

"PHP: Hypertext Preprocessor"

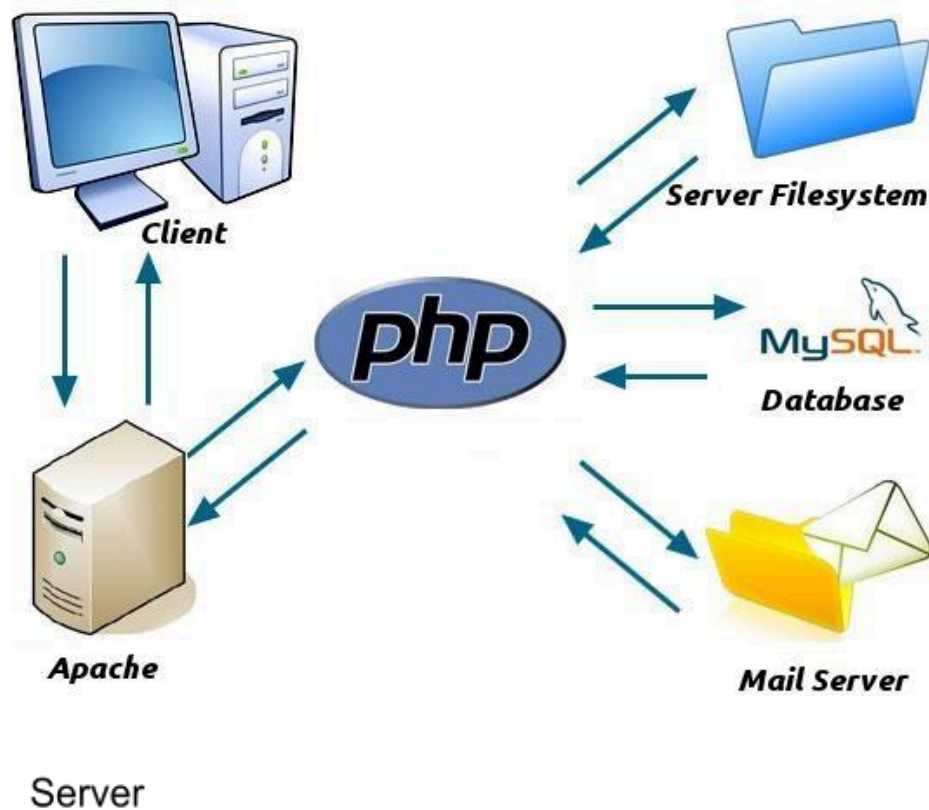
Características PHP

- PHP é uma linguagem de script que executa **no lado do servidor**
- Scripts PHP são **executados no servidor**
- PHP é livre para baixar e usar
- PHP é executado em várias plataformas (Windows, Linux, Unix, Mac OS X, etc.)
- PHP é compatível com quase todos os **servidores web** usados atualmente (Apache, IIS, etc.)
- PHP suporta uma **ampla gama de bancos de dados**

Funcionalidades PHP

- PHP pode gerar conteúdo de **página dinâmico**
- PHP pode criar, abrir, ler, escrever, excluir e fechar **arquivos no servidor**
- PHP pode **coletar dados de formulário**
- PHP pode enviar e receber cookies
- PHP pode **adicionar, excluir, modificar dados em seu banco de dados**
- PHP pode ser usado para controlar o acesso do usuário
- PHP pode criptografar dados

Arquitetura PHP (Back End).



PHP Versões 5 e 7,8

Versão 7 e 8 apresenta performance supostamente duas vezes superior à versão 5.

WAMP (<http://www.wampserver.com/en/>).

WampServer é um software que efetua a instalação automática de um conjunto de softwares no computador, de modo a facilitar a configuração de um software interpretador de scripts local e um banco de dados no sistema windows. Ele permite que a criação de aplicativos da Web com o **Apache2**, **PHP** e um banco de dados **MySQL**. Paralelamente, o PhpMyAdmin (<http://localhost/phpmyadmin>) permite gerenciar facilmente seus bancos de dados.



Utilização WAMP.

