Table of Contents

[Terms & definition 2](#_Toc26099854)

[Client 2](#_Toc26099855)

[Server 2](#_Toc26099856)

[Programming the web 3](#_Toc26099857)

[What is client side code? 3](#_Toc26099858)

[Why client side code? 3](#_Toc26099859)

[What is server side code? 3](#_Toc26099860)

[Why server side code? 3](#_Toc26099861)

[PHP Introduction 4](#_Toc26099862)

[Apache HTTP server 4](#_Toc26099863)

[What is PHP 4](#_Toc26099864)

[What does PHP code look like? 4](#_Toc26099865)

[PHP Introduction 5](#_Toc26099866)

[PHP Hello World 5](#_Toc26099867)

[Comment IN PHP 5](#_Toc26099868)

[PHP variable 5](#_Toc26099869)

[PHP URL 6](#_Toc26099870)

[PHP Operators 6](#_Toc26099871)

[PHP conditional statements 6](#_Toc26099872)

[PHP arrays 7](#_Toc26099873)

[PHP numeric arrays 7](#_Toc26099874)

[PHP Associative Arrays 8](#_Toc26099875)

[PHP Multidimensional Arrays 8](#_Toc26099876)

[PHP loops 9](#_Toc26099877)

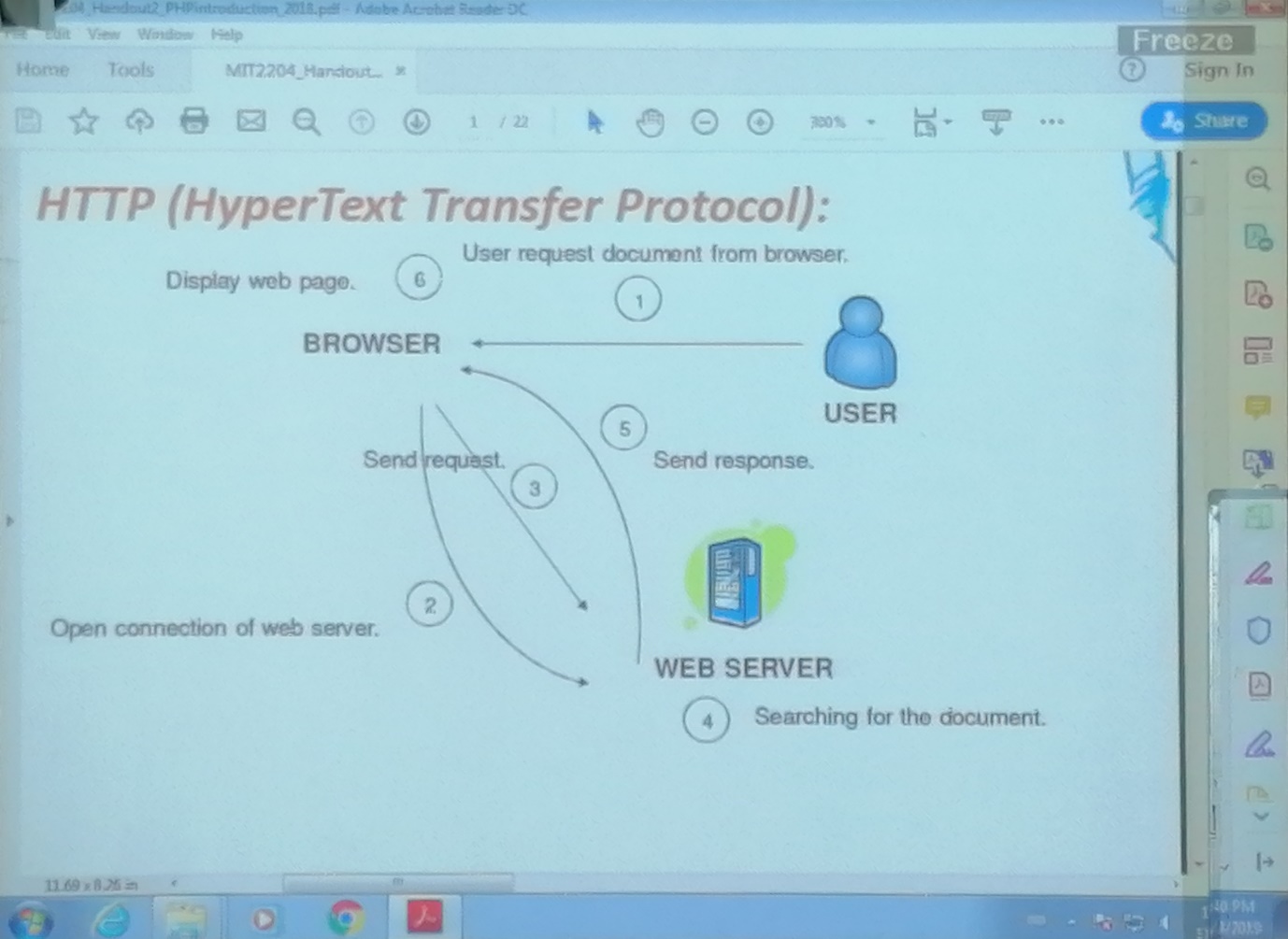
# Terms & definition

## Client

A **client** is the requesting program in a client/ server relationship, EG – the user of a web browser is effectively making client requests for pages from servers on the web.

## Server

I general, a **server** is an application that provides services to the other applications in the same or other computers.



# Programming the web

## What is client side code?

* Software that is downloaded from web server to browser and then executes on the browser client.

## Why client side code?

* Better scalability: less work done on server
* Better performance/ user experience(UX)
* Create UI constructs not inherent in HTML
  + Drop down and pull out menus
  + Tabbed dialogs
* Visual effects, EG – animation
* Data validation

## What is server side code?

* Software that runs on the server, not the client
* Receives input from
  + URL parameter
  + HTML form data
  + Cookies
* Can access server side databases, email servers, files, legacy applications, etc.
* Dynamically builds a custom HTML response for a client

## Why server side code?

* Accessibly
  + You can reach the internet from any browser, any device, anytime, anywhere.
* Manageability
  + Does not require distribution of application code
  + Easy to change code
* Security
  + Security code is not exposed
  + Once user is authenticated, can only allow certain actions.
* Scalability
  + Web based 3 tier architecture can scale out
  + **Three tier** architecture is a client server architecture in which the user interface (presentation), functional process logic (“business rules”), data storages and data access are developed and maintained as independent modules, most often on separate platforms. EG – Codelgnitor

