PHP

Hello Hackers , I am Adham Elhansye | 0xMr_Fr3on | I have been studying Cyber Security since I was 14 ,, Here's simple summary of PHP to help penetration testers understand what happens in the backend of websites or any system that uses the PHP Language .

We Will Start now …

1. What is PHP Langauge ?

PHP stands for Hypertext Preprocessor, and it is a widely-used open-source server-side scripting language. Originally created by Rasmus Lerdorf in 1994, PHP is designed for web development and can be embedded into HTML code. It is a powerful and versatile language that is particularly well-suited for building dynamic web pages and applications.PHP stands for Hypertext Preprocessor, and it is a widely-used open-source server-side scripting language. Originally created by Rasmus Lerdorf in 1994, PHP is designed for web development and can be embedded into HTML code. It is a powerful and versatile language that is particularly well-suited for building dynamic web pages and applications.

1. PHP File

When you create PHP file , start :

```
<?php
//code
?>
```

---

2. Print in PHP

```
<?php
echo “Hello”;
?>
```

Note:

Don’t Forget semicolon ;

---

3. Variables :

```
<?php
$var = “Hello”;
echo $var;
?>
```

Don’t Forget semicolon ;

4.String Functions :

- *substr — Return part of a string*
- *strpos — Find the position of the first occurrence of a substring in a string*
- *strlen — Get string length*
- *str_replace — Replace all occurrences of the search string with the replacement string*
- *str_repeat — Repeat a string*
- *ucwords — Uppercase the first character of each word in a string*
- *ucfirst — Make a string's first character uppercase*
- *str_word_count — Return information about words used in a string*
- *strtolower — Make a string lowercase*
- *strtoupper — Make a string uppercase*

Example :

```
<?php
$var = “CyberSecurity”;
echo substr($var, '4').'</br>';
echo strpos($var, 'e').'</br>';
echo strlen($var).'</br>';
echo str_replace("mad", "adham", $var).'</br>';
```

```
echo str_repeat($var, 5).'</br>';
echo ucwords($var).'</br>';
echo str_word_count($var).'</br>';
echo ucfirst($var).'</br>';
echo strtolower($var).'</br>';
echo strtoupper($var).'</br>';
?>
```

Note:

*
*

So that each command is printed on a new line

---

5. IF Function :

```
<?php
if ($grade >= 50)
echo "Passed";
else
echo "Failed";
?>
```

---

6. Switch Case :

The `switch` statement is similar to a series of IF statements on the same expression. In many occasions, you may want to compare the same variable (or expression) with many different values, and execute a different piece of code depending on which value it equals to. This is exactly what the `switch` statement is for.

```
<?php
$mon=10;
switch ($mon) {
    case 1: echo "jan"; break;
    case 2: echo "feb";  break;
    case 3: echo "mar"; break;
    case 4: echo "apr"; break;
    case 5: echo "may"; break;
    case 6: echo "jun"; break;
    case 7: echo "july"; break;
    case 8: echo "aug"; break;
    case 9: echo "sep"; break;
    case 10: echo "oct"; break;
    case 11: echo "nov"; break;
    default: echo "check your input ";
?>
```

7. Loops :

```
<?php
while ($num<=100) {
        echo $num."<br>";
        $num++
}


// for loop , continue , break
for ($a=10;$a<=100;$a++){
        echo $a."<br>";
}

$x=1;
for (;;){
        echo "Fsociety"."<br>";
        $x++;
        if($x > 5) break;

}
```

```
for ($id=0; $id <=10 ; $id++) {
        if ($id == 4)continue;
        echo "id= ".$id."<br>";
}
?>
```

8. User Defined Functions :

```
function Adham(){
        echo " My Name is Adham <br>";
}
adham();

function add($a,$b){
        echo $a+$b;
}
add(5,3);
```

---

8. Local and Global Variables :

$GLOBALS — References all variables available in global scope

```
<?php
function test(){
        $x=9;
        echo " Local Scope is ".$x.'<br>';
```

```
        echo " Global Scope is   ".$GLOBALS['x'];
}
test();
?>
```

---

9. Arrays :

```
<?php
$mon1='jan';
$mon2='feb';
$mon3='mar';
$mon4='april';
$mon5='may';
$mon6='jun';
$monts=array('jan','feb','mar','april','may','june','july','aug','sept','oct','nov','dec');
echo $monts[1];
print_r($monts)// arrays
$mon1='jan';
$mon2='feb';
$mon3='mar';
$mon4='april';
$mon5='may';
```

```
$mon6='jun';
$monts=array('jan','feb','mar','april','may','june','july','aug','sept','oct','nov','dec');
echo $monts[1];
print_r($monts)
?>
```

