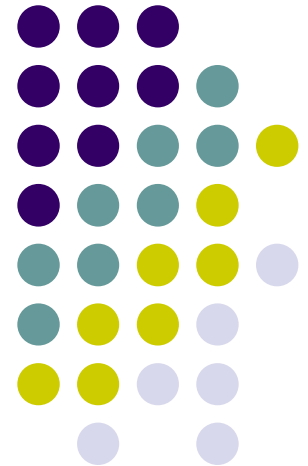
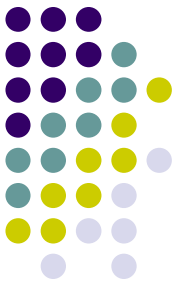


Linux
Apache
MySQL
PHP

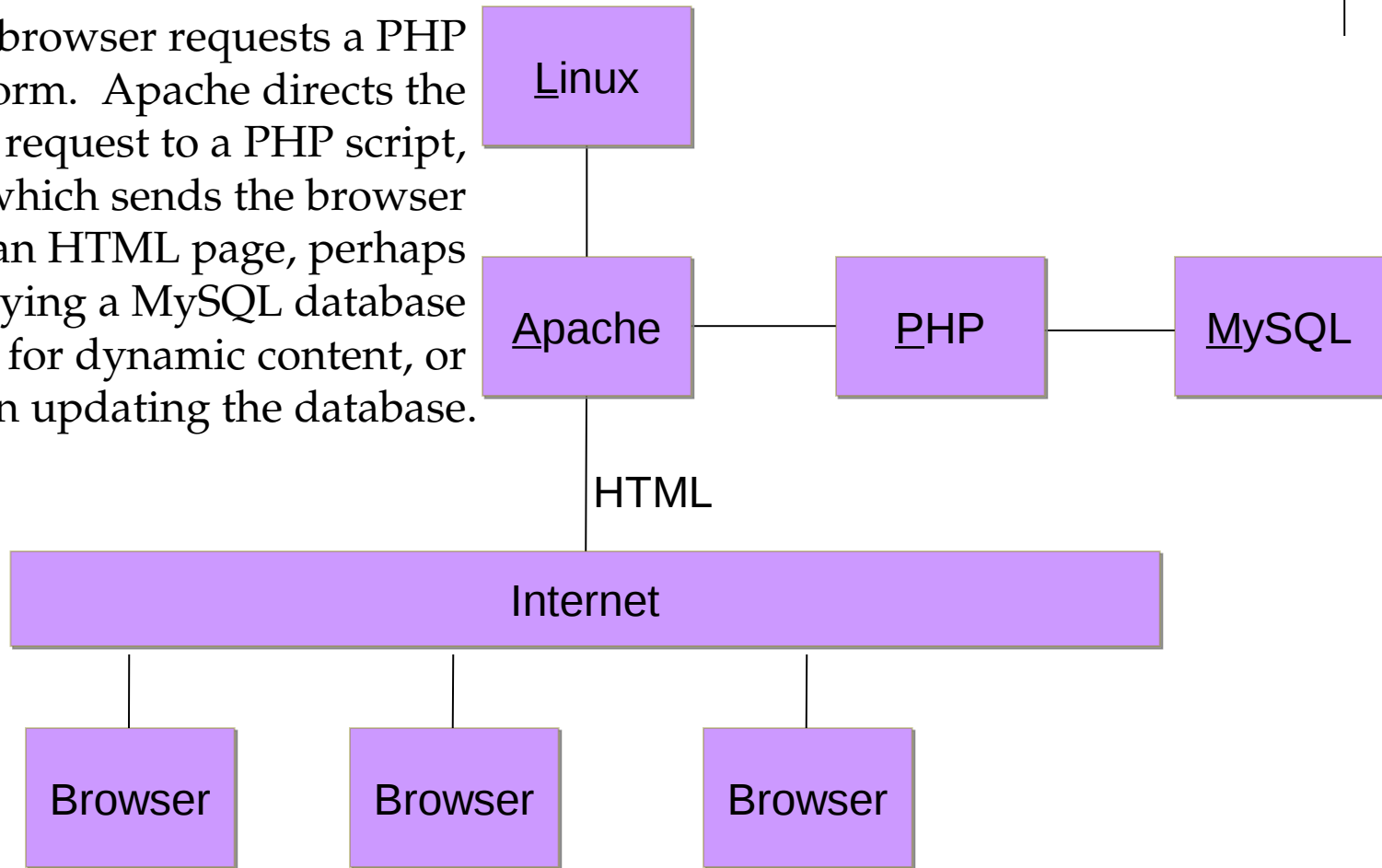
Part 2: PHP



A LAMP dynamic web server



A browser requests a PHP form. Apache directs the request to a PHP script, which sends the browser an HTML page, perhaps querying a MySQL database for dynamic content, or even updating the database.



The (free) LAMP package



- Installs
 - or updates your existing:
 - MySQL: An SQL database system
 - PHP: A programming language for web pages.
 - Apache: A web server
 - A control panel for Apache & MySQL servers.
 - PhpMyAdmin: A PHP forms interface to MySQL

Reminder: Products Table



Safari File Edit View History Bookmarks Window Help MAMP Mon 8:23:24 PM Admin

http://localhost:8888/MAMP/?language=English

MAMP Start phpInfo phpMyAdmin SQLiteManager FAQ Try MAMP PRO MAMP & MAMP PRO powered by appssolute GmbH

phpMyAdmin

Database: retrotech (4)

retrotech (4)

- customers
- Extended_Orders
- orders
- products

localhost > retrotech > products

Browse Structure SQL Search Insert Export Import Operations Empty Drop

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	Item	char(6)	ascii_general_ci		No	None		
