

# Introdução à Programação e Configuração do Ambiente

## O que é Programação?

Imagine que você quer dar instruções para um computador realizar uma tarefa. A programação é a forma de escrever essas instruções de maneira que o computador entenda e execute. É como se você estivesse ensinando o computador a fazer algo, passo a passo.

## Para que serve a Programação?

A programação está presente em tudo o que nos rodeia: desde os aplicativos que usamos no celular, até os sistemas complexos que controlam o tráfego aéreo.

Com a programação, podemos criar soluções para os mais diversos problemas, automatizar tarefas, criar jogos, websites e muito mais.

## Por que PHP?

PHP é uma linguagem de programação muito popular e utilizada para desenvolvimento web. Ela é relativamente fácil de aprender e possui uma grande comunidade de desenvolvedores dispostos a ajudar.

Além disso, o PHP é uma linguagem versátil que permite criar desde páginas web simples até sistemas complexos.

## Nosso Ambiente de Desenvolvimento

Para começarmos a programar em PHP, precisaremos configurar nosso ambiente de desenvolvimento. Utilizaremos as seguintes ferramentas:

- **VSCode:** Um editor de código poderoso e personalizável, onde escreveremos nossos programas.
- **XAMPP:** Um pacote de software que inclui o servidor web Apache, o banco de dados MySQL e o PHP, tudo o que precisamos para rodar nossos programas PHP.
- **Workbench:** Uma ferramenta para gerenciar o banco de dados MySQL, que utilizaremos mais adiante em nossos projetos.

## Instalando e Configurando as Ferramentas

### VSCode:

- Acesse o site oficial do VSCode (<https://code.visualstudio.com/>) e baixe a versão para o seu sistema operacional.
- Instale o VSCode seguindo as instruções do instalador.
- Após a instalação, abra o VSCode e instale a extensão "PHP Intelephense" para ter suporte completo à linguagem PHP.

### XAMPP:

- Acesse o site oficial do XAMPP (<https://www.apachefriends.org/index.html>) e baixe a versão para o seu sistema operacional.
- Instale o XAMPP seguindo as instruções do instalador.

- Após a instalação, abra o XAMPP Control Panel e inicie os serviços "Apache" e "MySQL".

#### **Workbench:**

- Acesse o site (<https://dev.mysql.com/downloads/workbench/>) e baixe a versão para o seu sistema operacional.
- Instale o Workbench seguindo as instruções do instalador.

## **Nosso Primeiro Programa**

Com o ambiente configurado, podemos escrever nosso primeiro programa em PHP. Abra o VSCode e crie um novo arquivo com o nome "hello.php". Dentro desse arquivo, escreva o seguinte código:

```
<?php  
echo "Olá, mundo!";?>
```

Salve o arquivo e abra o seu navegador. Na barra de endereço, digite "localhost/hello.php" e pressione Enter.

Se tudo estiver configurado corretamente, você verá a mensagem "Olá, mundo!" na tela.

## **Curiosidade**

Você sabia que o PHP foi criado em 1994 por Rasmus Lerdorf?

Inicialmente, era apenas um conjunto de scripts para monitorar o tráfego do seu site pessoal, mas com o tempo, evoluiu e se tornou uma das linguagens de programação mais utilizadas no mundo.

O PHP ainda é uma das linguagens de programação mais utilizadas na web.

#### **Porcentagem de sites e sistemas que utilizam PHP:**

- W3Techs: Estima que o PHP é usado por 78,9% de todos os sites que usam uma linguagem de programação do lado do servidor conhecida.
- Kinsta: Relata que o PHP está presente em mais de 70% dos sites na web atualmente.
- Hostinger: Afirma que o PHP é utilizado por 78,1% de todos os sites na internet, já que é a linguagem primária do WordPress.

#### **Observações:**

- A porcentagem exata de sites que utilizam PHP varia de acordo com a fonte e a metodologia de pesquisa.

#### **Fatores que influenciam a popularidade do PHP:**

- Facilidade de aprendizado: O PHP é considerado relativamente fácil de aprender, o que contribui para sua popularidade entre iniciantes.

- Grande comunidade: O PHP possui uma grande comunidade de desenvolvedores, o que facilita a busca por ajuda e recursos.
- WordPress: O WordPress, um dos CMSs mais populares do mundo, é escrito em PHP, o que garante a relevância da linguagem.
- Sistemas legados: Muitas empresas ainda utilizam sistemas legados escritos em PHP, o que mantém a demanda por desenvolvedores PHP.