# PHP Introduction

* PHP is a recursive acronym for “PHP: Hypertext Pre-Processor” It is a widely used open source general purpose scripting language that is especially suited for web development and can be embedded into HTML.
* PHP code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was.
* PHP is a server side scripting language
* PHP script are executed on the server
* PHP supports many databases (MySQL, MongoDB, Informix, Oracle, Sybase, Solid, etc.)
* PHP is an open source software

## Apache HTTP server

* This is an open source cross / platform web server software
* Apache wamp server
* Apache xamp server
* Apache lamp server

# What is PHP

* Interpreted language, scripts are parsed at run time rather than compiled beforehand
* Executed on the server side
* Source code can be made not visible by client
* Various built-in functions allow for fast development

## What does PHP code look like?

* Structurally similar to C/ C++
* Supports procedural and object oriented paradigm (to some degree)
* All PHP statements end with a semi colon
* Each PHP script must be enclosed in the reserved PHP tag

<?php

…………

…………

?>

# PHP Introduction

* Some info on MySQL which will be covered in the handout 3.
  + MySQL is a database server
  + MySQL is ideal for both small and large applications
  + MySQL supports standard SQL
  + MySQL complies on a number of platforms

## PHP Hello World

<html>

<head>

<title>PHP Test</title>

</head>

<body>

<?php echo “Hello World” ; ?>

</body>

</html>

## Comment IN PHP

//Single line comment

/\*Multiple

line comment\*/

## PHP variable

<?php

$txt=”Hello World!”;

$x=16;

?>

* In PHP a variable doesn’t not need to be declared before assigning a value to it.
* PHP automatically converts the variable to the correct data type, depending on its value.
* Must be begin from “$” sign.
* Case sensitive.

Predefine libraries E.g.:

* *$GLOBALS*
* *$\_SERVER*
* *$\_REQUEST*
* *$\_POST*
* *$\_GET*
* *$\_FILES*
* *$\_ENV*
* *$\_COOKIE*
* *$\_SESSION*

## PHP URL

<http://localhost:81/>

## PHP Operators

Operators are used to operate on values. There are four classifications of operators:

* Arithmetic
* Assignment
* Comparison
* Logical

## PHP conditional statements

* if statement - executes some code if one condition is true
* if...else statement - executes some code if a condition is true and another code if that condition is false
* if...elseif...else statement - executes different codes for more than two conditions
* switch statement - selects one of many blocks of code to be executed

**Activity 01**

Create a PHP program which finds whether assigned number is an odd number or an even number.

