



## Session Maintenance

# Session Maintenance

- HTTP is stateless by default
  - No memory of browsers state
- Motivations for preserving state?
- Examples:



# Where

- At client
  - Cookies
  - Hidden variables
  - URL rewriting
- At server
  - Database, file system, etc.





# Cookies

- Store at the browser
- Are **name=value** pairs (like parameters in a HTTP query)
- PHP script can create cookies and send them to the browser in the HTTP header.
- The browser maintains a list of cookies that belong to a particular Web server, and returns them to the web server in subsequent interactions.
- **Limitations:**
  - Browsers limit the size of cookies.
  - Users can refuse to accept them

# PHP Cookie API

## Create cookie

bool **setcookie** ( string \$name [, string \$value [, int \$expire = 0 [, string \$path [, string \$domain [, bool \$secure = false [, bool \$httponly = false ]]]]] ] )

**Note:** A cookie is a header field, and needs to be set before printing the response body

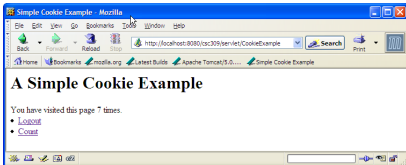
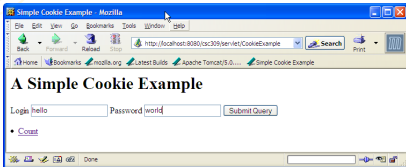
## Retrieve cookie value

\$value = \$\_COOKIE[\$name]

## Reset cookie

bool **setcookie** ( string \$name)

# cookie.php & logout.php



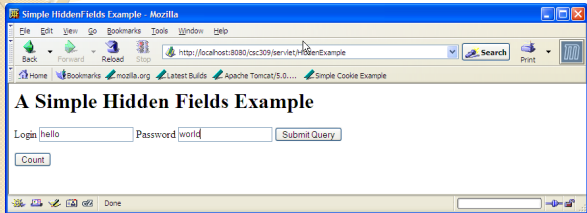
# Hidden Variables

- Store state in web pages: Add hidden variables.
- Supported by all browsers
- Requires all hidden variables to appear in all forms.

```
<input type="hidden" name="secret"  
      value="Don't tell anyone!!">
```

- State is sent inside each web page.
- For form based applications only. Following hyperlinks causes a loss of state.
- Current submitted page represents current state independent of what was done previously.

# hidden.php





# URL Rewriting

- **Store state in the URL:** Rewrite URLs so that they include state variables
- Each URL is now a CGI-get request
- Supported by all browsers
- Requires all URLs contain all state information (long URLs)
- Current submitted page represents current state independent of what was done previously.

# url.php



# Store State At Server

- Current state is stored at the server (i.e., in a file, database)
- Each request includes a token identifying the browsers session (tokens can be passed via cookies, hidden variables, URL rewriting).
- At each request, the executing script uses the token to fetch session state

# PHP Session API

## Initialize session

```
session_start()
```

## Set session attribute

```
$_SESSION["loggedIn"] = true
```

## Retrieve session attribute

```
$count = $_SESSION["count"]
```

## Invalidate session

```
$_SESSION = array()
```

# session.php

