



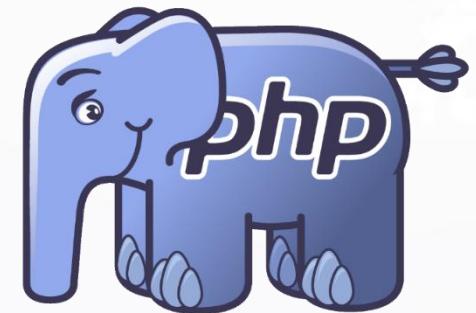
Lógica de Programação
Direcionada a PHP



Prof. Celso Caldeira
(11) 97110-9750

O que é PHP?

PHP significa: Hypertext Preprocessor. Realmente, o produto foi originalmente chamado de “Personal Home Page Tools”; mas como se expandiu em escopo, um nome novo e mais apropriado foi escolhido por votação da comunidade. Você pode utilizar qualquer extensão que desejar para designar um arquivo PHP, mas os recomendados foram .php , .phtml. O PHP está atualmente na versão 8, chamado de PHP8 ou, simplesmente de PHP.



PHP é uma linguagem de criação de scripts embutida em HTML no servidor.

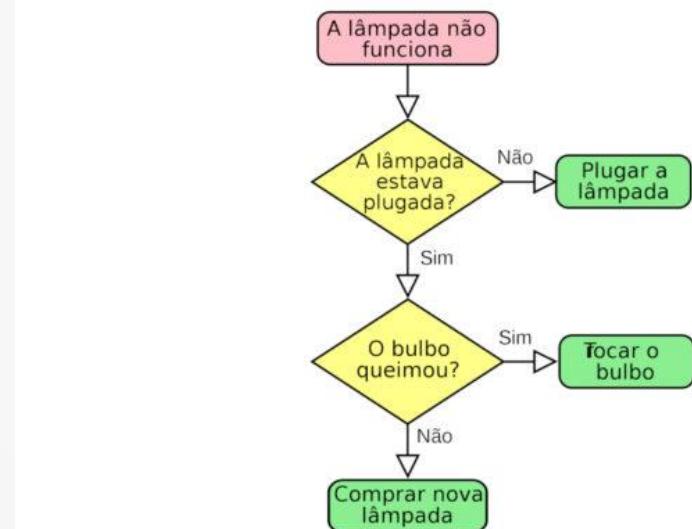
Com o PHP podemos criar uma série quase infinita de aplicações envolvendo ou não utilização de banco de dados. Até aplicativos para mobile do tipo PWA é possível fazer com essa linguagem.

O PHP é uma das linguagens mais utilizadas no mercado.

* PWA: Progressive Web App

Conceitos básicos de algoritmo

De maneira geral, ele compreende uma sequência finita de ações executáveis (passos) para resolver um problema, ou no caso mais comum em Ciência da Computação, executar uma tarefa. O algoritmo em si não é o programa, mas a sequência de ações e condições que devem ser obedecidas para que o problema seja resolvido.



Estruturas de controle

Qualquer script PHP é construído por uma série de instruções. Uma instrução pode ser uma atribuição, uma chamada de função, um laço de repetição, uma instrução condicional, ou mesmo uma instrução que não faz nada (um comando vazio). Instruções geralmente terminam com um ponto e vírgula. Além disso, as instruções podem ser agrupados em um grupo de comandos através do encapsulamento de um grupo de comandos com chaves. Um grupo de comandos é uma instrução também. Os vários tipos de instruções são descritos neste capítulo.



