

Connecting the Dots

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

John Valance
division 1 systems
johnv@div1sys.com

`<div1>`

www.div1sys.com

About John Valance



- 30+ years IBM midrange experience (S/38 thru IBM i)
- 17+ years of web development experience
- Independent consultant since early 2000
- Frequent presenter at IBM i groups nationwide
- COMMON Board of Directors
- Founder and CTO of Division 1 Systems
 - ▶ Web applications for IBMi
 - ▶ Full SDLC - design, project management, coding, training, consulting
 - ▶ Extended team - Can scale up/down to meet client needs
- Relationship with Zend / RogueWave
 - ▶ Teacher, Reseller, Zend Certified Engineer

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



These are universal - part of all browser based applications

- **Server side (IBM i):**

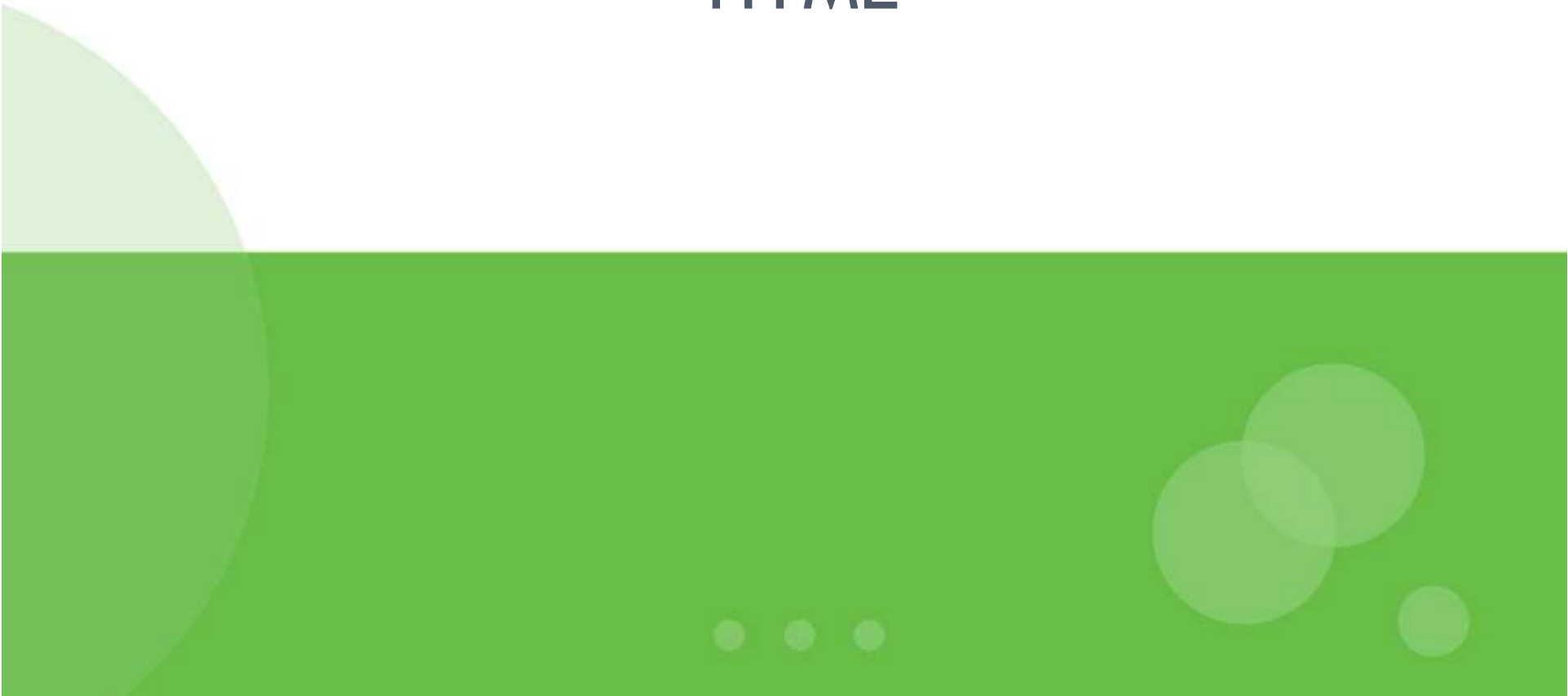
- ▶ PHP
- ▶ SQL (accessing DB2 tables)
- ▶ Possibly RPG & CL



Server side could be any languages, though SQL usually involved

- Called via stored procedures or Zend Toolkit for IBMi

HTML



HTML Sample Structure

```
<!DOCTYPE html>
<html>
  <head>
    <title>Stat
  </head>
  <body>
    <h1>Hello
  </body>
</html>
```

