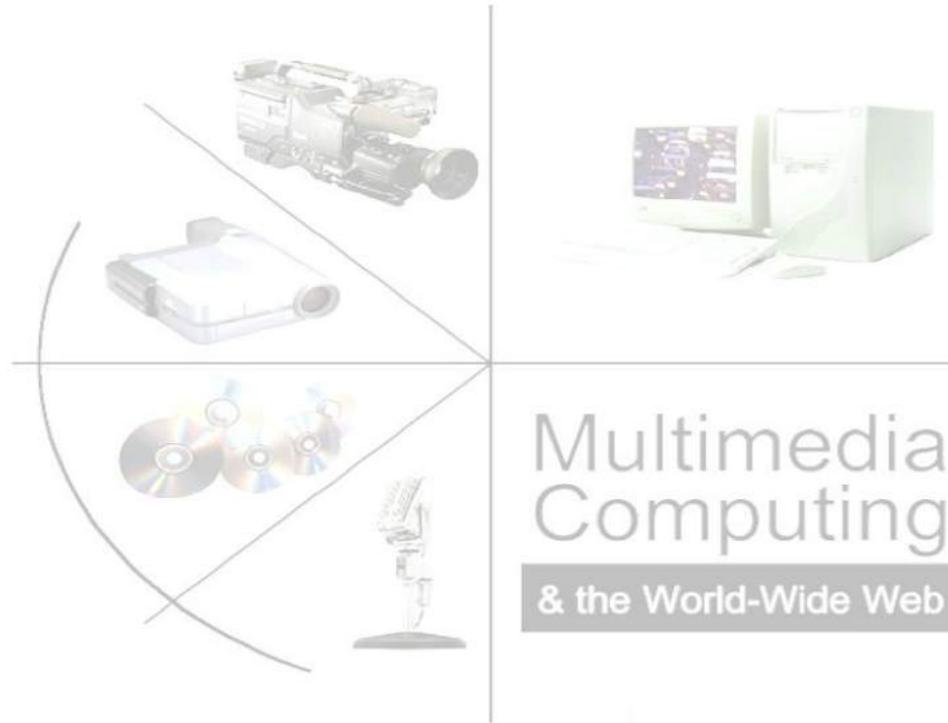


PHP Lecture 2



PHP Uses and Examples
Introduction to MySQL

Your Second PHP Page

```
<?php  
    phpinfo();  
?>
```

- ◆ Calls a single function (**phpinfo**)
 - this prints a lot of useful information about your system and setup
 - such as available predefined variables
 - loaded PHP modules
 - configuration settings

Headers and Footers

- ◆ PHP is a great way to implement templates on your website
 - templates give a consistent feel to our website
- ◆ Universal headers and footers
 - these can be used in every page in our web site

Headers

- ◆ We create a file called header.php
 - in it we put all common html that we want at the beginning of each web page

```
<html><head>  
  <title>Joe's Website</title>  
  <link rel="stylesheet" type="text/css"  
    href="mycssfile.css"/>  
</head><body>
```

Footers

- ◆ We create a file called footer.php

```
<footer>
  <span class="copyright">&copy; UCD School of Computer Science,
    <?php echo ("".date("Y")); ?>
    . All rights reserved. | University College Dublin, Belfield, Dublin
    4, Ireland |
    <a href="http://www.ucd.ie/"> ucd.ie </a>
  </span>
</footer>
</body>
```

Including our Header & Footer

- ◆ Any html document on this website will look like this
 - begins with a php include for the header
 - ends with a php include for the footer

```
1 1<!DOCTYPE html>
2 2<?php include('header.php'); ?>
3 
4 3<header>
5 4  <nav>This is where my links would go</nav>
6 5<section id="welcome">
7 6  | <h1>This is a heading</h1>
8 7  | <p>Aurora borealis at this time of year, at this time of day, in this part
9 8  . of the country, localized entirely within your kitchen?</p>
10 9</section>
11 10</header>
12 11<aside>Click the non-existent button to subscribe to this feed </aside>
13 12<article class="feed">
14 13  | <p> Some news item </p>
15 14</article>
16 15<?php include('footer.php'); ?>
17 16
18 17</html>
```

What it looks like

This is a heading

Aurora borealis at this time of year, at this time of day, in this part of the country, localized entirely within your kitchen?