Sintaxe Básica

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  
?>
```

O que precisaremos?

O PHP roda sobre o protocolo http, portanto a execução depende do navegador, mas não é totalmente necessário hospedar a aplicação, pois além da aplicação ser executada na internet, podemos trabalhar localmente e até mesmo montar uma intranet através de serviços especiais.

Nesse curso iremos usar o XAMPP e claro, um bom editor para codificar nossas aplicações e para isso usaremos o VsCode.



XAMPP



Vamos começar

Os equipamentos aqui no Senac já estão preparados e com todos os softwares necessários para começarmos a criar nossos códigos, mas em outras máquinas em que forem praticar, teremos que instalar o XAMPP. Basta ir no Google e pesquisar por XAMPP, acessar o primeiro link, fazer o download e instalar.

Faça o mesmo procedimento para instalar o VSCode.

Já que está tudo instalado....

Vamos agora subir o serviço que o XAMPP vai oferecer:

Procure por XAMPP na pesquisa e execute:

Depois clique em START em apache e mysql.



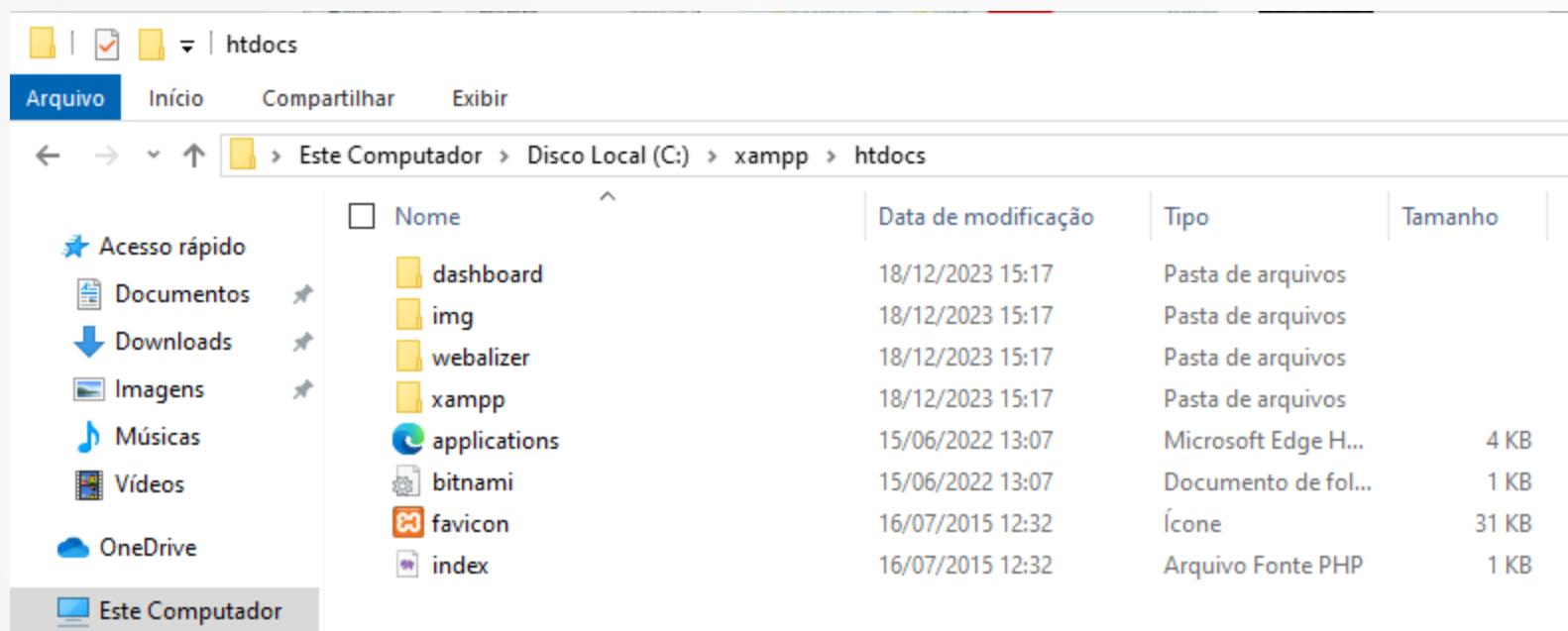
O Servidor Apache permite que rodemos a aplicação PHP localmente

O Servidor Mysql permite a criação e manipulação do banco de dados também localmente.

Onde ficarão nossos projetos

Nossos projetos ficarão gravados na pasta: c:\xampp\htdocs

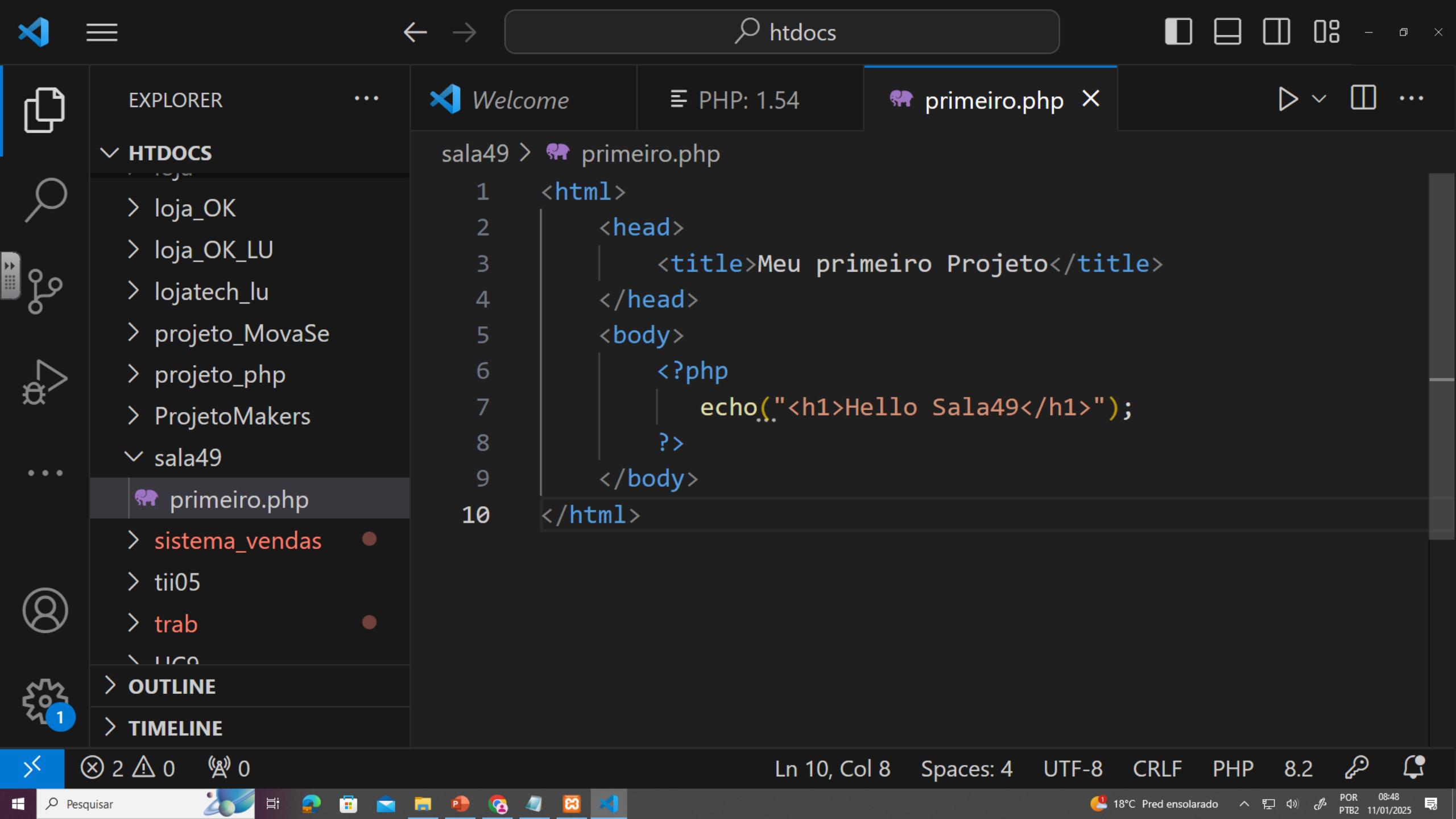
Criaremos as pastas e arquivos que forem necessários aqui, pois é o local destinado para que o XAMPP proceda a execução das nossas aplicações.



Antes de começar a codar

Em qualquer linguagem, a base da lógica de programação é a mesma:
Entrada, processamento e saída.

Portanto, entramos com o dado em uma posição da memória
denominada variável, efetuamos o processamento usando os comandos
e funções aritméticas e lógicas ou manipulando informações em banco
de dados e a saída será o resultado do que o usuário espera, seja na
tela do computador, celular ou impressora.



Tipos de variáveis

Para se declarar uma variável, usamos o símbolo: \$ antes do nome que daremos a variável e os tipos podem ser:

```
// variável do tipo String  
$nome = 'Meu nome 123';  
// variável do tipo inteiro  
$ano = 2017;  
// variável do tipo float  
$pi = 3.14159265;  
// variável do tipo booleano  
$sim = true;
```

Veja o exemplo:

```
<?php  
echo("<h1>Esse é o meu primeiro programa em PHP</h1>");  
echo("<br/>");  
$n1=1;  
$n2=2;  
echo($n1+$n2); // Irá mostrar 3  
echo("<br/>");  
$n3="1";  
$n4="2";  
echo($n3 . $n4); // Irá mostrar '12'  
?>
```

Para concatenar Strings, usamos o ponto “.”

```
<!-- aula1.php -->
<html>
<header>
    <title>Meu primeiro programa</title>
</header>
<body>
    <?php
        echo("<h1>Esse é o meu primeiro programa em PHP</h1><br/>");
        $n1=1;
        $n2=2;
        echo("O resultado é: ". $n1+$n2); // Irá mostrar 3
        echo("<br/>");
        $n3="1";
        $n4="2";
        echo("O resultado é: ". $n3 . $n4); // Concatenando string - Irá mostrar '12'
        echo("<br/>");
        $nome="José";
        $sobrenome="Manoel";
        echo("seja bem vindo " . $nome . " " . $sobrenome); // Concatenando string
    ?>
</body>
</html>
```

Operadores Aritméticos

Exemplo	Nome	Resultado
<code>+\$a</code>	Identidade	Conversão de <code>\$a</code> para int ou float conforme apropriado.
<code>-\$a</code>	Negação	Oposto de <code>\$a</code> .
<code>\$a + \$b</code>	Adição	Soma de <code>\$a</code> e <code>\$b</code> .
<code>\$a - \$b</code>	Subtração	Diferença entre <code>\$a</code> e <code>\$b</code> .
<code>\$a * \$b</code>	Multiplicação	Produto de <code>\$a</code> e <code>\$b</code> .
<code>\$a / \$b</code>	Divisão	Quociente de <code>\$a</code> e <code>\$b</code> .
<code>\$a % \$b</code>	Módulo	Resto de <code>\$a</code> dividido por <code>\$b</code> .
<code>\$a ** \$b</code>	Exponencial	Resultado de <code>\$a</code> elevado a <code>\$b</code> .

Os operadores matemáticos como Seno, Coseno, Tangente e demais, podem ser estudados no link:

https://www.php.net/manual/pt_BR/ref.math.php

Operadores Lógicos

Exemplo	Nome	Resultado
<code>\$a and \$b</code>	E	true se ambos <code>\$a</code> e <code>\$b</code> é true .
<code>\$a or \$b</code>	OU	true se <code>\$a</code> ou <code>\$b</code> é true .
<code>\$a xor \$b</code>	XOR	true se <code>\$a</code> ou <code>\$b</code> é true , mas não ambos ao mesmo tempo.
<code>! \$a</code>	Não	true se <code>\$a</code> não é true .
<code>\$a && \$b</code>	E	true se ambos <code>\$a</code> e <code>\$b</code> são true .
<code>\$a \$b</code>	OU	true se <code>\$a</code> ou <code>\$b</code> é true .

Note que o sinal `!` Significa negação, ou seja, não

Operadores de comparação

Operadores de comparação		
Exemplo	Nome	Resultado
<code>\$a == \$b</code>	Igual	true se <code>\$a</code> é igual a <code>\$b</code> após equalização de tipos.
<code>\$a === \$b</code>	Idêntico	true se <code>\$a</code> é igual a <code>\$b</code> , e eles são do mesmo tipo.
<code>\$a != \$b</code>	Diferente	true se <code>\$a</code> não é igual a <code>\$b</code> depois de equalização de ativos.
<code>\$a <> \$b</code>	Diferente	true se <code>\$a</code> não é igual a <code>\$b</code> depois de equalização de ativos.
<code>\$a !== \$b</code>	Não idêntico	true se <code>\$a</code> não é igual a <code>\$b</code> , ou eles não são do mesmo tipo.
<code>\$a < \$b</code>	Menor que	true se <code>\$a</code> é estritamente menor que <code>\$b</code> .
<code>\$a > \$b</code>	Maior que	true se <code>\$a</code> é estritamente maior que <code>\$b</code> .
<code>\$a <= \$b</code>	Menor ou igual	true se <code>\$a</code> é menor ou igual a <code>\$b</code> .
<code>\$a >= \$b</code>	Maior ou igual	true se <code>\$a</code> é maior ou igual a <code>\$b</code> .
<code>\$a <=> \$b</code>	Spaceship (nave espacial)	Um int menor que, igual a ou maior que zero quando <code>\$a</code> é, respectivamente, menor que, igual a ou maior que <code>\$b</code> .

O Comando IF

O construtor if é um dos recursos mais importantes em muitas linguagens, inclusive no PHP. Permite a execução condicional de fragmentos de código. O PHP apresenta uma estrutura if semelhante a da linguagem C:

```
<!-- notas.php -->
<html>
<head>
    <title>Meu segundo programa</title>
</head>
<body>
    <?php
        echo("<h1>Resultado</h1>");
        echo("<br/>");
        $nota = 6.5;
        if ($nota < 5) {
            echo("Aluno reprovado");
        } elseif ($nota < 7) {
            echo("Aluno em recuperação");
        } else {
            echo("Aluno aprovado");
        }
    ?>
</body>
</html>
```

Laço for

O laço for é uma estrutura de repetição muito utilizada nos programas, e é muito útil quando se sabe de antemão quantas vezes a repetição deverá ser executada. Este laço utiliza uma variável para controlar a contagem do loop, bem como seu incremento.

```
<!-- aula1b.php -->
<html>
<head>
    <title>Meu segundo programa</title>
</head>
<body>
    <?php
        for ($i = 1; $i <= 10; $i++) {
            echo $i . "<br/>";
        }
    ?>
</body>
</html>
```

Vamos fazer uma tabuada?

```
<!-- aula1c.php -->
<html>
<head>
    <title>Meu segundo programa</title>
</head>
<body>
    <?php
    $numero = 7;
    for ($i = 1; $i <= 10; $i++) {
        echo($numero . " X " . $i . " = " . $numero*$i . "<br/>");
    }
    ?>
</body>
</html>
```

Aqui fizemos a tabuada do 7. O
 é o comando do html para pular uma linha

O Laço while

O laço while também é uma estrutura de repetição muito utilizada nos programas, mas é diferente do for, pois não há incremento automático, é necessário declarar uma variável se for usar para incrementar e não é usada apenas para números, pois muitas vezes é utilizada para exibir os dados de uma tabela, como veremos mais pra frente.