← This ensures HTML5

<head>section not visible to user

<html> always outer-most tag. Defines document

<body>section contains all visible contents

Looks like this in browser:



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

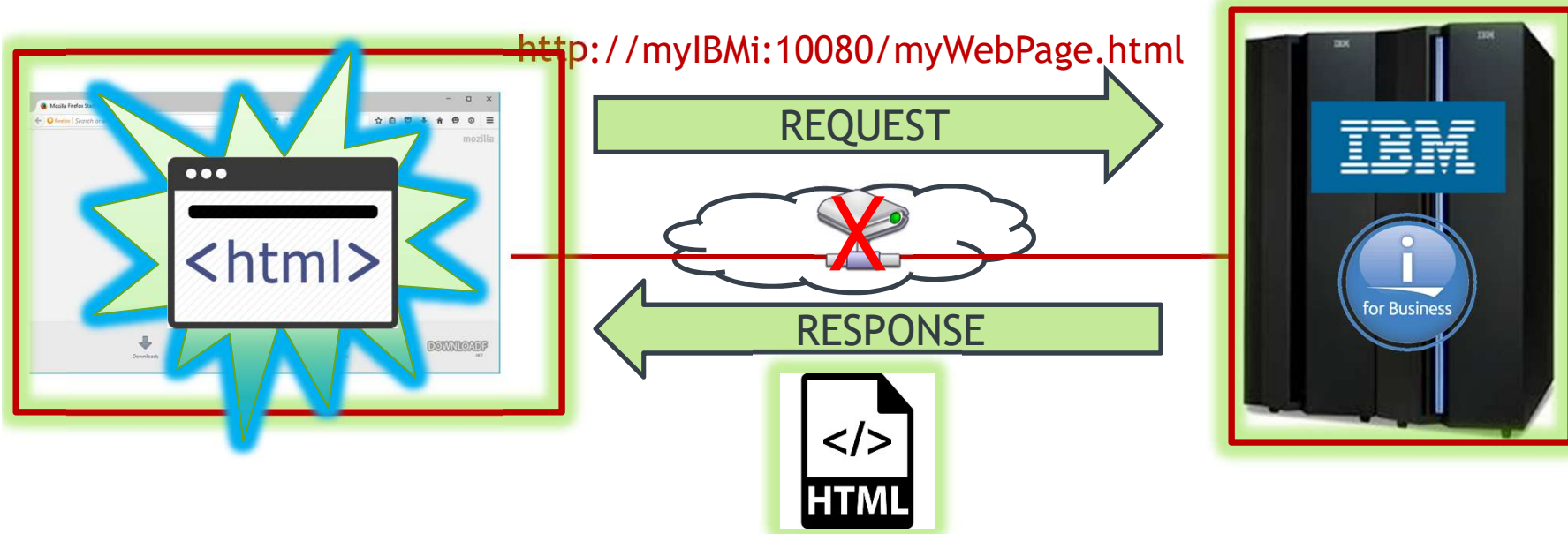
Done!

- ▶ Connection is dropped

Client (browser)

Internet (or LAN)

Server (IBM i)



Done!

- **REQUEST:** Connection is dropped
- **RESPONSE:** User types URL in browser
- **HTTP is a stateless protocol**
 - `http://www.mydomain.com/index.html`
 - Document root = `/www/zendsvr6/htdocs`
- Browser connects to server and requests files
- Apache sends file back to browser
- Apache receives request for requests on port 80
- Easy with PHP session variables

Where Are Web Files Stored?

- In the “Document Root”:
 - ▶ IFS Folder
 - ▶ For Zend Server on IBM i, 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`
- Can map doc root to a different folder in Apache config

