

PHP

Driblando desafios Iniciais



Guilherme da Silva



Bem-vindo!

Este e-book foi criado para ajudar iniciantes a entender e superar os problemas mais comuns no desenvolvimento com PHP. Se você já se sentiu perdido ao tentar configurar seu ambiente ou lidar com erros de banco de dados, você está no lugar certo. Vamos fazer isso juntos!

Por Que PHP?

PHP é uma linguagem de programação poderosa e popular, especialmente para desenvolvimento web. Ela alimenta sites como Facebook e WordPress, mostrando seu potencial e versatilidade. Aprender PHP pode abrir muitas portas no mundo do desenvolvimento web.

01

Configurando o Ambiente



Instalando o PHP

Para começar a desenvolver em PHP, você precisa instalar o PHP no seu computador. Aqui estão os passos para diferentes sistemas operacionais:

Windows:

Baixe o PHP do site oficial

(<https://www.php.net/downloads>).

Extraia os arquivos para uma pasta, por exemplo, C:\php.

Adicione C:\php ao PATH do sistema.

Teste a instalação abrindo o Prompt de Comando e digitando php -v.

macOS:

Instale o Homebrew, se ainda não tiver (<https://brew.sh/>).

Digite brew install php.

Verifique a instalação com php -v.

Linux:

Atualize seus pacotes com sudo apt update.

Instale o PHP com sudo apt install php.

Verifique a instalação com php -v.



Servidor Local

Para desenvolver e testar suas aplicações PHP localmente, você precisa de um servidor local. XAMPP e MAMP são ótimas opções.

XAMPP:

Baixe o XAMPP de

<https://www.apachefriends.org/index.html>.

Instale o XAMPP e inicie o Apache e MySQL através do painel de controle.

Coloque seus arquivos PHP na pasta htdocs (geralmente em C:\xampp\htdocs).

MAMP:

Baixe o MAMP de <https://www.mamp.info/en/>.

Instale e inicie o MAMP.

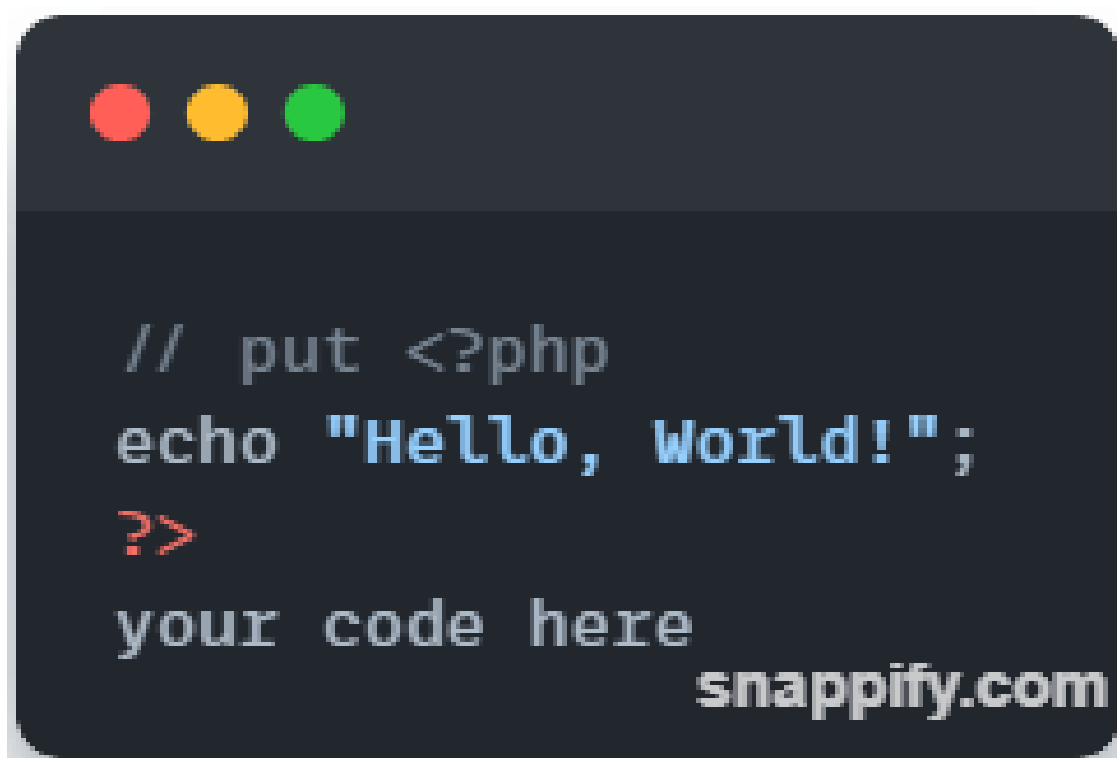
Coloque seus arquivos PHP na pasta htdocs (geralmente em /Applications/MAMP/htdocs).

