

Sistemas Multimédia

TPW-II

Tecnologia e Prática da Web II

13

PHP

□ PHP

- O que pode fazer
- Para usar o PHP
- Sintaxe
- Variáveis
- Exemplo

PHP



□ PHP: PHP Hypertext Preprocessor

- PHP é uma linguagem que permite criar sites web dinâmicos, possibilitando uma interação com o utilizador através de formulários, parâmetros da URL e links.
- A diferença de PHP em relação a linguagens semelhantes a Javascript é que o código PHP é **executado no servidor**, sendo enviado para o cliente apenas HTML puro.
- Desta maneira é possível interagir **com base de dados e aplicações existentes no servidor**, com a vantagem de **não expor o código fonte para o cliente**. Isso pode ser útil quando o programa processa dados confidenciais.
- O código **PHP** fica **embutido** no **próprio HTML**.

□ PHP: O que pode fazer

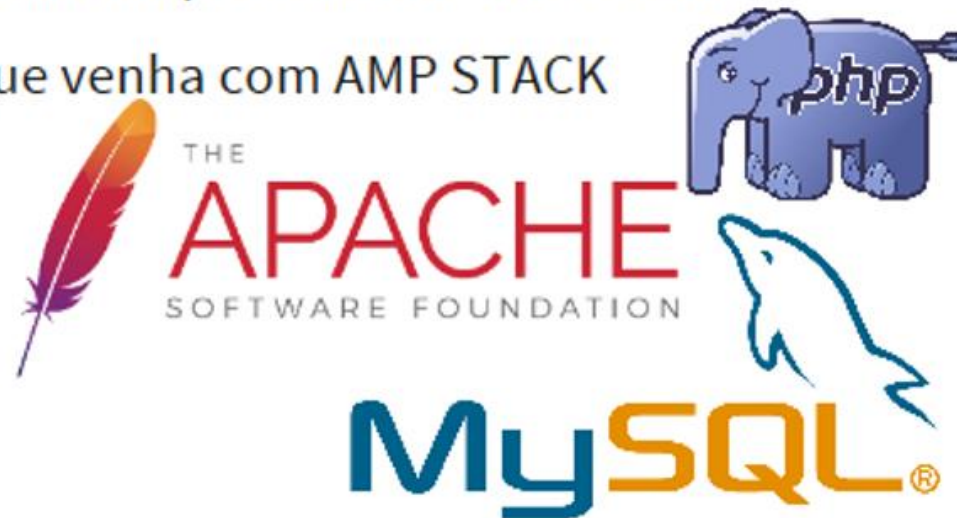
- Recolher dados de um formulário, gerar páginas dinamicamente ou enviar e receber cookies.
- Suporte a um grande número de base de dados, como dBase, Interbase, mSQL, MySQL, Oracle, Sybase, PostgreSQL e outros.

- ❑ Para usar o PHP

AMP STACK

- Para usar php no computador pessoal será necessário instalar um software que venha com AMP STACK

