

# Connecting the Dots

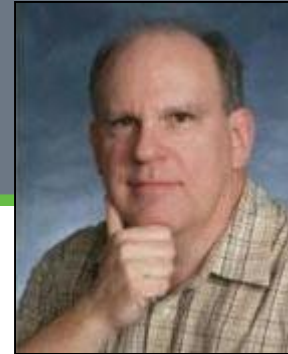
Building Web Applications with  
PHP, HTML, CSS, and JavaScript

**John Valance**  
division 1 systems  
johnv@div1sys.com

`<div1>`

[www.div1sys.com](http://www.div1sys.com)

# About John Valance



- **Independent Consultant for 15 years**
- **Founder and CTO of Division 1 Systems**
  - ▶ Helping IBM shops develop web applications and related skills
  - ▶ Extended team of 150+ technical people
  - ▶ Web and mobile systems development, design, project management
  - ▶ Training, mentoring, consultation and coding
- **30+ years IBM midrange experience (S/38 thru IBM i)**
- **15+ years of web development experience**
- **Frequent presenter on web development topics**
- **Working with Zend Technologies**
  - ▶ Teach Intro to PHP for RPG programmers
  - ▶ Zend Certified Engineer
  - ▶ Zend Reseller

# Goals of Presentation

- Introduce web development concepts to web beginner (experienced RPG programmer)
- Introduce major technical concepts and how components interact
- Introduce language syntax
- Show-and-tell demos and code examples (fun stuff)
- Prepare you for labs on HTML, CSS, PHP and JavaScript
- Come away with an idea of how to start

# Languages Involved in a PHP Database Application

- **Client side (web browser):**
  - ▶ HTML
  - ▶ CSS
  - ▶ JavaScript
- **Server side (IBM i):**
  - ▶ PHP
  - ▶ SQL (accessing DB2 tables)
  - ▶ Possibly RPG & CL
    - Called via stored procedures or Zend Toolkit for IBMi

# HTML



# HTML Sample Structure

```
<!DOCTYPE html>
```



This ensures HTML5

```
<html>
```

```
<head>
```

```
<title>Static Hello World</title>
```

```
</head>
```

```
<body>
```

```
<h1>Hello, World Wide Web!</h1>
```

```
</body>
```

Looks like this in browser:

```
</html>
```



96.95.131.213:10080/johnv/webdemos/hello.html

# Hello, World Wide Web!

<div1>

# HTTP Request/Response Cycle

- **REQUEST (Client) :**
  - ▶ User types URL in browser
    - `http://www.mydomain.com/index.html`
  - ▶ Browser connects to server and requests file
- **RESPONSE (Server):**
  - ▶ Apache server on `www.mydomain.com` listens for requests on port 80
  - ▶ Looks for `index.html` in web folder
  - ▶ If found, Apache retrieves file and sends it back to browser
- **Done!**
  - ▶ Connection is dropped

# Where Are Web Files Stored?

- In the “Document Root”:
  - ▶ IFS Folder
  - ▶ For Zend Server, doc root = `/www/zendsvr6/htdocs`
- Hello.html
  - ▶ IFS path: `/www/zendsvr6/htdocs/hello.html`
  - ▶ URL: `http://myibmi:10080/hello.html`
- Doc Root can have subfolders:
  - ▶ IFS path: `/www/zendsvr6/htdocs/ecommm/login.php`
  - ▶ URL: `http://myibmi:10080/ecommm/login.php`



# Basic Formatting - Lorem Ipsum

- Demo of unstyled HTML
  - ▶ look at source code
- Then we will add CSS



CSS

# Styling with CSS

- CSS = Cascading Style Sheets
- Extension to HTML as of HTML v 4
- Allows fine-grained control of visual elements on a page
- Simple, intuitive syntax

# CSS Syntax

```
selector {  
    property: value;  
    property: value;  
    ...  
}
```

- **selector**: identifies a part of the document to be styled  
HTML tag name, Class name, or a Unique ID
- **property**: A specific presentation attribute to be styled  
color, font-weight, border attributes, visibility
- **value**: How the presentation attribute should be styled  
color: red;  
font-weight: bold;  
border: 2px solid blue;

# CSS Style Sheet Example

```
1 body {  
2     font-family: arial, verdana, sans-serif;  
3     font-size: 12pt;  
4 }  
5  
6 h1, h2, h3 {  
7     color: #2E529C;  
8     font-family: verdana;  
9 }  
10 .error {  
11     color: red;  
12     background-color: yellow  
13 }  
14 p.big {  
15     font-size: 16pt;  
16 }
```

# Examples of CSS Selectors

- **HTML Tag Name:**

CSS: `BODY { font: arial; font-size: 12pt; color: navy }`

- *Can use any HTML tag name*
- *Applies to all occurrences of the tag throughout a document*

