Lecture 22

Javascript

Announcements

- Homework#7 now due 11/24 at noon
- Reminder: beginning with Homework #7, Javascript assignments must be submitted using a format described in an attachment to HW#7
- I will post a HW#8 due 12/3; HW#9 due 12/10; HW#10 (survey) due 12/15
Announcements

- **Office Hours:**
  - Tuesday, November 23
    - 11:00 AM to 12:00 PM RA Office Hours -- 310
    - 2:00 PM to 3:00 PM RA Office Hours -- 310
  - Wednesday, November 24
    - 11:00 AM to 12:00 PM Yariv Office Hours -- Dubois

Homework#7

- Q2A requires a simple loop and asks for a conversion “table” in the form of a **list**; not an html table
- Q2B asks you to use parseFloat() and while it is good practice, it is not really needed
  - What does parseFloat (parseInt) do?
Printing a table with JavaScript

```html
<html>
<head>
<title>Printing a Table</title>
</head>
<body>
<script type="text/javascript">
    document.write("<table><tr><th>Number</th><th>Number Squared</th></tr>\n");
    var tableEntry = 0;
    for(tableEntry=0;tableEntry<=20;tableEntry=tableEntry+2){
        var tableEntrySq = tableEntry*tableEntry;
        document.write("<tr><td>\n"+tableEntry+"</td><td>\n"+tableEntrySq+"</th></tr>\n");
    }
    document.write("</table>\n");
</script>
</body>
</html>
```

Sets up table

Computes table entries

Prints table entries

Closes table

---

Project

- **Project #2**
  - Did not get all of them graded
  - Hope to finish today, tomorrow

- **Grades**
  - OK, OK-, good, very good, excellent
  - See the comments
    - **OK** = satisfactory, but improve for final report
    - (very) good, excellent = representative of what I am looking for in final report
Final Project -- Due 12/15

- You should plan to have your final project accessible at
  - www-edlab.cs.umass.edu/~yourusername/index.html
- Final Report (P#3)
  1. Summary
  2. Initial design (2-3 paragraphs)
  3. Revised design based on FG (2-3 paragraphs)
  4. Final Design (2-5 pages)
     a) Audience
     b) CSS/HTML design decisions
     c) Javascript design
  5. Accessibility (if relevant)
  6. Conclusions

Final Project Grading

- Report including P1⇒P2⇒P3 evolution 25 pts;
  HTML/CSS 45 pts; JavaScript 30 pts
- HTML/CSS
  - Working website
  - Home page + 5 total linked pages (inc. homepage)
    with link back to homepage, navigation, images, text
  - Consistent layout, color, etc., browser compatibility,
    various subjective considerations
- JavaScript
  - Simple popup(s), buttons, clock, welcome, day of
    week, etc. [5-15 max]
  - Text box input, slide show [10-20 max]
  - More complex JavaScript, jQuery[15-30 max]
- Accessibility = 5-10 points EXTRA credit
Back to the DOM?

- The DOM defines the objects and properties of all document elements, and the methods (interface) to access them.

form Object

- The Form object represents an HTML form.
- For each instance of an HTML <form> tag in a document, a Form object is created.
- Forms are used to prompt users for input.
- The input data is normally posted to a server for processing.
HTML Forms and Input

- A form is an area that can contain form elements that allow the user to enter information
  - E.g. text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.
- A form is defined with the `<form>` tag
  
  ```html
  <form>
  
  input elements
  
  </form>
  ```
HTML Forms and Input

- The most used form tag is the `<input>` tag
- The type of input is specified with the type attribute.
  - *text fields* are used when you want the user to type letters, numbers, etc. in a form.

```
<form>
  First name: <input type="text" name="firstname" />
  Last name: <input type="text" name="lastname" />
</form>
```

- **Input fields**
  - **Password fields**
  - **Textarea**

- **Radio buttons** are used when you want the user to select one of a limited number of choices.

```
<form>
  <input type="radio" name="sex" value="male" /> Male
  <input type="radio" name="sex" value="female" /> Female
</form>
```

- **Checkboxes** are used when you want the user to select one or more options of a limited number of choices

```
<form>
  I have a bike: <input type="checkbox" name="vehicle" value="Bike" />
  I have a car: <input type="checkbox" name="vehicle" value="Car" />
  I have an airplane: <input type="checkbox" name="vehicle" value="Airplane" />
</form>
```

- **Mixed Forms**

```
Q11.5
```
HTML Forms and Input

- **Action Attribute and the Submit Button**
  - When the user clicks on the "Submit" button, the content of the form is sent to the server.
  - The form's action attribute defines the name of the file to send the content to.
  - The file defined in the action attribute usually does something with the received input.

```
<form name="input" action="html_form_submit.asp" method="get">
  Username: <input type="text" name="user" />
  <input type="submit" value="Submit" />
</form>
```

If you type some characters in the text field below, and click the "Submit" button, the browser will send your input to a page called `html_form_submit.asp`. The page will show you the received input.