- Apache
- MySQL
- PHP



- Para usar o PHP

AMP STACK

- **XAMP**

<https://www.apachefriends.org/index.html>

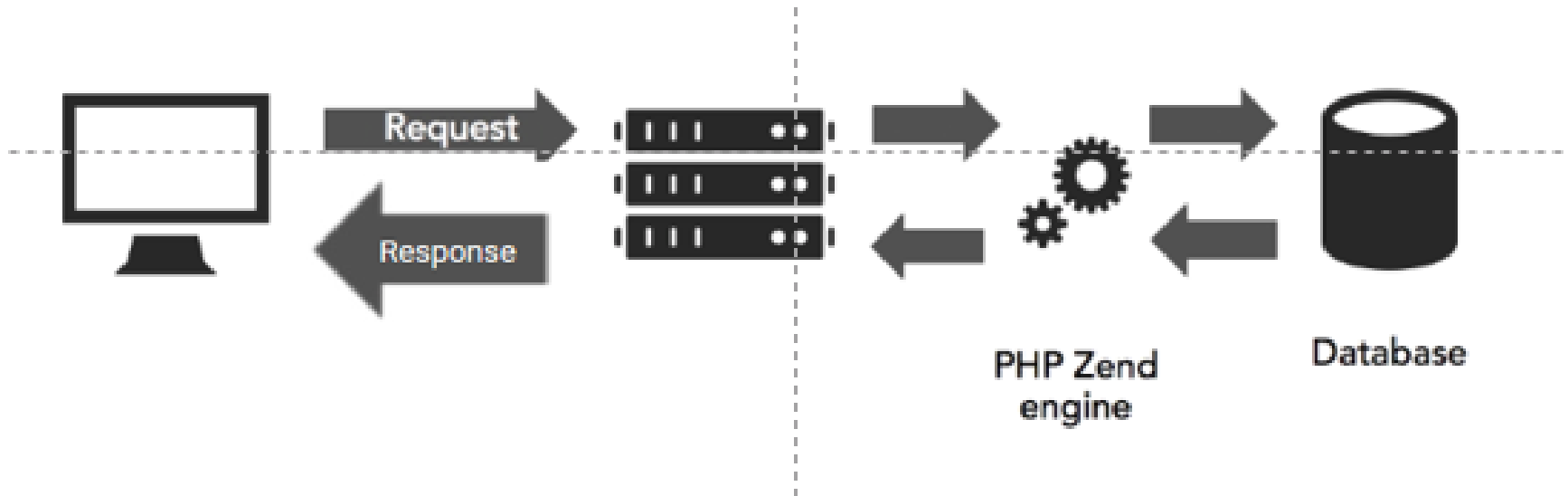
- **WAMP**

<http://www.wampserver.com/en/>

- **AMPPS**

<http://www.ampps.com/>

❑ Modelo de funcionamento



PHP: sintaxe

□ Sintaxe base

- O código PHP fica embutido no próprio HTML. O interpretador identifica quando um código é PHP pelas seguintes tags:

```
<?php  
    comandos  
?>
```

```
<script language="php">  
    comandos  
</script>
```

```
<?  
    comandos  
?>
```

- Usar a extensão de ficheiro **.php**
- Usar **.php** mesmo em páginas com apenas HTML
- Entre cada instrução em PHP é preciso utilizar o ponto-e-vírgula ;

PHP: variáveis

❑ Variáveis

- No PHP, variáveis começam sempre com \$
- Letras, números e *underscore* são permitidos
- Primeiro caractere depois do \$ não pode ser um número
- Sem espaços, hífen, ponto final ou caracteres especiais
- Nomes são sensíveis a maiúsculas e minúsculas

PHP: exemplo

□ Exemplo

```
<?php
```

```
//define a variavel texto e escreve no ecra o seu valor  
$texto = "Primeiro Script";  
echo $texto;
```

```
?>
```

PHP: estruturas

□ Comentários

```
<!DOCTYPE html>
<html>
<body>

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```


□ Variáveis

```
<?php
```

```
    $txt = "Hello world!";
```

```
    $x = 5;
```

```
    $y = 10.5;
```

```
?>
```

```
<?php
```

```
    $txt = "W3Schools.com";
```

```
    echo "I love $txt!";
```

```
?>
```

```
<?php
```

```
    $txt = "W3Schools.com";
```

```
    echo "I love " . $txt . "!";
```

```
?>
```

□ Variáveis

```
<?php
```

```
$x = 5; // global scope
```

```
function myTest() {  
    // using x inside this function will generate an error  
    echo "<p>Variable x inside function is: $x</p>";  
}  
myTest();
```

```
echo "<p>Variable x outside function is: $x</p>";
```

```
?>
```

```
<?php
```

```
function myTest() {  
    $x = 5; // local scope  
    echo "<p>Variable x inside function is: $x</p>";  
}  
myTest();
```