- **Class Name - precede with period (.) :**

CSS: `.error { color: red; font-weight: bold }`

HTML: `<p class="error">Invalid email address</p>`

- *Can specify the same class on many different HTML tags*

- **Unique ID - precede with hash (#):**

CSS: `#shipto { visibility: hidden }`

HTML: `<div id="shipto"> <table>... </div>`

- *ID name should only occur once in HTML document*

# Where Can Styles Be Defined?

- Inside a single HTML element

```
<table style="border:none; color:blue">
```

- Inside the <head> element of an HTML page

```
<head>
```

```
  <style type="text/css">
```

```
    table { border:none; color:blue }
```

```
  </style>
```

```
</head>
```

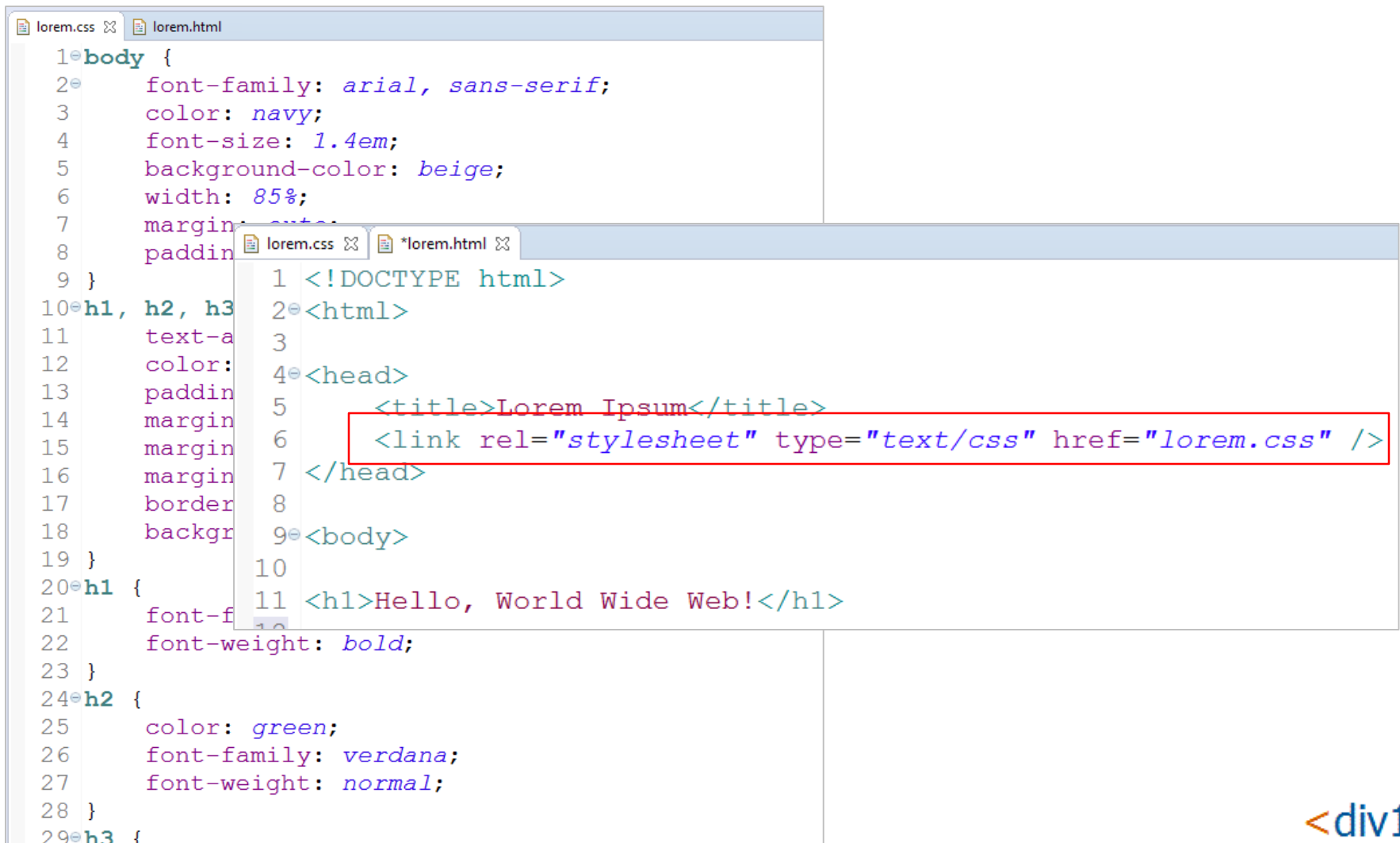
- In an external CSS file

```
<head>
```

```
  <link rel="stylesheet" type="text/css" href="siteStyle.css" />
```

```
</head>
```

