

Aula 5 – A Linguagem de Programação PHP

Objetivo: Conhecer a linguagem de programação PHP.

5.1. – PHP - definição

PHP é uma linguagem de script executada do lado do servidor e é uma poderosa ferramenta para a criação de páginas web dinâmicas e interativas. PHP significa “**P**HP: **H**ypertext **P**reprocessor”. É aberto para quem se interessar em melhorá-lo e gratuito para uso.

A versão atual do PHP é a 7 e pode ser baixada no site www.php.net. Lá, você também pode encontrar novidades sobre o desenvolvimento, documentação, links para a comunidade de desenvolvedores e o código-fonte do PHP. Por ser uma linguagem de programação que é executada do lado do servidor, ela deve ser instalada junto a um servidor de páginas web. Se você for contratar um serviço de hospedagem de websites, deve verificar se o PHP está instalado no servidor e qual a versão.

5.2. - Servidor de Teste

Um servidor é um programa de computador que serve algo. Podemos ter um servidor de e-mail, que serve e-mails; um servidor de vídeos, que é quem envia o vídeo, como no YouTube, e também temos um servidor que serve as páginas web. A instalação do PHP deve ser feita em um servidor web já instalado.

Então, para testarmos nossas páginas PHP precisamos de uma empresa que forneça isso? Não. Podemos instalar um servidor em nosso computador junto com o PHP e testarmos nosso código PHP localmente. A isso damos o nome de servidor local. Atente para o fato de que nosso servidor é local, ou seja, as páginas apenas serão vistas em nosso computador.

Quando nossa programação estiver pronta, podemos enviá-la para o servidor que contratarmos para que seja disponibilizada mundialmente.

Parece muito complicado? Existe um pacote de software que facilita esse trabalho, seu nome é XAMPP. Ele é todo em português e existe uma versão *portable*, ou seja, leve e própria para um pendrive, por exemplo. A figura 6 reproduz a tela para download e logo abaixo dela está disponível o link para download, tanto da versão *portable* quanto da versão para seu sistema operacional.

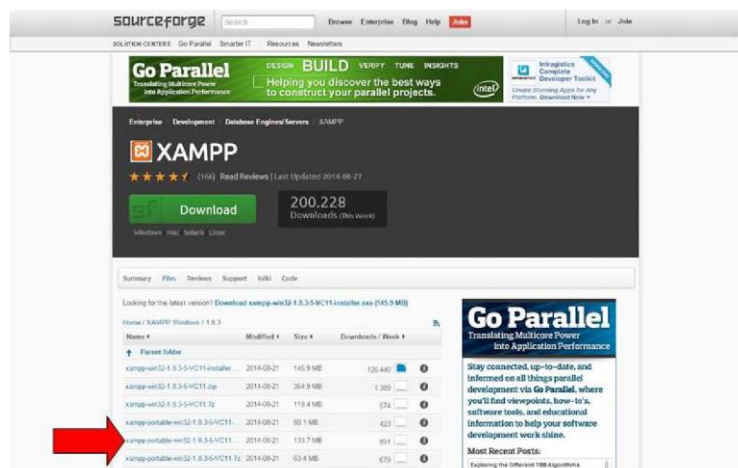


Figura 6 - Tela para download XAMPP. Fonte: Source Forge.

Link para baixar a versão para seu sistema operacional:

https://www.apachefriends.org/pt_br/index.html.

Link para baixar a versão *portable*:

<https://sourceforge.net/projects/xampp/files/XAMPP%20Windows/1.8.3/xampp-portable-win32-1.8.3-5-VC11.zip/download>

Descompacte o arquivo em um bom lugar, preferencialmente na raiz de seu sistema operacional, e memorize sua localização. Dentro há vários programas, entre eles o Apache, que é um servidor de páginas web com o PHP já instalado e o banco de dados web MySQL. Além disso, há ainda o phpMyAdmin, um aplicativo web feito com PHP para gerenciar os bancos do MySQL. Com ele você pode criar tabelas, campos, visualizar e gerenciar tudo no MySQL.

Antes de podermos testar o servidor, temos que configurar sua localização. Não se preocupe que tudo será muito fácil. Dentro da pasta que você descompactou, procure o arquivo **setup_xampp.bat** e execute-o. O que ele faz é descobrir onde estão os servidores. Isso só é preciso porque você pode levar esta pasta no pen drive para outros lugares, assim, a localização pode mudar. Sempre que você mudar a pasta de lugar, vai ter que executar este arquivo para configurar a localização, mas, se não mudar de lugar não precisará repetir o processo.