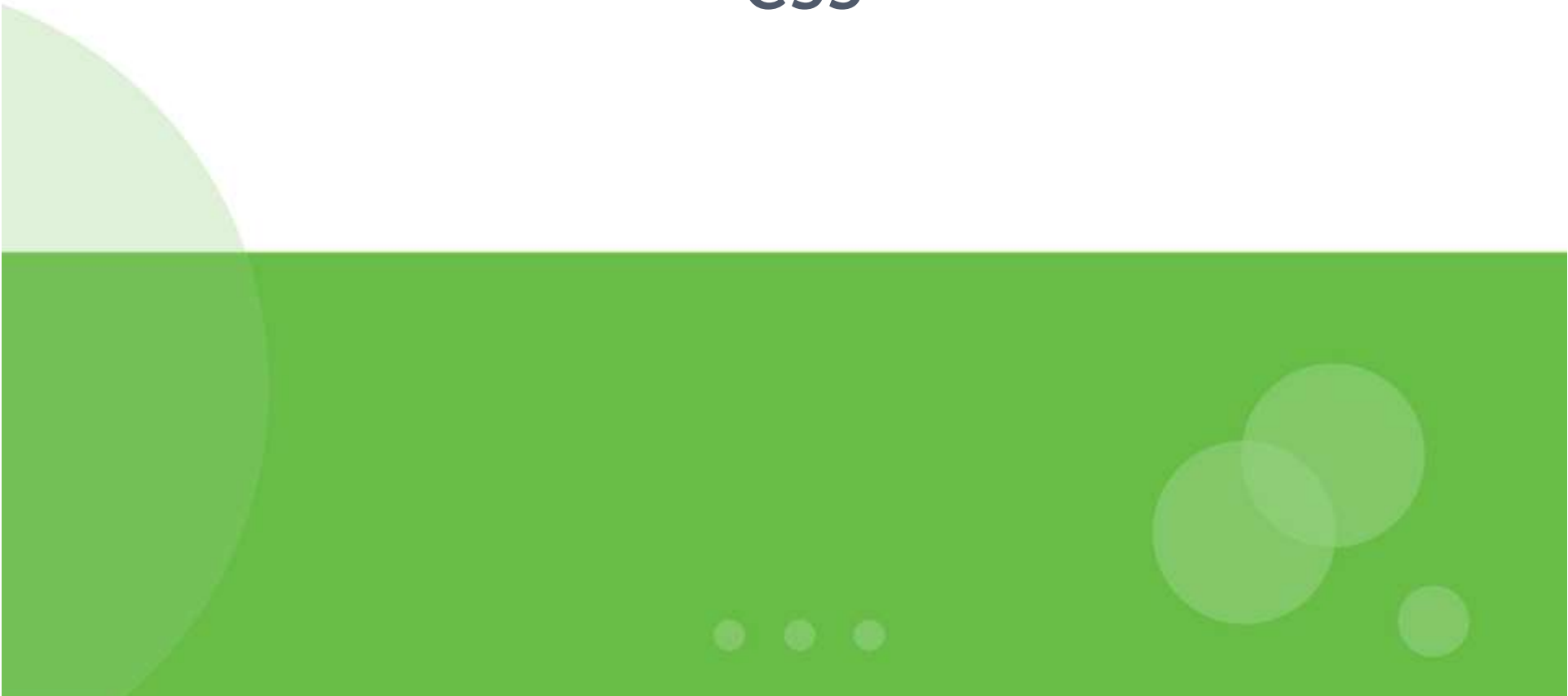
Basic Formatting - Lorem Ipsum

- 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
- Useful for default styling

- **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
- Useful for creating reusable styling rules

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

CSS: **#shipto { visibility: hidden }**

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

- ID name should only occur once in HTML document
- Useful for identifying individual elements in the document to style (or script)

Where Can Styles Be Defined?

- Inside a single HTML element

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

Applies only to this one element and its descendants

- Inside the <head> element of an HTML page

```
<head>
```

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

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

```
  </style>
```

```
</head>
```

Applies to the entire document (web page)

