



Lecture 30

11/6/15

Instructor: Yu-San Lin
yusan@psu.edu

Course Website: <http://www.cse.psu.edu/~yul189/cmpsc431w>

Slides based on McGraw-Hill & Dr. Wang-Chien Lee



Overview of the Presentation Tier

- Functionality of the presentation tier
 - Primary interface to the user
 - Adapt to different display devices
 - Simple functionality, e.g., field validity checking
- Technologies:
 - _____: pass data to the middle tier
 - _____: simple functionality at the presentation tier
 - _____: separate data from formatting

HTML Forms

- Common way of passing data from _____ to _____

- General format of a form:

```
<FORM ACTION = "page.jsp"  
      METHOD = "GET"  
      NAME = "LoginForm">
```

...

```
</FORM>
```

Input Fields

- INPUT, SELECT, TEXTAREA are used to specify user input elements
- SELECT: specify a drop-down list
- INPUT (stand alone without end tag) has attributes:
 - TYPE: type of input field
 - NAME: symbolic name, used to identify field value at the middle tier
 - VALUE: default value (and default label for submit button)

JavaScript

- **Goal:** add functionality to the presentation tier
- Sample applications:
 - **Browser detection:** detect browser type and load browser-specific page
 - **Form validation:** validate form input fields
 - **Browser control:** open new windows, close existing windows (e.g., pop-up ads)
- Usually embedded in HTML with `<SCRIPT>...</SCRIPT>` tag

JavaScript Attributes

- LANGUAGE: specifies language of the script
- SRC: external file with script code, e.g.,

```
<SCRIPT LANGUAGE="JavaScript" SRC="validate.js">  
</SCRIPT>
```

- If <SCRIPT> does not have SRC, then JavaScript is directly in the HTML file, e.g.,

```
<SCRIPT LANGUAGE = "JavaScript">  
<!--  
alert("Welcome to our bookstore")  
//-->  
</SCRIPT>
```

JavaScript: Example

HTML Form

```
<form name="LoginForm" method="POST"
  action="TableOfContents.jsp"
  onSubmit="return testLoginEmpty()">
  Userid:
  <input type="text" name="userid">
  Password:
  <input type="password" name="password">
  <input type="submit" value="Login"
  name="submit">
  <input type="reset" value="Clear">
</form>
```

Associated JavaScript

```
<script language="javascript">
function testLoginEmpty()
{
  loginForm = document.LoginForm
  if ((loginForm.userid.value == "") ||
      (loginForm.password.value == ""))
  {
    alert('Please enter values for userid
          and password.');
```

Style Sheets

- **Idea:** separate display from contents, and adapt display to different presentation formats
- Two aspects:
 - **Documents transformations:** decide what parts of the document to display in what order
 - **Document rendering:** decide how each part of the document is displayed

Style Sheets (cont.)

- Why use style sheets?
 - Reuse of the same document for different displays
 - Tailor display to user's preferences
 - Reuse of the same document in different contexts
- Two style sheets languages
 - Cascading style sheets (____): for HTML documents
 - Extensible stylesheet language (____): for XML documents

Cascading Style Sheets (CSS)

- Defines how to display HTML elements
- CSS files contain style definitions
- Many different HTML documents can refer to the same CSS file
 - Can change the format of a website by changing a single file
- Considered as the first step toward the separation of content from presentation

Cascading Style Sheets (CSS)

- CSS file is included into an HTML file with the following line:

```
<LINK REL="style sheet" TYPE="text/css" HREF="books.css" />
```

- Each line in CSS follows the format:

```
Selector {property: value}
```

- Example:

```
BODY {BACKGROUND-COLOR: yellow}
```

- Has the same effect as the following in HTML:

```
<BODY BACKGROUND-COLOR="yellow">
```

Bootstrap



- Originally named Twitter Blueprint, developed by Mark Otto and Jacob Thornton at Twitter for internal use
- An open-source collection of tools for creating websites and web applications
- Contains _____ and _____ based design templates for typography, forms, buttons, navigation and other interface components

Bootstrap: Example

Project name

Home

About

Contact

Hello, world!

This is an example to show the potential of an offcanvas layout pattern in Bootstrap. Try some responsive-range viewport sizes to see it in action.

Link

Link

Link

Link

Link

Link

Link

Link

Link

Link

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

Heading

Donec id elit non mi porta gravida at eget metus. Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Etiam porta sem malesuada magna mollis euismod. Donec sed odio dui.

[View details »](#)

Heading

Donec id elit non mi porta gravida at eget

Heading

Donec id elit non mi porta gravida at eget

Heading

Donec id elit non mi porta gravida at eget

Overview of the Middle Tier

- Functionality of the middle tier
 - Encodes business logic
 - Connects to database systems
 - Accepts form input from the presentation tier
 - Generates output for the presentation tier

Overview of the Middle Tier (cont.)

- Technologies:
 - Ruby on Rails
 - Java EE
 - ASP.NET
 - PHP
 - ColdFusion
 - Perl
 - Python

Maintaining State

- HTTP is stateless
- Advantages
 - Easy to use
 - Great for static-information applications
 - Requires no extra _____
- Disadvantages
 - No record of previous requests, such as
 - Shopping cart
 - User logins
 - Custom or dynamic content

Application State

- **Server-side state**
 - Information is stored in a database, or in the application layer's local memory
- **Client-side state**
 - Information is stored on the client's computer in the form of a cookie
- **Hidden state**
 - Information is hidden within dynamically created web pages

Server-side State

- Store information in a database
 - Data will be safe in the database
 - BUT: requires a database access to query or update the information
- Use application layer's local memory
 - Can map the user's IP address to some state
 - BUT: this information is volatile and takes up lots of server main memory

Server-side State (cont.)

- Should use server-side state maintenance for information that needs to persist, such as
 - Old customer orders
 - “Click streams” of a user’s movement through a site
 - Permanent choices a user makes

Client-side State: Cookies

- Storing text on the client which will be passed to the application with every HTTP request
 - Can be disabled by the client
 - Are wrongfully perceived as “dangerous”, and therefore will scare away potential site visitors if asked to enable cookies

Hidden State

- Often users will disable cookies
- You can “hide” data in two places:
 - Hidden fields within a form
`<input type=“hidden” name=“user” value=“username”/>`
 - Using the path information
`index.html?user=jeffd&preference=pepsi`
- Requires no storage of information because the state information is passed inside of each web page

Summary

- Internet Concepts (URIs, HTTP)
- Web data formats
 - HTML, XML, JSON
- Web scraping
- Three-tier architectures
- The presentation layer
 - HTML forms; Javascript; Style sheets; Bootstrap
- The middle tier
- Maintaining states

Don't Forget

- Next week:
 - Mon (11/9)
 - Review session
 - Homework #4 due
 - Tue (11/10)
 - Midterm 2 part 1, evening
 - Wed (11/11)
 - No class
 - Fri (11/13)
 - Midterm 2 part 2, in class