Seu Primeiro Script Php

Abra seu editor de texto favorito (VSCode, Sublime Text, etc.).

Crie um novo arquivo chamado index.php.

Adicione o seguinte código:



```
// put <?php
echo "Hello, World!";
?>
your code here
snappify.com
```

Salve o arquivo na pasta htdocs do XAMPP ou MAMP.

Abra seu navegador e acesse <http://localhost/index.php>.

Você deve ver "Hello, World!" na tela.

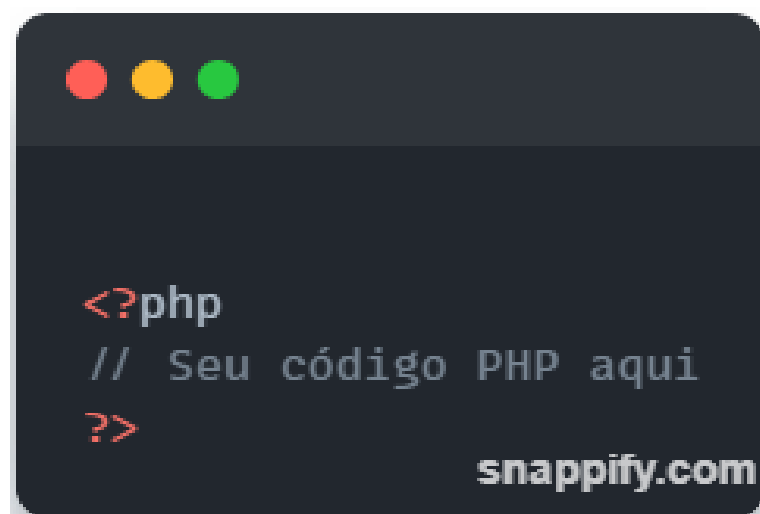
02

ENTENDENDO A ESTRUTURA DO PHP

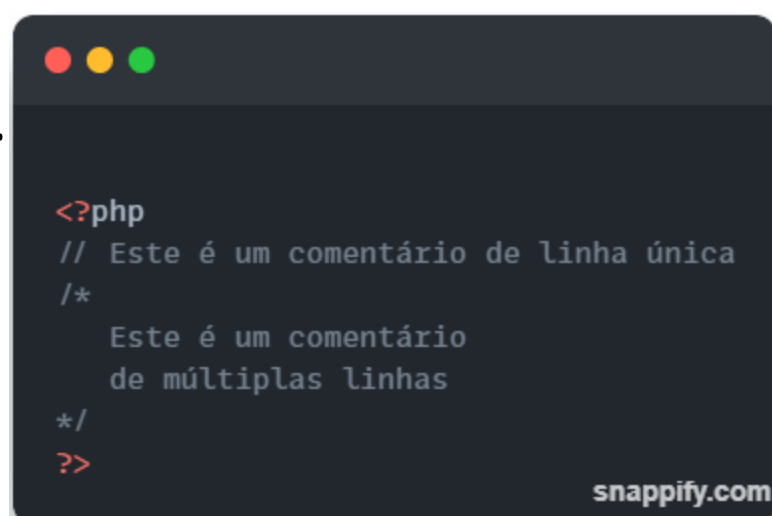
Sintaxe Básica

Vamos entender a estrutura básica de um script PHP. PHP é uma linguagem de scripts que roda no servidor, o que significa que o código é executado no servidor e o resultado HTML é enviado ao navegador.

