SWINBURNE UNIVERSITY OF TECHNOLOGY

COS10005 Web Development

Module 10 - Part B – jQuery and Server Side Scripting

Contents

- Client-Side Scripting
  - jQuery
    - JavaScript to jQuery Example
    - jQuery Effects
    - Handling Arrays
    - Handling Objects
- Server-Side Web Development
  - Introduction
  - A Quick Look at PHP

CLIENT SIDE SCRIPTING

SWINBURNE UNIVERSITY OF TECHNOLOGY

jQuery

- jQuery is a library (a collection of pre-written JavaScript code) that simplifies common tasks in web development, primarily on the frontend.
- jQuery it is specialised for updating webpage documents on the fly.
- Library can be downloaded from www.jquery.com
- Download the compressed (production) version. This is a “minified” version intended for speed Example `jquery-3.6.0.min.js`
-

Framework: jQuery

- The filename format is jquery-<version>.min.js
- `<script src="jquery-3.6.0.min.js"></script>`
- `<script src="myscript.js"></script>` Don’t forget your own JavaScript!
    - Preferably the ***first*** script defined in the HTML
- Include the jQuery library
- Selecting element(s) using jQuery
    - `$(<selector>)`

jQuery: Selection

- `$(<selector>)`

Selector	selects
"*"	all elements
".*class*"	all elements that use a specified class name
"*element*"	All elements that match the specified element name
"#*id*"	The first element that matches the specified id attribute
"selector1, selector2, …"	All elements that match the results of one or more specified selectors

jQuery: Selection

JavaScript:

- `document.getElementById("regform")`

Using jQuery: **`$("#regform")`** Like CSS!

- `document.getElementsByTagName("p")`

Using jQuery: **`$ ("p")`**

I probably should have told you this sooner.
Please don't hate me …

jQuery: Selection

- Contextual Selection

How to select all the `<section>` elements in the `<article>` element?

In CSS:

?

Using jQuery:

?

```
<section>Apple</section>
<article id="myArticle">
  <section id="s2">
       iPhone 5
  </section>
  <section id="s3">
       iPhone 6
  </section>
</article>
<section id="s3">
       iPad Air
</section>
```

jQuery: Variable Assignment

- Using JavaScript

```
var regForm =
    document.getElementById("regform");
```

```
regForm.onsubmit = validate;
```

- Using jQuery

```
$regForm = $("#regform");
```

```
$regForm.submit(validate);
```

jQuery: Property and Method

All about functions!

Using JavaScript	**Using jQuery**
element	*element*
• .onclick = *function*	• .click(*function*)
• .onsubmit = *function*	• .submit(*function*)
• .value	• .val()
• *.checked*	• .attr("checked")
• .style.*color*	• .css("color")
• .style.display="none"	• .css("display", "none")

JQUERY INSTEAD OF JAVASCRIPT - EXAMPLES

JavaScript to jQuery: Step 1

- Ensure HTML elements are given IDs,
  if the element will be accessed in JavaScript
- For example,

```
<form id="surveyForm" … >
        …
</form>


<h1 id="companyName">…</h1>


<p id="pIntro">…</p>
```

JavaScript to jQuery: Step 2

- Create necessary JavaScript functions that perform specific tasks

```
function functionName () {
    /* JavaScript codes; */
}
```

- For example,

```
function validateForm () {
    alert ("All good!");
    return true;
}
```

JavaScript to jQuery: Step 2

- For example, using JavaScript

```
function updateHTML () {
    var h1 =
document.getElementById("companyName");
    h1.innerHTML = "Apple";
}
```

- For example, **using jQuery**

```
function updateHTML () {
    $("#companyName").html("Apple");
}
```

JavaScript to jQuery: Step 3

- Creating an initialisation function
- For example, in JavaScript

```
function init () {
    /* obtain HTML element objects
    assign functions to the objects' events */
}
window.onload = init;
```

- In, jQuery

```
function init () {
    /* obtain HTML element objects AND assign
          functions to the objects' events */
}
$(document).ready(init);
```

JavaScript to jQuery: Step 3

- JavaScript
  - Obtain HTML element object

```
var surveyForm=
document.getElementById("surveyForm");
```

