Oum El Bouaghi University
Faculty of Exact Sciences, Natural and Life Science
Department of Mathematics and Computer Science

Advanced Web programming

PHP

Dr. Saighi Asma

Concerned students

Faculty	Department	Level	Speciality
ESNL	MI	B3	ISSE

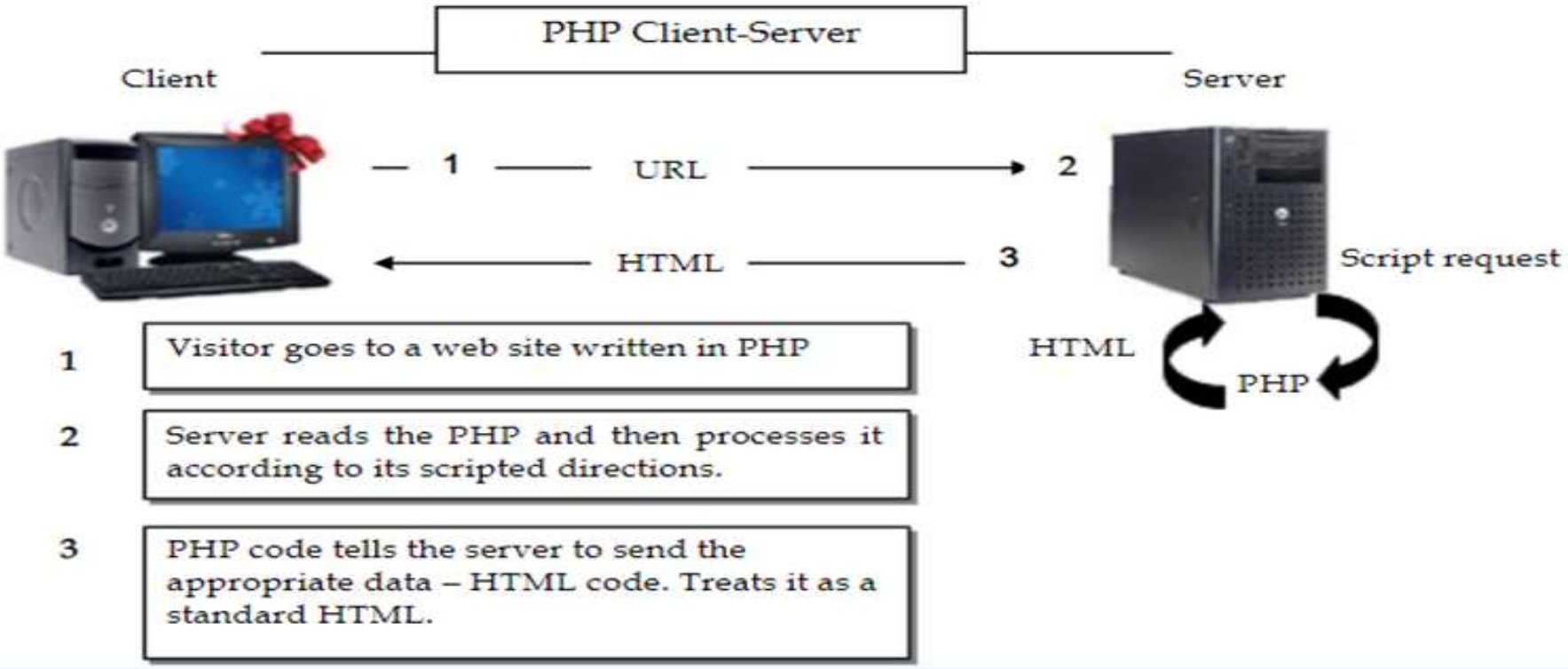
What is PHP?

- Is a **server** scripting language.
- Was designed to **create dynamic websites**.
- **Generates HTML code** which is sent to the client.
- Can work alone, but it is more useful when **combined with MySQL**.
- PHP and MySQL are **open-source software**.
- HTML and CSS have no competitors because **they are standards**.
- **PHP has many competitors**: ASP.NET, Ruby on Rails, Django, Java, and JSP (Java Server Pages).

  - *There is no best choice*, the language that one has programming knowledge is the best.
  - *PHP is a good choice for beginners* in this sense.

How PHP works?

- PHP: **Server side language.**
- The code written in PHP resides on a host computer (i.e. server).
- The server sends Web pages to visitor's requests: clients, your



- A web server: software application or a system that **serves web pages**, content, or services to users over the internet or a local network.
- It **handles** incoming network **requests** from clients (such as web browsers).
- **Response: delivers requested web resources**, which could include HTML pages, images, CSS files, JavaScript, and other multimedia content.