Click the non-existent button to subscribe to this feed

Some news item

© UCD School of Computer Science
University College Dublin

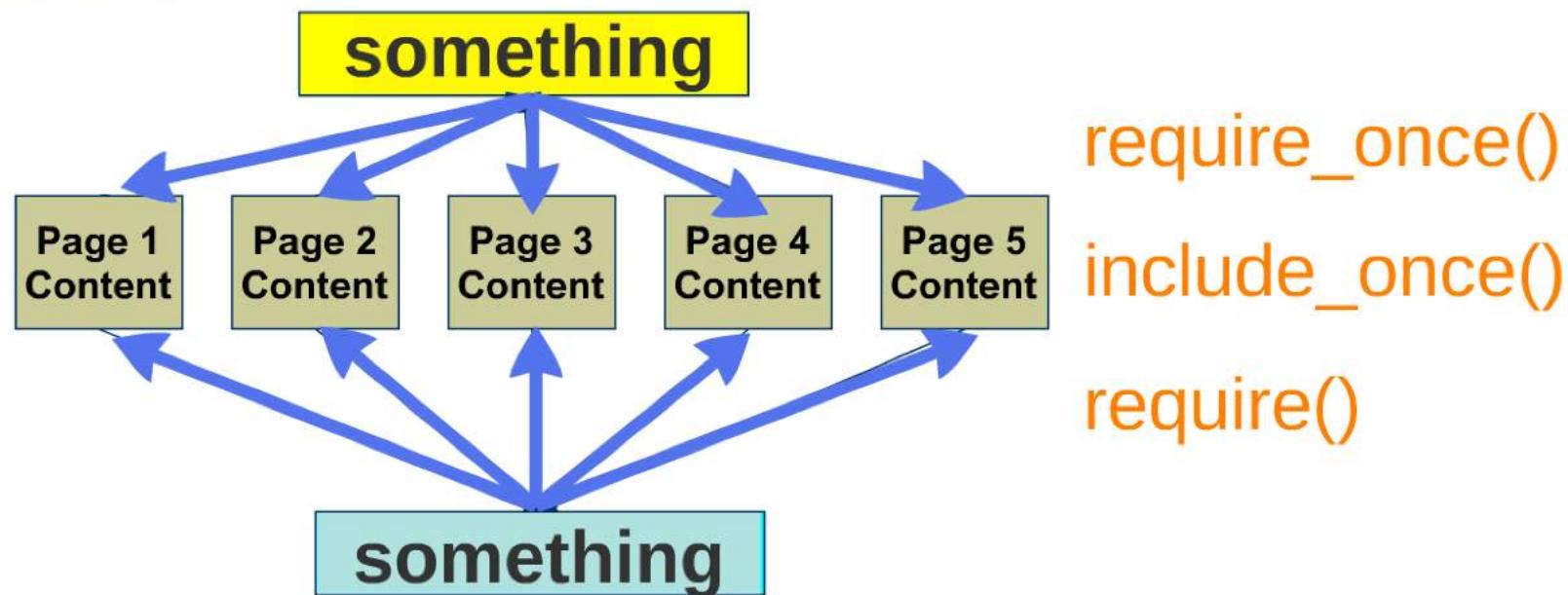
```
1 <!DOCTYPE html>
2 <html><head>
3   <title>Joe's Website</title>
4   <link rel="stylesheet" type="text/css" href="mycssfile.css" />
5 </head><body>
6
7 <header>
8   <nav>This is where my links would go</nav>
9   <section id="welcome">
10    <h1>This is a heading</h1>
11    <p>Aurora borealis at this time of year, at this time of day, in this part
12      of the country, localized entirely within your kitchen?</p>
13  </header>
14  <aside>Click the non-existent button to subscribe to this feed</aside>
15  <article class="feed">
16    <p>Some news item </p>
17  </article>
18
19 <footer>
20   <span class="copyright">&copy; UCD School of Computer Science, 2018. All
21     rights reserved. | University College Dublin, Belfield, Dublin 4, Ireland |
22     <a href="http://www.ucd.ie/"> ucd.ie </a>
23   </span>
24 </footer>
25 </body>
26 </html>
```

header.php

footer.php

Why do this?

- ◆ Changes to header or footer only require editing of a single file
- ◆ Helps separate the content and design for easier maintenance



Functions

- ◆ PHP has functions just like other languages
- ◆ Defined with the **function** keyword
- ◆ May take arguments
- ◆ May return a value
- ◆ Function names are not case-sensitive (foo = Foo)

```
<?php  
    function foo ($arg1, $arg2){  
        echo "A function.<br/>";  
        return 7;  
    }  
?> $i = foo("some words", 8);  
$i == 7
```

```
optional type return  
function sum($a, $b): float {  
    return $a + $b;  
}  
  
optional default parameter  
function userType($type="1")  
{  
    return "User is of type ".$type."\n";  
}  
echo (userType()); User is of type 1  
echo (userType(4)); User is of type 4  
echo (userType(1)); User is of type 1
```

Processing Forms

- ◆ When a form is submitted on the web what happens?
 - HTTP Request is sent to the web server
 - request contains a set of variables with values
 - these represent the entries in the form
 - text fields, password fields, radio buttons, etc
- ◆ Web server uses these variables to perform some custom action
 - e.g. send a mail, book flights, etc
- ◆ Web server returns some HTML to the client

HTTP

- ◆ HTTP is the language that web clients and web servers use to talk to each other
- ◆ Each message, whether a request or a response, has three parts:
 - the request or the response line
 - a header section
 - the body of the message
- ◆ We will talk about the two most common request types
 - GET
 - POST



WIKIPEDIA
The Free Encyclonenedia

Main Page Talk

Read Vi

Welcome to Wikipedia,

• Arts

Elements | Network | Sources | Timeline | Profiles | Resources | Audits | Console

Preserve log Disable cache

Name

Path

Main_Page

/wiki

load.php?debug=false&lan...
bits.wikimedia.org/en.wikip...

load.php?debug=false&lan...
bits.wikimedia.org/en.wikip...

load.php?debug=false&lan...
bits.wikimedia.org/en.wikip...

100px-Milla_Jovovich_Cann...
upload.wikimedia.org/wikip...

75px-Muhammadu_Buhari...
upload.wikimedia.org/wikip...

78px-Busta_Jan_z_Jen%C5...
upload.wikimedia.org/wikip...

70px-Rakesh_sharma.jpg
upload.wikimedia.org/wikip...

39 requests | 16.9 KB transferred | 1...