  - Assign event to function

```
formElement.onsubmit = validateForm;
```

- jQuery
  - Obtain and Assign

```
$("#surveyForm").submit(validateForm);
```

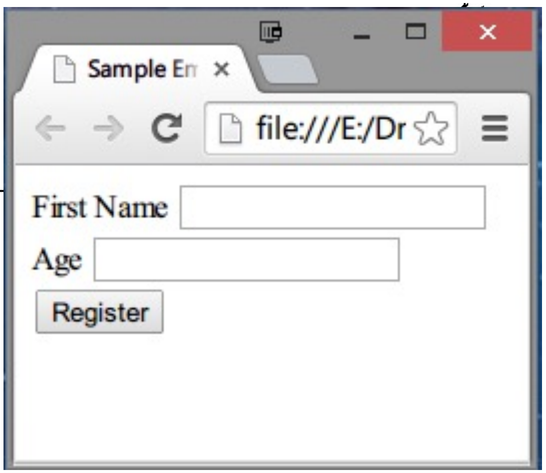
JavaScript to jQuery: Events

Mouse Event	The event occurs when the user
click	clicks on an element
dblclick	double-clicks on an element
hover	Moves in and/or out an element
mouseup	releases a mouse button over an element
mousedown	presses a mouse button over an element
mousemove	moves while it is over an element
mouseover	moves into an element
mouseout	moves out of an element
mouseenter	moves into a bound not descendant element
mouseleave	moves out of bound not descendant element

JavaScript to jQuery: Events

keydown	is pressing a key
keypress	presses a key
keyup	releases a key
blur	a form element loses focus
focus	an element gets focus
focusin	an element gets focus
focusout	a form element loses focus
change	the content of a form element, the selection, or the checked state have changed
select	a user selects some text
submit	a form is submitted

jQuery: Form Data Validation



Sample En

file:///E:/Dr

First Name

Age

Given the following HTML form:

```
<form id="regForm" method="post"
   action="process.php">
   <div class="textinput">
      <label for="firstname">First Name</label>
      <input type="text" name="firstname"
                               id="firstName" />
   </div>
   <div class="textinput">
      <label for="age">Age</label>
      <input type="text" name="age" id="age" />
   </div>
   <div class="buttoninput">
      <input type="submit" value="Register" >
   </div>
</form>
```

jQuery: Form Data Validation (continued)

```
function validateForm () {
/* validation code here */
  return true/false;
}
```

Write the validation code, and return **`true`** if valid, otherwise **`false`.**

```
function init () {
 $("#regForm").submit(validateForm);
}
```

Form ID

JavaScript: **var** regForm = document.getElementById("**regForm**");
regForm.onsubmit=validateForm;

```
$(document).ready(init);
```

Don't forget this!

JavaScript: window.onload(init);

jQuery: Form Data Validation (continued)

- Validation Function – Part 1

```
function validateForm() {
 var firstName = $("#firstName").val();
 var age = $("#age").val();
 var errMsg = "";
 var result = true;
```

Get the `value` property of an input element, e.g., text box or select.

jQuery: Form Data Validation (continued)

- Validation Function– Part 2

```
  if (firstName == "") {
    errMsg += "First Name cannot be empty.\n";
 }
 if (age == "") {
     errMsg += "Age cannot be empty.\n";
 }
 if (isNaN(age)) {
     errMsg += "Age is not a valid number.\n";
 }
 if (errMsg != "") {
     alert (errMsg);
     result = false;
 }
 return result;
}
```

If **`false`** is returned here, the form would not be submitted.