<input type="checkbox"/>	Description	varchar(40)	ascii_general_ci		No	None		
<input type="checkbox"/>	Stock	decimal(4,0)			No	None		
<input type="checkbox"/>	Price	decimal(6,2)			No	None		

Check All / Uncheck All With selected:

Print view Relation view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After Item Go

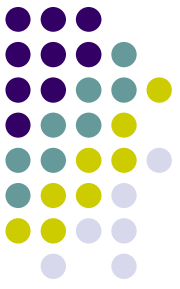
- Details...

Indexes:

Action	Keyname	Type	Unique	Packed	Field	Cardinality	Collation	Null	Comment
	PRIMARY	BTREE	Yes	No	Item	45	A		

Create an index on 1 columns Go

Space usage			Row Statistics	
Type	Usage		Statements	Value
Data	16,384	B	Format	Compact
Index	0	B	Collation	ascii_general_ci
Total	16,384	B	Creation	Apr 05, 2010 at 02:22 PM



What is HTML?

- HTML = Hyper-Text Mark-up Language
- It is used to create web pages and forms.
- HTML is rendered by a browser, e.g. *Safari*
- HTML *tags* tell the browser what to do: e.g. `<i> Italic text is bracketed by I tags </i>`



Example HTML Form

```
<html>
  <head>
    <title>Product Search by Keyword</title>
  </head>
  <form action="http://localhost:8888/show-product.php"
    method="post">
    <input type="text" name="keyword"
      value="Please type a keyword" />
    <input type="submit" value="Search" />
  </form>
</html>
```