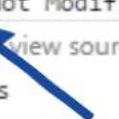
Headers Preview Response Cookies Timing

General

Remote Address: [REDACTED]:80
Request URL: http://en.wikipedia.org/wiki/Main_Page
Request Method: GET
Status Code: 304 Not Modified

Response Headers

view source
Accept-Ranges: bytes
Age: 714
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Connection: keep-alive
Content-Encoding: gzip
Content-Language: en
Content-Type: text/html; charset=UTF-8
Date: Thu, 02 Apr 2015 10:19:07 GMT
Last-Modified: Thu, 02 Apr 2015 10:07:12 GMT
Server: Apache
Vary: Accept-Encoding,Cookie
Via: 1.1 varnish, 1.1 varnish
X-Analytics: page_id=15580374;ns=0
X-Cache: cp1055 hit (9), cp1054 frontend hit (4571)



HTTP Message Types

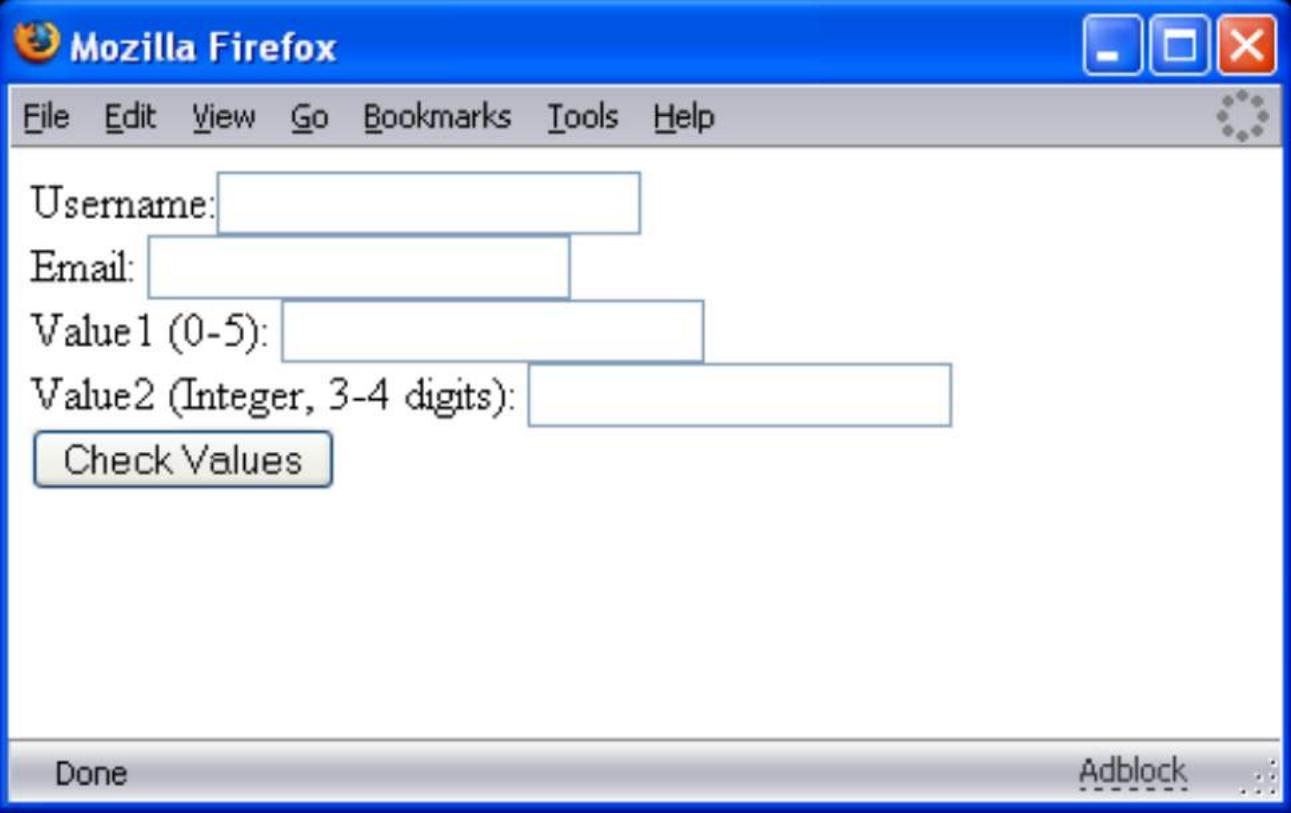
- ◆ GET is the default HTTP message type
 - usually when you click on a link
 - whenever you enter an address and hit go
- ◆ Both GET and POST may^{be} used when submitting forms
 - usually POST
- ◆ There are other HTTP message types
 - e.g. HEAD, PUT, DELETE
 - we won't cover them in this course

GET and POST

- ◆ GET and POST come with usage recommendations
 - GET should be used when the form processing is *idempotent*
 - otherwise POST should be used
- ◆ As a simplification, we might say that **GET** is used for getting data whereas **POST** may involve making state changes to the web server
 - storing or updating data
 - ordering a product
 - sending E-mail

viewing vs changing

A Simple Form



The image shows a screenshot of a Mozilla Firefox browser window displaying a simple HTML form. The window title is "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "Go", "Bookmarks", "Tools", and "Help". The toolbar includes standard icons for minimize, maximize, and close. The main content area contains four input fields and one button:

- "Username:" followed by an empty text input field.
- "Email:" followed by an empty text input field.
- "Value1 (0-5):" followed by an empty text input field.
- "Value2 (Integer, 3-4 digits):" followed by an empty text input field.

Below the input fields is a blue "Check Values" button. At the bottom of the form window are "Done" and "Adblock" buttons.

Form – HTML

```
<html><body>
<form method="GET" action="welcome.php"
      onsubmit="return validateAll(this)"
      name="detectform">
    Username:<input type="text" name="Username"> <br/>
    Email: <input type="text" name="Email"><br/>
    Value1 (0-5): <input type="text" name="Value1"><br/>
    Value2 (Integer, 3-4 digits):
    <input type="text" name="Value2"><br/>
    <input type="submit" value="Check Values"
          name="Button1"><br/>
</form>
</body></html>
```

So What Happens?