- O diretório "www" é criado automaticamente (geralmente c:\wamp64\www)
- Crie um subdiretório em "www" e coloque seus arquivos .php dentro (colocar também HTML, js, css).
- Clique no link "localhost" no menu do WampSever ou abra seu navegador de internet e acesse a URL: <http://localhost>

Configurar Visual Studio Code:

Menu: **Run -> Add Configuration**

launch.json

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Launch Chrome",
      "request": "launch",
      "type": "chrome",
      "url": "http://localhost/${workspaceFolderBasename}/${relativeFile}",
      "webRoot": "${workspaceFolder}"
    },
    {
      "type": "msedge",
      "request": "launch",
      "name": "Launch Edge",
      "url": "http://localhost/${workspaceFolderBasename}/${relativeFile}",
      "webRoot": "${workspaceFolder}"
    }
  ]
}
```

Conceitos Básicos de PHP

Um script PHP pode ser colocado em qualquer parte do documento HTML. Um script PHP começa com `<?php` e termina com `?>`. **A extensão de arquivo padrão para arquivos PHP é ".php"(como .html não funciona)**. Um arquivo PHP normalmente contém tags HTML e algum código de script PHP.

```
<!DOCTYPE html>
<html>
<body>
<h1>My first PHP page</h1>

<?php
//comentários em PHP são iguais a javascript

echo "<h2>PHP is Fun!</h2>";
echo "Hello World! <br>";
?>
</body>
</html>
```

Variáveis

No PHP, uma variável começa com o sinal **\$**, seguido pelo nome da variável. As variáveis em PHP são **case-sensitive**. PHP é uma linguagem **fracamente tipada**.

```
<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;
?>
```

Saída de variáveis de texto

```
<?php
$txt = "IFRS Erechim";
echo "I love " . $txt . "!" . "<br>"; //"."->concatenar string
ou
echo "I love $txt! <br>";
?>
```

OPERADORES

Aritméticos

Operador	Nome	Exemplo
+	Addition	\$x + \$y
-	Subtraction	\$x - \$y
*	Multiplication	\$x * \$y
/	Division	\$x / \$y
%	Modulus	\$x % \$y
**	Exponentiation	\$x ** \$y

Comparação

Operator	Name	Example
==	Equal	\$x == \$y
!=	Not equal	\$x != \$y
<>	Not equal	\$x <> \$y
>	Greater than	\$x > \$y
<	Less than	\$x < \$y
>=	Greater than or equal to	\$x >= \$y
<=	Less than or equal to	\$x <= \$y

Lógicos

Operator	Name	Example
&&	And	\$x && \$y
	Or	\$x \$y
!	Not	!\$x

Funções de String

Função	Descrição
explode()	Quebra uma string em um array
implode()	Retorna uma string dos elementos de um array
lcfirst()	Converte o primeiro caractere de uma string em minúscula
ltrim()	Remove espaços em branco ou outros caracteres do lado esquerdo de uma string
rtrim()	Remove espaços em branco ou outros caracteres do lado direito de uma string
str_getcsv()	Analisa uma string CSV em uma matriz
str_replace()	Substitui alguns caracteres em uma string (diferencia maiúsculas de minúsculas)
str_word_count()	Conta o número de palavras em uma string
strcasecmp()	Compara duas cadeias (sem distinção entre maiúsculas e minúsculas)
strcmp()	Compara duas strings (diferencia maiúsculas de minúsculas)
stripos()	Retorna a posição da primeira ocorrência de uma string dentro de outra string (não diferencia maiúsculas de minúsculas)
stristr()	Encontra a primeira ocorrência de uma string dentro de outra string (sem distinção entre maiúsculas e minúsculas)
strlen(string)	Retorna o comprimento de uma string
strpos(string, localizar)	Retorna a posição da primeira ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas). Se não encontrar retorno "False"
strrchr()	Encontra a última ocorrência de uma string dentro de outra string
stripos(string, localizar)	Encontra a posição da última ocorrência de uma string dentro de outra string (sem distinção entre maiúsculas e minúsculas)
strrpos(string, localizar)	Localiza a posição da última ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas)
strstr()	Encontra a primeira ocorrência de uma string dentro de outra string (diferencia maiúsculas de minúsculas)

<u>strtolower()</u>	Converte uma string em letras minúsculas
<u>strtoupper()</u>	Converte uma string em letras maiúsculas
<u>substr(string, start, length)</u>	Retorna uma parte de uma string
<u>trim()</u>	Remove espaços em branco ou outros caracteres de ambos os lados de uma string
<u>ucwords()</u>	Converte o primeiro caractere de cada palavra em uma string para maiúscula

Exemplos de funções de string