JQUERY
- JQUERY EFFECTS

jQuery Effects

- Effects without writing the CSS code

Effect	Description
.hide()	Hide the selected elements
.show()	Display the selected elements
.toggle()	Display or hide the selected element
.fadeIn()	Display the selected elements by fading in
.fadeOut()	Hide the selected elements by fading out
.slideDown()	Display in sliding motion the selected elements
.slideUp()	Hide in sliding motion the selected elements
.slideToggle()	Display or hide in sliding motion selected element
.animate()	Display with customs css property value

jQuery Effects

Example

- $("sidTip").show();
- $("sidTip").hide();
- $("sidTip").fadeIn();
- $("sidTip").fadeOut();
- $("sidTip").slideUp();
- $("sidTip").slideDown();
- $("sidTip").animate({height:"300px"});

Duration parameters on all effects such as "slow", "fast", or a number can be specified.

jQuery Effects

JavaScript

```
function showTip () {
  sidTip.style.display
     = "inline";
}

function hideTip () {
  sidTip.style.display
     = "none";
}
```

jQuery 1

```
function showTip () {
  $("#sidTip").show();
}

function hideTip () {
  $("#sidTip").hide();
}
```

jQuery Effects

jQuery 2

```
function showTip () {
  $("#sidTip").fadeIn();
}
```

```
function hideTip () {
  $("#sidTip").fadeOut();
}
```

jQuery 3

```
function showTip () {
  $("#sidTip").slideUp();
}
```

```
function hideTip () {
$("#sidTip").slideDown();
}
```

JQUERY
- HANDLING ARRAYS

Array Objects

- Selecting Multiple Elements

```
$h1s = $("h1");
$elements =$(".red");
```

- jQuery provides methods that access elements in an array:

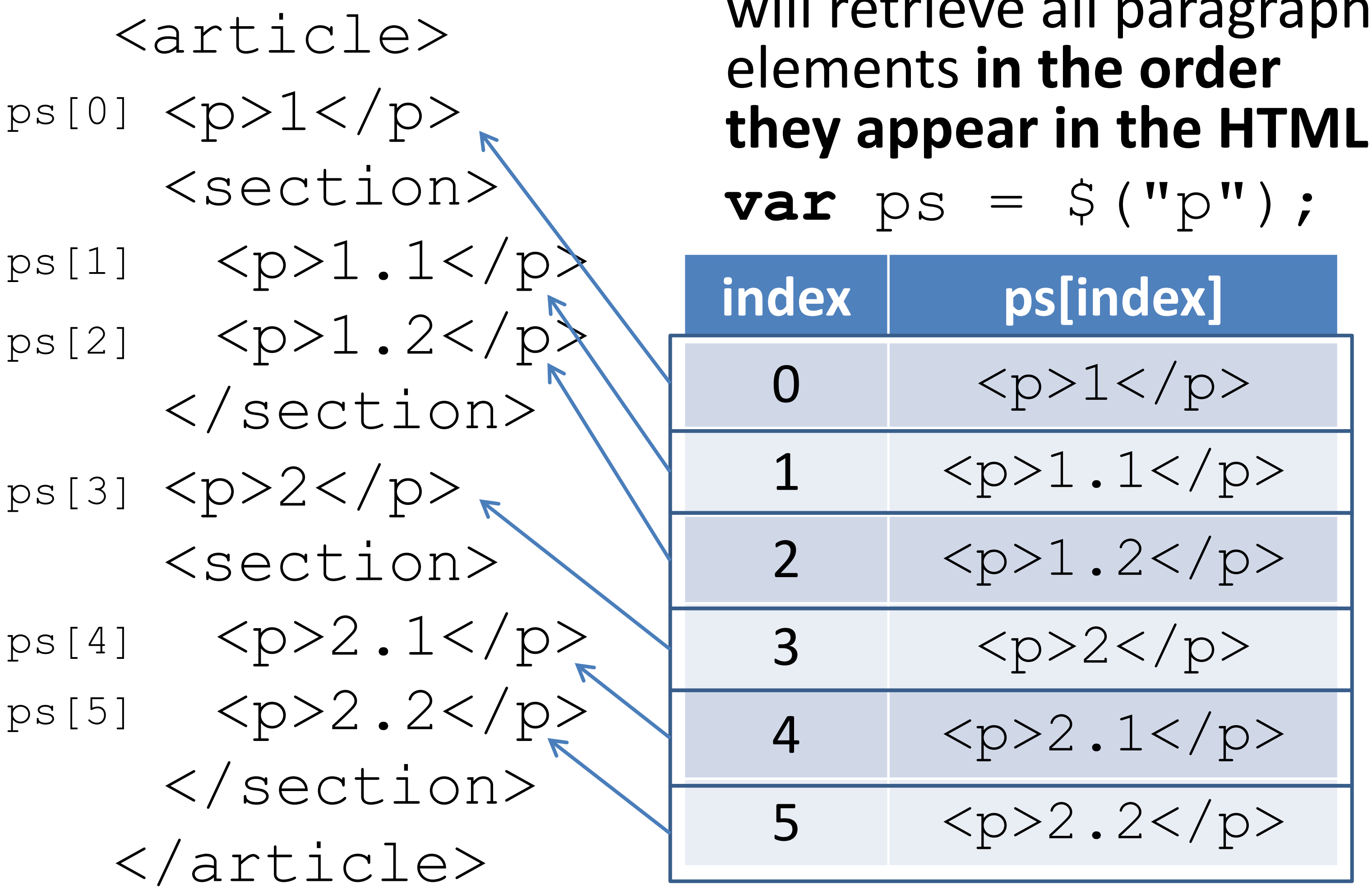
`$element.`**`first()`** – get first element