Aparecerá a janela conforme ilustrado na Figura 7. Aguarde até aparecer uma mensagem dizendo que a atualização foi realizada. Pressione, então, qualquer tecla para que a janela desapareça.

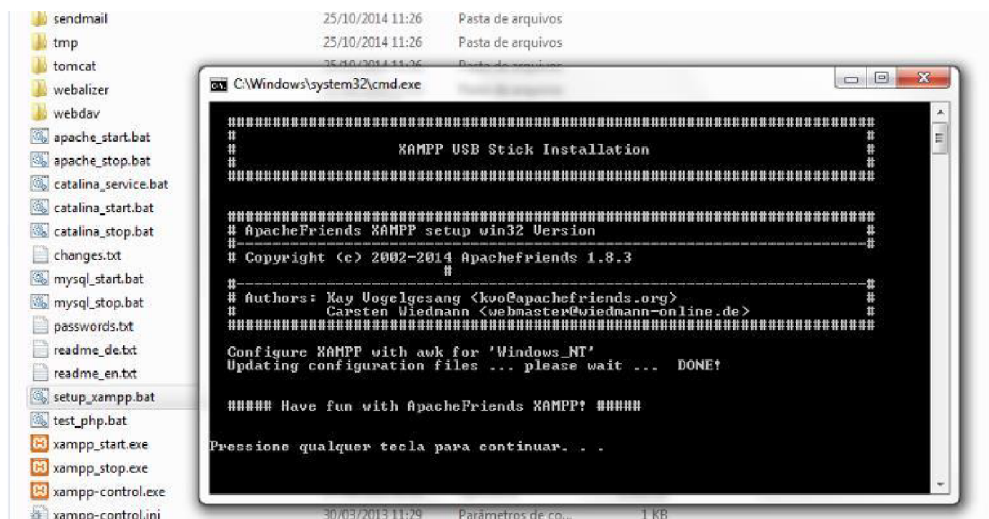


Figura 7 - Menu do setup_xampp.bat. Fonte: Próprio Autor.

Uma vez configurado, vamos executar o arquivo **xampp-control.exe**. Ele é um programa que gerencia os servidores. Na primeira execução, ele pergunta qual o idioma que será utilizado, escolha a bandeira americana e pressione “Save”.

Com a tela do “Control Panel” aberta, pressione os botões “Start” para ligar o Apache e o MySQL. Aparecerá a tela de permissão do Windows, permita o acesso. Só é necessário fazer isso na primeira vez.

Espere algum tempo para os servidores começarem e você verá a tela da Figura 8, com os nomes dos aplicativos destacados em verde.

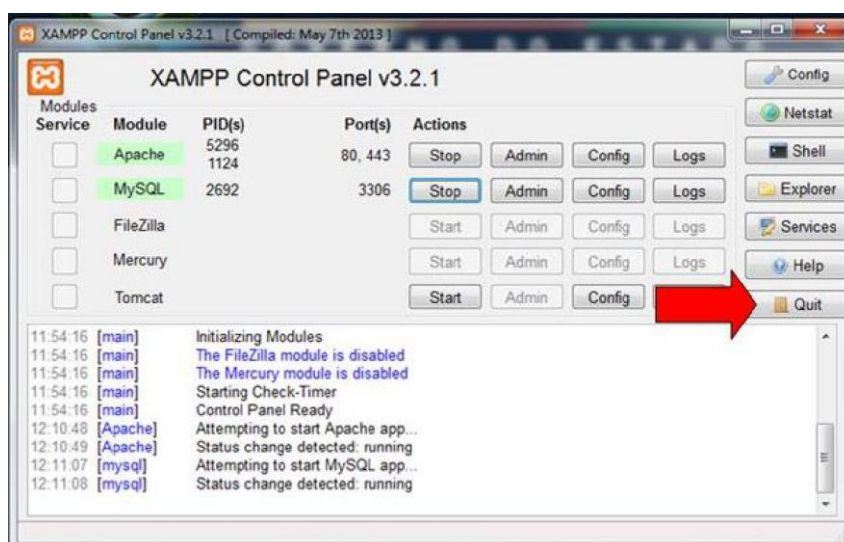


Figura 8 – Tudo OK. Fonte: Próprio Autor.

Agora que tudo está instalado, configurado e funcionando, vamos testar nosso ambiente de teste. Abra seu navegador preferido e digite o endereço **http://localhost/xampp/**. Deve abrir uma tela

como a da Figura 9. Selecione o idioma Português (Brasil) e aparecerá uma página similar à imagem da Figura 10. Esta página não está na internet, foi servida pelo seu servidor local.