- In an external CSS file

```
<head>
```

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

```
</head>
```

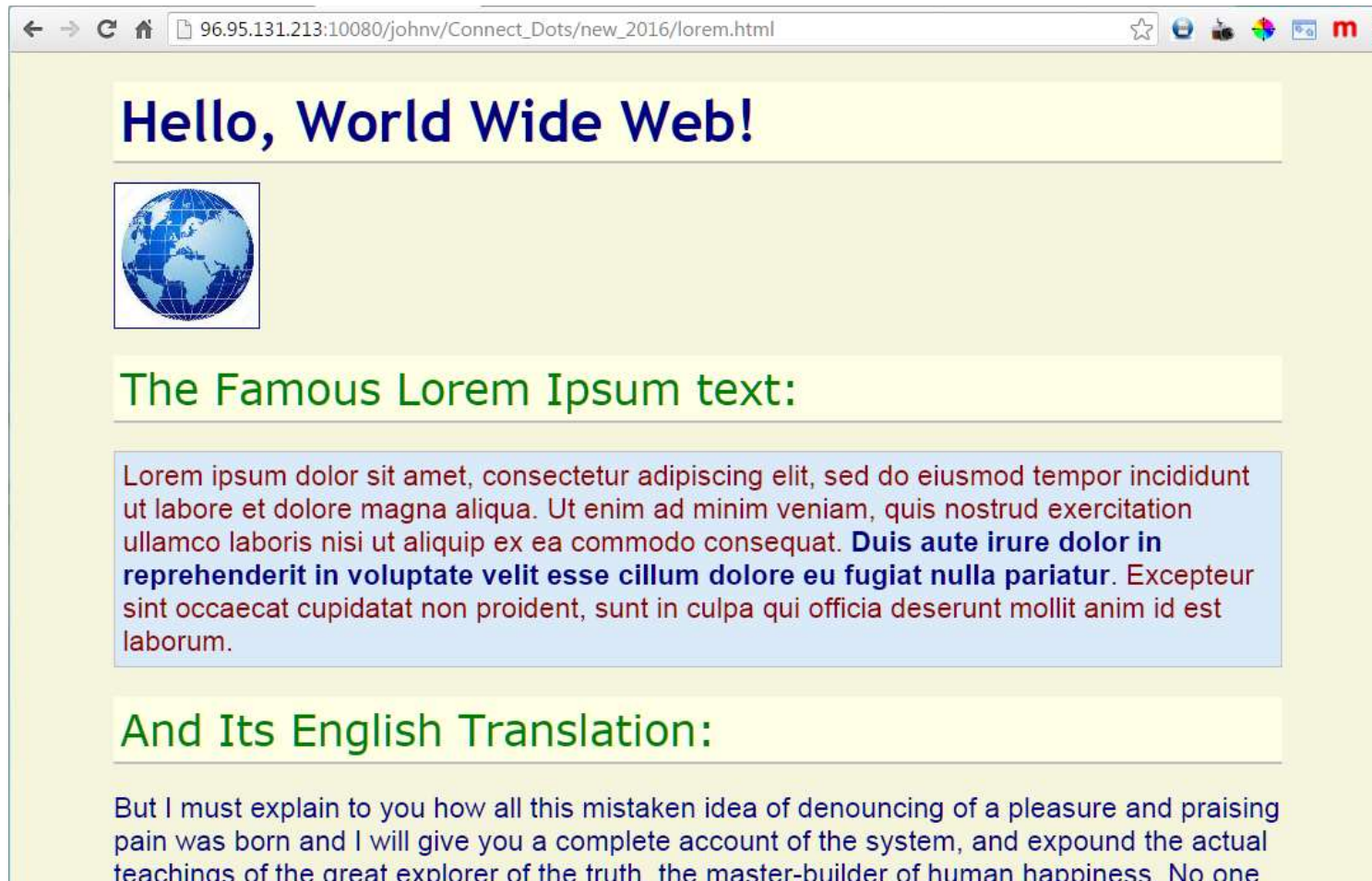
Can be used on every page of an entire site/application

Adding External Style Sheet to lorem.html

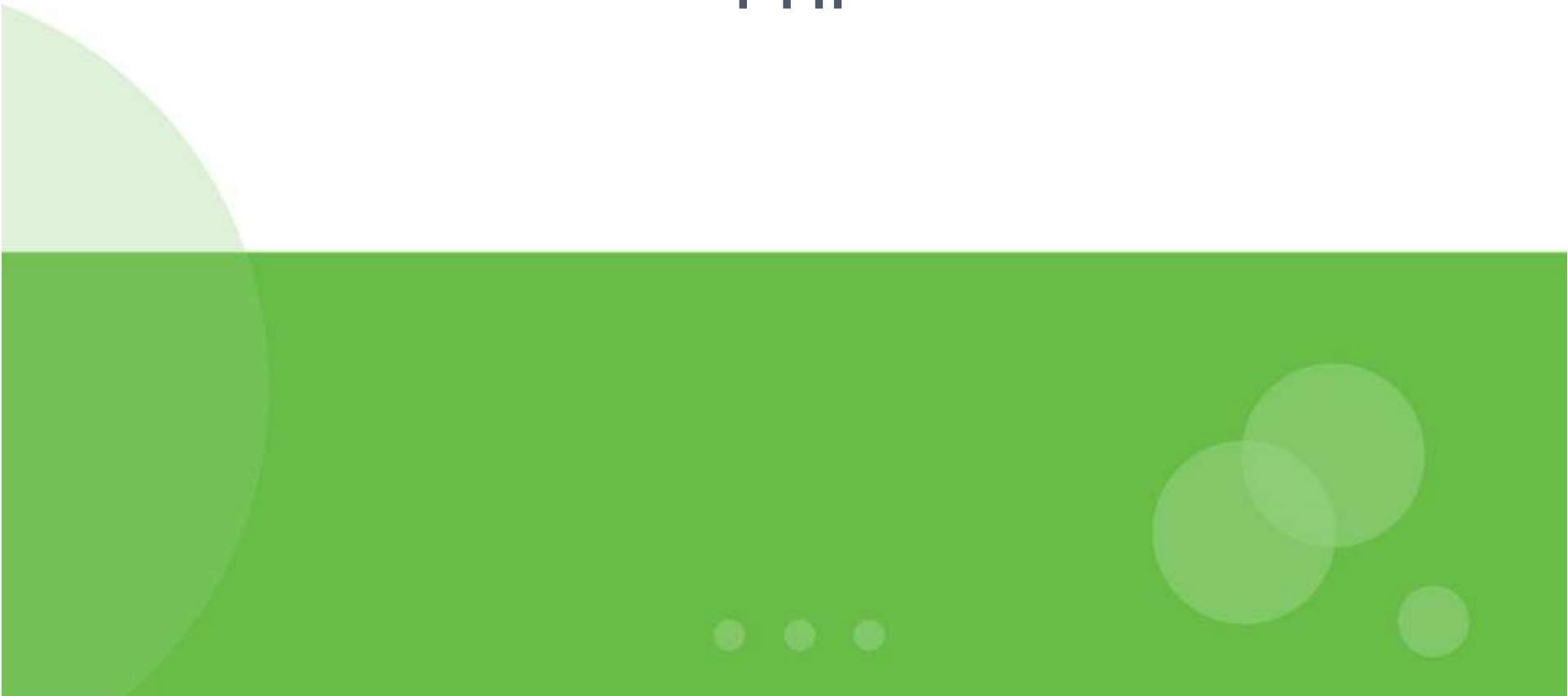
```
lorem.css x lorem.html
1 body {
2   font-family: arial, sans-serif;
3   color: navy;
4   font-size: 1.4em;
5   background-color: beige;
6   width: 85%;
7   margin: auto;
8   padding: 10px;
9 }
10 h1, h2, h3 {
11   text-align: center;
12   color: red;
13   padding: 10px;
14   margin: 10px 0;
15   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 {
28 }
29
```

```
lorem.css x *lorem.html x
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>
12
```