- ◆ So what happens when we click on the button?
 - the JavaScript validation script is called
 - if the JavaScript finds no problems (i.e. returns true) a HTTP request is sent to the web server requesting the document purchase.php
 - the following parameters could be sent
 - Username=joe
 - Email=joe@bloggs.com
 - Value1=1
 - Value2=123
- ◆ The request type is GET
 - method = "GET"

HTTP GET Requests

- ◆ In the address bar the URL looks like this:
`http://secure.ucd.ie/~lcoyle/welcome.php?
Username=joe&Email=joe@bloggs.com&Value1=1&Value2
=123`
- ◆ There is now a fourth component to the URL – the parameter list
 - this is a list of parameters after the URL that begins with a question mark
 - the list is delineated with ampersands (&)
 - spaces in the values are replaced with either “+” or “%20”

What Can the Web Server Do?

- ◆ The web server now has access to these parameters
- ◆ PHP can be used to extract them
 - `$_GET` is an associative array holding all the GET parameters received by the web server
- ◆

```
echo $_GET['Username']
```

 - joe ← value ← key
- ◆ PHP can be used to do whatever the developer wants to do with this value
 - save it to a file
 - enter it to a database
 - email it to someone

Getting Client Data

- ◆ `$_GET` and `$_POST` are associative arrays of variables passed to the current script
 - depending on the method that was used to transmit the HTTP message only one of these will be available
- ◆ Format of these arrays is key -> value
 - key is the name of the variable from the form
 - value is the actual value that came from the form
 - e.g. Username -> joe
- ◆ `$_GET` and `$_POST` are **superglobals**

Separating the Variables from \$_GET (or \$_POST)

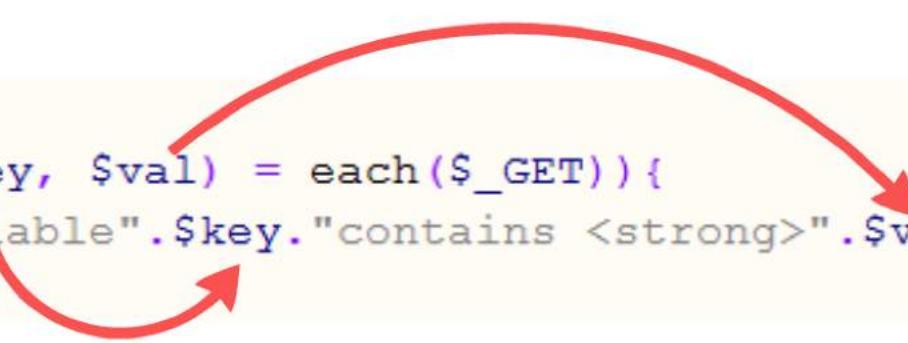
- Unless we know the variable names in advance from the GET request we will need to extract them
 - for the most part we will know the variable names since we probably wrote the form

```
list($key, $val) = each($_GET);
```

- This creates two lists of variables called \$key and \$val
 - these are the keys and values of the associative array \$_GET respectively

Lets Print out the Variables

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>I'm printin</title>
5      </head>
6      <body>
7          <p>
8              <?php
9                  while(list($key, $val) = each($_GET)) {
10                      echo("Variable". $key. "contains <strong>". $val. "</strong><br/>\n");
11                  }
12              ?>
13          </p>
14      </body>
15  </html>
```



Source View

Mozilla Firefox

File Edit View Go Bookmarks Tools Help

Variable Username contains joe
Variable Email contains joe@bloggs.com
Variable Value1 contains 1
Variable Value2 contains 123
Variable Button1 contains Check Values

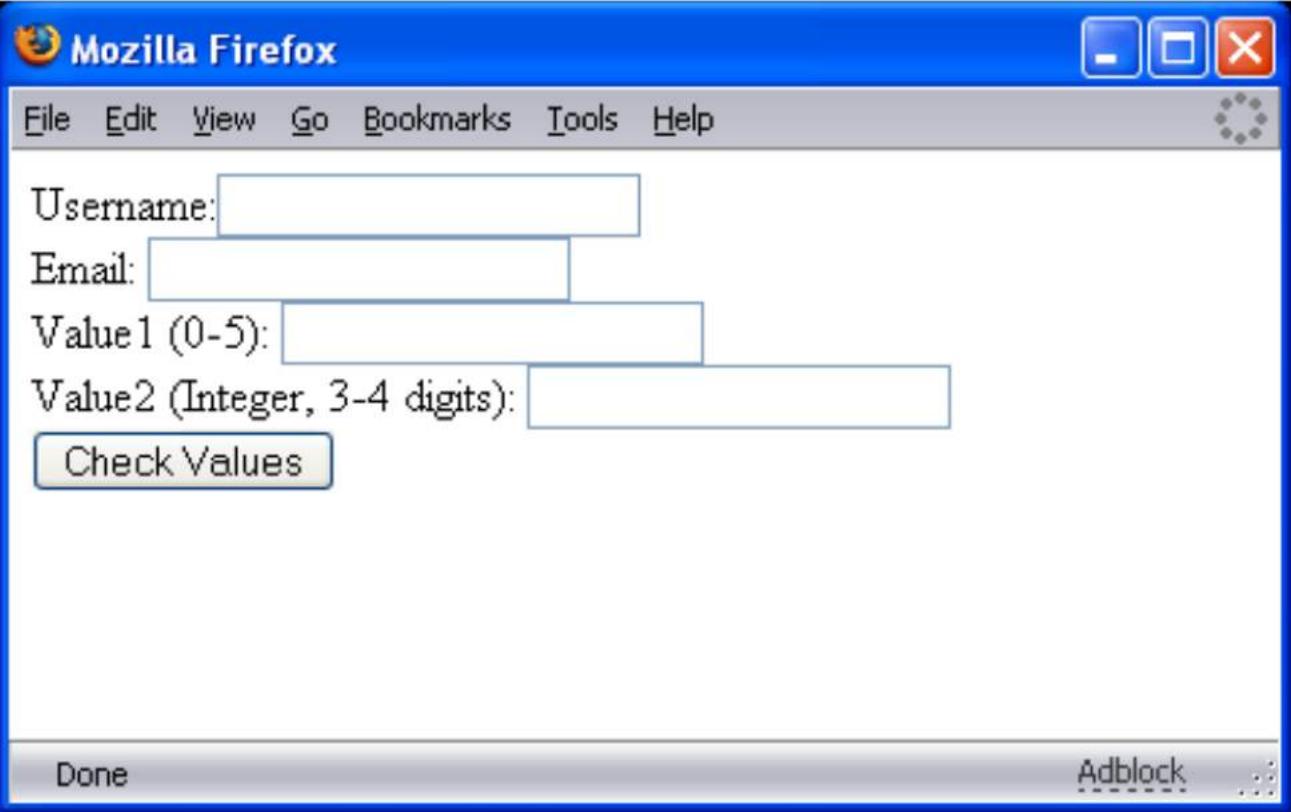
view-source: - Source of: http://secure.ucd.ie/~lcoyle/snoopyvariables.php?User... -

File Edit View