# Adding External Style Sheet to lorem.html



The image shows a code editor with two files open: `lorem.css` and `lorem.html`. The `lorem.css` file contains the following CSS rules:

```
1 body {
2   font-family: arial, sans-serif;
3   color: navy;
4   font-size: 1.4em;
5   background-color: beige;
6   width: 85%;
7   margin: auto;
8   padding: 10px;
9 }
10 h1, h2, h3 {
11   text-align: center;
12   color: green;
13   padding: 10px;
14   margin: 10px 0;
15   border: 1px solid black;
16   background-color: #f0f0f0;
17 }
18 h1 {
19   font-size: 2em;
20   font-weight: bold;
21 }
22 h2 {
23   color: green;
24   font-family: verdana;
25   font-weight: normal;
26 }
27 h3 {
```

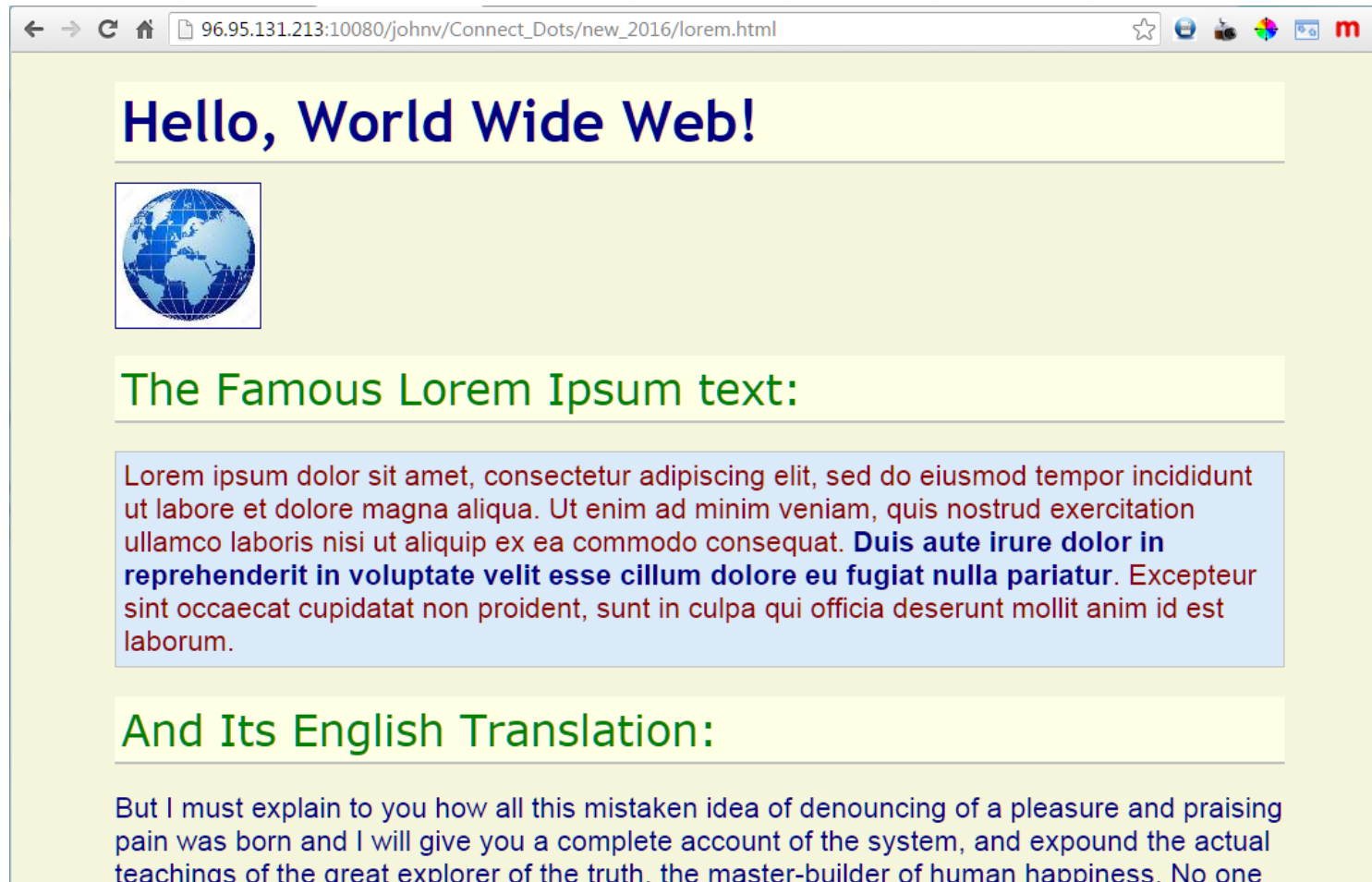
The `lorem.html` file contains the following HTML code:

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <title>Lorem Ipsum</title>
6   <link rel="stylesheet" type="text/css" href="lorem.css" />
7 </head>
8
9 <body>
10
11 <h1>Hello, World Wide Web!</h1>
```

The `<link rel="stylesheet" type="text/css" href="lorem.css" />` line in the `lorem.html` file is highlighted with a red box, indicating the step of linking the external CSS file.



# Styled HTML using lorem.css style sheet



# PHP



# HTTP Request/Response - *PHP File*

- Client requests file `myApp.php` from web server
- Apache sees `‘.php’` file request
  - ▶ File is retrieved and *handed to PHP processor*
    - PHP file may *combine HTML with embedded PHP code*.
    - Embedded PHP code is executed, which *may retrieve information from database*, and merge database content with HTML
- Apache receives document (HTML) back from PHP
- Apache sends HTML back to browser
- Done!

# Simple PHP - Dynamic Content

```
<!DOCTYPE html>
<html>
<head>
<title>Hello world</title>
<link rel="stylesheet" type="text/css" href="lorem.css" />
</head>

<body>
<h2>Hello world wide web!</h2>

<p class="box">
<?php
echo 'This is a constant string.<br>';
$dateFormatted = date('D M d, Y \a\t g:i:s A');
echo "The current date and time is <b>$dateFormatted</b>.";
?>
</p>
</body>
</html>
```

- **PHP code block**

1. Processing instructions
2. Use **echo** to add dynamic content to HTML

# PHP Code Block details

```
<?php
echo 'This is a constant string.<br>';
$dateFormatted = date('D M d, Y \a\t g:i:s A');
echo "The current date and time is <b>$dateFormatted</b>.";
?>
```

- All php code blocks are surrounded by `<?php` and `?>`
- PHP code will never be seen in the browser
- Only output from an `echo` or `print` statement will be seen in the browser (and a few other functions).
- Variables all start with '\$'
- Rich string handling capabilities
  - ▶ variable interpolation

# Arrays in PHP

- **Many features of PHP implemented as arrays**
  - ▶ Over 60 array handling functions
  - ▶ Very powerful aspect of PHP
- **Two types of arrays:**
  - ▶ **Numeric Array:**
    - index is an integer
    - starts at zero
  - ▶ **Associative arrays:**
    - index is character string
    - “key => value” lists

# Numeric vs. Associative Arrays

## Numeric array (zero-based):

```
$fruit = array('apples', 'oranges', 'bananas');  
echo $fruit[0]; // apples  
echo $fruit[2]; // bananas  
$fruit[100] = 'grapes';  
$fruit[] = 'pears'; // assigned to $fruit[101]
```

## Associative array (character index):

```
$states = array(  
    'CT' => 'Connecticut',  
    'RI' => 'Rhode Island',  
    'MA' => 'Massachusetts'  
);  
echo $states['RI']; // Rhode Island  
$states['VT'] = 'Vermont'; // add new element
```

# Other features of PHP arrays

- **Multi-dimensional arrays**
  - ▶ any depth
- **Mixed data types in one array**
  - ▶ any combination of data types (including arrays - see above)
- **Numeric and Associative keys in same array**
- **Can add elements at run time**
  - ▶ arrays can grow infinitely
- **Useful for passing multiple values in/out of functions**



# PHP Database Access

*List all records from DB table:*