`$element.`**`eq(<#>)`** – get element by index #

`$element.`**`last()`** – get last element

Array Objects: HTML Code

Given the following



```
      <article>
ps[0]   <p>1</p>
        <section>
ps[1]     <p>1.1</p>
ps[2]     <p>1.2</p>
        </section>
ps[3]   <p>2</p>
        <section>
ps[4]     <p>2.1</p>
ps[5]     <p>2.2</p>
        </section>
      </article>
```

- Retrieve all <p> elements will retrieve all paragraph elements **in the order they appear in the HTML**

```
var ps = $("p");
```

index	ps[index]
0	<p>1</p>
1	<p>1.1</p>
2	<p>1.2</p>
3	<p>2</p>
4	<p>2.1</p>
5	<p>2.2</p>

Array Objects: JavaScript

- Retrieve all <p> elements using

```
var p1s =
document.getElementsByTagName("p");
```

i	ps[i].innerHTML
0	1
1	1.1
2	1.2
3	2
4	2.1
5	2.2

- Sample loop code

```
var p1s =
document.getElementsByTagName
("p");
var i;

for(i = 0;i < p1s.length;i++)
{
  alert(p1s[i].innerHTML);
}
```

Array Objects: jQuery

- Retrieve all <p> elements using

```
var $p3s = $("p");
```

i	$ps.eq(i).html()
0	1
1	1.1
2	1.2
3	2
4	2.1
5	2.2

- Sample loop code

```
$p3s = $("p");
var i;
for(i = 0;i<$p3s.length;i++)
{
 alert($p3s.eq(i).html());
}
```

JavaScript:
p2s[i].innerHTML

`$p3s.first()`

`$p3s.last()`

JQUERY
- HANDLING OBJECTS

Handling Objects

- jQuery provides methods that allows one to add or remove HTML elements
- First, get access to the target element: `$e = $(…);`
- Methods used to dynamically insert elements
  - `$e.before("<p>Paragraph</p>");`
  - `$e.after("<hr />");`
- Methods used to select previous and next sibling elements
  - `$e.prev():` get the previous sibling element
  - `$e.next():` get the next sibling element

Handling Objects

- For example to interactively create a tooltip on focus, we have

```
function showAgeTip() {
  $(this).after("<span class='tooltip'>
     must be above 18yo</span>");
}
function hideAgeTip() {
  $(this).next().remove();
}
function init () {
  $("#tbAge").focusin(showAgeTip);
  $("#tbAge").focusout(hideAgeTip);
}
```