```
<html><body>
Variable Username contains <strong>joe</strong><br />
Variable Email contains <strong>joe@bloggs.com</strong><br />
Variable Value1 contains <strong>1</strong><br />
Variable Value2 contains <strong>123</strong><br />
Variable Button1 contains <strong>Check Values</strong><br />
</body></html>
```

Done Lorcan has forgotten the doctype and head... as usual...

A Simple Form



The image shows a screenshot of a Mozilla Firefox browser window. The title bar reads "Mozilla Firefox". The menu bar includes "File", "Edit", "View", "Go", "Bookmarks", "Tools", and "Help". Below the menu bar are four buttons: a minus sign, a square, a red X, and a circular refresh icon. The main content area contains four input fields and one button. The first field is labeled "Username:" with an empty text input. The second field is labeled "Email:" with an empty text input. The third field is labeled "Value1 (0-5):" with an empty text input. The fourth field is labeled "Value2 (Integer, 3-4 digits):" with an empty text input. Below these fields is a blue rectangular button with the text "Check Values". At the bottom of the window is a toolbar with "Done" on the left and "Adblock" with three dots on the right.

Username:

Email:

Value1 (0-5):

Value2 (Integer, 3-4 digits):

Check Values

Done Adblock ::

Let's Print out a Welcome Screen

```
<html><body><p>  
    Welcome <?php echo $_GET['Username'] ;?>.\n  
    We have registered your email address (<?  
    php echo $_GET['Email'] ?>).\n  
    The magic numbers you entered were <?php  
    echo $_GET['Value1'] ?> and <?php echo  
    $_GET['Value2'] ?>.  
</p></body></html>
```

A More Complicated Example

- ◆ Imagine we are building a web page that takes in orders for t-shirts and cds for a band's website
 - the website only stocks one type of t-shirt and one type of CD
 - very simple interface, the user can place an order
- ◆ Lets assume there is a PHP function that takes in a username and automatically debits their account for the amount they owe

```
debitAccount ($username, $cost) ;
```
- ◆ Lets assume that t-shirts cost €20 and CDs cost €15

The HTML

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Can I ask how you can afford to buy all these amazing products?</title>
5          <script type="text/javascript" src="myscripts/validator.js" async></script>
6      </head>
7      <body>
8          <form action="/purchase.php" method="post" onsubmit="return validateAll(this)"
9              name="detectform" id="detectform">
10             Username:
11             <input type="text" name="username" id="username" size="20"/> <br />
12             Number of T-shirts: (&euro;20) <br />
13             0 <input type="radio" name="t-shirt" value="0"/> <br />
14             1 <input type="radio" name="t-shirt" value="1"/> <br />
15             2 <input type="radio" name="t-shirt" value="2"/> <br />
16             Number of "cds": (&euro;15) <br />
17             0 <input type="radio" name="cd" value="0"/> <br />
18             1 <input type="radio" name="cd" value="1"/> <br />
19             2 <input type="radio" name="cd" value="2"/> <br />
20             <input type="submit" id="submitter" value="Check Values"/>
21         </form>
22     </body>
23 </html>
```

The web page

Username:

Number of T-shirts: (€20)

- 0
- 1
- 2

Number of "cds": (€15)

- 0
- 1
- 2

Check Values

```
<!DOCTYPE html>
<html>
    <head>
        <title>Can I ask how you can afford to buy all these amazing products?</title>
        <script type="text/javascript" src="myscripts/validator.js" async></script>
    </head>
    <body>
        <form action="/purchase.php" method="post" onsubmit="return validateAll(this)" name="detectform" id="detectform">
            Username: <input type="text" name="username" id="username" size="20"/> <br />
            Number of T-shirts: (&euro;20) <br />
            0 <input type="radio" name="t-shirt" value="0"/> <br />
            1 <input type="radio" name="t-shirt" value="1"/> <br />
            2 <input type="radio" name="t-shirt" value="2"/> <br />
            Number of "cds": (&euro;15) <br />
            0 <input type="radio" name="cd" value="0"/> <br />
            1 <input type="radio" name="cd" value="1"/> <br />
            2 <input type="radio" name="cd" value="2"/> <br />
            <input type="submit" id="submitter" value="Check Values"/>
        </form>
    </body>
</html>
```