```
// using x outside the function will generate an error  
echo "<p>Variable x outside function is: $x</p>";
```

```
?>
```

□ Variáveis

```
<?php
```

```
$x = 5;  
$y = 10;
```

```
function myTest() {  
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];  
}
```

```
myTest();  
echo $y; // outputs 15
```

```
?>
```

```
<?php
```

```
function myTest() {  
    static $x = 0;  
    echo $x;  
    $x++;  
}
```

```
myTest();  
myTest();  
myTest();
```

```
?>
```

□ Tipos de dados

■ PHP is a Loosely Typed Language

```
<?php
```

```
$x = "Hello world!";  
$y = 'Hello world!';
```

```
echo $x;  
echo "<br>";  
echo $y;
```

```
?>
```

PHP String

```
<?php
```

```
$x = 5985;  
var_dump($x);
```

```
?>
```

PHP Integer

```
<?php
```

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

```
?>
```

PHP Array

var_dump(): returns the data type and value

□ Tipos de datos

```
<?php
```

```
class Car {  
    function Car() {  
        $this->model = "VW";  
    }  
}
```

PHP Object

```
// create an object  
$herbie = new Car();  
  
// show object properties  
echo $herbie->model;
```

```
?>
```

PHP NULL Value

```
<?php
```

```
$x = "Hello world!";  
$x = null;  
var_dump($x);
```

```
?>
```

□ Constantes

```
<?php
    define("GREETING", "Welcome to W3Schools.com!");
    echo GREETING;
?>
```

```
<?php
    define("cars", [
        "Alfa Romeo",
        "BMW",
        "Toyota"
    ]);
    echo cars[0];
?>
```

❑ Operadores

PHP Arithmetic Operators

Operator	Name	Example	Result
+	Addition	$\$x + \y	Sum of $\$x$ and $\$y$
-	Subtraction	$\$x - \y	Difference of $\$x$ and $\$y$
*	Multiplication	$\$x * \y	Product of $\$x$ and $\$y$
/	Division	$\$x / \y	Quotient of $\$x$ and $\$y$
%	Modulus	$\$x \% \y	Remainder of $\$x$ divided by $\$y$
**	Exponentiation	$\$x ** \y	Result of raising $\$x$ to the $\$y$ 'th power

PHP Assignment Operators

Assignment	Same as...	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \% = y$	$x = x \% y$	Modulus

❑ Operadores

PHP Comparison Operators

Operator	Name	Example	Result
==	Equal	\$x == \$y	Returns true if \$x is equal to \$y
===	Identical	\$x === \$y	Returns true if \$x is equal to \$y, and they are of the same type
!=	Not equal	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Not equal	\$x <> \$y	Returns true if \$x is not equal to \$y
!==	Not identical	\$x !== \$y	Returns true if \$x is not equal to \$y, or they are not of the same type
>	Greater than	\$x > \$y	Returns true if \$x is greater than \$y
<	Less than	\$x < \$y	Returns true if \$x is less than \$y
>=	Greater than or equal to	\$x >= \$y	Returns true if \$x is greater than or equal to \$y
<=	Less than or equal to	\$x <= \$y	Returns true if \$x is less than or equal to \$y
<=>	Spaceship	\$x <=> \$y	Returns an integer less than, equal to, or greater than zero, depending on if \$x is less than, equal to, or greater than \$y. Introduced in PHP 7.

❑ Operadores

PHP Increment / Decrement Operators

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments \$x by one, then returns \$x
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x
<code>\$x--</code>	Post-decrement	Returns \$x, then decrements \$x by one

PHP Logical Operators

Operator	Name	Example	Result
<code>and</code>	And	<code>\$x and \$y</code>	True if both \$x and \$y are true
<code>or</code>	Or	<code>\$x or \$y</code>	True if either \$x or \$y is true
<code>xor</code>	Xor	<code>\$x xor \$y</code>	True if either \$x or \$y is true, but not both
<code>&&</code>	And	<code>\$x && \$y</code>	True if both \$x and \$y are true
<code> </code>	Or	<code>\$x \$y</code>	True if either \$x or \$y is true
<code>!</code>	Not	<code>!\$x</code>	True if \$x is not true

❑ Operadores

PHP String Operators

Operator	Name	Example	Result
.	Concatenation	\$txt1 . \$txt2	Concatenation of \$txt1 and \$txt2
.=	Concatenation assignment	\$txt1 .= \$txt2	Appends \$txt2 to \$txt1

PHP Array Operators

Operator	Name	Example	Result
+	Union	\$x + \$y	Union of \$x and \$y
==	Equality	\$x == \$y	Returns true if \$x and \$y have the same key/value pairs
===	Identity	\$x === \$y	Returns true if \$x and \$y have the same key/value pairs in the same order and of the same types
!=	Inequality	\$x != \$y	Returns true if \$x is not equal to \$y
<>	Inequality	\$x <> \$y	Returns true if \$x is not equal to \$y
!==	Non-identity	\$x !== \$y	Returns true if \$x is not identical to \$y

PHP Conditional Assignment Operators

Operator	Name	Example	Result
?:	Ternary	\$x = <i>expr1</i> ? <i>expr2</i> : <i>expr3</i>	Returns the value of \$x. The value of \$x is <i>expr2</i> if <i>expr1</i> = TRUE. The value of \$x is <i>expr3</i> if <i>expr1</i> = FALSE
??	Null coalescing	\$x = <i>expr1</i> ?? <i>expr2</i>	Returns the value of \$x. The value of \$x is <i>expr1</i> if <i>expr1</i> exists, and is not NULL. If <i>expr1</i> does not exist, or is NULL, the value of \$x is <i>expr2</i> . Introduced in PHP 7

□ PHP if...else...elseif