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 Programming



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:



The screenshot shows a web browser window with the address bar containing the URL: 96.95.131.213:10080/johnv/Connect_Dots/new_2016/cust_list_DB2.php. The page content is displayed on a light yellow background and features a title "Customer Listing" in a large, bold, blue font. Below the title, there are four light blue rectangular boxes, each containing customer information in a red font. The information is as follows:

Customer ID	Customer Name	Company	Country
1221	LINA Norman	Kauai Dive Shoppe	US
1231	George Weathers	Unisco	Bahamas
1351	Phyllis Spooner	Sight Diver	Cyprus
1354	Joe Bailey	Cayman Divers World Unlimited	British West Indies

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

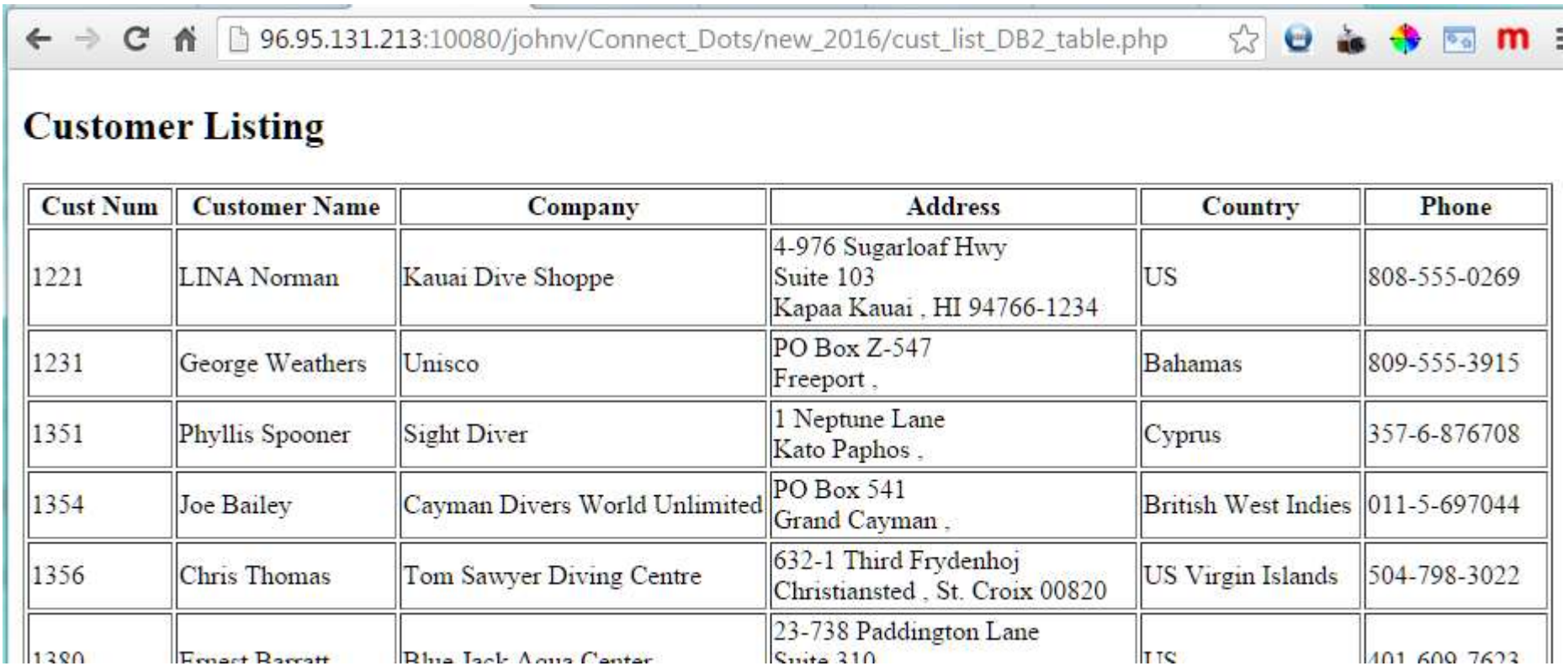
<td> - Table data - One for each column (cell) in each row