Figura 9 – Página Inicial do XAMPP. Fonte: Próprio Autor



Figura 10 – XAMPP para Windows em Português. Fonte: Próprio Autor

5.3. – IDE

Uma IDE é um ambiente integrado de desenvolvimento, ou seja, um ambiente que reúne vários aplicativos que são necessários e úteis para o desenvolvimento. Existem diversas IDEs para desenvolvimento em PHP, algumas pagas, outras gratuitas. Entre elas está o **Net Beans**.

Antes de instalá-la, precisaremos instalar a máquina virtual Java, porque o Net Beans foi desenvolvido em Java. Clique no link a seguir e escolha a opção JRE: <http://www.oracle.com/technetwork/pt/java/javase/downloads/index.html>

Aceite os termos de uso e clique na versão referente ao seu sistema operacional, conforme a Figura 11.

Java SE Runtime Environment 8u171		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.		
Product / File Description	File Size	Download
Linux x86	64.47 MB	jre-8u171-linux-i586.rpm
Linux x86	80.38 MB	jre-8u171-linux-i586.tar.gz
Linux x64	61.43 MB	jre-8u171-linux-x64.rpm
Linux x64	77.41 MB	jre-8u171-linux-x64.tar.gz
Mac OS X x64	74.58 MB	jre-8u171-macosx-x64.dmg
Mac OS X x64	66.21 MB	jre-8u171-macosx-x64.tar.gz
Solaris SPARC 64-bit	52.13 MB	jre-8u171-solaris-sparcv9.tar.gz
Solaris x64	50.03 MB	jre-8u171-solaris-x64.tar.gz
Windows x86 Online	1.79 MB	jre-8u171-windows-i586-iftw.exe
Windows x86 Offline	61.66 MB	jre-8u171-windows-i586.exe
Windows x86	64.84 MB	jre-8u171-windows-i586.tar.gz
Windows x64	68.5 MB	jre-8u171-windows-x64.exe
Windows x64	68.92 MB	jre-8u171-windows-x64.tar.gz

Dê um duplo clique no arquivo que você baixou e o processo de instalação terá início. Aceite os próximos dois passos clicando em “Next”.

Logo em seguida, faça o download do Net Beans. A instalação é bem simples.
https://netbeans.org/downloads/start.html?platform=windows&lang=pt_BR&option=php&bits=x64

5.4. - Testando o Ambiente

Agora que temos tudo pronto, vamos testar o nosso ambiente de desenvolvimento. Para isso, vamos fazer uma página simples em PHP, o nosso “Hello world!”.

No Net Beans, cada site é um novo projeto, por isso abra o menu Arquivo => Novo Projeto, será aberta a tela a seguir, onde você escolherá a opção PHP.