## Empresas que utilizam PHP

- **Facebook:** A maior rede social do mundo, o Facebook, utiliza PHP em grande parte de sua estrutura. A linguagem é utilizada para gerar páginas dinâmicas, processar dados de usuários e interagir com o banco de dados.
- **Yahoo:** Um dos pioneiros da internet, o Yahoo utiliza PHP em diversos de seus serviços, como o portal de notícias, o serviço de e-mail e a plataforma de buscas.
- **Wikipedia:** A enciclopédia online mais famosa do mundo, a Wikipedia, utiliza PHP para gerar suas páginas, exibir conteúdo dinâmico e interagir com o banco de dados.
- **WordPress:** A plataforma de blogs e sites mais utilizada do mundo, o WordPress, é totalmente escrita em PHP. Isso garante a flexibilidade e a facilidade de uso da plataforma.

## Empresas brasileiras

- **Locaweb:** Uma das maiores empresas de hospedagem de sites e serviços de internet do Brasil, a Locaweb utiliza PHP em seus sistemas para gerenciar contas de clientes, processar pagamentos e oferecer suporte técnico.
- **TOTVS:** Uma das maiores empresas de software da América Latina, a TOTVS utiliza PHP em alguns de seus sistemas, como o sistema de gestão empresarial RM.

## Observação

É importante lembrar que as empresas podem utilizar diversas linguagens de programação em seus sistemas. PHP é apenas uma delas.

Espero que esta informação tenha sido útil. Se tiver mais alguma dúvida, pode perguntar!

# Variáveis e Tipos de Dados em PHP

## *Variáveis*

### O que são?

Variáveis são como caixas onde podemos armazenar informações. Cada caixa tem um nome único que a identifica, e podemos colocar diferentes tipos de informações dentro dela.

### Para que servem?

As variáveis nos permitem guardar dados temporariamente durante a execução do programa. Esses dados podem ser números, textos, resultados de cálculos, etc.

### Como criar?

Em PHP, as variáveis são representadas por um símbolo de dólar (\$) seguido do nome da variável. O nome da variável deve começar com uma letra ou sublinhado, e pode conter letras, números e sublinhados.

### Exemplos

```
$nome = "João"; // Variável que armazena um texto
$idade = 30; // Variável que armazena um número inteiro
$altura = 1.80; // Variável que armazena um número decimal
$casado = true; // Variável que armazena um valor booleano (verdadeiro ou falso)
```

### Regras importantes

- **Nomes:** Devem começar com \$ seguido de uma letra ou sublinhado.
- **Case-sensitive:** PHP diferencia letras maiúsculas e minúsculas (\$nome é diferente de \$Nome).
- **Atribuição:** O sinal de igual (=) é usado para atribuir um valor a uma variável.

### Tipos de Dados

#### O que são?

Os tipos de dados definem o tipo de informação que uma variável pode armazenar. PHP possui diversos tipos de dados, como:

- **Inteiros (int):** Números inteiros (ex: 10, -5, 0).
- **Decimais (float):** Números com casas decimais (ex: 3.14, -2.5).
- **Textos (string):** Sequências de caracteres (ex: "Olá", "PHP").
- **Booleanos (bool):** Valores verdadeiro (true) ou falso (false).
- **Arrays:** Coleções de dados.
- **Objetos:** Instâncias de classes.
- **Nulo (null):** Valor que representa a ausência de um valor.

#### Para que servem?

Os tipos de dados nos ajudam a organizar e manipular as informações de forma correta. Cada tipo de dado possui características e operações específicas.

## Como descobrir o tipo de uma variável?

Podemos usar a função `gettype()` para descobrir o tipo de uma variável.

```
$numero = 10;echo gettype($numero); // Saída: integer
$texto = "Olá";echo gettype($texto); // Saída: string
$booleano = true;echo gettype($booleano); // Saída: boolean
```

## Conversão de Tipos (Type Casting)

### O que é?

A conversão de tipos é a forma de transformar um dado de um tipo para outro.

### Para que serve?

A conversão de tipos é útil quando precisamos realizar operações entre dados de tipos diferentes.

### Como fazer?

Podemos usar o operador de cast (tipo) para converter um dado para um tipo específico.

### Exemplos

```
$numero = 10;
$texto = (string) $numero; // Converte o número para texto
echo gettype($texto); // Saída: string

$texto = "3.14";
$decimal = (float) $texto; // Converte o texto para decimal
echo gettype($decimal); // Saída: double
```

### Dicas

- Escolha nomes de variáveis que sejam claros e descritivos.
- Utilize os tipos de dados corretos para cada situação.
- Faça a conversão de tipos quando necessário.

### Curiosidade

Você sabia que PHP é uma linguagem de tipagem dinâmica? Isso significa que o tipo de uma variável é definido pelo valor que ela armazena, e pode mudar durante a execução do programa.