Tags PHP:



Comentários:

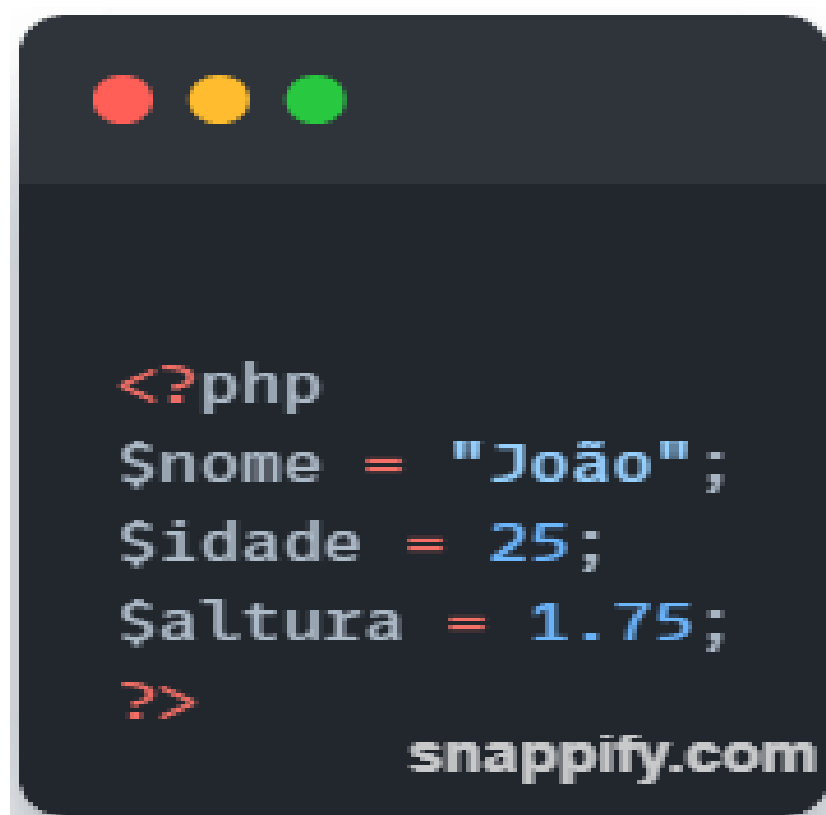


Comentários são essenciais para tornar seu código legível e fácil de manter. Use comentários para explicar o que o código faz ou por que certas decisões foram tomadas

Váriaveis e Tipos de dados

Variáveis em PHP são usadas para armazenar dados. Elas começam com um \$ seguido do nome da variável.

Declarando Variáveis:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) at the top. It contains PHP code for declaring variables: \$nome, \$idade, and \$altura. The code is color-coded: opening and closing PHP tags are red, variable names are blue, and string literals and numeric values are green. The website 'snappify.com' is visible in the bottom right corner of the editor.

```
<?php
$nome = "João";
$idade = 25;
$altura = 1.75;
?>
```

snappify.com

Tipos de Dados Comuns:

Strings: "\$nome"

Inteiros: \$idade

Float: \$altura

Boolean: true ou false

03

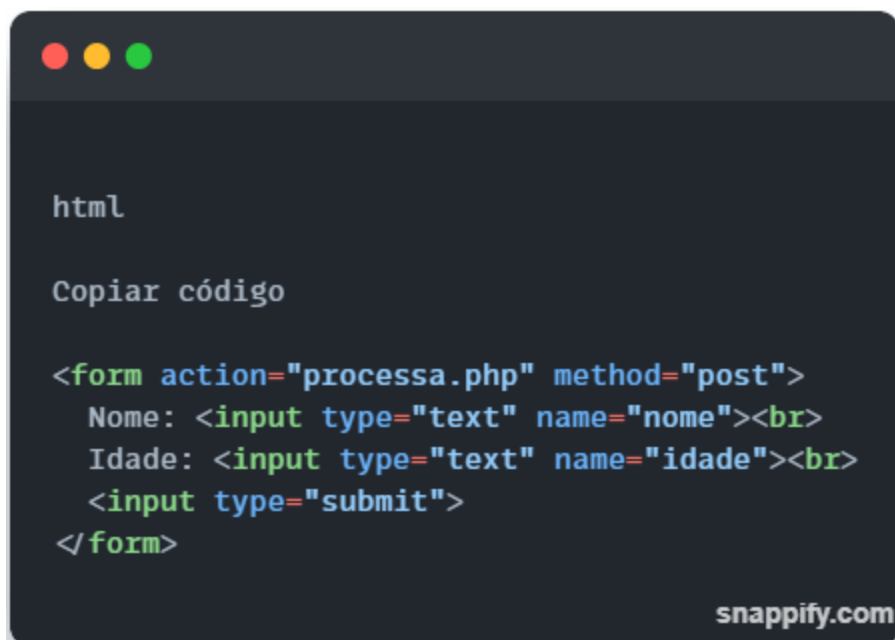
**MANIPULAÇÃO
DE
FORMULÁRIOS**

Manipulação de Formulários

Recebendo Dados de Formulários

Formulários HTML são usados para coletar dados do usuário. PHP pode processar esses dados usando `$_GET` ou `$_POST`.