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



The screenshot shows a web browser window with the address bar displaying the URL: 96.95.131.213:10080/johnv/Connect_Dots/new_2016/cust_list_DB2_table.php. The page title is "Customer Listing". Below the title is a table with the following data:

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	ITC	401-600-7623

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

- echo the HTML using the “here-doc” string syntax

```
echo <<<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>

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

Query String - Multiple Parameters

Name/Value Pairs, Separated by ‘&’

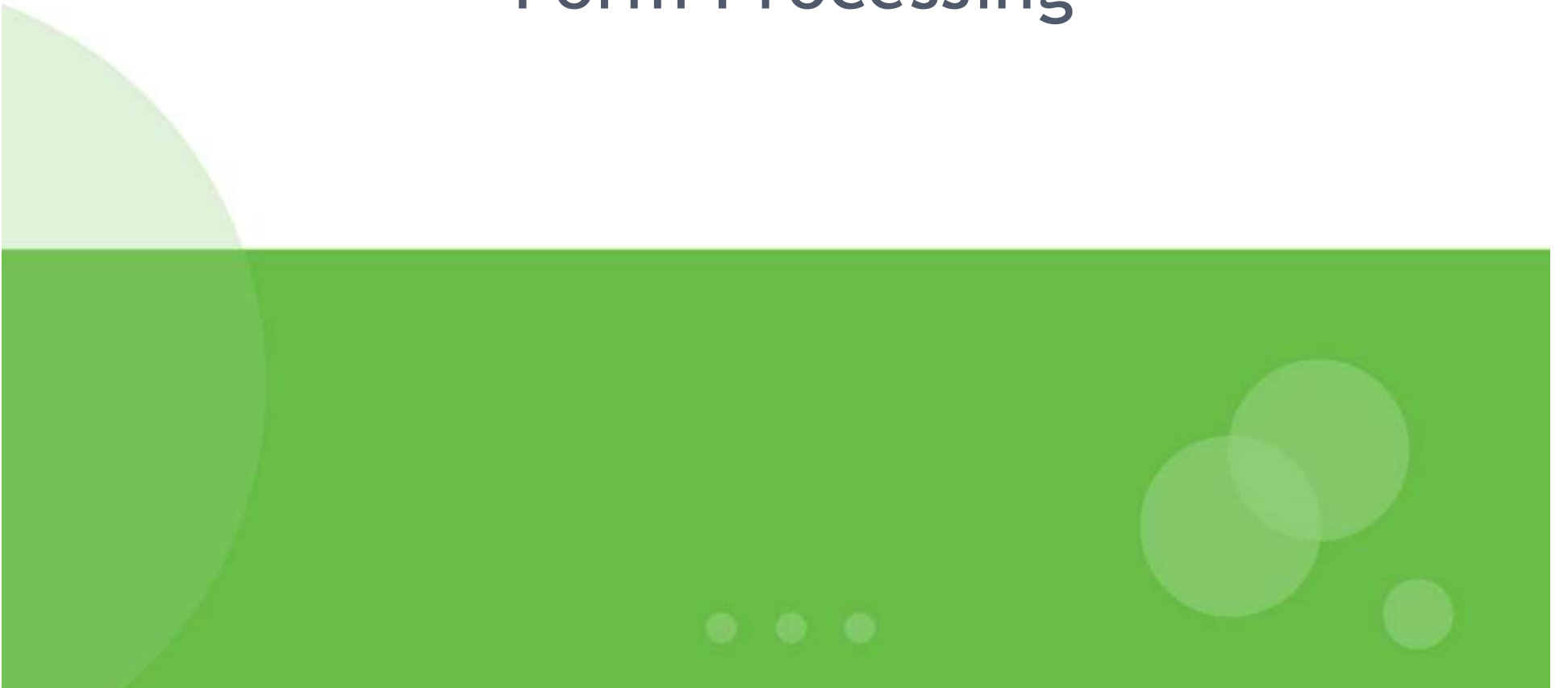
```
script.php?name1=value1&name2=value2...
```