```
<!-- lacowhile.php -->
<html>
<head>
    <title>Meu segundo programa</title>
</head>
<body>
<?php

$numero = 9;
$multiplicador = 1;
while ($multiplicador <= 10) {
    echo $numero . "X" . $multiplicador . " = " .
$numero*$multiplicador;
    if ($numero*$multiplicador % 2 == 0) {
        echo " - Esse número é par<br/>";
    } else {
        echo " - Esse número é ímpar<br/>";
    }
    $multiplicador = $multiplicador + 1;
}
?>
</body>
</html>
```

Switch case

A switch instrução é semelhante a uma série de instruções IF na mesma expressão. Em muitas ocasiões, você pode querer comparar a mesma variável (ou expressão) com muitos valores diferentes e executar um trecho de código diferente dependendo de qual valor ele é igual. É exatamente para isso que serve o Switch case.

```
<!-- switchcase.php -->
<html>
<head>
    <title>Meu segundo programa</title>
</head>
<body>
<?php
$dia_da_semana = "quarta";

switch ($dia_da_semana) {
    case "segunda":
        echo "Hoje é segunda-feira!";
        break;
    case "terça":
        echo "Hoje é terça-feira!";
        break;
    case "quarta":
        echo "Hoje é quarta-feira!";
        break;
    case "quinta":
        echo "Hoje é quinta-feira!";
        break;
    case "sexta":
        echo "Hoje é sexta-feira!";
        break;
    case "sábado":
        echo "Hoje é sábado!";
        break;
    case "domingo":
        echo "Hoje é domingo!";
        break;
    default:
        echo "Dia inválido!";
}
?>
</body>
</html>
```

Funções String

Principais funções:

rtrim – Remove espaços em branco a direita de uma string

ltrim – Remove espaços em branco a esquerda de uma string

str_replace – Troca uma palavra por outra dentro de uma string

Ver mais em: https://www.php.net/manual/pt_BR/ref.strings.php

Funções definidas pelo desenvolvedor

Uma função é um bloco de código que executa alguma operação.

Opcionalmente, uma função pode definir parâmetros de entrada que permitem que os chamadores passem argumentos para a função.

Uma função também pode retornar um valor como saída.

```
/*Declaração da função soma  
e os parâmetros $num_1 e $num_2*/  
function soma($num_1,$num_2){  
  
    $total = $num_1 + $num_2;  
  
    return $total;  
}  
  
//definido os valores para as variáveis  
echo soma(5,3);
```

Exercício

Criar uma função para cálculo do IMC, sendo : peso / (altura * altura)

IMC (kg/m ²)	Classificação
Menor que 16,9	Muito abaixo do peso
17 a 18,4	Abaixo do peso
18,5 a 24,9	Peso normal
25 a 29,9	Acima do peso
30 a 34,9	Obesidade grau I
35 a 40	Obesidade grau II
Maior que 40	Obesidade grau III

Vamos Calcular o Delta da Equação?

$$3x^2 - 2x - 8 = 0$$

$$\Delta = b^2 - 4ac$$

$$a = 3$$

$$\Delta = (-2)^2 - 4 \cdot 3 \cdot (-8)$$

$$b = -2$$

$$\Delta = 4 + 96$$

$$c = -8$$

$$\Delta = 100$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

$$x = \frac{-(-2) \pm \sqrt{100}}{2 \cdot 3}$$

$$x = \frac{2 \pm 10}{6}$$

$$x_1 = \frac{2 - 10}{6} = -\frac{8}{6} = -\frac{4}{3}$$

$$x_2 = \frac{2 + 10}{6} = \frac{12}{6} = 2$$

O que estudamos até agora

- Variaveis
- Array
- Operadores (matemáticos, lógicos e comparação)
- Condicionais if e case
- Laços for e while
- Funções numéricas e string
- Funções do desenvolvedor

Banco de dados Mysql

Inclusão:

```
Insert into <nometabela> ('campo1','campo2','campo3')  
values ('valor1','valor2','valor3')
```

Alteração:

```
Update <nometabela> set campo = 'valor',campo2='valor2' Where id=1
```

Exclusão

```
Delete from <nometabela> Where id=1
```

Consulta

```
Select * from <nometabela> // Opcional colocar o Where e selecionar o que quer ver
```