SERVER-SIDE WEB DEVELOPMENT
- A QUICK LOOK AT PHP

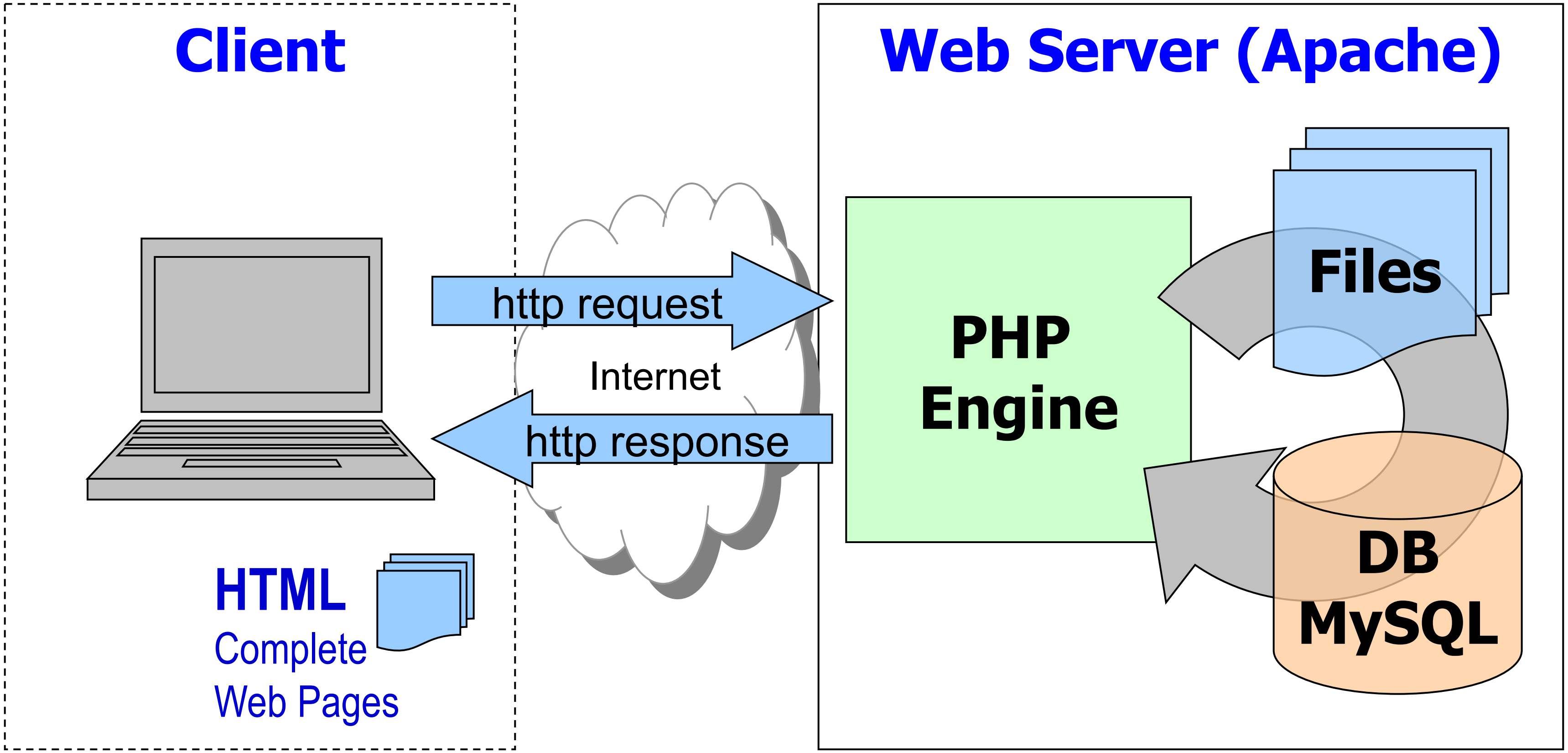
What is Embedded Scripting ?

- ***Embedded Scripts*** are scripts that are ***embedded*** or linked into HTML documents, and stored on the server.
- In response to client requests, the ***called pages are "parsed"*** by the ***server software***, the ***embedded scripts are "processed"*** and the requested information or content is ***returned as formatted html***.
- Client requests usually include parameters (key=value pairs) that are passed to the server, so the embedded scripts might query databases, or retrieve other dynamic information.
- The client response is *(usually)* ***browser independent,*** as it returns ***"plain html".***
- The embedded script ***is not visible to the client***

  - the client only sees the completed html page.