Examples

- create a simple drop-down box on an HTML page. A drop-down box is a selectable list.
  - Simple drop down box
  - Another drop down box
- draw a border with a caption around your data
  - Fieldset around data
- add a form to a page. The form contains two input fields and a submit button
  - Form with input fields and a submit button
- extended forms
  - Form with checkboxes
  - Form with radio buttons
- how to send e-mail from a form
  - Send e-mail from a form
  - e-mail validation
**Example**

```html
<form name="formtest">
  Please enter your name: <br>
  <input type="text" size="50" name="user_name">
</p>
  Please enter your phone: <br>
  <input type="text" size="30" name="user_phone">
</p>
  <input type="button" value="show form data"
  onClick="showForm(this.form)">
</form>

function showForm(myform) {
  NewWin=window.open('', '', 'width=300, height=200');
  name_input="<b>Your name: " + myform.user_name.value + "</b><br;">
  NewWin.document.write(name_input);
  phone_input="<b>Your phone: " + myform.user_phone.value + "</b><br;">
  NewWin.document.write(phone_input);
```
Example

```javascript
function display(){
    var output="";
    output+="First Name: "+document.aform.first.value;
    output+="Last Name: "+document.aform.last.value;
    output+="Comments: "+document.aform.comments.value;
    output+="Check box checked: "+document.aform.check.checked;
    output+="Selection box: "+document.aform.sel.value;
    alert(output);
}
```

```html
<form name="aform">
    First Name:<br>
    <input type="text" name="first" value="bob" onFocus="this.select()"><br>
    Last Name:<br>
    <input type="text" name="last" value="smith" onFocus="this.select()"><br>
    Comments:<br>
    <textarea name="comments" rows="5" cols="100">
        text in text area
    </textarea><br>
    check box<br>
    <input type="checkbox" name="check"><br>
    pick one<br>
    <select name="sel">
        <option value="hw">homework</option>
        <option value="quiz">quiz</option>
        <option selected value="exam">exam</option>
    </select><br>
    <input type="button" value="click to display information" onClick="display()" name="button1">
</form>
</html>
```

Pulldown

```html
<html>
<head><title>Drop-Down Menus</title>
<script language="JavaScript">
    function schedule(f){
        if(f.menu1.selectedIndex == 0){
            // Could also say: document.form1.menu1.selectedIndex
            f.text1.value="PL100, Feb 3-7, 9am to 5pm, Room 2133, Dr. Baloney 
            // Could also say: document.form1.text1.value
        }
        if(f.menu1.selectedIndex == 1){
            f.text1.value="PL200 Feb 10-13 9am to 5pm, Room 209B, Ms. Eclectic"
        }
        if(f.menu1.selectedIndex == 2){
            f.text1.value="UX101 Mar 2-6 9am to 5pm, Room 209, Mr. Nerdly"
        }
        if(f.menu1.selectedIndex == 3){
            f.text1.value="SH201 Apr 10-13 9am to 5pm, Room 209B, Miss Bashing"
        }
    }
</script>
<body bgcolor=lightgreen>
    <font face=arial>
        <form name="form1">
            Select a Course<br>
            <select name="menu1" size="4" onChange="schedule(this.form)">
                <option name="choice1" value="Perl1">Intro to Perl</option>
                <option name="choice2" value="Perl2">Advanced Perl</option>
                <option name="choice3" value="Unix1">Shell Programming</option>
            </select><br>
            <input type="text" name="text1" size=60 />
        </form>
</body>
</html>
```
Accessing images & creating rollovers

- As with forms, images are accessible through "image object"