## Formulário: Fazer exemplo com get e post

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!-- criar um formulario com os campos nome, email e senha -->
  <form action="b.php" method="post">
    <input type="text" name="nome" placeholder="Nome">
    <input type="email" name="email" placeholder="Email">
    <input type="password" name="senha" placeholder="Senha">
    <button type="submit">Enviar</button>
  </form>
</body>
</html>
```

=====

### Sem validar

```
<?php
    $nome = $_POST["nome"];
    $email = $_POST["email"];
    $senha = $_POST["senha"];

    echo "Nome: {$nome}<br>";
    echo "Email: {$email}<br>";
    echo "Senha: {$senha}<br>";
?>
```

### Validação básica

```
<?php
    // valide se o formulario foi enviado
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
        $nome = $_POST["nome"];
        $email = $_POST["email"];
        $senha = $_POST["senha"];

        echo "Nome: {$nome}<br>";
        echo "Email: {$email}<br>";
        echo "Senha: {$senha}<br>";
    }else{
        echo "Formulario não enviado";
    }
?>
```

## Saída de dados, concatenação e interpolação

Em programação, **saída de dados** refere-se ao processo de exibir ou retornar informações para o usuário ou para outro sistema. Em PHP, a saída de dados é comumente realizada utilizando a função `echo`, que imprime uma ou mais strings na tela.

### Exemplo básico de saída de dados:

```
<?php
echo "Olá, mundo!";
?>
```

Neste exemplo, a função `echo` exibe a mensagem "Olá, mundo!" na tela.

## Concatenação de Strings:

A **concatenação** é o processo de unir duas ou mais strings em uma única. Em PHP, o operador de concatenação é o ponto (`.`).

### Exemplo de concatenação:

```
<?php
$nome = "João";
$idade = 25;
echo "Meu nome é " . $nome . " e tenho " . $idade . " anos.";
?>
```

Saída:

Meu nome é João e tenho 25 anos.

Neste exemplo, as variáveis `$nome` e `$idade` são concatenadas com strings literais para formar uma frase completa.

## Interpolação de Strings:

A **interpolação** permite inserir o valor de variáveis diretamente dentro de uma string. Em PHP, isso é possível quando se utiliza aspas duplas (`"`) para definir a string.

### Exemplo de interpolação:

```
<?php
$nome = "João";
$idade = 25;
echo "Meu nome é $nome e tenho $idade anos.";
?>
```

Saída:

Meu nome é João e tenho 25 anos.

Neste exemplo, as variáveis \$nome e \$idade são inseridas diretamente na string, sem a necessidade de concatenação explícita.

### Observações Importantes:

**Aspas Simples vs. Aspas Duplas:** Quando se utiliza aspas simples ('), o PHP não realiza a interpolação de variáveis. Já com aspas duplas ("), a interpolação é realizada.

### Exemplo com aspas simples:

```
<?php
$nome = "João";
echo 'Meu nome é $nome.'; // Saída: Meu nome é $nome.
?>
```

**Uso de chaves para variáveis dentro de strings:** Para evitar ambiguidades, especialmente quando a variável é seguida por caracteres alfanuméricos, é recomendável utilizar chaves {} ao redor da variável.

### Exemplo com chaves:

```
<?php
$nome = "João";
echo "Meu nome é {$nome}123."; // Saída: Meu nome é João123.
?>
```

**Desempenho:** Em termos de desempenho, a concatenação utilizando o operador . pode ser mais eficiente em alguns casos, especialmente quando se trabalha com muitas variáveis.

## Operadores Aritméticos em PHP

### Operadores Aritméticos

#### O que são?

Os operadores aritméticos são símbolos que nos permitem realizar operações matemáticas básicas, como adição, subtração, multiplicação, divisão e resto da divisão.

#### Para que servem?



Os operadores aritméticos são essenciais para realizar cálculos em nossos programas.

## Exemplos

Operador	Nome	Exemplo	Resultado
+	Adição	$\$a + \$b$	Soma de $\$a$ e $\$b$
-	Subtração	$\$a - \$b$	Diferença entre $\$a$ e $\$b$
*	Multiplicação	$\$a * \$b$	Produto de $\$a$ e $\$b$
/	Divisão	$\$a / \$b$	Quociente de $\$a$ e $\$b$
%	Módulo (resto)	$\$a \% \$b$	Resto da divisão de $\$a$ por $\$b$
**	Exponenciação	$\$a ** \$b$	$a$ elevado à potência de $b$

```
$a = 10;
$b = 5;
echo $a + $b;      // Saída: 15
echo $a - $b;      // Saída: 5
echo $a * $b;      // Saída: 50
echo $a / $b;      // Saída: 2
echo $a % $b;      // Saída: 0
echo $a ** $b;     // Saída: 100000
```

## Operadores de Atribuição Combinados

### O que são?

Os operadores de atribuição combinados são formas abreviadas de realizar uma operação aritmética e atribuir o resultado à mesma variável.

### Para que servem?

Os operadores de atribuição combinados tornam o código mais conciso e legível.

## Exemplos

Operador	Equivalente a	Exemplo
+=	$\$a = \$a + \$b$	$\$a += \$b$
-=	$\$a = \$a - \$b$	$\$a -= \$b$
*=	$\$a = \$a * \$b$	$\$a *= \$b$
/=	$\$a = \$a / \$b$	$\$a /= \$b$
%=	$\$a = \$a \% \$b$	$\$a \% = \$b$