- There are numerous web servers available. Some of the most commonly used web servers include:

1. **Apache HTTP Server:** Among the oldest and most utilized web servers, known for its flexibility, robustness, and extensive module support.

2. **Nginx:** Highly performant, often used as a reverse proxy, load balancer, and for serving static content due to its efficiency in handling concurrent connections.

3. **Microsoft Internet Information Services (IIS):** Developed by Microsoft for Windows servers, IIS is integrated with Windows environments and supports various web protocols.

4. **LiteSpeed Web Server:** Similar to Apache but known for its speed and ability to handle high loads efficiently, often used in high-traffic websites and hosting environments.

5. **Lighttpd (Lighty):** Known for its speed and low resource usage, Lighttpd is often used for serving static content and as a reverse proxy.

6. **Node.js:** While not a traditional web server, Node.js enables JavaScript to run on the server side, facilitating the creation of web servers using JavaScript.

- **Note:** These servers vary in terms of *performance*, *resource usage*, *ease of configuration*, etc. Developers often choose a web server based on the specific requirements of their projects.

What will you need?

Install Web server

Objective: **simplify the process of setting up a local web development environment.**

- **WAMP** – Windows, Apache, MySQL, PHP
- **LAMP** – Linux, Apache, MySQL, PHP
- **MAMP** – Mac, Apache, MySQL, PHP
- **XAMPP** – Cross-platform (X), can run on Windows, Linux, and macOS. It includes Apache, MariaDB, PHP, Perl

PHP Syntax

- PHP code is inserted into the HTML code.
- The PHP code snippets are **the dynamic parts** of the page.
- These pages are saved with the **.php** extension.
- To use of PHP code, introduce a tag: **<?php and ?>**, placed anywhere.
- The echo statement inserts text (within quotes) into the web page:

<?php echo " This is text \" only\""; ?>

PHP Syntax

- Statement Separators: **semicolon**.
- Semicolon must never be forgotten, otherwise, you get an error message: “error parse”.
- Comments are two types:
  - Single-line comments: //
  - Multiple line comments: /*closed by*/

Types, Variables, and Operators

- Strings **(string)**: A string written between " " or ' '.
- Integers **(int)**: 1,2,3,4… or -1, -2, -3,…
- Booleans **(bool):** true or false.
- Null **(NULL):** Absence of type.
- The **$** symbol precedes the name of a variable.

```php
$variable = 'My "name" is Brahim';
$age_du_visiteur = 17;
$poids = 57.3;
$je_suis_bon_en_php = false;
$pas_de_valeur = NULL;

echo "The visitor is $age_du_visiteur years old";
echo 'The visitor is ' . $age_du_visiteur . ' years old';
```

- **Basic operations**: addition (+, ++, +=), soustraction (-, --, -=), Multiplication (*, *=), Division (/, /=), modulo (%), …

- **Test Structures**

```
if (Condition1) // Comparison symbols are used: ==, <, >, <=, >=, !=
{
}
elseif (Condition2) // and multiple conditions with AND (or &&) and OR (or ||)
{
}
else
{
}

switch ($variable)
{
    case 0:
        // operation 1; break;
        // ...
    default:
        // operation n;
}
```

- **Loops**

```
while (Condition)
{
   // instructions
}

for (initialization; Condition; Suite)
{
   // instructions
}
```

- **Arrays**

```
$prenoms = array ('Brahim', 'Karim');
// or
$prenoms[0] = 'Brahim';
$prenoms[1] = 'Karim';

// To display:
echo $prenoms[1];
```

Functions

- **Some important functions**

Function	Description	Example
strlen	length of a string	strlen($phrase);
str_replace	Search and replace	str_replace('b', 'p', 'bim bam boum');
str_shuffle	Shuffle letters	str_shuffle($chaine);
strtolower	Convert to lowercase	strtolower($chaine);
strtoupper	Convert to uppercase	strtoupper($chaine);
date	Retrieve the date	$jour = date('d'); $mois = date('m'); $year = date('Y'); $heure = date('H'); $minute = date('i');

- **Personal functions**

```
function SayHello($name)
{
   echo 'Hello ' . $name . ' !<br />';
}
SayHello('Brahim');
```