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt</td>
<td>Sets or returns how to align an image according to the surrounding text</td>
</tr>
<tr>
<td>all</td>
<td>Sets or returns an alternate text to be displayed, if a browser cannot show an image</td>
</tr>
<tr>
<td>border</td>
<td>Sets or returns the border around an image</td>
</tr>
<tr>
<td>complete</td>
<td>Returns whether or not the browser has finished loading the image</td>
</tr>
<tr>
<td>height</td>
<td>Sets or returns the height of an image</td>
</tr>
<tr>
<td>hover</td>
<td>Sets or returns the white space on the left and right side of the image</td>
</tr>
<tr>
<td>id</td>
<td>Sets or returns the id of the image</td>
</tr>
<tr>
<td>align</td>
<td>Returns whether or not an image is a server-side image map</td>
</tr>
<tr>
<td>longDesc</td>
<td>Sets or returns a URL to a document containing a description of the image</td>
</tr>
<tr>
<td>lowRes</td>
<td>Sets or returns a URL to a low-resolution version of an image</td>
</tr>
<tr>
<td>name</td>
<td>Sets or returns the name of an image</td>
</tr>
<tr>
<td>src</td>
<td>Sets or returns the URL of an image</td>
</tr>
<tr>
<td>useMap</td>
<td>Sets or returns the value of the usemap attribute of an inline image map</td>
</tr>
<tr>
<td>width</td>
<td>Sets or returns the width of an image</td>
</tr>
</tbody>
</table>

Creating an image rollover effect

- assigning a new image to the src property of the image each time the onMouseover event handler is fired inside an image link, we change the image onMouseover.

```html
<html>
<head><title>Preloading Images</title></head>
<h2>This Is Baby William</h2>
<script language="JavaScript">
  if(document.images){
    var baby1=new Image(); // Preload an image
    baby1.src="baby1image.jpg";
  }
  if (document.images){
    var baby2=new Image(); // Preload an image
    baby2.src="baby2image.jpg";
  }
</script>
<body bgcolor="cornflowerblue">
  <a href="#" onMouseOver="document.willy.src=baby2.src;"
    onMouseOut="document.willy.src=baby1.src;">
    <img name="willy" src="baby1image.jpg" align="left" alt="baby" border=2
      hspace="10" width="220" height="250">
  </a>
</body>
</html>
```

Sequencing Images Q11.33
Selecting an image Q11.34

Changing on a click Q11-6
Preloading images

- load an image into cache before being used, so it appears instantaneously when needed
  - good for change of image in effects like rollover images and image slideshows.
  - create an instance of the image object in the HEAD section of the page, and assigning the image we wish to preload to its src property
- example:
  ```html
  <head>
  <script type="text/javascript">
  <!--
  image01= newImage()
  image01.src="1.gif"
  image02= new Image()
  image02.src="3.gif"//-->
  </script>
  </head>
  ```
- repeat this for every image you wish to preload.

jQuery

- jQuery is a JavaScript Library
  - What is a library?

- jQuery is a library of JavaScript functions.
  - a lightweight "write less, do more" library.
  - greatly simplifies JavaScript programming
  - easy to learn
jQuery features

- HTML element selections
- HTML element manipulation
- CSS manipulation
- HTML event functions
- JavaScript Effects and animations
- HTML DOM traversal and modification
- AJAX
- Utilities

Adding jQuery Library

- The jQuery library is stored a single JavaScript file, containing all the jQuery functions.
- It can be added to a web page with the following mark-up:

```html
<head>
    <script type="text/javascript" src="jquery.js"></script>
</head>
```
- Please note that the `<script>` tag should be inside the page's `<head>` section
Basic jQuery Example

- The following example demonstrates the jQuery hide() function, hiding all <p> elements in an HTML document.

```html
<html>
<head>
<script type="text/javascript" src="jquery.js"></script>
<script type="text/javascript">
$(document).ready(function(){
  $('button').click(function(){
    $('p').hide();
  });
});
</script>
</head>
<body>
<h2>This is a heading</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button>
</body>
</html>
```

jQuery Syntax

- The jQuery syntax is tailor made for selecting HTML elements and perform some action on the element(s).

- Basic syntax is: $(selector).action()
  - A dollar sign to define jQuery
  - A (selector) to "query (or find)" HTML elements
  - A jQuery action() to be performed on the element(s)

- Examples:
  - `$(this).hide()` - hides current element
  - `$('p').hide()` - hides all paragraphs
  - `$('p.test').hide()` - hides all paragraphs with class="test"
  - `$('#test').hide()` - hides the element with id="test"
The Document Ready Function

- A document.ready() function is used to prevent any jQuery code from running before the document is finished loading (is ready).
  
  ```javascript
  $(document).ready(function(){
    // jQuery functions go here...
  });
  ```

- examples of actions that can fail if functions are run before the document is fully loaded:
  - Trying to hide an element that doesn't exist
  - Trying to get the size of an image that is not loaded

Hiding - Sliding - Fading

- jQuery fadeOut()
  - Example
- Hide explanations
  - Example
- Slide panel
  - Example
- jQuery animate()
  - Example
  - Example