```
$conn = db2_connect ( "*LOCAL", "PHPUSER", "PSWD1" );
```

```
$query = "SELECT * FROM PHPTEST.MEMBERSHIP";
```

```
$stmt = db2_prepare( $conn, $query );
```

```
db2_execute( $stmt );
```



*Returns associative array:*

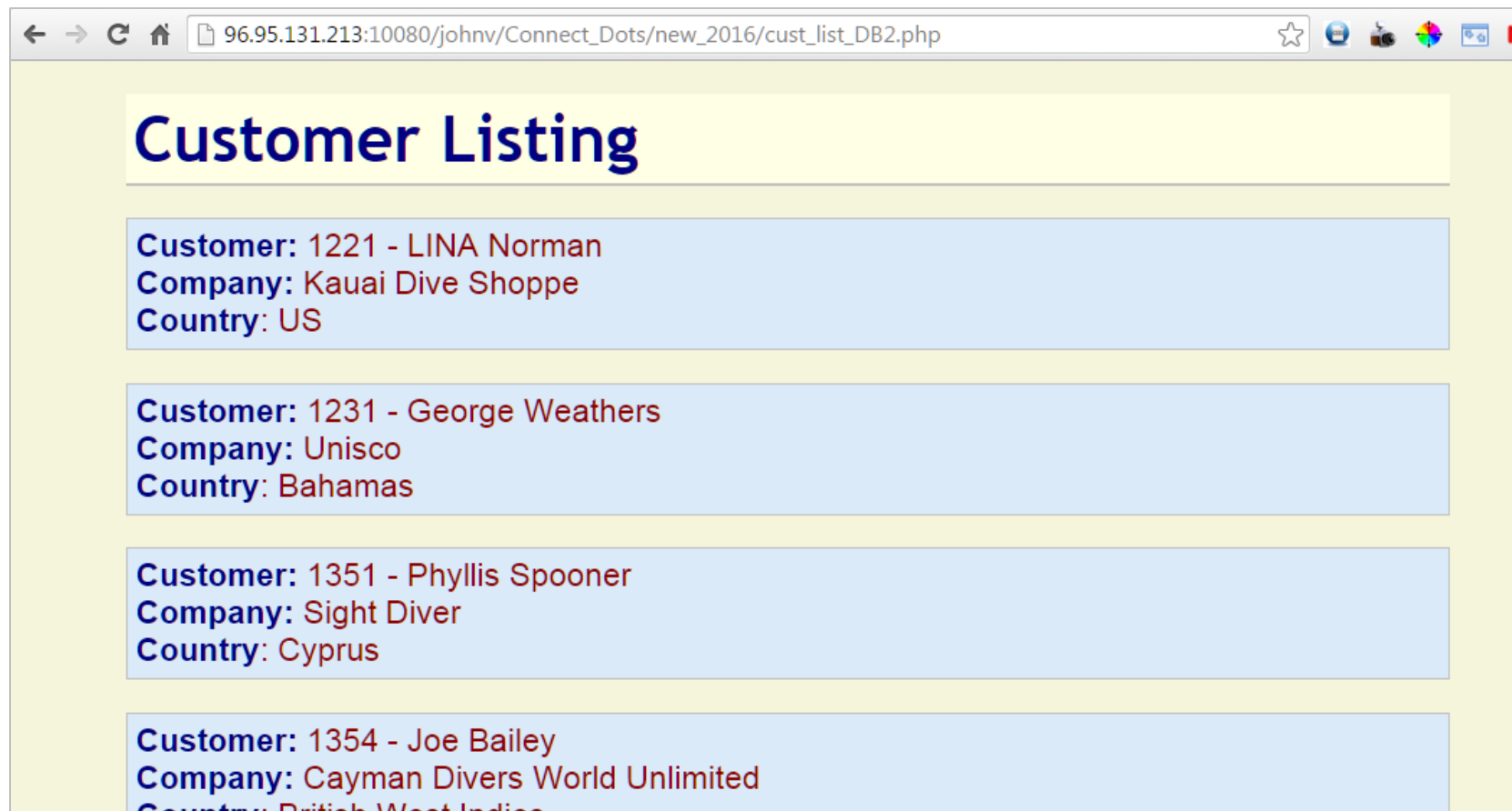
`$row['column-name'] => $row['column-value'];`

```
while ( $row = db2_fetch_assoc( $stmt ) ) {  
    $memberId = $row['MEMBERID'];  
    $name = "{$row['FIRST_NAME']} {$row['LAST_NAME']}";  
    echo "Member ID $memberId; $name<br>";  
}
```

```
db2_close ( $conn );
```

# Customer Listing PHP (using <p> tags)

cust\_list\_DB2.php:



# HTML Tables

Let's change the listing to show fields in a grid of rows and columns.

**<table>** - Defines entire table

**<tr>** - One for each table row

**<td>** - Table data - One for each column (cell) in each row

*Tables can be nested – can start a new table within a <td>*

```
<table>
  <tr>
    <td>Col 1</td> <td>Col 2</td> <td>Col 3</td>
  </tr>
  <tr>
    <td>Col 1</td> <td>Col 2</td> <td>Col 3</td>
  </tr>
</table>
```

# Customer Listing using <table>

- No styling

← → ↻ 🏠 96.95.131.213:10080/johnv/Connect\_Dots/new\_2016/cust\_list\_DB2\_table.php ☆ 🌐 🖨️ 🎨 📄 📧

## Customer Listing

Cust Num	Customer Name	Company	Address	Country	Phone
1221	LINA Norman	Kauai Dive Shoppe	4-976 Sugarloaf Hwy Suite 103 Kapaa Kauai , HI 94766-1234	US	808-555-0269
1231	George Weathers	Unisco	PO Box Z-547 Freeport ,	Bahamas	809-555-3915
1351	Phyllis Spooner	Sight Diver	1 Neptune Lane Kato Paphos ,	Cyprus	357-6-876708
1354	Joe Bailey	Cayman Divers World Unlimited	PO Box 541 Grand Cayman ,	British West Indies	011-5-697044
1356	Chris Thomas	Tom Sawyer Diving Centre	632-1 Third Frydenhoj Christiansted , St. Croix 00820	US Virgin Islands	504-798-3022
1380	Ernest Barrett	Blue Jack Aqua Center	23-738 Paddington Lane Suite 310	TTX	101 600 7623

# Customer Listing HTML (using <table>)