Exemplo de Formulário HTML:

A code editor window with a dark background and light-colored text. It features a title bar with three colored circles (red, yellow, green) on the left. The code is as follows:

```
html

Copiar código

<form action="processa.php" method="post">
  Nome: <input type="text" name="nome"><br>
  Idade: <input type="text" name="idade"><br>
  <input type="submit">
</form>
```

In the bottom right corner, the text "snappify.com" is visible.

Processando os Dados com PHP:



```
<?php
$nome = $_POST['nome'];
$idade = $_POST['idade'];
echo "Nome: $nome<br>";
echo "Idade: $idade";
?>
```

snappify.com

Validação de Dados

Validação é importante para garantir que os dados recebidos são corretos e seguros.

Exemplo de Validação Simples:



```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nome = $_POST['nome'];
    $idade = $_POST['idade'];
    if (empty($nome) || empty($idade)) {
        echo "Todos os campos são obrigatórios.";
    } else {
        echo "Nome: $nome<br>";
        echo "Idade: $idade";
    }
}
?>
```

snappify.com

Segurança

Proteja seu site contra SQL Injection validando e sanitizando os dados de entrada.

Exemplo de Sanitização:

```
<?php
$nome = htmlspecialchars($_POST['nome']);
$idade = (int) $_POST['idade'];
?>
```

snappify.com

Processando os Dados com PHP:



```
<?php
$nome = $_POST['nome'];
$idade = $_POST['idade'];
echo "Nome: $nome<br>";
echo "Idade: $idade";
?>
```

snappify.com

Validação de Dados

Validação é importante para garantir que os dados recebidos são corretos e seguros.

Exemplo de Validação Simples:



```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nome = $_POST['nome'];
    $idade = $_POST['idade'];
    if (empty($nome) || empty($idade)) {
        echo "Todos os campos são obrigatórios.";
    } else {
        echo "Nome: $nome<br>";
        echo "Idade: $idade";
    }
}
?>
```

snappify.com

04

**TRABALHANDO
COM BANCO DE
DADOS**

Trabalhando com Banco de Dados

Conectando ao Banco de Dados

Para conectar-se a um banco de dados MySQL, você pode usar MySQLi ou PDO.

Usando MySQLi:

```
<?php
$conn = new mysqli("localhost", "username", "password", "database");
if ($conn->connect_error) {
    die("Conexão falhou: " . $conn->connect_error);
}
?>
```



X

Usando PDO:

```
php
Copiar código

<?php
try {
    $conn = new PDO("mysql:host=localhost;dbname=database", "username", "password");
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
} catch(PDOException $e) {
    echo "Conexão falhou: " . $e->getMessage();
}
?>
```



X

Processando os Dados com PHP:



```
<?php
$nome = $_POST['nome'];
$idade = $_POST['idade'];
echo "Nome: $nome<br>";
echo "Idade: $idade";
?>
```

snappify.com

Validação de Dados

Validação é importante para garantir que os dados recebidos são corretos e seguros.

Exemplo de Validação Simples:



```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nome = $_POST['nome'];
    $idade = $_POST['idade'];
    if (empty($nome) || empty($idade)) {
        echo "Todos os campos são obrigatórios.";
    } else {
        echo "Nome: $nome<br>";
        echo "Idade: $idade";
    }
}
?>
```

snappify.com

05

SESSÕES E COOKIES



O Que São Sessões e Cookies?

Sessões e cookies são métodos para armazenar dados do usuário entre diferentes páginas.

Sessões:

Armazenadas no servidor.

Expiram quando o navegador é fechado (a menos que configurado de outra forma).

Cookies:

Armazenados no navegador do usuário.

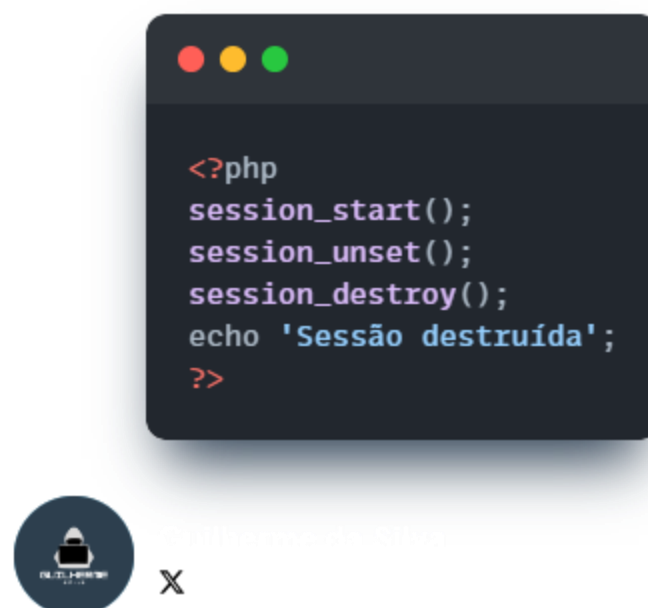
Podem ter uma duração específica.