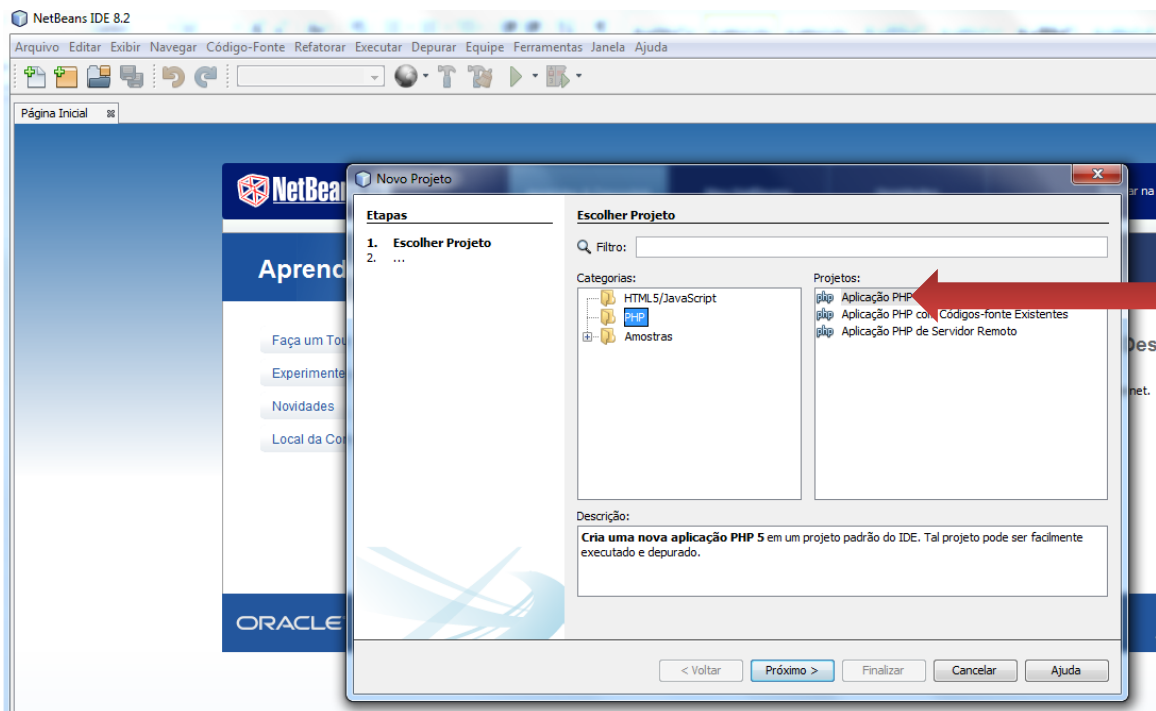


Figura 12 – Tela do Net Beans, escolhendo a categoria PHP. Fonte: Próprio Autor.

Clique no botão “Próximo”. Na tela a seguir, dê um nome ao projeto: “HelloMundo”. Em seguida, você define o seu Local Host, ou seja, o local onde estará sua página, sua URL local, seu endereço local. As duas próximas telas poderão ser desconsideradas neste momento, portanto, clique no botão “Finalizar”.

A tela do Net Beans será como a que segue:

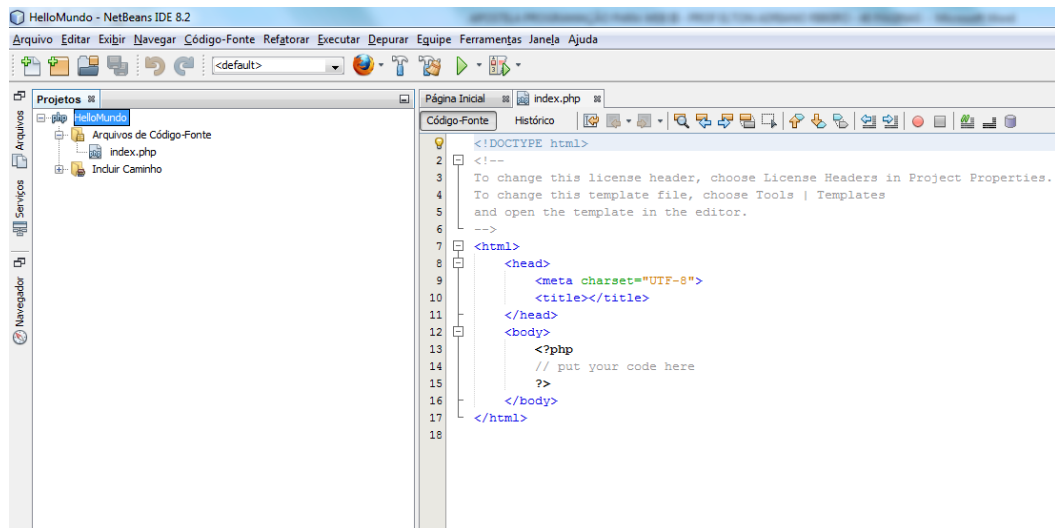


Figura 13 – Projeto HelloMundo, no Net Beans. Fonte: Próprio Autor.

Para criar nossa primeira página, digite o seguinte código:

```
<html>
<head>
  <meta charset="UTF-8">
  <title></title>
</head>
<body>
  <?php
    echo "<p>Olá Mundo</p>"
  ?>
</body>
</html>
```

Figura 14 – Primeiro código PHP. Fonte: Próprio Autor.

Salve o arquivo.

Abra seu navegador e digite o seguinte endereço: <https://127.0.0.1/HelloMundo/>. Se tudo foi configurado corretamente, será exibida sua primeira página PHP. Conforme mostra a figura 15:

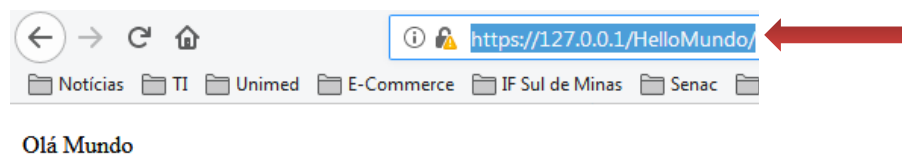


Figura 15 – Primeira página Web. Fonte: Próprio Autor.

Aula 6 – Sintaxe Básica e Estudo das Variáveis

Objetivo: Conceituar Sintaxe Básica e conhecer as variáveis do PHP.