Processing the Form Data

- ◆ Lets print out a confirmation message saying that the order was received
 - includes the username
 - the total price
 - the list of items purchased
- ◆ We could also print out a different page if the user didn't purchase anything (0 cds and 0 t-shirts)

First – getting the variables

- ◆ We create three global variables
 - for the username, number of t-shirts and number of CDs

```
<?php  
    global $username, $numberOfTShirts,  
    $numberOfCDs;  
  
    $username = $_POST['Username'];  
    $numberOfTShirts = $_POST['t-shirt'];  
    $numberOfCDs = $_POST['cd'];  
?  
>
```

Calculating the Total Cost

```
function getCost($cds, $tshirts) {  
    $cost = 0;  
    //CDs cost £15  
    $cost = 15 * $cds;  
    // T shirts cost £20  
    $cost = $cost + (20 * $tshirts);  
    return $cost;  
}
```

or

```
function getCost($cds, $t-shirts) {  
    return(15 * $cds) + (20 * $t-shirts);  
}
```

Debiting the Account Balance

```
<?php  
    function debitAccount($user, $euros) {  
        // perform some actions here  
        // get credit card from database  
        // send visa details of the transaction  
    }  
?>
```

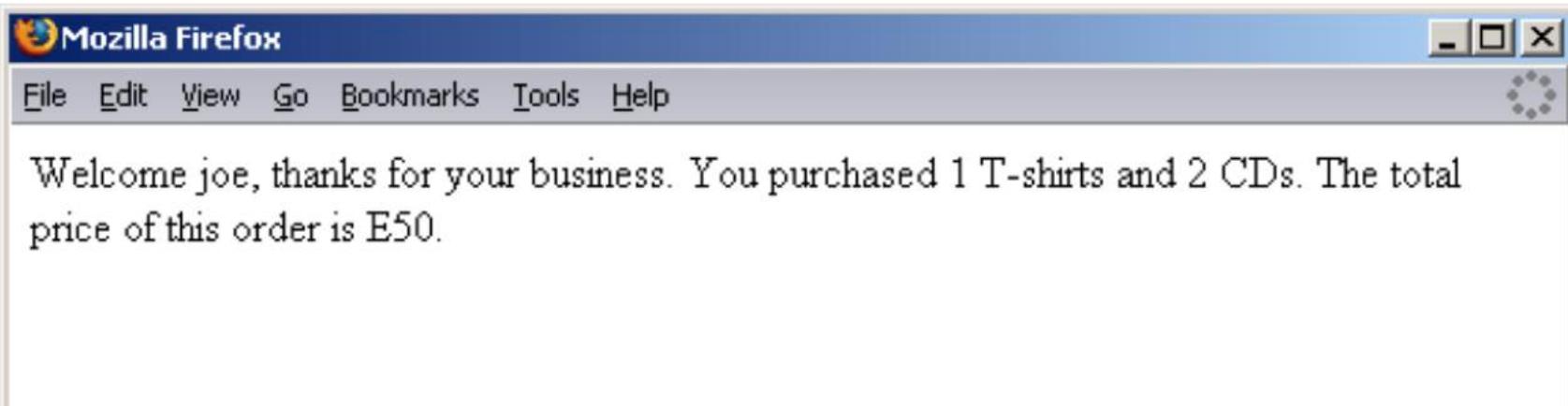
- ◆ Embed the following function in the page

```
debitAccount($username, $cost);
```

The HTML

```
<html><body><p>  
Welcome <?php echo $username ?>,  
thanks for your business. You purchased  
<?php echo $numberOfTShirts ?> T-shirts and  
<?php echo $numberOfCDs ?> CDs. The total price of  
this order is E  
<?php  
    $cost = getCost($numberOfCDs, $numberOfTShirts);  
    debitAccount($Username, $cost);  
    echo $cost;  
?>. Thank you!  
</p></body></html>
```

The Purchase Page

A screenshot of a Mozilla Firefox browser window titled "view-source: - Source of: http://secure.ucd.ie/~lcoyle/purchase.php - Mozilla Firefox". The menu bar shows "File", "Edit", and "View". The main content area displays the raw HTML code:

```
<html><body>
<p>
Welcome joe, thanks for your business. You purchased 1 T-shirts and 2 CDs.
</p>
</body></html>
```