```
$a = 10;
$b = 5;
$a += $b;  // $a = $a + $b;
echo $a;   // Saída: 15
```

```
$a -= $b;    // $a = $a - $b;
echo $a;    // Saída: 10

$a *= $b;    // $a = $a * $b;
echo $a;    // Saída: 50

$a /= $b;    // $a = $a / $b;
echo $a;    // Saída: 10

$a %= $b;    // $a = $a % $b;
echo $a;    // Saída: 0
```

## Operadores de Incremento e Decremento

### O que são?

Os operadores de incremento e decremento são utilizados para aumentar ou diminuir o valor de uma variável em 1.

### Para que servem?

Os operadores de incremento e decremento são úteis em loops e outras situações onde precisamos alterar o valor de uma variável de forma rápida.

### Exemplos

Operador	Nome	Exemplo
++	Pré-incremento	++\$a
--	Pré-decremento	--\$a
++	Pós-incremento	\$a++
--	Pós-decremento	\$a--

```
$a = 10;

echo ++$a; // Saída: 11 (pré-incremento)
echo $a;   // Saída: 11

echo $a--; // Saída: 11 (pós-decremento)
echo $a;   // Saída: 10
```

# Lista de exercícios

## 01 - Cálculo de Distância Percorrida

Crie um programa que leia o **tempo** (em horas) e a **velocidade média** (em km/h) e calcule a **distância percorrida** utilizando a fórmula:

$$\text{distância} = \text{Velocidade média} / \text{tempo}$$

*Exemplo:*

Digite o tempo (em horas): 3

Digite a velocidade média (km/h): 60

Distância percorrida: 180.00 km

## 02 - Cálculo da Área de uma Esfera

Crie um programa que leia o **raio** (r) de uma esfera e calcule a **área superficial** da esfera usando a fórmula:

$$A = 4 * \pi * r^2$$

Onde  $\pi$  é aproximadamente 3,14159.

*Exemplo:*

Digite o raio da esfera: 5

Área superficial: 314.16

## 03 - Cálculo do Volume de um Cone

Crie um programa que leia o **raio** (r) e a **altura** (h) de um cone e calcule o **volume** utilizando a fórmula:

$$V = 1/3 * \pi * r^2 * h$$

*Exemplo:*

Digite o raio do cone: 3

Digite a altura do cone: 5

Volume: 141.37

## 04 - Conversão de Temperatura (Escala Fahrenheit para Celsius)

Crie um programa que leia uma **temperatura** em **Fahrenheit** e converta para **Celsius** usando a fórmula:

$$C = 5 / 9 * (F - 32)$$

*Exemplo:*

Digite a temperatura em Fahrenheit: 100

Temperatura em Celsius: 37.8

## 05 - Cálculo do Salário com Desconto de Impostos

Crie um programa que leia o **salário bruto** de um trabalhador e aplique os seguintes descontos:

- Desconto de **INSS** de 8%
- Desconto de **Imposto de Renda** de 12%

Exiba o **salário líquido** após os descontos.

*Exemplo:*

Digite o salário bruto: 3000

Desconto de INSS: 240

Desconto de Imposto de Renda: 360

Salário líquido: 2400

## 06 - Cálculo da Média Ponderada

Crie um programa que leia **três notas** e seus respectivos **pesos**, e calcule a **média ponderada** utilizando a fórmula:

$$Mp = [ (N1 * P1) + (N2 * P2) + (N3 * P3) ] / (P1 + P2 + P3)$$

*Exemplo:*

Digite a primeira nota: 7

Digite o peso da primeira nota: 3

Digite a segunda nota: 8

Digite o peso da segunda nota: 2

Digite a terceira nota: 9

Digite o peso da terceira nota: 1

Média ponderada: 7.67

## 07 - Cálculo da Perda de Massa de um Corpo

Crie um programa que leia a **massa inicial** de um corpo (em kg) e calcule a **massa final** após perder 10% de sua massa inicial, 20% e 30%.

*Exemplo:*

Digite a massa inicial: 80

Massa final após perder 10%: 72.00

Massa final após perder 20%: 64.00

Massa final após perder 30%: 56.00

### Observação:

Trabalhe cada exercício de forma independente, realizando as **operações matemáticas** corretas e utilizando os operadores de forma precisa.

Ao final, **teste os cálculos** com diferentes valores para garantir que o programa está funcionando corretamente.

Faça **reflexões** sobre como os operadores podem ser usados de forma eficiente para resolver problemas matemáticos e lógicos.