6.1 - Página PHP

O PHP é uma linguagem de programação que é executada do lado do servidor. Ela é diferente, por exemplo, do JavaScript, que também é uma linguagem interpretada, mas executada do lado do cliente, ou seja, no navegador do usuário e não no servidor.

Vamos ver o que acontece quando se usam páginas PHP em um site. Acompanhe o fluxo na Figura 16.

Primeiramente o usuário faz um pedido de uma página pelo endereço DNS, ou clicando em um link que possui um endereço DNS (1). Quando a solicitação chega ao servidor, ele verifica se o que foi pedido é uma página PHP e entrega para o interpretador (2). O interpretador lê a página e quando encontra um código PHP o executa (3). Quando termina de ler, o interpretador constrói uma página HTML como resultado da computação realizada e entrega ao servidor (4). Finalmente, o servidor envia a página HTML resultante para o usuário. Todas as vezes que são feitas requisições de página PHP, esse processo é repetido.

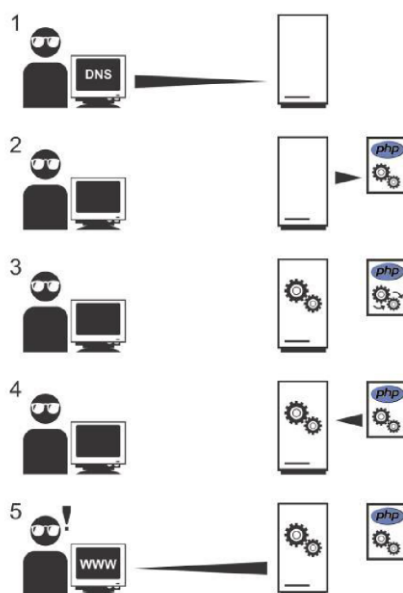


Figura 16 – Fluxo de execução páginas PHP. Fonte: Loop Infinito

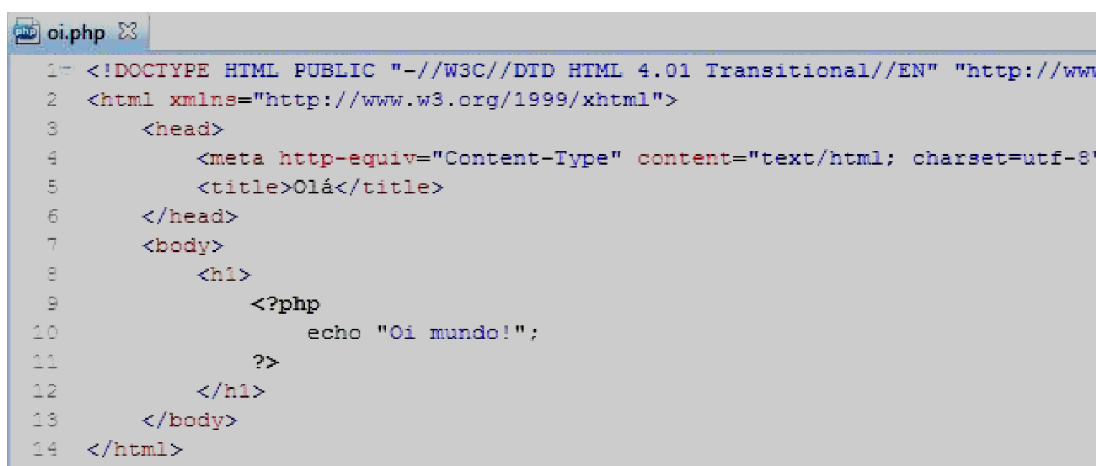
Mais adiante, observe que as páginas PHP são páginas HTML com código PHP misturado. Para que o servidor saiba que são páginas PHP, ao invés de colocarmos a extensão .html, colocamos .php. Podemos enviar para o servidor uma página PHP contendo apenas código HTML. O servidor vai ter todo o trabalho descrito na Figura 16 e irá retornar à mesma página, sem realizar qualquer computação. Quando formos fazer nosso sistema de locadora teremos casos como este.

6.2. - Sintaxe Básica

Para que uma página seja interpretada pelo PHP, ela deve ter a extensão **.php**, mas não só isso, todo código PHP deve estar entre dois sinais que indicam o início e o fim de um pedaço de computação. Assim, o interpretador vai saber que o que está dentro é comando PHP.

Então, para marcar o início utilizamos **<?php** e para marcar o final utilizamos **?>**.

O comando “echo” escreve o que está logo após, entre aspas duplas. Vamos escrever “Oi mundo!”, utilizando PHP, conforme demonstrado na Figura 17.