D

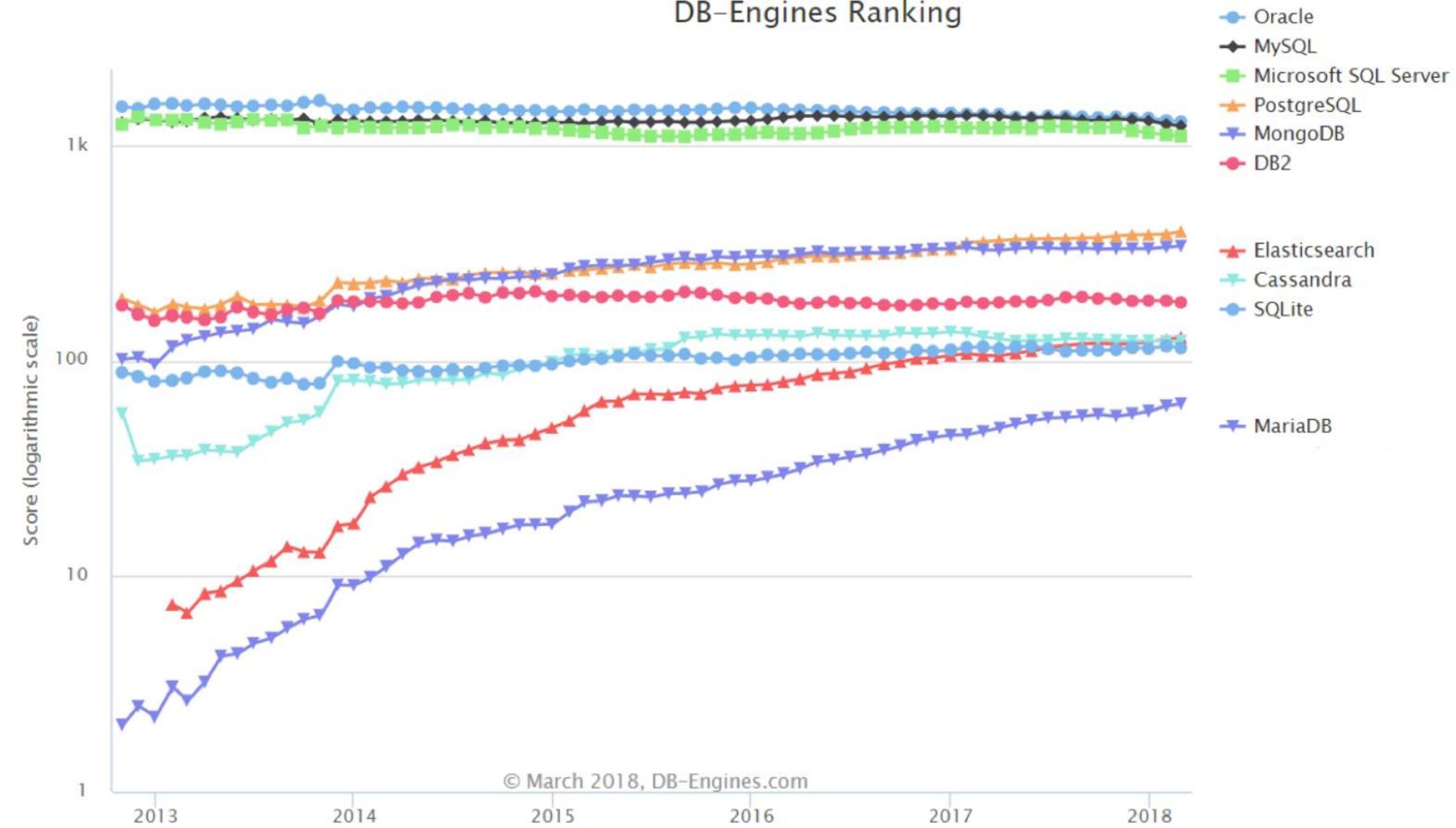
How are the website owners to keep track of purchases?

PHP Database Support - MySQL

- ◆ PHP and MySQL work very well together
 - in addition to the speed and features of each individual tool
 - in some ways they have been developed with the other in mind
- ◆ Both are open source and free
- ◆ MySQL is available here: mysql.com/

Databases

DB-Engines Ranking



MySQL Overview

- ◆ Fast, free, stable
- ◆ Relational database management system (RDBMS)
- ◆ Uses SQL
 - Syntax is similar to Oracle
- ◆ Many of the same features as Oracle
- ◆ Production version has subqueries, stored procedures, and triggers
 - not important for casual users

Introduction to Databases

- ◆ A relational database manager (MySQL) manages databases which holds tables which has records (rows) with attributes (columns)
- ◆ In order to be indexable, each record must have a unique ID
 - also known as a Primary Key
- ◆ Each attribute has to have a data type. (e.g. int, text, varchar)
- ◆ A database language (SQL) is used to create and delete databases and manage data

An Example Database Table

- ◆ Table is called *users*
- ◆ Columns are *Username*, *Email*, *Value1* and *Value2*
 - the table is shown below
- ◆ Lets get this data
- ◆ Lets add to the data

Username	Email	Value1	Value2
Joe	joe@bloggs.com	1	123
Bob	bob@bloggs.com	3	567

SELECT and INSERT

- ◆ Just the basics of the programming language SQL
 - SELECT queries a database
 - INSERT inserts new data into a database
- ◆ SELECT
 - `SELECT columnname FROM tablename WHERE conditional;`
`SELECT Username, Email FROM users WHERE`
`Username="joe";`
`joe, joe@bloggs.com`
 - `INSERT (columnname,...) VALUES (value1,...) INTO tablename;`
`INSERT (Username, Email, Value1, Value2) VALUES`
`('bill', 'bill@bloggs.com', 3, 543);`

Database Example

```
<html> <body>
<?php
    connectToDatabase();
    $sql = "SELECT Username, Email FROM users"; SQL
    $results = getResults($sql);
    echo "<table><tr>";
    echo "<th>Username</th>"; HTML
    echo "<th>Email</th></tr>";
    while (getNextRow($results)) {
        $name=getResult($result,"Username");
        $email=getResult($result,"Email");
        echo "<tr><td>$name</td>";
        echo "<td>$email</td></tr>";
    }
    closeDatabase();
    echo "</table>";
?>
</body> </html>
```