Classes and Objects

- Class: Defined by the keyword **class**, the name of the class and curly braces (**{}**);
- Its properties and methods go **inside the braces**;
- **Two methods: set_name()** and **get_name()** to set and get the property **$name**.
- Objects of a class are created using the **new** keyword.

```php
<?php
class Fruit {
  // Properties
  public $name;
  public $color;

  // Methods
  function set_name($name) {
    $this->name = $name;
  }
  function get_name() {
    return $this->name;
  }
}
$apple = new Fruit();
$banana = new Fruit();
$apple->set_name('Apple');
$banana->set_name('Banana');

echo $apple->get_name();
echo "<br>";
echo $banana->get_name();
?>
```

- **Error handling:**
  - When an **error** occurs, PHP shows a **blanc page**.
  - PHP needs to be configured to **display errors**:
    - "**php.ini**" file;
    - Create **.php** file containing the **phpinfo()** function **(<?php phpinfo(); ?>)**;
    - Find the line "**Loaded Configuration File**";
    - Open the indicated file and modify it;
    - Change the values of the configuration keys *error_reporting* and *display_errors* to **E_ALL**.

- **File Upload Management:**

  - File Upload via **POST Method**.
  - **Documentation:** *https://www.php.net/manual/fr/features.file-upload.post-method.php*

  - **Example: File Upload Form**

```
<form enctype="multipart/form-data" action="_URL_" method="post">
<!-- MAX_FILE_SIZE must precede the file input field -->
<input type="hidden" name="MAX_FILE_SIZE" value="30000" />
<!– The name of the input element determines the name in the $_FILES array-->
Send this file: <input name="userfile" type="file" />
<input type="submit" value="Envoyer le fichier" />
</form>
```

  - URL in the previous example must be replaced and pointed to a PHP file.

- **Existing File on the Server:**
  - PHP allows for saving data into files on the server's hard disk.
  - File manipulation:

```php
$monfichier = fopen('compteur.txt', 'r+'); // opens the file
// file operations would be performed here…
fclose($monfichier);  // close the file
```

Mode	Explanation
r	Opens the file in read-only mode.
r+	Opens the file in read and write mode.
a	Opens the file in write only mode. If the file does not exist , it is automatically created.
a+	Opens the file in write and read mode.

Features

- **Existing File on the Server: Read and Modify a file**

  - Read character by character with the **fgetc** function.
  - Read line by line with **fgets**.
  - Write to a file with **fputs**.
  - Move the cursor to a position in the file with **fseek**.

- **Example:** Counting the number of visits to a page.

```php
<?php
$monfichier = fopen('compteur.txt', 'r+');
$pages_vues = fgets($monfichier);
$pages_vues += 1;
fseek($monfichier, 0);
fputs($monfichier, $pages_vues);
fclose($monfichier);
echo '<p>This page has been visited ' . $pages_vues . ' times!</p>';
?>
```

- **Using Remote Files**
  - **Example:** Open a file on a remote web server, parse results and use it in a database query**.**
  - **Example 1:** To get a remote page title.

```php
<?php
$file = fopen("http://www.example.com/", "r");
if (!$file) {
    echo "<p>Unable to read the page.\n";
    exit;
}

while (!feof($file)) {
    $line = fgets($file, 1024);
    /* This works only if Title tags are correctly used */
    if (preg_match("@\<title\>(.*)\</title\>@i", $line, $out)) {
        $title = $out[1];
        break;
    }
}
fclose($file);
?>
```

- **Using Remote Files**
  - **Example 2: storing data on a remote server**

```
<?php
$file = fopen("ftp://ftp.example.com/incoming/outputfile", "w");
if (!$file) {
   echo "<p>Unable to open the remote file for writing.\n";
   exit;
}
/* Writing data. */
fputs($file, $_SERVER['HTTP_USER_AGENT'] . "\n");
fclose($file);
?>
```

- **Connection Management**
  - **The status of connections is internally maintained by PHP:**
    - **0 – NORMAL:** PHP script is running
    - **1 – ABORTED:** Remote client disconnects
    - **2 – TIMEOUT:** Maximum execution time of PHP exceeded
    - **3 – ABORTED and TIMEOUT**

- **Persistent Database Connection to SQL**

  - PHP ensures that there is no other **identical** connection (same hostname, username, and password).