This title will appear at the top of the browser window

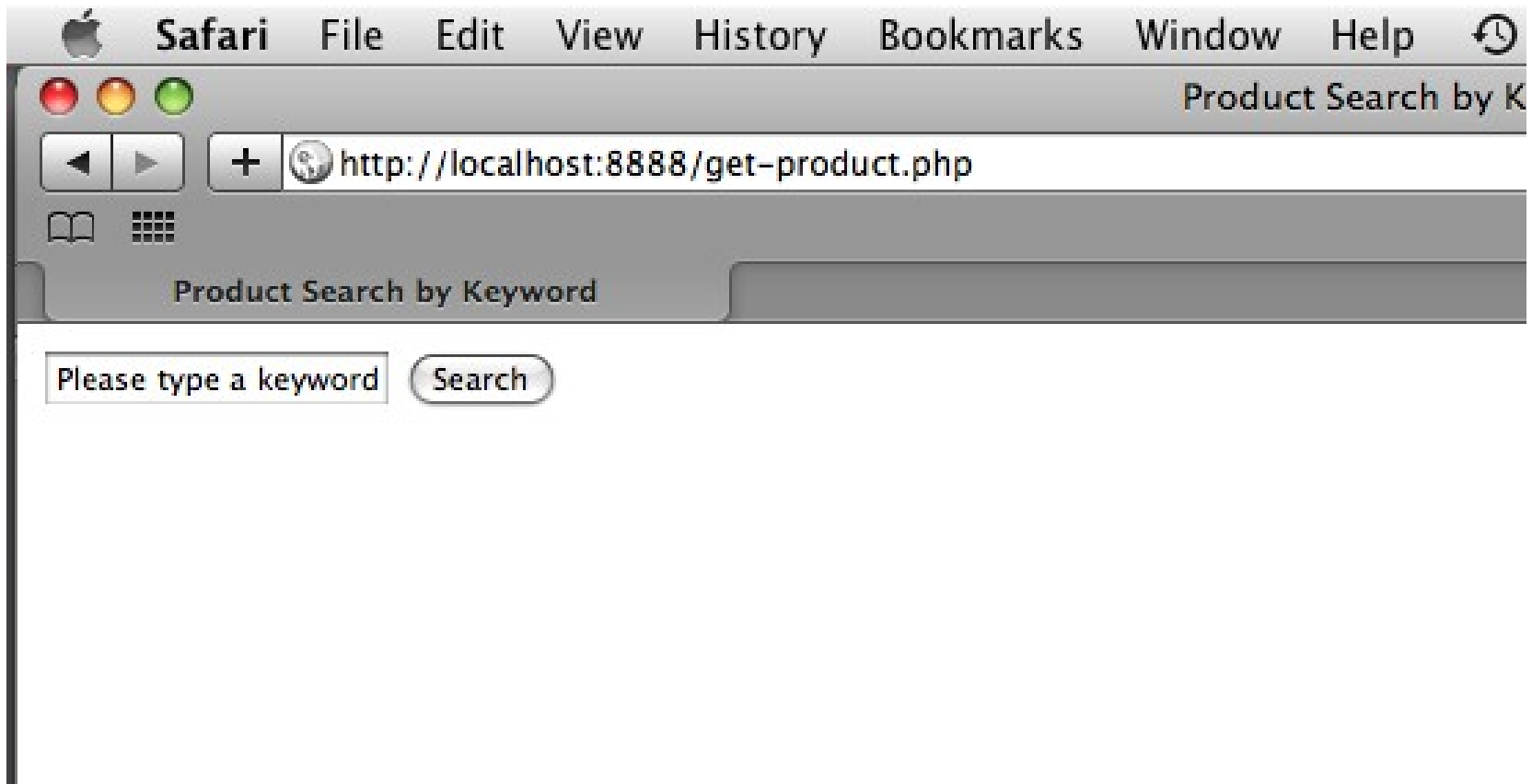
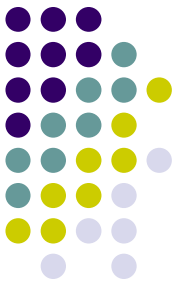
Submitting the form will open the URL below ...

... passing data using the POST method.