defined (and presumably imported from) elsewhere

Database Example

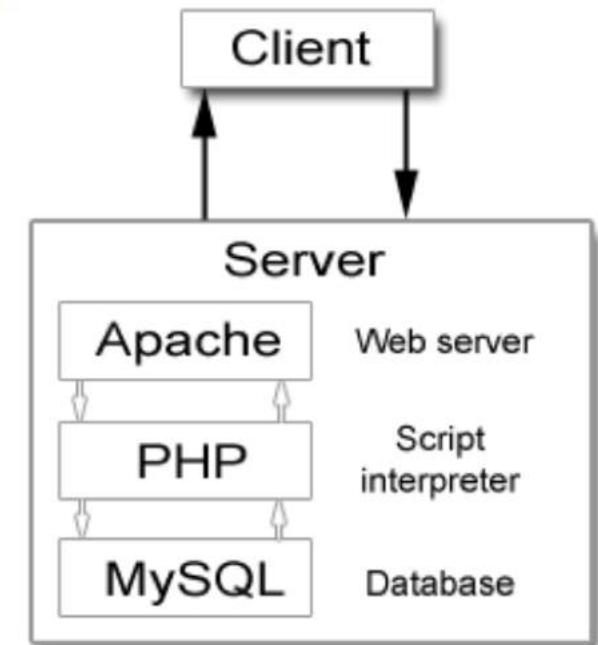
The screenshot shows two windows side-by-side. The left window is a Mozilla Firefox browser displaying a simple HTML table with two rows. The first row contains 'joe' and 'joe@bloggs.com'. The second row contains 'bob' and 'bob@bloggs.com'. The right window is a 'view-source:' window showing the raw HTML source code of the page, which includes the table structure and the data.

joe joe@bloggs.com
bob bob@bloggs.com

```
<html><body>
<table><tr>
<tr><td>joe</td><td>joe@bloggs.com</td></tr>
<tr><td>bob</td><td>bob@bloggs.com</td></tr>
</table>
</body> </html>
```

LAMP

- ◆ Set of free software programs commonly used together to run dynamic Web sites or servers
 - Linux
 - Apache Web Server (<http://apache.org/>)
 - MySQL database
 - PHP, Perl and or Python scripting
- ◆ Alternatives:
 - WAMP (Windows)
 - WIMP (Windows, IIS)



XAMPP: Like LAMP, but works on everything

XAMPP Control Panel v3.2.2 [Compiled: Nov 12th 2015]



XAMPP Control Panel v3.2.2

Modules

Service	Module	PID(s)	Port(s)	Actions			
<input type="checkbox"/>	Apache	9424 11312	80, 443	Stop	Admin	Config	Logs
<input type="checkbox"/>	MySQL	8408	3306	Stop	Admin	Config	Logs
<input type="checkbox"/>	FileZilla			Start	Admin	Config	Logs
<input type="checkbox"/>	Mercury			Start	Admin	Config	Logs
<input type="checkbox"/>	Tomcat			Start	Admin	Config	Logs

- Config
- Netstat
- Shell
- Explorer
- Services
- Help
- Quit

```

11:57:12 [main] XAMPP Installation Directory: "c:\xampp\" 
11:57:12 [main] Checking for prerequisites
11:57:12 [main] All prerequisites found
11:57:12 [main] Initializing Modules
11:57:12 [main] Starting Check-Timer
11:57:12 [main] Control Panel Ready
11:57:23 [Apache] Attempting to start Apache app...
11:57:24 [Apache] Status change detected: running
11:57:27 [mysql] Attempting to start MySQL app...
11:57:28 [mysql] Status change detected: running
  
```



XAMPP Apache + MariaDB + PHP + Perl

Welcome to XAMPP for Windows 7.2.3

You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

XAMPP is meant only for development purposes. It has certain configuration settings that make it easy to develop locally but that are insecure if you want to have your installation accessible to others. If you want have your XAMPP accessible from the internet, make sure you understand the implications and you checked the [FAQs](#) to learn how to protect your site. Alternatively you can use [WAMP](#), [MAMP](#) or [LAMP](#) which are similar packages which are more suitable for production.

Start the XAMPP Control Panel to check the server status.

Community

XAMPP has been around for more than 10 years – there is a huge community behind it. You can get involved by joining our [Forums](#), adding yourself to the [Mailing List](#), and liking us on [Facebook](#), following our exploits on [Twitter](#), or adding us to your [Google+](#) circles.

Contribute to XAMPP translation at translate.apachefriends.org.

Can you help translate XAMPP for other community members? We need your help to translate XAMPP into different languages. We have set up a site, translate.apachefriends.org, where users can contribute translations.

Install applications on XAMPP using Bitnami

Apache Friends and Bitnami are cooperating to make dozens of open source applications available on XAMPP, for free. Bitnami-packaged applications include Wordpress, Drupal, Joomla! and dozens of others and can be deployed with one-click installers. Visit the [Bitnami XAMPP page](#) for details on the currently available apps.

The screenshot shows the phpMyAdmin interface on a local host. At the top, the URL bar displays the address: `localhost/phpmyadmin/#PMAURL-0:index.php?db=&table=&server=1&target=&lang=en&collation_connect`. The main header includes the "phpMyAdmin" logo and a navigation menu with links for Databases, SQL, Status, Users, Export, Import, and Settings.

The left sidebar lists various MySQL databases:

- cdcol
- information_schema
- mysql
- performance_schema
- phpmyadmin
- test
- webauth

Two large black arrows point from the text "General Settings" and "Appearance Settings" located in the bottom right corner of the image towards the "General Settings" and "Appearance Settings" sections in the screenshot.

General Settings

- Server connection collation: utf8_general_ci

Appearance Settings

- Language: English
- Theme: pmahomme
- Font size: 82%
- [More settings](#)



(Recent tables) ...

+ -	cdcol
+ -	information_schema
+ -	mysql
+ -	performance_schema
+ -	phpmyadmin
+ -	test
+ -	webauth

Browse

Structure

SQL

Search

Insert

Export

⚠ This table does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete buttons will not work.

✓ Showing rows 0 - 1 (2 total, Query took 0.0010 sec)

```
SELECT *  
FROM `users`  
LIMIT 0 , 30
```



Show : Start row: Number of rows: Headers every

+ Options

Username	Email	Value1	Value2
joe	joe@bloggs.com	1	123
bill	bill@bloggs.com	3	543



Show : Start row: Number of rows: Headers every

Query results operations