```
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 Processing



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 in body of request
 - *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:

http://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>:
```

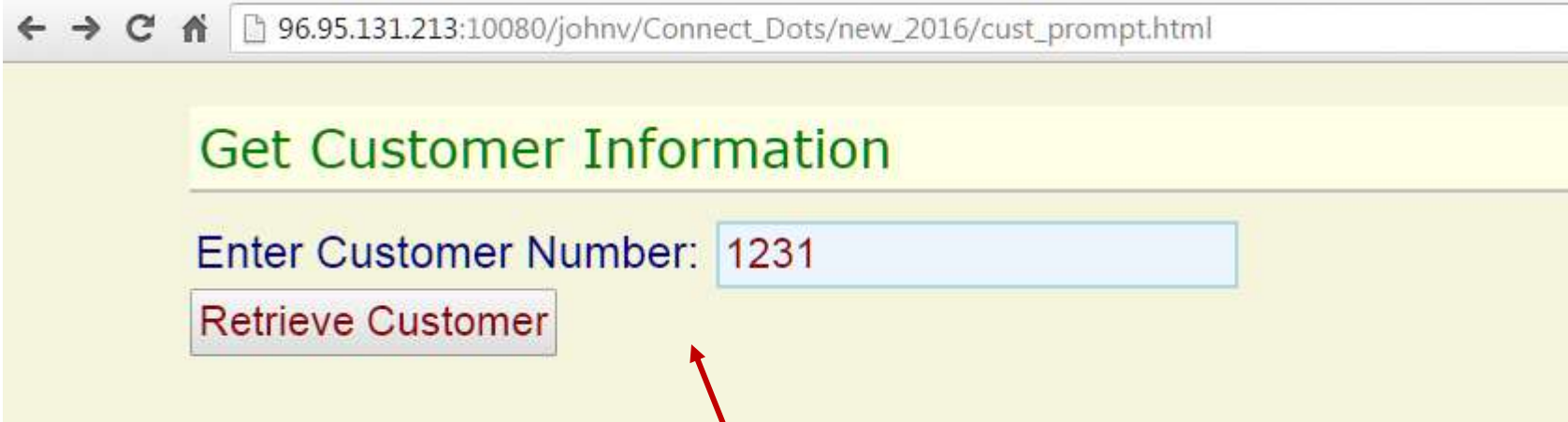


```
<input type="password">:
```

```
<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>
```


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

Session Management



Sessions and Cookies

- **HTTP is stateless**
 - ▶ Each request from client to server is unique and independent of previous requests
- **Applications need to maintain state between requests**
 - ▶ Store information about the user and application variables
 - UserID, Customer#, Order#, etc...
- **Need to simulate session state in the browser**
- **PHP has Session variables**
 - ▶ Session variables are stored on server
 - ▶ Cookie containing session ID is stored on client
 - ▶ Simply use `session_start()` function in PHP, and add values to `$_SESSION` array. They're available on subsequent requests.

Sessions are Controlled by Cookies

- **Cookies: What are they? and How do they work?**
 - ▶ Cookie is a piece of information (name/value pair)
 - Userid=JVALANCE
 - SessionID=284672ab93f98d7ce12718a4bd
 - ▶ Set by the server application
 - PHP: `setcookie('UserID', 'JVALANCE');`
 - Returned to browser on the response
 - Stored by the browser (storage location depends on browser)
 - ▶ Sent to server automatically
 - After being set, cookies are automatically sent by the browser on subsequent requests to the same server

PHP `session_start()` function

Was session cookie sent with request?

- **Yes:**

- ▶ Retrieve session ID
- ▶ Load `$_SESSION` array for session ID

- **No:**

- ▶ Generate session ID
- ▶ Initialize `$_SESSION` array
- ▶ Send session cookie on response

JavaScript



What is JavaScript?

- It isn't Java! (but similar syntax, based on C).
- Runs on the client-side (usually) i.e. in browser
 - node.js is server-side JavaScript
- 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 checkInput() {
    var custNo = document.getElementById('custNo');
    if (custNo.value == '') {
      alert('Customer number is required.');
```

```
    } else {
      document.getElementById('myForm').submit();
    }
  }
</script>
</head>
<body> <form id="myform" action="cust_retrieve.php">...
  <input id="custNo" /> <input type="button" onclick="checkInput()">
... </form>
</body></html>
```


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?