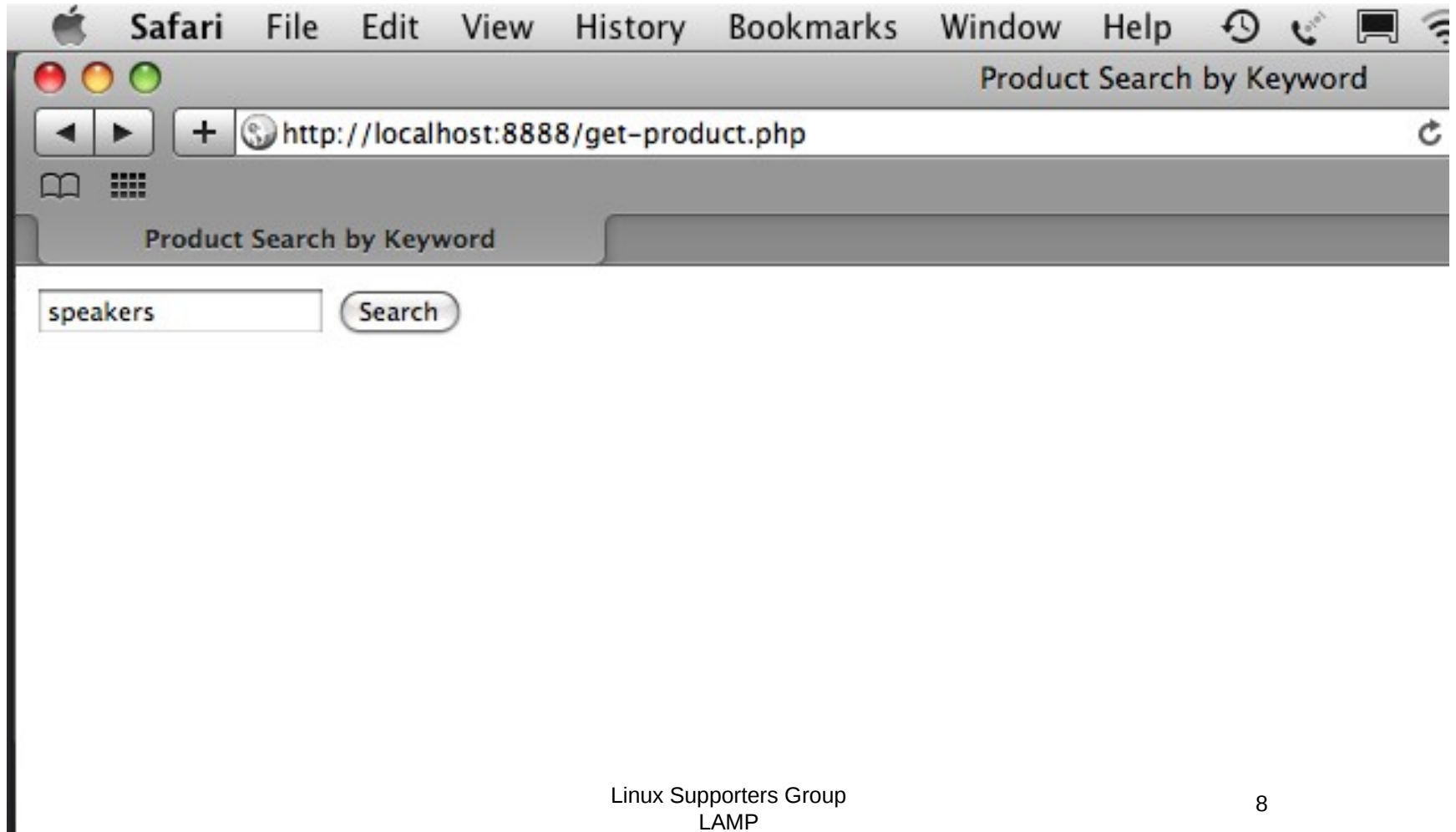
This creates a text box called "keyword"

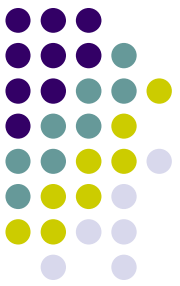
This creates a button labelled "Search". When clicked it will submit the updated contents of the text box to the URL

How the form is rendered



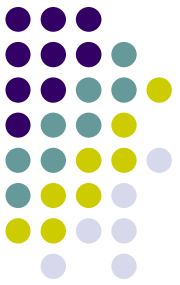
After Entering a Keyword





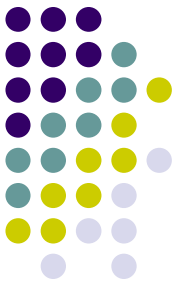
Resulting Display

Item	Description	Stock	Price
AUGBOX	Audio Gods Box Speakers	4	35.00
FDM6SS	Freedom 6.5" Stereo Speakers	124	18.50
HOMPYS	Home Party Speakers	8	88.50
SYOCSS	Sanyo Deluxe Car Stereo Speakers	20	12.50



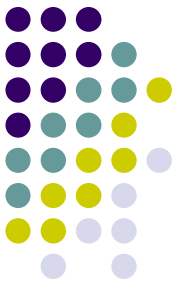
What happens?