```
<?php
echo strlen("Hello world!")."<br/>"; // resultado 12
$pos= strpos("Hello world!", "world"); // resultado 6, se não
encontrar retorna 0 ou false
if($pos===false){
    echo "String não encontrada";
} else {
    echo "Posição da string ".$pos;
}
echo str_replace("world", "Dolly", "Hello world!"); //
resultado Hello Dolly!
echo substr("Hello world!",2,4);
?>
```

Expressões Condicionais:

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

if ($t < 12) {
    echo "Have a good morning!";
} elseif ($t < 20) {
    echo "Have a good afternoon!";
} else {
    echo "Have a good night!";
}
?>
```

Funções Matemáticas

```
echo(round(4.96754,2) . "<br>"); //arredondar 2 decimais
echo(rand(10,100) . "<br>"); // número randômico entre 10 e
100
echo(ceil(0.60) . "<br>"); // número inteiro acima
echo(floor(0.60) . "<br>"); // número inteiro abaixo
```

```
echo(pow(2,4) . "<br>"); //potência  
echo(sqrt(9) . "<br>"); // raiz quadrada
```

Data e Hora

A função PHP `date()` formata um timestamp para uma data e hora. A sintaxe da função `date()` é a seguinte. Se não for informado timestamp será retornada a data e hora atual do **servidor**.

`date(format,timestamp)`

Parâmetro	Descrição
format	Requerido. Especifica o formato do registro de data e hora
timestamp	Opcional. Especifica um timestamp de data e hora. O padrão é a data e hora atuais.

Opções para o parâmetro format de data.

d - Representa o dia do mês (01 a 31)

m - representa um mês (01 a 12)

Y - representa um ano (em quatro dígitos)

l - representa o dia da semana como texto

w - representa o dia da semana como número (0-domingo a 6-sábado)

t- último dia do mês

Opções para o parâmetro format de hora.

H - formato de 24 horas de uma hora (00 a 23)

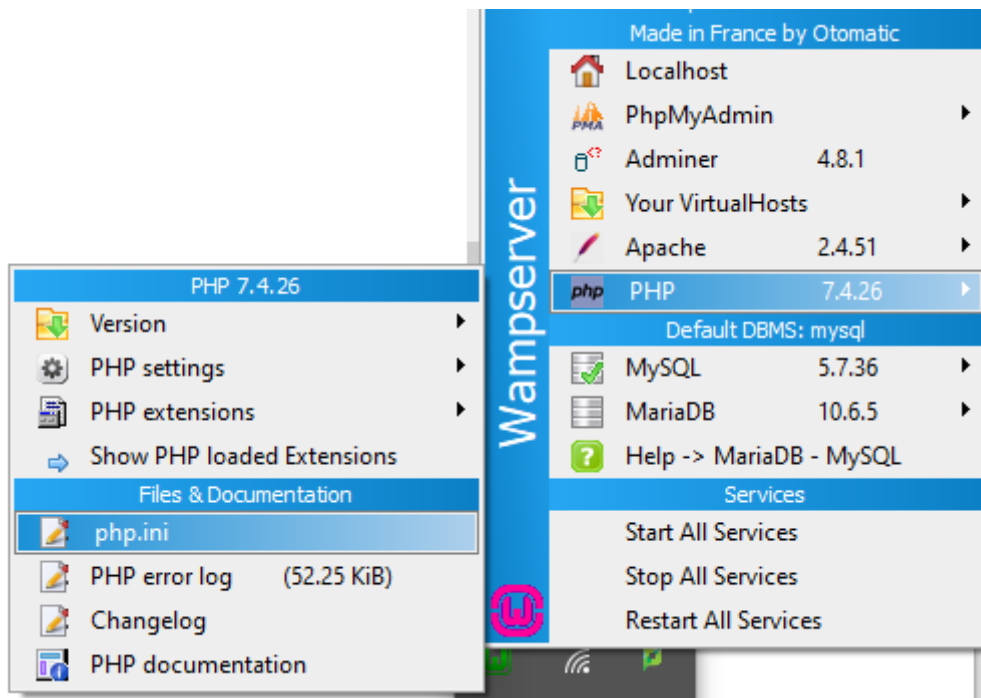
h - formato de 12 horas de uma hora com zeros à esquerda (01 a 12)

i - Minutos com zeros à esquerda (00 a 59)

s - segundos com zeros à esquerda (00 a 59)

a - Em minúsculas Ante meridiem e Post meridiem (am ou pm)

OBS: alterar o arquivo php.ini: date.timezone ="America/Sao_Paulo"