  - **Using persistent connections:** ways to use PHP to generate pages:

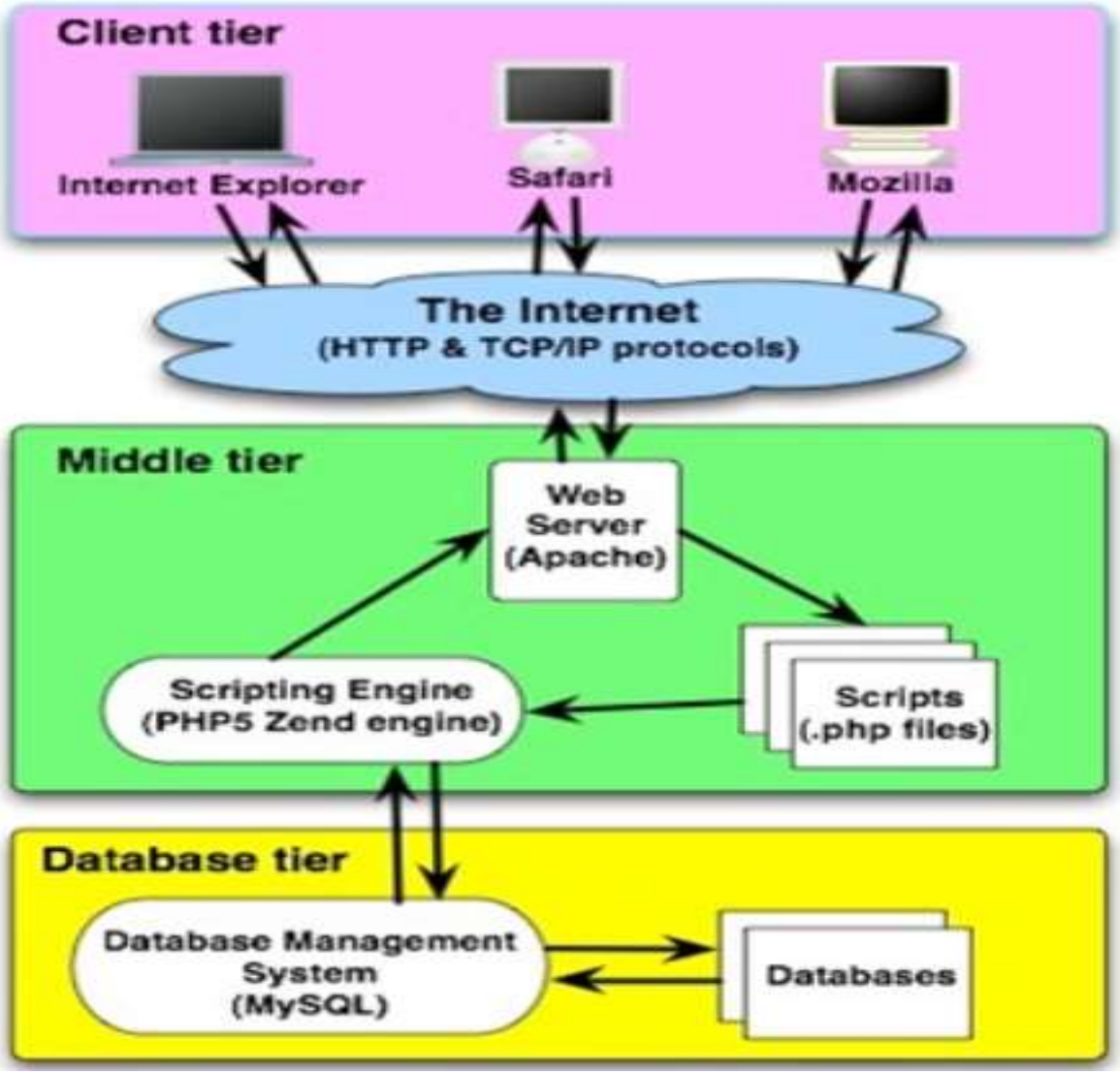
    1. Use PHP as a common **Gateway Interface** (CGI).
    2. Run PHP as a **module** on a multi-process server (Apache).
    3. Use PHP as a **multithreaded server module**.

  - Persistent connections are good way to **speed up access** to an **SQL database.**

Features

- **Session Management:**
  - Not easy with GET and POST to transmit variables across all pages of a site for a duration of a visitor's presence.
  - **Principle:** Visitor arrive to a site ➡ Request to create a session.
  - PHP generates a unique number 'session ID'.
  - The session is closed when the visitor logs out of the site.

- **Web based 3-tier Architecture**



References

- https://slideplayer.com/slide/5180908/
- https://www.w3schools.com/php/php_syntax.asp
- https://assets.ctfassets.net/nkydfjx48olf/5qFMF3mvitLMahX67i7iOb/028229996c13cbc27a0538f055a41b46/php_cookbook.pdf