- Browser requests *Get-Product* form.
- Server sends form, and browser renders it.
- User enters a keyword, and submits form.
- PHP *Show-Product* page is given the keyword.
- PHP processor queries database and constructs HTML page.
- Server sends page, and browser renders it.



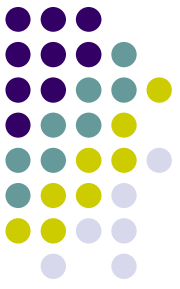
HTML that was Received

```
<html>
<head>
  <title>Product Details</title>
</head>
<body>
  <table>      ← Start a table, (default auto-layout)
    <tr>      ← Start heading row
      <th>Item</th>
      <th>Description</th>
      <th>Stock</th>      ← Define 4 column headings
      <th>Price</th>
    </tr>      ← End heading row
    <tr>      ← Start 1st data row
      <td>AUGBOX</td>
      <td>Audio Gods Box Speakers</td>      ← Define 4 data columns
      <td>4</td>
      <td>35.00</td>
    </tr>      ← End 1st data row
    ... and so on for 3 more rows ...
  </table>
</body>
</html>      ← End the table
```



What is PHP?

- PHP = **P**HP **H**ypertext **P**rocessor
(originally “*Personal* Hypertext Processor)
- Does *nothing* to HTML code.
- Is active only between `<?php` and `?>`
- Is a programming language, like **C** or a shell.
- Is HTML and MySQL friendly.
- Generates tailor-made HTML (using ‘**echo**’).

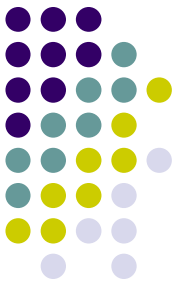


PHP that Created the HTML

```

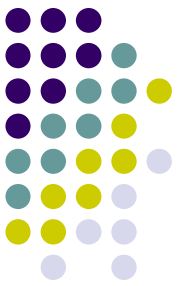
<html><head><title>Product Details</title></head>
<body> <table><tr><th>Item</th><th>Description</th><th>Stock</th><th>Price</
th></tr> <?php
    $code = $_POST['keyword']; Start PHP code
    $link = mysql_connect('localhost', 'root', 'root'); Get the keyword passed from the form
    or die("Unable to connect: ".mysql_error()); Connect to MySQL host
    mysql_select_db("retrotech") or die("Could not open db.</br>"); Choose database
    $key="'%".$code."'"; Construct search pattern
    $buffer=mysql_query($query) or die("Query failed: ".mysql_error()); Construct query
    while ($row=mysql_fetch_assoc($buffer)){
        echo "<tr>"; Fetch each row of query from buffer
        foreach($row as $colvalue){echo "<td>".$colvalue."</td>"; Generate some HTML
        echo "</tr>\n";
    };
    echo "</table>\n"; End while loop
    mysql_free_result($result); mysql_close($link); Release resources
?>
</body></html> End PHP, back to HTML

```



Notes on the PHP

- Variables start with \$, e.g., \$buffer
- Variables don't need to be declared.
- PHP statements end with ;
- A dot is used to join strings, e.g.,
`$key=""%".$code."%";`
- "\n" stops HTML output being one long line.
- Use single quotes inside double quotes, or *v.v.*