```
<?php
```

```
    $t = date("H");
```

```
    if ($t < "20") {  
        echo "Have a good day!";  
    } else {  
        echo "Have a good night!";  
    }
```

```
?>
```

```
<?php
```

```
    $t = date("H");
```

```
    if ($t < "10") {  
        echo "Have a good morning!";  
    } elseif ($t < "20") {  
        echo "Have a good day!";  
    } else {  
        echo "Have a good night!";  
    }
```

```
?>
```

□ PHP if...else...elseif

```
<?php
    $t = date("H");

    if ($t < "10") {
        echo "Have a good morning!";
    } elseif ($t < "20") {
        echo "Have a good day!";
    } else {
        echo "Have a good night!";
    }

?>
```

□ PHP while Loop

```
<?php
    $x = 1;

    while($x <= 5) {
        echo "The number is: $x <br>";
        $x++;
    }
?>
```

□ PHP do while Loop

```
<?php
    $x = 1;

    do {
        echo "The number is: $x <br>";
        $x++;
    } while ($x <= 5);

?>
```

□ PHP do while Loop

```
<?php
    for ($x = 0; $x <= 10; $x++) {
        echo "The number is: $x <br>";
    }
?>
```

□ PHP foreach Loop

```
<?php
    $colors = array("red", "green", "blue", "yellow");

    foreach ($colors as $value) {
        echo "$value <br>";
    }
?>
```


□ PHP Functions

```
<?php
function writeMsg() {
    echo "Hello world!";
}

writeMsg(); // call the function
?>
```

□ PHP Arrays

```
<?php
$cars = array("Volvo", "BMW", "Toyota");
echo "I like " . $cars[0] . ", " . $cars[1] . " and " . $cars[2] . ".";
?>
```

PHP: referencias

- ❑ <https://www.w3schools.com/php/>
- ❑ <https://www.php.net/docs.php>