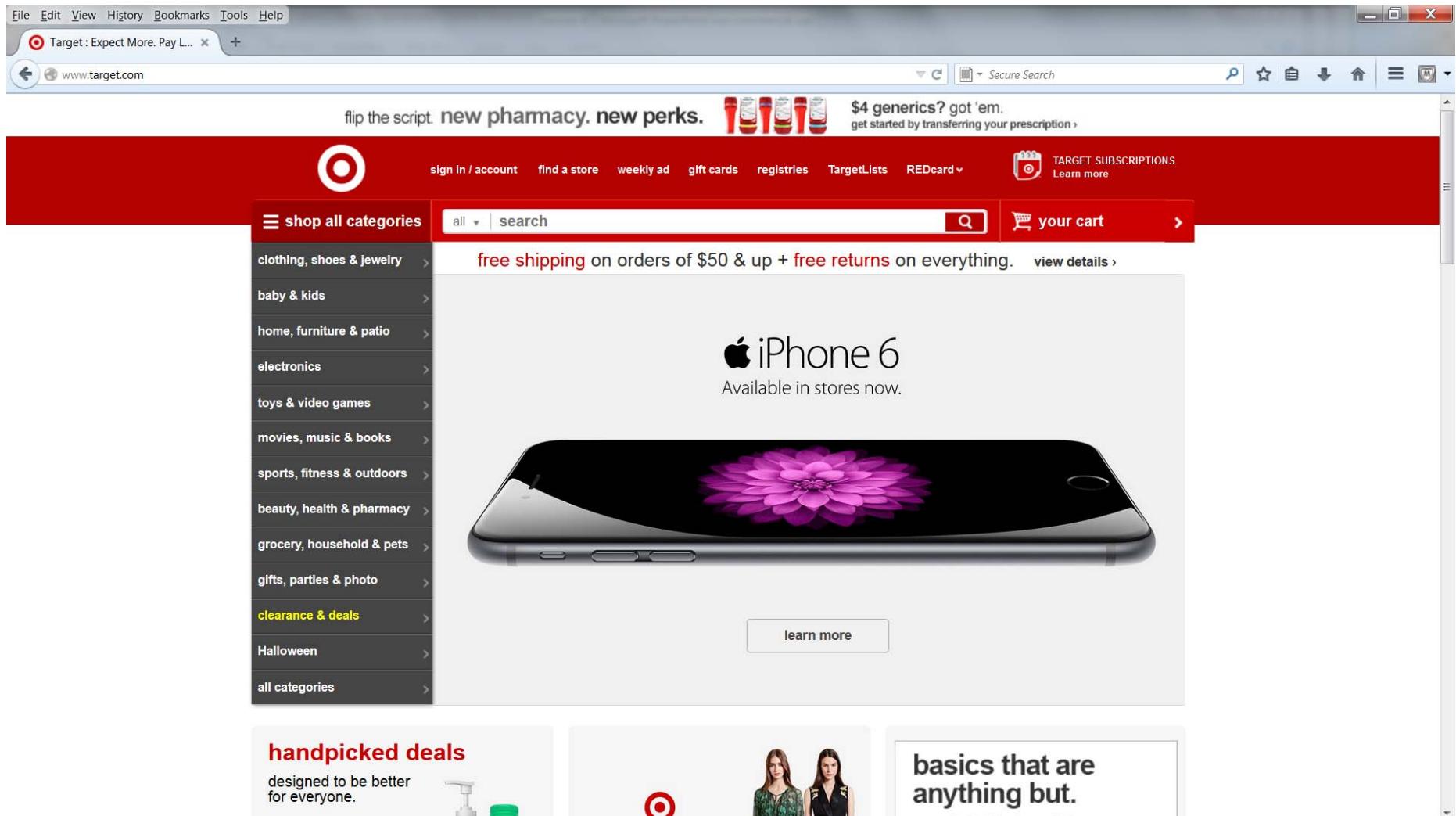
free shipping on orders of \$50 & up + free returns on everything. view details >

iPhone 6  
Available in stores now.

learn more

handpicked deals designed to be better for everyone.

basics that are anything but.



# Webpage Layout

The screenshot shows the Apple website homepage for the iPhone 6. The main headline reads "iPhone 6" followed by the tagline "Bigger than bigger". Below the headline are two links: "Learn more > Watch the film" and "Experience the keynote". To the right of the text are two side-by-side iPhone 6 devices, both displaying a purple flower wallpaper and the time "9:41" and date "Tuesday, September 9". At the bottom of the page, there is a horizontal navigation bar with four items: "WATCH", "iOS 8 Now Available.", "The new album from U2 is here.", and "A message from Tim Cook about Apple's commitment to your privacy".

iPhone 6  
Bigger than bigger

Learn more > Watch the film  
Experience the keynote

WATCH

Our most personal device yet.

iOS 8 Now Available.

The new album from U2 is here.  
And it's already in your iTunes music library.

A message from Tim Cook about Apple's commitment to your privacy.

[www.apple.com/iphone-6/](http://www.apple.com/iphone-6/)

# Webpage Layout

The screenshot shows the Microsoft Store homepage. At the top, there's a navigation bar with links for File, Edit, View, History, Bookmarks, Tools, and Help. Below that is a header bar with the Microsoft logo, a search bar, and a "Secure Search" link. The main content area features a large banner with a person using a Surface Pro 3 tablet. Text on the banner includes "Starting at \$799" and "The only school supplies you'll need". A call-to-action button says "Shop Surface Pro 3 >". To the right of the banner, there are two promotional boxes: one for software discounts and another for Xbox game pre-orders.

File Edit View History Bookmarks Tools Help

Microsoft Store - Xbox, Sur... +

www.microsoftstore.com/store/msusa/en\_US/home

Microsoft Find a store Contact us Sign in Cart (0) Search by keyword, SKU # or item #

Free shipping. Free returns.

Starting at \$799

The only school supplies you'll need

Shop Surface Pro 3 >

\$25 off next order when you buy select software

Shop now >

Office 365 Home Premium

Windows 8.1 Pro

Pre-order any Xbox game and get a \$10 gift card

Shop now >

Call of Duty: Advanced Warfare

Grand Theft Auto V

Tomb Raider

Xbox One console and controller

Shop categories

Surface

Xbox

Computers

Windows Phone

Spotlight

Office 365 Home Premium

Laptop

Minecraft characters

Tablet and accessories