Exemplo

```
<?php
echo "Data de Hoje " . date("d/m/Y") . "<br>";
echo "Dia do mês " . date("m") . "<br>";
echo "Ano " . date("Y") . "<br>";
echo "Dia da semana " . date("w") . "<br>";
echo "Data/Hr atual " . date("d/m/Y H:i:s") . "<br>";
echo '<br>'. strtotime('01-01-1970 01:00:00');//representação
numérica da data
?>
```

Criação de variável do tipo data a partir da data atual:

A função PHP **date_create()** pode ser usada para criar uma variável do tipo data com a data atual do servidor

```
$hj=date_create();
//se não for alterado date.timezone no php.ini
$hj = date_create('now',timezone_open("America/Sao_Paulo"));
echo date_format($hj,"d/m/Y H:i:s");
```

Criação de variável do tipo data a partir de string:

A função PHP **date_create_from_format()** pode ser usada para converter uma string legível para um timestamp.

```
$dt1=date_create_from_format("d/m/Y H:i:s","02/05/2019  
00:00:00");  
echo date_format($dt1,"d/m/Y H:i:s");
```

Checagem de validade de data

```
$month = 02;  
$day = 28;  
$year = 2020;  
  
// retorna um boolean após a checagem da data  
if (checkdate($month,$day,$year)) {/  
    echo "data válida " . "<br>";  
} else {  
    echo "data inválida " . "<br>";  
}
```

Diferença entre duas datas

```
$date1=date_create_from_format("d/m/Y", "15/02/2020");  
$date2=date_create_from_format("d/m/Y", "15/03/2020");  
$diff=date_diff($date1,$date2);  
echo $diff->format("%a"); // número de dias  
echo $diff->format("%Y"); // número de anos
```

Adicionar valores a uma data

A função `date_add()` pode adicionar dias, meses, anos, horas, minutos e segundos a uma data.

```
<?php  
$date=date_create_from_format("d/m/Y","22/03/2020");  
$interval = new DateInterval('P5D'); //5 dias  
date_add($date,$interval);  
echo date_format($date,"d/m/Y") . '<br>';  
  
$date=date_create_from_format("d/m/Y H:i:s","22/03/2020");  
$interval = new DateInterval('P4DT6H'); //4dias e 6 horas  
date_add($date,$interval);  
echo date_format($date,"d/m/Y H:i:s") . '<br>';  
?>
```

Loop while

```
<?php
$x = 1;

while($x <= 5) {
    echo " O número é: $x <br>";
    $x++;
}
?>
```

Loop do .. while

```
<?php
$x = 1;

do {
    echo "O número é: $x <br>";
    $x++;
} while ($x <= 5);
?>
```

Loop for

```
<?php
for ($x = 0; $x <= 10; $x++) {
    echo "O número é: $x <br>";
}
?>
```

ARRAY

Array Indexados

```
<?php
$cars = [];
$cars[0] = "Volvo";
$cars[1] = "BMW";
$cars[2] = "Toyota";
$arlength = count($cars); //número de elementos do array

echo $cars[0]."<br>";

for($x = 0; $x < $arlength; $x++) {
    echo $cars[$x];
}
```

```

        echo "<br>";
    }

    var_dump($cars);
?>

```

Arrays Associativos

Arrays associativos são matrizes que usam chaves nomeadas que você atribui a elas.

```

<?php
$age = [];
$age['Peter'] = "35";
$age['Ben'] = "37";
$age['Joe'] = "43";
echo "Peter is " . $age['Peter'] . " years old." . '<br/>';

foreach($age as $x => $x_value) {
    echo "Key=" . $x . ", Value=" . $x_value;
    echo "<br>";
}
//testar a existência de uma chave
if (array_key_exists('Peter',$age)) {
    echo "Idade do Peter ". $age['Peter'] . '<br/>';
}
?>

```

Array Multidimensional

```

<?php
$arr = array (array());
$arr[0][0]="Maria"; //nome
$arr[0][1]=42; //idade
$arr[1][0]="Paulo";
$arr[1][1]=45;

echo $arr[1][1] . '<br/>';
?>

```

As seguintes funções são utilizadas para ordenar arrays:

- `sort()` - ordena arrays em ordem crescente
- `rsort()` - ordena arrays por ordem decrescente

```

sort($cars);
var_dump($x);

```

Loop foreach (para arrays)

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

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

Exercício 1 PHP