```

<body>
  <h2>Customer Listing</h2>
  <table border=1>
    <tr>
      <th width="8%">Cust Num</th>
      <th width="12%">Customer Name</th>
      <th width="20%">Company</th>
      <th width="20%">Address</th>
      <th width="12%">Country</th>
      <th width="10%">Phone</th>
    </tr>
    <tr>
      <td class="center">1221</td>
      <td class="left">LINA Norman</td>
      <td class="left">Kauai Dive Shoppe</td>
      <td class="left">4-976 Sugarloaf Hwy</td>
      <td class="left">US</td>
      <td class="center">808-555-0269</td>
    </tr>
    <tr>
      <td class="center">1231</td>
      <td class="left">George Weathers</td>
      <td class="left">Unisco</td>
      <td class="left">PO Box Z-547 <br>Fr

```

- Headings row

- Data rows

- ▶ Create one template
- ▶ Repeat for each row from database

# Repeating Table Rows in PHP

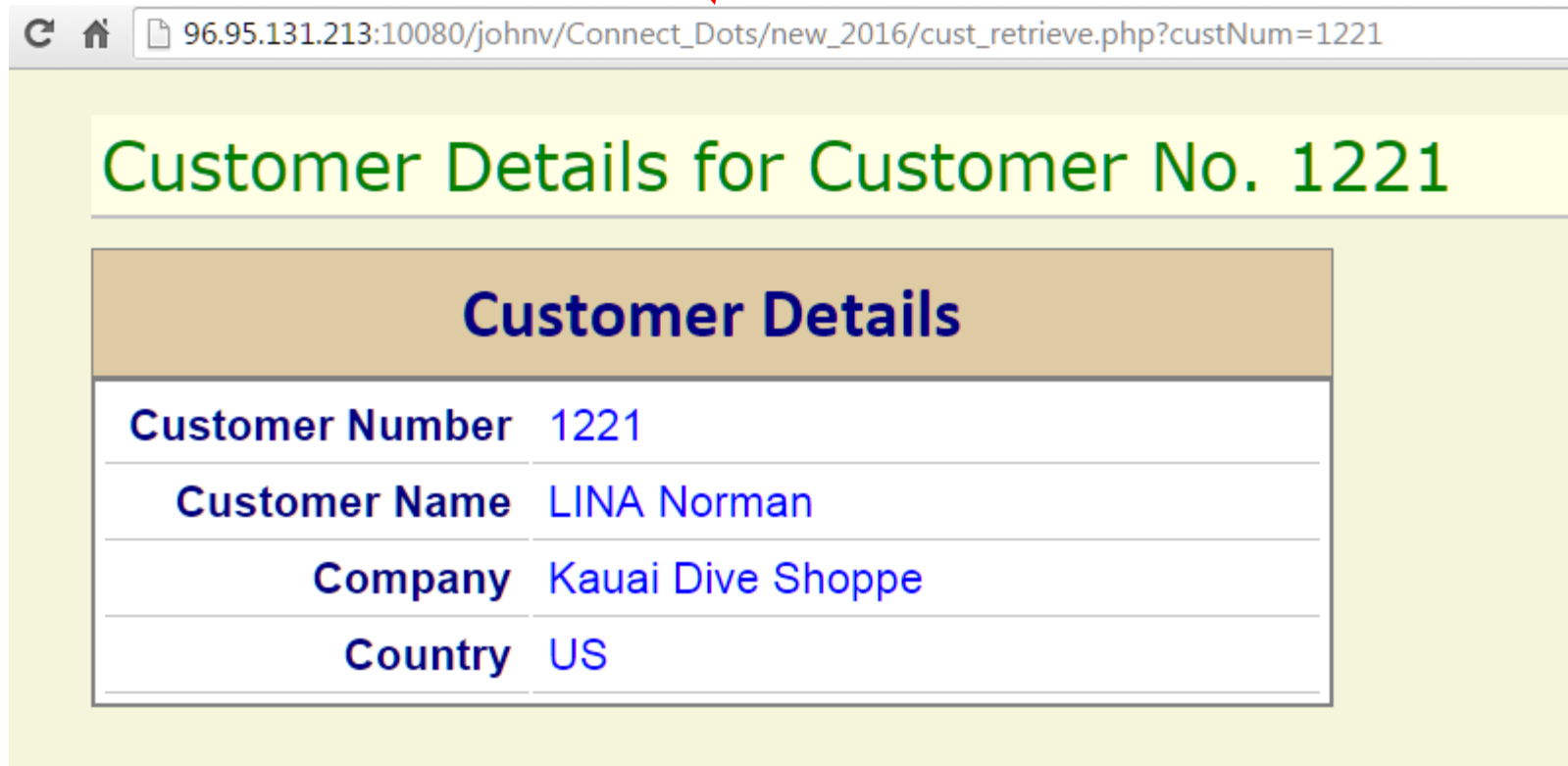
- Uses the “here-doc” string syntax

<<<DATA\_ROW ... DATA\_ROW;