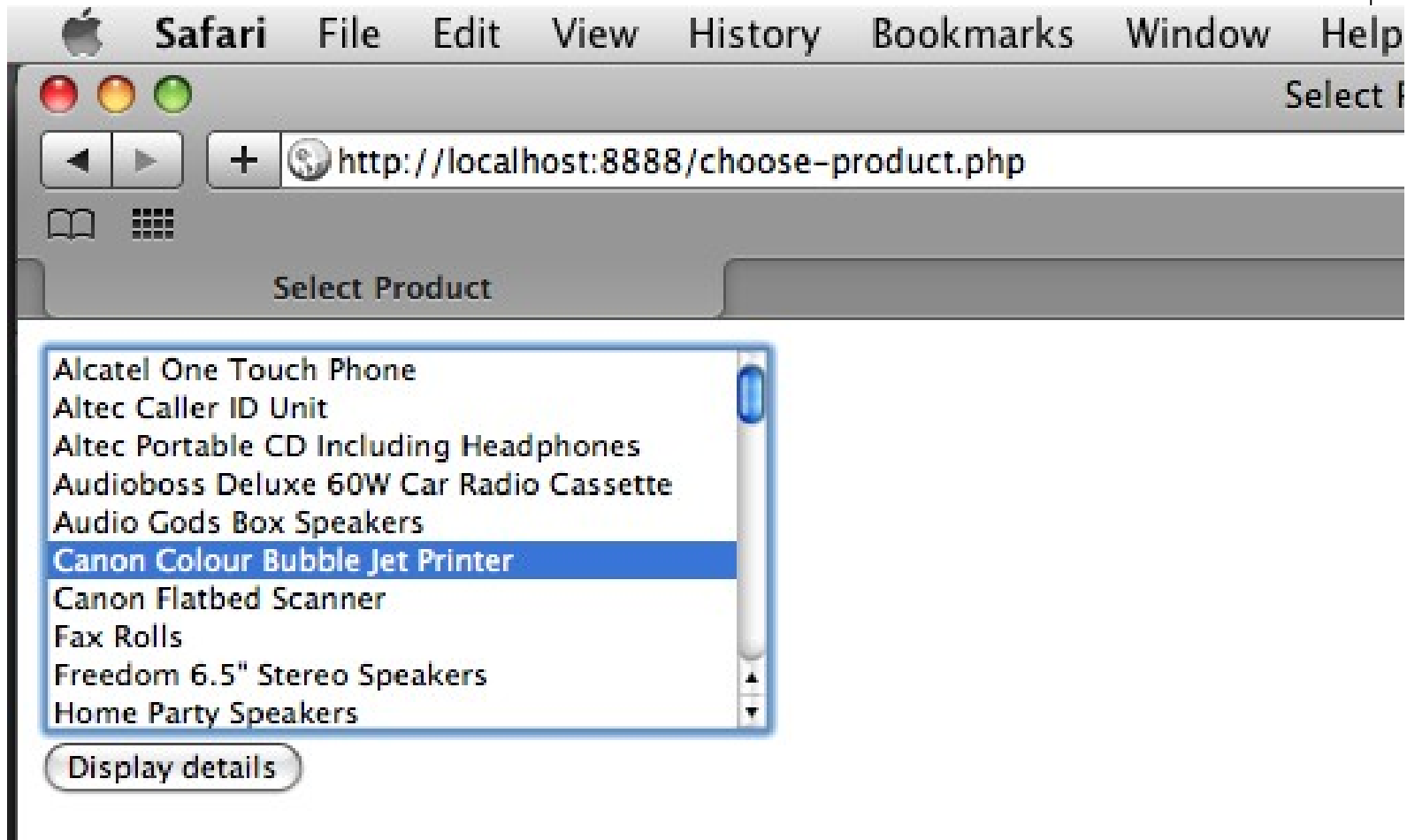
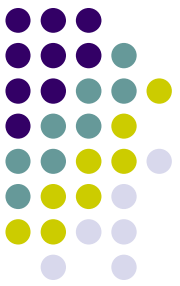


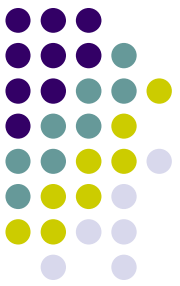
Arrays: Simple & Associative

- Simple array: `$var = $array[0];`
- The result of the SQL query is placed in `$buffer`
 - Each row can be retrieved as an **associative array** using `mysql_fetch_assoc()`;
 - Columns can then be referenced by name e.g., `$stock = $row['Stock'];`
 - The type and contents of any variable can be debugged using `print_r($row);`

```
Array([Item]=>ACLOTP [Description]=>Alcatel One Touch Phone [Stock]=>9 [Price]=>69.50)
```

A More Sophisticated Approach





Result of Submitting Form

The screenshot shows a Safari browser window with the following elements:

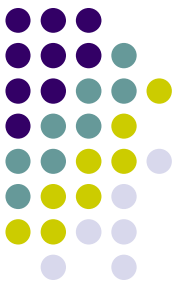
- Menu bar: Safari, File, Edit, View, History, Bookmarks, Window, Help
- Address bar: <http://localhost:8888/display-product.php>
- Page title: Product
- Tab: Product Details
- Table content:

Item	Description	Stock	Price
CANCBJ	Canon Colour Bubble Jet Printer	14	\$89.00

How would you like to maintain this HTML?