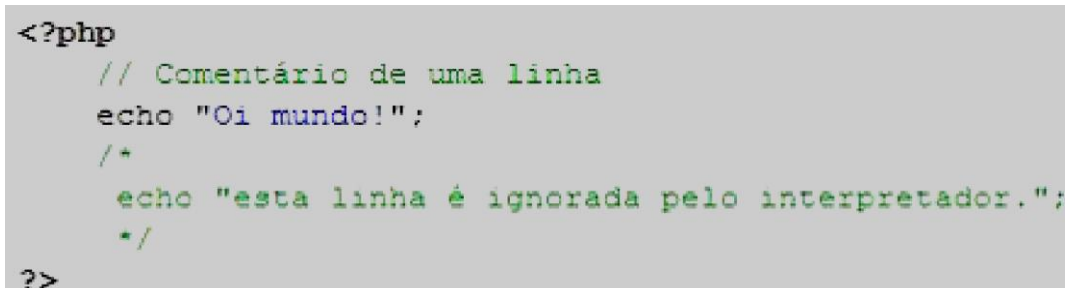


```
1- <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www
2 <html xmlns="http://www.w3.org/1999/xhtml">
3     <head>
4         <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
5         <title>Olá</title>
6     </head>
7     <body>
8         <h1>
9             <?php
10                 echo "Oi mundo!";
11             ?>
12         </h1>
13     </body>
14 </html>
```

Figura 17 – Sintaxe básica em PHP. Fonte: Próprio Autor

Anteriormente, vimos como testar o ambiente, tenha certeza de que o XAMPP esteja ligado e faça o mesmo com o código acima.

Quando escrevemos sistemas, podemos chegar a uma quantidade enorme de códigos e esquecermos porque criamos um ou outro, para isso utilizamos os “comentários” em uma linha, colocando o // no final dela ou para um intervalo utilizamos /* no começo e */ no final. Como pode ser visto na Figura 18.



```
<?php
    // Comentário de uma linha
    echo "Oi mundo!";
    /*
        echo "esta linha é ignorada pelo interpretador.";
    */
?>
```

Figura 18 – Comentários no código. Fonte: Próprio Autor

6.3. – Variáveis

De acordo com Miles (2008), variáveis são como “caixas” na memória do computador, onde armazenamos algo. Para trabalharmos com dados, precisamos armazenar seus valores em um lugar que seja fácil de encontrar quando necessitarmos. Essas “caixas” chamam-se variáveis. O conceito é o mesmo das variáveis matemáticas, onde utilizamos letras para representar um valor, por exemplo, Z ou Y.

Da mesma forma que na matemática, as variáveis em PHP podem ser utilizadas para guardar valores ($X=2$) e expressões ($Z=X+Y$).

6.3.1 - Nome das Variáveis

Ao invés de utilizarmos letras, podemos dar nomes às variáveis, por exemplo: sexo, cor, modelo, idade, etc. As variáveis devem obedecer às seguintes regras:

- Uma variável deve começar com o sinal de cifrão \$, seguido pelo nome da variável;
- O nome de uma variável deve começar com uma letra ou com os caracteres de sublinhado;
- O nome de uma variável só pode conter caracteres alfanuméricos e sublinhados (Az, 0-9 e _);
- O nome de uma variável não deve conter espaços;
- PHP distingue maiúsculas de minúsculas, dessa forma \$nome, \$Nome, \$NOME e \$nOmE são variáveis diferentes.

6.3.2 – Declaração de Variáveis

Toda variável deve ter um valor atribuído a ela, para isso utilizamos o sinal de igual (=), por exemplo:

```
$nome = "IF Sul de Minas";  
$idade = 10;
```

Quando executarmos o programa, o computador guardará na memória o nome IF Sul de Minas e o número 10, para que usemos a qualquer momento.

6.3.3 – Escopo de uma variável

Não podemos acessar nossas variáveis em qualquer lugar do código. Elas pertencem a um determinado local, a que chamamos de escopo. Temos quatro tipos de escopo: Local, Global, Estático e Parâmetro.