<?php

$x=5;

if ($x%2==0){echo "Even number";}

else{echo "Odd number";}

?>

## PHP arrays

In PHP, there are three types of arrays:

* **Indexed arrays** - Arrays with a numeric index
* **Associative arrays** - Arrays with named keys
* **Multidimensional arrays** - Arrays containing one or more arrays

## PHP numeric arrays

* The index can be assigned automatically (index always starts at 0), like this:

$cars = array("Volvo", "BMW", "Toyota");

* or the index can be assigned manually:

$cars[0] = "Volvo";

$cars[1] = "BMW";

$cars[2] = "Toyota";

## PHP Associative Arrays

* Associative arrays are arrays that use named keys that you assign to them.
* There are two ways to create an associative array:

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

* or:

$age['Peter'] = "35";

$age['Ben'] = "37";

$age['Joe'] = "43";

## PHP Multidimensional Arrays

* In a multidimensional array, each element in the main array can also be an array.
* And each element in the sub-array can be an array, and so on.

|  |  |  |
| --- | --- | --- |
| **Name** | **Stock** | **Sold** |
| Volvo | 22 | 18 |
| BMW | 15 | 13 |
| Saab | 5 | 2 |
| Land Rover | 17 | 15 |

$cars = array  
  (  
  array("Volvo",22,18),  
  array("BMW",15,13),  
  array("Saab",5,2),  
  array("Land Rover",17,15)  
  );

<?php  
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";  
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";  
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";  
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";  
?>

## PHP loops

* while - loops through a block of code as long as the specified condition is true
* do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
* for - loops through a block of code a specified number of times
* foreach - loops through a block of code for each element in an array

PHP loops - foreach

* The foreach loop works only on arrays, and is used to loop through each key/value pair in an array.

foreach ($array as $value)

{code to be executed;}

* For every loop iteration, the value of the current array element is assigned to $value and the array pointer is moved by one, until it reaches the last array element.

Example

<?php  
$colors = array("red", "green", "blue", "yellow");  
  
foreach ($colors as $value) {  
  echo $value."<br>";  
}  
?>

**Activity 02**

Create an array (numeric arrays) which contains following strings.

* One, Tow, Three

Write a PHP script which displays all array elements within a web page.

<?php

$numericArray = array("one", "two", "three");

foreach ($numericArray as $value) {

echo $value."<br/>";}

?>

## PHP form validation

* The PHP superglobals $\_GET and $\_POST are used to collect form-data.
* Why???? – Proper validation of form data is important to protect your form from hackers and spammers.

### PHP simple form validation

* When the user fills out the form above and clicks the submit button, the form data is sent for processing to a PHP file named “welcome.php”. The form data is sent with the HTTP POST method.
* To display the submitted data, you could simply echo all the variables. The “welcome.php”.

## Difference between GET vs POST

* The GET method sends the encoded user information appended to the page request. The page and the encoded information are separated by the ? character.

Ex: http://www.test.com/index.htm?name1=value1&name2=value2

* The POST method transfers information via HTTP headers. The information is encoded as described in case of GET method and put into a header called QUERY\_STRING.

## Server side validation

In the Server Side Validation, the input submitted by the user is being sent to the server and validated using one of server. After the validation process on the Server Side, the feedback is sent back to the client by a new dynamically generated web page. It is better to validate user input on Server Side because you can protect against the malicious users, who can easily bypass your Client Side scripting language and submit dangerous input to the server.

Ex: <form method=”POST” action=”<?php echo htmlspecialchars($\_SERVER[“PHP\_SELF”]);?>”>

* When the form is submitted, the form data is sent with method=”POST”.
* The $\_SERVER[“PHP\_SELF”] is a super global variable that returns the filename of the currently executing script. Therefore the $\_SERVER[“PHP\_SELF”] sends the submitted form data to the page itself, instead of jumping to a different page. This way, the user will get error messages on the same page as the form.

## What is the htmlspecialchars() function?

The htmlspecialchars() function converts special characters to HTML entities. This means that it will replace HTML characters like < and > with &lt; and &gt;. This prevents attackers from exploiting the code by injecting HTML or JavaScript code (Cross-site Scripting attacks) in forms.