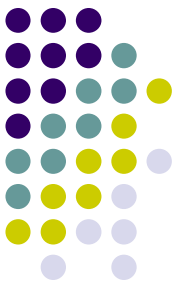


```
<html>
<head>
  <title>Select Product</title>
</head>
<form action="http://localhost:8888/display-product.php" method="post">
<select name = 'product' size = 10> ← A box with 10 scrollable options
<option value = 'ACLOTP'>Alcatel One Touch Phone</option>
<option value = 'ALTCIU'>Altec Caller ID Unit</option> ← The options
<option value = 'ALTPCD'>Altec Portable CD Including Headphones</option>
... and so on, and so on
and so on ... ← What is passed in product
<option value = 'TVLCDC'>Travel CD Case</option> ← What is displayed
<option value = 'UDNCID'>Uniden Cordless Phone With Caller ID</option>
<option value = 'VCAPLE'>Voca Phone Line Extension Cord</option>
</select></br>
  <input type="submit" value="Display details" /></p>
</form>
</html>
```



PHP won't need maintaining!

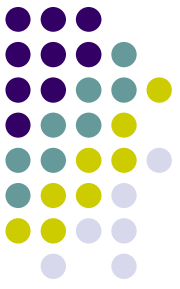
```
<html>
<head><title>Select Product</title></head>
<form action="http://localhost:8888/display-product.php" method="post">
<?php
    $link = mysql_connect('localhost', 'root', 'root')
        or die("Unable to connect: ".mysql_error());
    mysql_select_db("retrotech") or die("Could not open db.</br>");
    $query="select * from products;" ;
    $buffer=mysql_query($query) or die("Query failed: ".mysql_error()."</br>");
    echo "<select name = 'product' size = 10>\n";
    while ($row = mysql_fetch_assoc($buffer)){
        $description = $row['Description'];
        $item = $row['Item'];
        echo "<option value = '". $item. "'>". $description. "</option>\n";
    };
    echo "</select></br>\n";
    mysql_free_result($result);
    mysql_close($link);?>
    <input type="submit" value="Display details" /></p>
</form>
</html>
```



Display-Product Page

```
<html>
<head>
<title>Product Details</title></head><body>
<?php
    $code = $_POST['product']; ← Get selected product code
    $link = mysql_connect('localhost', 'root', 'root') or die("Unable to connect: ".mysql_error());
    mysql_select_db("retrotech") or die("Could not open db.</br>");
    $query="select * from products where Item = ".$code.";";
    $buffer=mysql_query($query) or die("Queryfailed: ".mysql_error()."</br>");
    echo "<table>\n";
    echo "<tr><th>Item</th><th> Description</th>";<th>Stock</th>"<th>Price</th></tr>\n";
    while ($row = mysql_fetch_assoc($buffer)){ ← Should get one row (or none)
        echo "<tr>\n<td>".$row['Item']."</td>";
        echo "<td>".$row['Description']."</td>";
        echo "<td align='right'>".$row['Stock']."</td>";
        echo "<td align='right'>".$row['Price']."</td></tr>\n";
    };
    echo "</table>\n";
mysql_free_result($result); mysql_close($link);?>
</body>
</html>
```

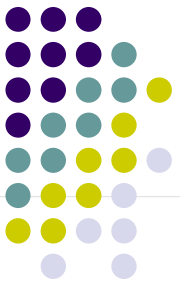
A little bit fancy



Examining Built-in Variables ...

```
<?php session_start();?>
<?php
if (isset($_SESSION['count'])){$_SESSION['count']++;}else
    {$_SESSION['count']=1;};
?>
<html>
<head><title>PHP Variables</title></head>
<body>
<h2>Built-in Variables</h2>
<?php
echo "<p>This page has been visited ".$_SESSION['count']."' times.&nbsp;&nbsp;&nbsp;";
echo "Last modified:&nbsp;&nbsp;&nbsp;".date('j M Y',getlastmod())."</p>";
?>
<h3>$_SERVER</h3><?php print_r($_SERVER); ?>
<h3>$_ENV</h3><?php print_r($_ENV); ?>
<h3>$_SESSION</h3><?php print_r($_SESSION); ?>
<h3>$_REQUEST</h3><?php print_r($_REQUEST); ?>
</body>
</html>
```

... not to Mention \$_POST



Built-in Variables

This page has been visited 10 times. Last modified: 1 Sep 2010.