---

11. Foreach :

The `foreach` construct provides an easy way to iterate over arrays. `foreach` works only on arrays and objects

```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
$value = $value * 2;
}
?>
```

12. Globals :

$_SERVER — Server and execution environment information

```
echo $_SERVER['PHP_SELF'].'<br>';
echo $_SERVER['SERVER_NAME'].'<br>';
echo $_SERVER['HTTP_USER_AGENT'].'<BR>';
echo $_SERVER['HTTP_HOST'].'<BR>';
echo $_SERVER['REMOTE_ADDR'].'<BR>';
print_r($_SERVER).'<BR>';
```

$_GET — HTTP GET variables

```
<?php
echo 'Hello ' .
htmlspecialchars($_GET["name"]) . '!';
?>
```

$_POST — HTTP POST variables

```
<?php
echo 'Hello ' .
htmlspecialchars($_POST["name"]) . '!';
?>
```

$_FILES — HTTP File Upload variables

There is more Globals Variables , so I recommend visit this site :

“ https://www.php.net/manual/en/reserved.variables.php “

13. Include and Require :

INCLUDE

The `include` expression includes and evaluates the specified file.

REQUIRE

*`require` is identical to include except upon failure it will also produce a fatal **`E_COMPILE_ERROR`** level error. In other words, it will halt the script whereas include only emits a warning (**`E_WARNING`**) which allows the script to continue.*

Example :

```
<!-- contents of required_file.php -->
<?php
    $message = "Hello, this is from required_file.php!";
?>

<!-- main_file.php -->
<?php
    require 'required_file.php'; // include the contents of required_file.php

    echo $message; // output: Hello, this is from required_file.php!
?>
```

14. CTYPE Function :

ctype_alnum — Check for alphanumeric character(s)

ctype_alpha — Check for alphabetic character(s)

ctype_cntrl — Check for control character(s)

ctype_digit — Check for numeric character(s)

ctype_graph — Check for any printable character(s) except space

ctype_lower — Check for lowercase character(s)

ctype_print — Check for printable character(s)

ctype_punct — Check for any printable character which is not whitespace or an alphanumeric character

ctype_space — Check for whitespace character(s)

ctype_upper — Check for uppercase character(s)

ctype_xdigit — Check for character(s) representing a hexadecimal digit

---

15 .Filters :

String Filters

- Conversion Filters
- Compression Filters
- Encryption Filters

The following is a list of a few built-in stream filters for use with stream_filter_append(). Your version of PHP may have more filters (or fewer) than those listed here.

It is worth noting a slight asymmetry between stream_filter_append() and stream_filter_prepend(). Every PHP stream contains a small *read buffer* where it stores blocks of data retrieved from the filesystem or other resource in order to process data in the most efficient manner. As soon as data is pulled from the resource into the stream's internal buffer, it is immediately processed through any attached filters whether the PHP application is actually ready for the data or not. If data is sitting in the read buffer when a filter is *appended*, this data will be immediately processed through that filter making the fact that it was sitting in the buffer seem transparent. However, if data is sitting in the read buffer when a filter is *prepended*, this data will *NOT* be processed through that filter. It will instead wait until the next block of data is retrieved from the resource.

For a list of filters installed in your version of PHP use stream_get_filters().

16. Sessions :

Session Handling ¶

- Introduction
- Installing/Configuring
  - Requirements
  - Installation
  - Runtime Configuration
  - Resource Types
- Predefined Constants
- Examples
  - Basic usage
  - Passing the Session ID
  - Custom Session Handlers
- Session Upload Progress
- Sessions and Security

        - SessionIdInterface::create_sid — Create session ID
    - SessionUpdateTimestampHandlerInterface — The SessionUpdateTimestampHandlerInterface interface
        - SessionUpdateTimestampHandlerInterface::updateTimestamp — Update timestamp
        - SessionUpdateTimestampHandlerInterface::validateId — Validate ID

Resources :

https://www.php.net

https://www.youtube.com/watch?v=3YEZsMIETiw&list=PLxofFKbtL6_1W1pzynkwS2bdVfwjm22WK

https://www.youtube.com/watch?v=xcg9qq6SZ0w&list=PLDoPjvoNmBAy41u35AqJUrI-H83DObUDq