Web Programming

Kristina Striegnitz

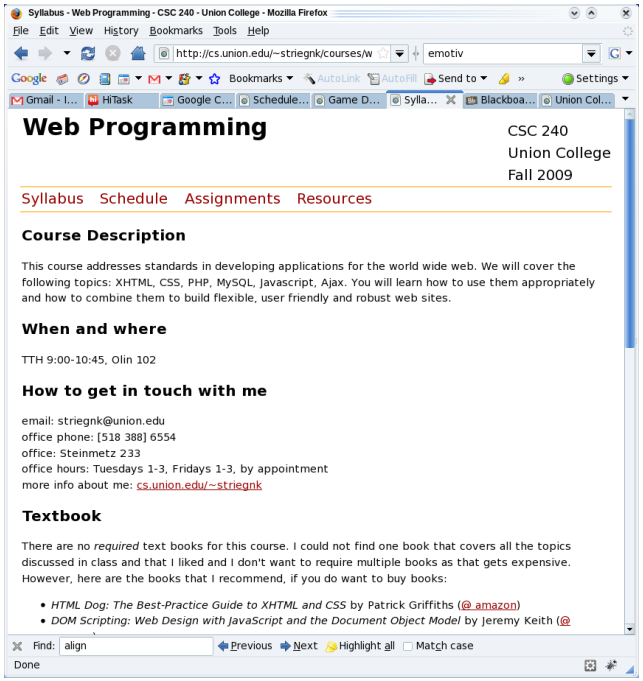
cs.union.edu/~striegnk

striegnk@union.edu

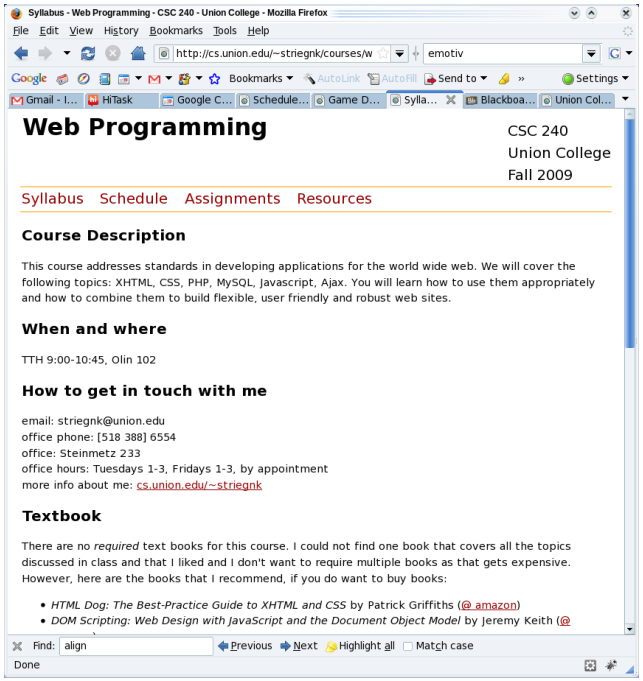
Steinmetz 233

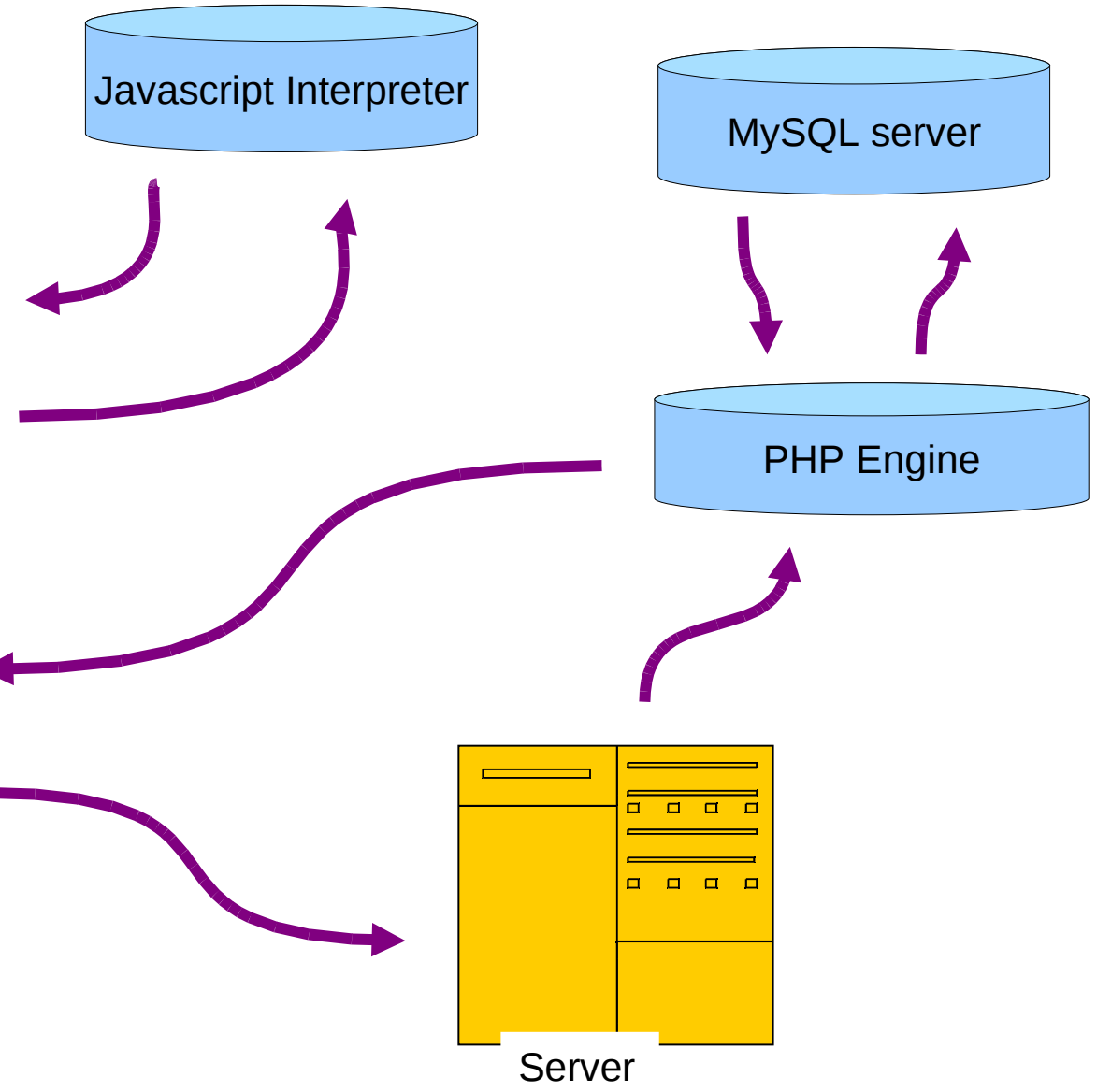
phone: x6554

PHP / MySql – client server interaction



Javascript – client server interaction





Embedding Javascript into HTML

- internal:

  <script type=”text/javascript”>

  java script code here

  </script>

- external:

  <script type=”text/javascript”

  scr = "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.)

Javascript - statements

- 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:

```
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.)

Javascript - datatypes

- nothing surprising:
- strings: “hallo”, 'hallo'
- numbers: 4, 5.6, -3.87
- boolean: true, false

Javascript - datatypes

- arrays (indexed):

```
var some_numbers = [1, 2, 3, 4];

var more_numbers = Array(5, 6, 7);

var another_array = Array();

another_array[0] = 1;
```

- arrays (associative):

```
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.)

Javascript - operators

- Nothing unusual.
- string concatenation: +

Javascript - Overview

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

(See chapter on Blackboard.)

Javascript - conditionals

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

optional

optional

Javascript - Overview

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

(See chapter on Blackboard.)

Javascript – loops - while

while (*condition*) {

...

}

do {

...

} while (*condition*);

Javascript – loops - for

```
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.)

Javascript - functions

```
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 `max` that takes two numbers as arguments and returns the larger one.
- Write two functions `sum` and `multiply`. `sum` should add all numbers in an array of numbers. `multiply` should multiply them.
- Write a function `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.)

Javascript - objects

- What are objects?
- creating a new object instance:

  var var_name = new Object_Type();

  e.g.: var today = new Date();
- accessing properties/attributes/instance variables:

  obj_var_name.attribute_name
- calling methods/functions

  obj_var_name.method_name(...)

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:

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

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.