Usando Sessões Sessões são úteis para armazenar informações de login ou preferências do usuário.

Iniciando uma Sessão:



Destruindo uma Sessão:



Usando Cookies

Cookies são úteis para armazenar dados do usuário por longos períodos.

Criando um Cookie:

```
<?php
setcookie("username", "João", time() + (86400 * 30), "/"); // 86400 = 1 dia
echo 'Cookie criada';
?>
```



X

Acessando um Cookie:

```
<?php
if(isset($_COOKIE["username"])) {
    echo 'Bem-vindo ' . $_COOKIE["username"];
} else {
    echo 'Cookie não encontrado';
}
?>
```



X

06

**DEPURAÇÃO E
TRATAMENTO
DE ERROS**



Tipos de erros em PHP

Erros comuns em PHP incluem erros de sintaxe, runtime e lógica.

Erros de Sintaxe:

Ocorrências comuns: falta de ponto e vírgula, parênteses não fechados.

Mensagem de erro típica: "Parse error: syntax error, unexpected..."

Erros de Runtime:

Ocorrências comuns: tentativa de acessar uma variável indefinida, divisão por zero.

Mensagem de erro típica: "Warning: Undefined variable..."

Erros de Lógica:

Ocorrências comuns: lógica incorreta em loops e condicionais.

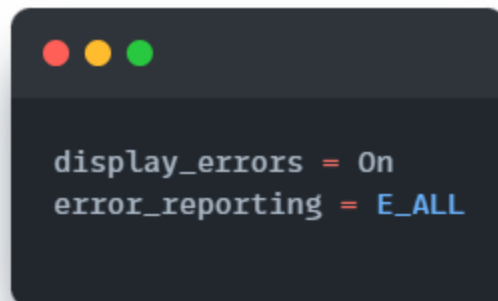
Mensagem de erro típica: Não há mensagens automáticas, apenas resultados inesperados.

Exibindo Erros Configure o PHP para mostrar erros durante o desenvolvimento.

Exibindo Erros

Configure o PHP para mostrar erros durante o desenvolvimento.

Exibir Erros no php.ini:




```
display_errors = On  
error_reporting = E_ALL
```



Ultimate Programmer



Exibir Erros no Código:



```
<?php  
ini_set('display_errors', 1);  
ini_set('display_startup_errors', 1);  
error_reporting(E_ALL);  
?>
```



Ultimate Programmer



Depuração com var_dump e print_r

Use essas funções para inspecionar variáveis e estruturas de dados.

Usando var_dump:

```
<?php
$arr = array(1, 2, 3);
var_dump($arr);
?>
```



FullstackSolutions



Usando print_r:

```
<?php
$arr = array(1, 2, 3);
print_r($arr);
?>
```



FullstackSolutions



Tratamento de Exceções

Use try-catch para capturar e tratar exceções.

Exemplo de Exceção:

```
<?php
function dividir($a, $b) {
    if ($b == 0) {
        throw new Exception("Divisão por zero.");
    }
    return $a / $b;
}

try {
    echo dividir(5, 0);
} catch (Exception $e) {
    echo 'Erro: ' . $e->getMessage();
}

?>
```





AGRADECIMENTOS

AGRADECIMENTOS



Obrigado Por Ler até aqui!

Este foi meu primeiro e-book, gerado por IA, e diagramado Por mim, foi uma desafio proposto durante um Bootcamp da [Dio](#)

Por Que PHP?

PHP é a linguagem de programação que eu mais tive contato durante os meus anos de estudo, a qual gosto muito e desenvolvi grande partes dos meus projetos optando por ela. E por isso já tive muitos problemas com erros simples que custão a ser solucionados, e alguns deles estão neste e-book, que me ajudariam muito caso existissem quando foi a minha vez.



<https://github.com/guilhermedasilvaa/guilhermedasilvaa.git>