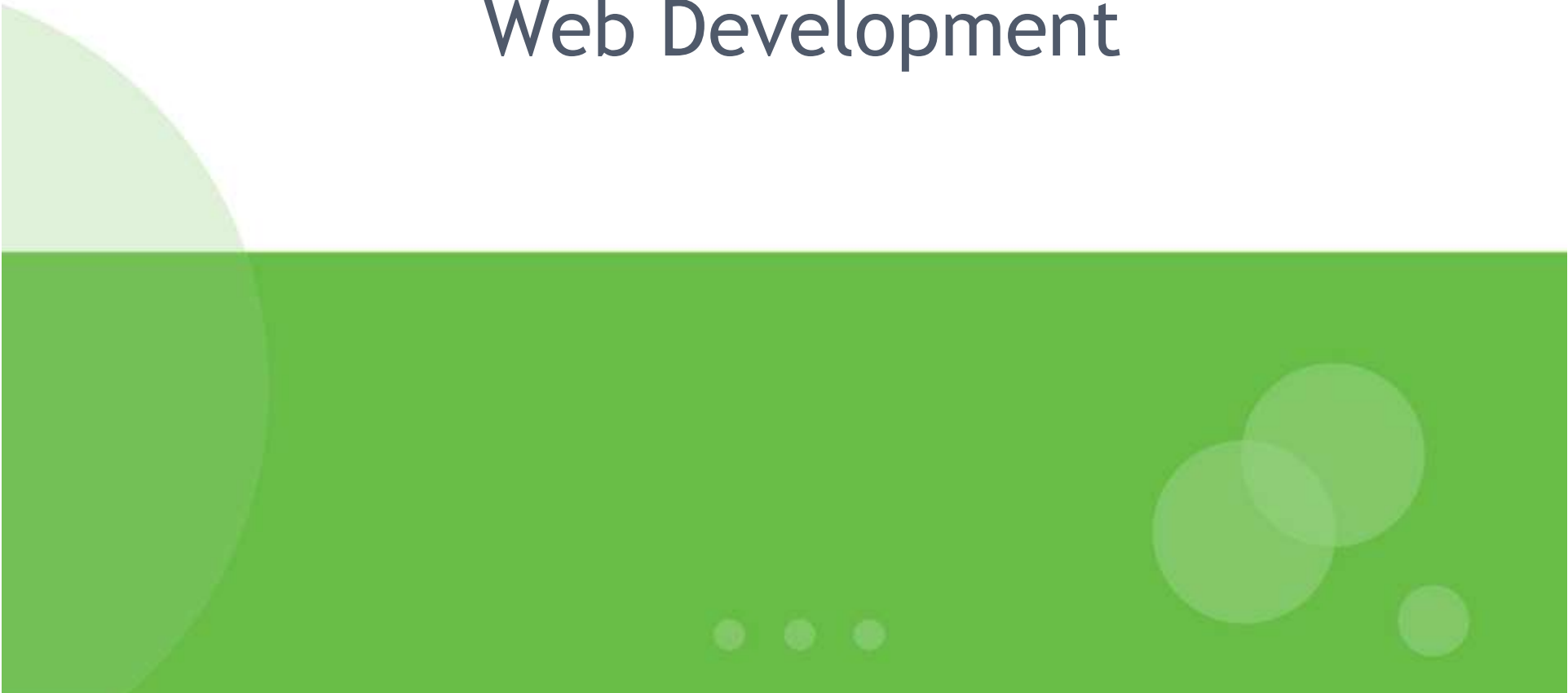
- Can be inserted just about anywhere, but must be enclosed in `<script> </script>` tag
- Typically, functions are defined in `<head>` section.
- Can also be included as external file
 - Function libraries, Frameworks
 - Linked to document in `<head>` section
- Can also be included as event handler in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

- **Application-oriented web**
 - HTML 5 / CSS 3
- **Mobile First - Responsive Design**
 - Apps on multiple devices / different orientations (landscape / portrait)
 - CSS Frameworks (Twitter Bootstrap, LESS, others...)
- **JavaScript Ascendance**
 - Ajax - JSON
 - Node.js
- **Application Architecture**
 - APIs and Service Oriented Architecture (SOA)
 - Object Oriented code base / Frameworks - many (especially JavaScript)
 - Shift from server side control (PHP) to Client side (JavaScript)
 - JavaScript on server (NodeJS)

More Information

- Email me if you would like the source code:
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>