Uma variável declarada dentro de uma função é **local** e só pode ser acessada dentro dessa função. Uma função é um pequeno programa, basicamente.

```
1 <?php
2 function teste() {
3     $x = 2 + 2; // A variável $x só existe localmente.
4 } // Quando a função termina a variável $x é apagada da memória.
5
6 echo $x; // Quando esta linha for lida a variável $x não existe.
7 ?>
```

Figura 19 – Exemplo de variável local. Fonte: Próprio Autor

Uma variável que é declarada fora de uma função tem um escopo mais abrangente, sendo classificada como **global**. Uma variável global pode ser acessada em qualquer lugar do programa, exceto dentro de uma função.

```
<?php
$x=2; // Escopo global
$y=5; // Escopo global

function teste() {
    global $x, $y; // Aqui chamamos as variáveis que estão fora da função.
    $y = $x + $y; // Aqui somamos o conteúdo de $x e $y e colocamos dentro de $y
}

teste();
echo $y; // Escreve 7
?>
```

Figura 20 – Escopo global. Fonte: Próprio Autor.

Quando o escopo de uma variável local ou global termina, ela é excluída, mas você pode querer que ela continue a existir por mais tempo. Sendo assim, utilize a palavra **static** quando declarar a variável.

```
<?php

function teste() {
    static $x=0; // Escopo estático
    echo $x; // Escreve seu valor
    $x++; // Soma mais um ao seu valor
}

teste(); // Aqui, quando chegasse no final, a variável $x era para ser excluída.
teste(); // Mas como ela foi declarada com static, não é apagada.
teste(); // E chamada novamente mostrando o valor acrescido.

?>
```

Figura 21 – Escopo estático. Fonte: Próprio Autor.

6.4. – Strings

O PHP disponibiliza várias funções para a manipulação de strings. As strings são os conteúdos existentes nas variáveis. Vamos abordar algumas das mais corriqueiras.

6.4.1 – Operação de Concatenação

Concatenar é unir, juntar. O operador de concatenação realiza uma operação de unir duas strings. Seu símbolo é o ponto e podemos unir duas ou mais strings na mesma expressão.

```
<?php
$txt1 = "Oi!";
$txt2 = "Fulando de Tal.";

echo $txt1 . " " . $txt2;
?>
```

Figura 22 – Exemplo de concatenação. Fonte: Próprio Autor.

Será escrito “Oi! Fulano de Tal”. Neste caso, concatenamos um espaço em branco no meio de duas variáveis.

6.4.2 – Função strlen ()

A função strlen conta a quantidade de caracteres. Será escrito como saída, 9.

```
<?php
$txt1 = "Oi!";
$txt2 = "Fulando de Tal.";

echo $txt1 . " " . $txt2;
?>
```

Figura 23 – Exemplo da função strlen(). Fonte: Próprio Autor

6.4.3 - Função strpos()

Usada para procurar uma string dentro de uma string. Caso ache a função, retorna o número da posição do primeiro caractere da string procurada, caso não ache, retorna **falso**. O exemplo abaixo retornará o valor 10.

```
<?php
echo strpos("Fulano de Tal","Tal");
?>
```

Figura 24 – Exemplo de uso da função strpos(). Fonte: Próprio Autor.

6.5. – Operadores

Os operadores servem para executar uma operação em uma expressão, como as operações de soma, subtração, multiplicação e divisão. Em PHP temos os seguintes operadores: Operadores Aritméticos, Operadores de Atribuição, Operadores de Incremento e Decremento, Operadores de Comparação e Operadores Lógicos.

6.5.1 – Operadores Aritméticos

OPERADOR	NOME	DESCRIÇÃO	EXEMPLO	RESULTADO
$x + y$	Adição	Soma de x e y	$2 + 2$	4
$x - y$	Subtração	Subtração de x e y	$7 - 2$	5
$x * y$	Multiplicação	Produto de x por y	$3 * 2$	6
x / y	Divisão	Quociente de x por y	$4 / 2$	2
$x \% y$	Módulo	Resto da divisão de x por y	$5 \% 2$	1
$-x$	Negativo	Inverte o sinal de x	-2	2

Figura 24 – Operadores Aritméticos. Fonte: Próprio Autor

6.5.2 – Operadores de Atribuição

OPERADOR	COMO SE FOSSE...	DESCRIÇÃO
$x = y$	$x = y$	Coloca no operando da esquerda o resultado ou valor da direita
$x += y$	$x = x + y$	Adiciona y a x
$x -= y$	$x = x - y$	Subtrai y de x
$x *= y$	$x = x * y$	Multiplica x por y
$x /= y$	$x = x / y$	Divide x por y
$x \% = y$	$x = x \% y$	Coloca em x o resto da divisão de x por y
$x . = y$	$x = x . y$	Concatena duas strings

Figura 25 – Operadores de Atribuição. Fonte: Próprio Autor.