Embedded Scripting

Apache/PHP/MySQL example



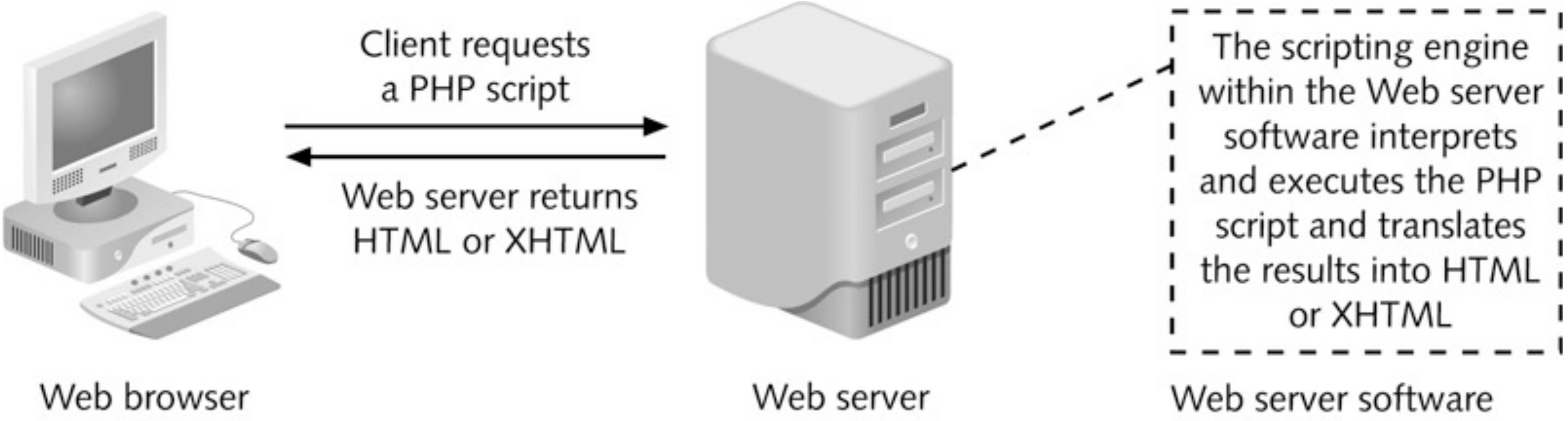
Embedded Scripting and PHP

- **Server-side scripting** refers to a scripting language that is executed from a Web server
- **PHP** is a server-side *embedded* scripting language that is used to develop interactive Web sites
    - Is easy to learn
    - Includes object-oriented programming capabilities
    - Supports many types of databases (MySQL, Oracle, Sybase, ODBC-compliant)

Embedded Scripting and PHP (continued)

- **PHP** (continued):
  - PHP is an open source programming language
    - Open source refers to software where source code can be freely used and modified
  - Can’t access or manipulate a Web browser, like JavaScript
  - Exists and executes solely on a Web server, where it performs various types of processing or accesses databases

Embedded Scripting and PHP (continued)


Client requests a PHP script
Web server returns HTML or XHTML
The scripting engine within the Web server software interprets and executes the PHP script and translates the results into HTML or XHTML
Web browser
Web server
Web server software

How a Web server processes a PHP script

- **General rule:**
  Use *client-side scripting* to handle user interface processing and light processing, such as data validation; use *server-side scripting* for intensive calculations and data storage.

Quick PHP Intro

What is PHP? http://www.php.net

- PHP stands for **P**HP: **H**ypertext **P**reprocessor
- PHP is a server-side scripting language, like ASP
- PHP scripts are executed on the server
- PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP is an open source software (OSS)
- PHP is free to download and use
- PHP filename .php

Source: w3schools

PHP Example…

```
<html>
...
<body>
<h1>Hello World!</h1>
<?php
 echo "<p>";
 $i=1;
 while($i<=5) {
   echo "The number is " . $i . "<br />";
   $i++;
 }
 echo "</p>";
?>
</body>
</html>
```

You are not expected to be able to create any PHP scripts. This example is just to help you understand Embedded Scripting Concepts.

Embedded PHP Script is processed on the Server, before it is sent to the client

NEXT LECTURE:

INTRODUCTION TO XML and JSON