\$_SERVER

```
Array ( [HTTP_HOST] => localhost:8888 [HTTP_USER_AGENT] => Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10_5_8; en-gb)
AppleWebKit/533.17.8 (KHTML, like Gecko) Version/5.0.1 Safari/533.17.8 [HTTP_ACCEPT] =>
application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5 [HTTP_ACCEPT_LANGUAGE] => en-gb
[HTTP_ACCEPT_ENCODING] => gzip, deflate [HTTP_COOKIE] => PHPSESSID=4d1c000e144515b6043c9b5d5289765b [HTTP_CONNECTION]
=> keep-alive [PATH] => /usr/bin:/bin:/usr/sbin:/sbin [SERVER_SIGNATURE] =>
Apache/2.0.63 (Unix) PHP/5.2.11 DAV/2 Server at localhost Port 8888
[SERVER_SOFTWARE] => Apache/2.0.63 (Unix) PHP/5.2.11 DAV/2 [SERVER_NAME] => localhost [SERVER_ADDR] => ::1 [SERVER_PORT]
=> 8888 [REMOTE_ADDR] => ::1 [DOCUMENT_ROOT] => /Applications/MAMP/htdocs [SERVER_ADMIN] => you@example.com
[SCRIPT_FILENAME] => /Applications/MAMP/htdocs/php-variables.php [REMOTE_PORT] => 51853 [GATEWAY_INTERFACE] => CGI/1.1
[SERVER_PROTOCOL] => HTTP/1.1 [REQUEST_METHOD] => GET [QUERY_STRING] => [REQUEST_URI] => /php-variables.php
[SCRIPT_NAME] => /php-variables.php [PHP_SELF] => /php-variables.php [REQUEST_TIME] => 1283297318 [argv] => Array ( ) [argc] => 0 )
```

\$_ENV

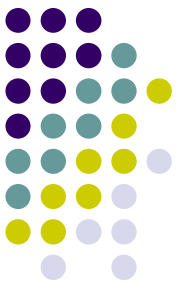
```
Array ( [SHELL] => /bin/bash [TMPDIR] => /var/folders/Nd/Nd8qjZiX2RaKwU+F70yBIU+++TI/-Tmp-/ [Apple_PubSub_Socket_Render] =>
/tmp/launch-rqB5MV/Render [USER] => dwyer [COMMAND_MODE] => legacy [SSH_AUTH_SOCK] => /tmp/launch-MfExdc/Listeners
[_CF_USER_TEXT_ENCODING] => 0x1F5:0:2 [PATH] => /usr/bin:/bin:/usr/sbin:/sbin [PWD] => / [HOME] => /Users/dwyer [SHLVL] => 2
[DYLD_LIBRARY_PATH] => /Applications/MAMP/Library/lib: [LOGNAME] => dwyer [DISPLAY] => /tmp/launch-mnxKXD:/0
[SECURITYSESSIONID] => a2f370 [_] => /Applications/MAMP/Library/bin/httpd )
```

\$_SESSION

```
Array ( [count] => 10 )
```

\$_REQUEST

```
Array ( [PHPSESSID] => 4d1c000e144515b6043c9b5d5289765b )
```



Notes & Resources

- Security can be a big issue if used on-line!
- Read *MySQL & PHP for Dummies*
- <http://v3.thewatchmakerproject.com/journal/276/building-a-simple-php-shopping-cart>
- PHP files are stored in LAMP/htdocs/
- Easy to get mixed up between PHP & HTML
 - (Not to mention SQL)
- Syntax errors are sent to LAMP/php_error.log