6.5.3 – Operadores de Incremento e Decremento

OPERADOR	NOME	DESCRIÇÃO
$++x$	pré-incremento	Soma um ao valor de x e depois retoma x
$x++$	pós-incremento	Retoma x e depois soma um ao valor de x
$--x$	pré-decremento	Subtrai um do valor de x e depois retoma x
$x--$	pós-decremento	Retoma x e depois subtrai um do valor de x

Figura 26 – Operadores de Incremento e Decremento. Fonte: Próprio Autor.

6.5.4 - Operadores de Comparação

OP.	NOME	DESCRIÇÃO	EX.	RES.
$x == y$	Igual	Retorna verdadeiro se for igual.	$2 == 3$	false
$x != y$	Não igual	Retorna verdadeiro se for diferente.	$2 != 3$	true
$x <> y$	Diferente	Retorna verdadeiro se for diferente.	$2 <> 3$	true
$x > y$	Maior que	Retorna verdadeiro se x for maior que y	$3 > 2$	true
$x < y$	Menor que	Retorna verdadeiro se x for menor que y	$2 < 3$	true
$x >= y$	Maior que ou igual	Retorna verdadeiro se x for maior que ou igual a y	$3 >= 3$	true
$x <= y$	Menor que ou igual	Retorna verdadeiro se x for menor que ou igual a y	$2 <= 3$	true

Figura 27 – Operadores de Comparação. Fonte: Próprio Autor.

6.5.5 - Operadores Lógicos

OPERADOR	NOME	DESCRIÇÃO	EXEMPLO	RESULTADO
$x \text{ and } y$	e	Retorna verdadeiro se x e y forem verdadeiro	$x=1; y=9$ $(x < 3 \text{ and } y > 5)$	true
$x \text{ or } y$	ou	Retorna verdadeiro se houver um verdadeiro	$x=1; y=9$ $(x == 1 \text{ or } y == 2)$	true
$x \&\& y$	e	Retorna verdadeiro se x e y forem verdadeiro	$x=1; y=9$ $(x < 3 \&\& y > 5)$	true
$x \ \ y$	ou	Retorna verdadeiro se houver um verdadeiro	$x=1; y=9$ $(x == 1 \ \ y == 2)$	true
$!x$	não	Retorna verdadeiro se x for falso	$x = 1, y = 9$ $!(x == y)$	true

Figura 28 – Operadores Lógicos. Fonte: Próprio Autor.

6.6. – If...Else

Linguagens de programação têm duas estruturas em comum: estruturas de repetição e estruturas de condicionais. O “if...else” é uma estrutura condicional. Ela verifica uma condição, se for verdadeira faz algo, senão faz outra coisa.

If (condição) comando;

If (condição) {

// Bloco de código. Aqui pode ter várias linhas.

}

Nos casos acima, o “if” verifica a condição, se for verdade ela executa um comando. Caso queira mais de um comando, você deve criar um bloco de comando utilizando { e }. Caso a condição seja falsa, o comando ou bloco será ignorado. O “else” significa “senão”. Utilize-o para colocar um comando ou bloco de comandos, caso a condição seja falsa. Conforme mostra a figura 29.

```

<?php
$x = 10;
$y = 100;
if ($x < $y) echo "<p>x é menor do que y<p>";

if ($y > $x) {
    echo "<p>y é maior do que x, ";
    echo "o valor de y é " . $y . "</p>";
} else {
    echo "x é maior que y";
}
?>

```

Figura 29 – Estrutura Condicional. Fonte: Próprio Autor

O exemplo acima escreve “**x é menor do que y**”, passa à próxima linha e escreve “**y é maior do que x, o valor de y é 100**”.

6.7. – Switch

É uma estrutura de condição, assim como o if...else, que verifica uma condição com várias possibilidades de resposta.

switch (n) {

case x:

código a ser executado se n igual a x;

break;

case y:

código a ser executado se n igual a y;

break;

...

default:

código a ser executado se n não for nenhuma das opções anteriores.

}

```
<?php
$musica = "rock";
switch ($musica) {
    case "sertanejo":
        echo "Adoro sertanejo!";
        break;
    case "rock":
        echo "Rock'n Roll, baby!";
        break;
    case "samba":
        echo "Samba é o que tem de melhor!";
        break;
    default:
        echo "Não gosto nem de sertanejo, nem de rock, nem de samba, prefiro funk!";
}
?>
```

Figura 30 – Exemplo de estrutura switch. Fonte: Próprio Autor.

O exemplo acima retorna “**Rock’n Roll, baby!**”. Você pode mudar o valor da variável **\$musica** para ver os outros resultados. Também pode colocar um valor que não exista nas opções, para exibir a opção **default**.