```
while ($row = db2_fetch_assoc($stmt)) {
    $address = formatAddress($row);
    echo <<<DATA_ROW
    <tr>
        <td class="center">{$row['CUST_ID']}</td>
        <td class="left">{$row['FIRSTNAME']}
            {$row['LASTNAME']}</td>
        <td class="left">{$row['COMPANY']}</td>
        <td class="left">$address</td>
        <td class="left">{$row['COUNTRY']}</td>
        <td class="center">{$row['PHONE']}</td>
    </tr>
    DATA_ROW;
} // end of while loop
```

# Customer Detail Page

[http://96.95.131.213:10080/johnv/Connect\\_Dots/new\\_2016/  
cust\\_retrieve.php?custNum=1221](http://96.95.131.213:10080/johnv/Connect_Dots/new_2016/cust_retrieve.php?custNum=1221)



96.95.131.213:10080/johnv/Connect\_Dots/new\_2016/cust\_retrieve.php?custNum=1221

## Customer Details for Customer No. 1221

Customer Details	
Customer Number	1221
Customer Name	LINA Norman
Company	Kauai Dive Shoppe
Country	US

# Anatomy of a Request URL

<http://www.mydomain.com/pubapps/myScript.php?cust=10357>

<a href="http://www.mydomain.com/">http://www.mydomain.com/</a>	Protocol // domain
<a href="#">pubapps/</a>	Path to the script (relative to the web root folder)
<a href="#">myScript.php</a>	Script file name
<a href="#">?</a>	Delimiter (separates script name from the query string)
<a href="#">cust=10357</a>	Query string (i.e. parameters the script can access)



# Query String - Multiple Parameters

Name/Value Pairs, Separated by '&'

```
script.php?name1=value1&name2=value2...
```

```
http://www.myComp.com/  
myScript.php?cust=12345&action=update
```

PHP parses query string into \$\_GET array

```
$custNo = $_GET['cust']; // 12345  
$action = $_GET['action']; // update
```

# Form Tag

```
<form action="myScript.php" method="post">  
  <input> tags...  
</form>
```

---

**<form>** - defines a group of input fields  
Makes user input easier than typing query string in URL

- **action** attribute
  - tells what PHP script will receive input values
- **method** attribute
  - defines how values are delivered to action script
  - `method="get"` - send inputs on URL, as a query string
    - *Limited data length*
  - `method="post"` - send inputs with HTTP headers
    - *Allows unlimited data to be sent*
    - *Typically used when updating the server*

# Form Example

```
<form method="get" action="form_process.php">
  Enter your name:
  <input type="text" name="nameFld" value="John" />
  <br>
  <input type="submit">
</form>
```

Looks like this in browser:

Enter your name:

Clicking Submit button creates request for:  
`/mydomain.com/form_process.php?nameFld=John`

```
<?php
  $name = $_REQUEST['nameFld'];
  echo "Hello $name! <br>";
?>
```

⇒ PHP can read form inputs  
via the `$_REQUEST` array

# HTML 4 <input> Types

## HTML Input Field Types

