Web Programming

Kristina Striegnitz

cs.union.edu/~striegnk

striegnk@union.edu

Steinmetz 233

phone: x6554
PHP / MySQL – client server interaction
Javascript – client server interaction

Web Programming
CSC 240
Union College
Fall 2009

Syllabus  Schedule  Assignments  Resources

Course Description
This course addresses standards in developing applications for the world wide web. We will cover the following topics: XHTML, CSS, PHP, MySQL, Javascript, Ajax. You will learn how to use them appropriately and how to combine them to build flexible, user friendly and robust web sites.

When and where
TTTH 9:00-10:45, Clin 103

How to get in touch with me
email: streign@union.edu
office phone: (518) 388-6554
office: Steinmetz 213
office hours: Tuesdays 1-3, Fridays 1-3, by appointment
more info about me: streign@union.edu

Textbook
There are no required text books for this course. I could not find one book that covers all the topics discussed in class and that I liked and don’t want to require multiple books as that gets expensive. However, here are the books that I recommend, if you do want to buy books:

* HTML 101: The Best-Practice Guide to XHTML and CSS by Patrick Griffith (at amazon)
* DOM Scripting: Web Design with JavaScript and the Document Object Model by Jeremy Keith (at amazon)
Embedding Javascript into HTML

- internal:
  ```html
  <script type="text/javascript">
    javascript code here
  </script>
  ```

- external:
  ```html
  <script type="text/javascript" src="first_example.js">
  </script>
  ```

(See hello_javascript_1/2/3/4.html)
Javascript – simple output (for now)

- `alert(string)` – produces pop-up
- `document.write(string)` – writes to the html document
Javascript - Overview

• comments
• statements
• variables
• datatypes
• operators
• control flow (conditionals, loops)
• functions
• objects

(See chapter on Blackboard.)
Javascript - Comments

- // single line comment
- /*
  
multi line comment
  */
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
• don't have to end in ';

• but it is good practice to use ';
  to end them
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
Javascript - variables

- don't have to be declared, but it is good practice to declare them:
  
  ```javascript
  var some_variable;
  some_variable = 6;
  
  var another_variable = 5;
  ```
Javascript – simple input (for now)

- var name = prompt(“What is your name”?, “default name”);
  - opens pop up asking for input
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
nothing surprising:
strings: “hallo”, 'hallo'
numbers: 4, 5.6, -3.87
boolean: true, false
Arrays (indexed):

```javascript
var some_numbers = [1, 2, 3, 4];
var more_numbers = Array(5, 6, 7);
var another_array = Array();
another_array[0] = 1;
```

Arrays (associative):

```javascript
var numbers_and_letters = Array();
numbers_and_letters["a"] = 97
numbers_and_letters["z"] = 122
```
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
Nothing unusual.

string concatenation: +
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
Javascript - conditionals

```javascript
if (condition) {
    ...
}
else if (condition) {
    ...
}
else {
    ...
}
```

Optional
Javascript - Overview

• comments
• statements
• variables
• datatypes
• operators
• control flow (conditionals, loops)
• functions
• objects

(See chapter on Blackboard.)
while (condition) {
...
}

do {
...
} while (condition);
for (var count=0; count <= 10; count++) {
...
}

for (var element in some_array) {
...
}
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
function name (parameter, another parameter, ...) {
    ...
}

local vs. global scope:
• local: only visible inside function
• var inside a function creates a local variable
• global scope: visible both outside of functions and inside
Some simple Javascript exercises

• Write a function \( \text{max} \) that takes two numbers as arguments and returns the larger one.

• Write two functions \( \text{sum} \) and \( \text{multiply} \). \( \text{sum} \) should add all numbers in an array of numbers. \( \text{multiply} \) should multiply them.

• Write a function \( \text{happyBirthday} \) that takes a number as argument and writes “Happy Birthday!” onto the screen as many times.
Javascript - Overview

- comments
- statements
- variables
- datatypes
- operators
- control flow (conditionals, loops)
- functions
- objects

(See chapter on Blackboard.)
What are objects?
creating a new object instance:

\[
\text{var } \text{var}_\text{name} = \text{new Object}_\text{Type}();
\]
\[
\text{e.g.: var } \text{today} = \text{new Date}();
\]
accessing properties/attributes/instance variables:

\[
\text{obj}_\text{var}_\text{name}.\text{attribute}_\text{name}
\]
calling methods/functions

\[
\text{obj}_\text{var}_\text{name}.\text{method}_\text{name}(\ldots)
\]
Javascript – kinds of objects

• native objects
  – date, strings, arrays

• host objects
  – document (as in document.write(...))
  – we will see much more of these

• user defined objects
  – we are not going to use those
The date object

- var date = new Date()
- methods:

<table>
<thead>
<tr>
<th>Methods</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date()</td>
<td>Returns a Date object</td>
</tr>
<tr>
<td>getDate()</td>
<td>Returns the date of a Date object (from 1-31)</td>
</tr>
<tr>
<td>getDay()</td>
<td>Returns the day of a Date object (from 0-6. 0=Sunday, 1=Monday, etc.)</td>
</tr>
<tr>
<td>getMonth()</td>
<td>Returns the month of a Date object (from 0-11. 0=January, 1=February, etc.)</td>
</tr>
<tr>
<td>getFullYear()</td>
<td>Returns the year of the Date object (four digits)</td>
</tr>
<tr>
<td>getHours()</td>
<td>Returns the hour of the Date object (from 0-23)</td>
</tr>
<tr>
<td>getMinutes()</td>
<td>Returns the minute of the Date object (from 0-59)</td>
</tr>
<tr>
<td>getSeconds()</td>
<td>Returns the second of the Date object (from 0-59)</td>
</tr>
</tbody>
</table>
Exercise

- Create a page that always displays the current date in the following format:

  Thursday, October 22, 2009

- Use a date object and its methods, and use associative arrays to map days and months as numbers to their names.