Guia Simples de PHP

# 1. O que é PHP?

PHP (Hypertext Preprocessor) é uma linguagem de programação que roda no servidor e é usada para criar sites dinâmicos. Isso significa que as páginas podem mudar de acordo com as informações do usuário ou do sistema. É gratuita, fácil de começar e está presente em grande parte da internet (WordPress, Joomla, Drupal).

# 2. Como Instalar o PHP

Para começar, você precisa de um servidor local. As opções mais usadas são:  
• XAMPP (Windows, Linux, Mac)  
• WAMP (Windows)  
• MAMP (Mac)  
• Laragon (Windows)  
  
Passos básicos:  
1. Baixe e instale o XAMPP (recomendado para iniciantes).  
2. Abra o programa e inicie o Apache (servidor web) e o MySQL (banco de dados, se precisar).  
3. Crie seus arquivos com extensão .php dentro da pasta htdocs (ex: C:/xampp/htdocs/).  
4. Abra no navegador digitando: http://localhost/nomedoarquivo.php

# 3. Primeiro Código em PHP

Crie um arquivo chamado index.php e escreva o seguinte código:

<?php  
echo "Olá, mundo!";  
?>

Ao abrir no navegador (http://localhost/index.php), aparecerá a frase: Olá, mundo!

# 4. Variáveis

Variáveis são como caixinhas onde você guarda informações para usar depois. No PHP, toda variável começa com o símbolo $.

Exemplo:

<?php  
$nome = "Maria"; // Texto  
$idade = 20; // Número  
$estudante = true; // Verdadeiro ou falso  
  
echo "Nome: $nome";  
echo "<br>Idade: $idade";  
?>

# 5. Conceitos Básicos

## 5.1 Operadores Aritméticos (contas)

<?php  
$a = 10;  
$b = 3;  
echo $a + $b; // Mostra 13  
echo $a - $b; // Mostra 7  
echo $a \* $b; // Mostra 30  
echo $a / $b; // Mostra 3.33  
echo $a % $b; // Mostra 1  
?>

## 5.2 Operadores de Comparação (perguntas)

<?php  
$x = 5;  
$y = 10;  
var\_dump($x < $y); // true (verdadeiro)  
var\_dump($x == $y); // false (falso)  
?>

## 5.3 Operadores Lógicos (regras)

<?php  
$idade = 20;  
$temCNH = true;  
  
if ($idade >= 18 && $temCNH) {  
 echo "Pode dirigir";  
} else {  
 echo "Não pode dirigir";  
}  
?>

# 6. Criando Backend para Website – HTML e CSS

Um backend em PHP pode receber informações de formulários e responder dinamicamente.

Exemplo simples de formulário:

<!DOCTYPE html>  
<html>  
<head><title>Formulário</title></head>  
<body>  
 <form method="post" action="">  
 Nome: <input type="text" name="nome"><br>  
 <input type="submit" value="Enviar">  
 </form>  
 <?php  
 if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
 $nome = htmlspecialchars($\_POST["nome"]);  
 if (!empty($nome)) {  
 echo "<p>Olá, $nome!</p>";  
 } else {  
 echo "<p style='color:red;'>Por favor, preencha seu nome.</p>";  
 }  
 }  
 ?>  
</body>  
</html>

Exemplo de CSS para melhorar o visual:

<style>  
 body { font-family: Arial, sans-serif; padding: 20px; }  
 form { margin-bottom: 10px; }  
 input[type="text"] { padding: 5px; margin: 5px 0; }  
 input[type="submit"] { padding: 5px 10px; cursor: pointer; }  
 p { font-weight: bold; }  
</style>

# 7. Boas Práticas

• Sempre valide os dados enviados pelo usuário.

• Use comentários no código para se lembrar do que cada parte faz.

• Organize seus arquivos em pastas (css/, js/, includes/).

• Separe o máximo possível HTML do PHP para ficar mais claro.

# 8. Colocando seu Projeto Online

Quando terminar seu projeto, você pode colocar ele online em uma hospedagem. Isso pode ser feito enviando os arquivos por FTP ou usando o painel de controle do servidor. A maioria das hospedagens já suporta PHP e banco de dados MySQL.

# 9. Recursos Extras

• Documentação oficial: https://www.php.net/docs.php

• Tutoriais no YouTube sobre PHP para iniciantes.

• Comunidades online de programação em PHP.

# 10. Conclusão

PHP é uma linguagem simples de aprender e muito poderosa. Com este guia você já sabe como instalar, criar seu primeiro código, trabalhar com variáveis, operadores e até criar um formulário simples com HTML, CSS e PHP.