```
<input type="text">
```

```
<input type="radio">:
```

☐ Choice 1 ☒ Choice 2 ☐ Choice 3

```
<input type="checkbox">:
```

☒ Choice 1 ☐ Choice 2 ☒ Choice 3

```
<select>:
```

```
<textarea>:
```

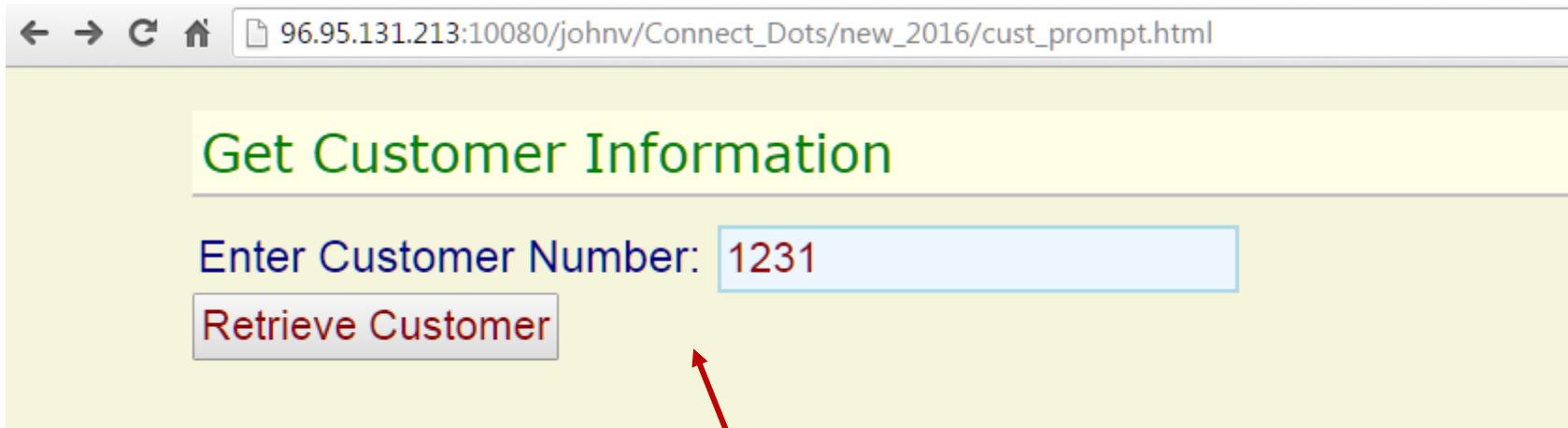
Line 1  
Line 2

```
<input type="password">:
```

```
<input type="hidden">:
```

```
<input type="submit">:
```

# Customer Prompt Form



← → ↻ 🏠 96.95.131.213:10080/johnv/Connect\_Dots/new\_2016/cust\_prompt.html

## Get Customer Information

Enter Customer Number:

```
<body>
  <h2>Get Customer Information</h2>

  <form action="cust_retrieve.php" method="get">
    <label for="custNum">Enter Customer Number:</label>
    <input type="text" name="custNum" />
    <br>
    <input type="submit" value="Retrieve Customer" />
  </form>
</body>
```

# JavaScript



# What is JavaScript?

- It isn't Java! (but similar syntax, based on C).
- Runs on the client-side (usually) i.e. in browser
- Scripting language for web browsers
- All browsers have built-in JavaScript interpreter – you don't buy it or install it.
- Interpreted at run-time (as page loads)
- JavaScript code is downloaded with the HTML document, but only runs in the browser.

# JavaScript Sample

```
<html>
<head>
<title>JavaScript Example</title>

<script>
    function checkCustNo() {
        if (document.myForm.custNo.value == '') {
            alert('Customer number is required.');
        } else {
            alert('Customer No. entered was: ' +
                document.myForm.custNo.value);
        }
    }
</script>
</head>
```



# What Can JavaScript Do?

- Validate input data
- Handle events
  - e.g.: mouse clicks or cursor movement into/out of fields
- Control Dynamic HTML
  - make things move around, appear and disappear
- Read and alter document elements, including HTML tags and CSS attributes
- Open & close windows, and communicate between windows.
- *Key technology in Ajax and Web 2.0 applications*

# Where Is JavaScript Coded in HTML?

- Can be inserted just about anywhere, but must be enclosed in `<script> </script>` tag
- Typically, functions are defined in `<head>` section.
- Can also be included as external file
  - Function libraries, Frameworks
  - Linked to document in `<head>` section
- Can also be included as action in certain HTML tags:

```
<form action="checkInputs();">
```

```
<button onclick="alert('You clicked me.')">
```

```
<a href="javascript:openHelpWindow();">
```

# Current State of Web Development



# Present/Future State of Web Development

- **Mobile is King**
  - HTML 5 / CSS 3
- **Responsive Design**
  - apps on multiple devices
  - = more CSS
- **Ajax - Asynchronous JavaScript and XML**
  - Service Oriented Architecture (SOA)
- **Application Architecture**
  - Shift from server side (PHP) to Client side (JavaScript)
  - Framework du Jour (expecially JavaScript)

## More Information

- Email me if you would like the source code:  
[johnv@div1sys.com](mailto:johnv@div1sys.com)
- Attend the hands-on Labs for more details!

*Thank you!*

# Contact Info

**John Valance**  
**Division 1 Systems**

**johnv@div1sys.com**

**802-355-4024**

**<http://www.div1sys.com>**

