



**PHP: A Magia do Código que
Conecta Dois Mundos**

1: Introdução ao PHP

Neste capítulo, você será apresentado ao PHP e sua importância no desenvolvimento web. Veremos como configurar o ambiente de desenvolvimento PHP e entender a estrutura básica de um script PHP.

```
<?php
// Exemplo de código PHP simples
echo "Olá, Mundo!";
?>
```

1: Introdução ao PHP

Tipos de input

String: \$nome = "João"; // Armazena um nome

Int: \$idade = 25; // Armazena uma idade

Float: \$altura = 1.75;

```
<?php
// Exemplo de código PHP simples
echo "Olá, Mundo!";
?>
```

1: Introdução ao PHP

Tipos de input

Boolean: True ou False,

Array: array("maçã", "banana", "laranja")

Null: Null

```
<?php
// Exemplo de código PHP simples
echo "Olá, Mundo!";
?>
```

1: Introdução ao PHP

Tipos de input

Boolean: True ou False,

Array: array("maçã", "banana", "laranja", 1, 2, 3)

Null: Null

```
<?php
// Exemplo de código PHP simples
echo "Olá, Mundo!";
?>
```

1: Introdução ao PHP

Exemplos tipos de input

```
// Exibindo os valores das variáveis
echo "Nome: " . $nome . "\n";
echo "Sobrenome: " . $sobrenome . "\n";
echo "Idade: " . $idade . "\n";
echo "Altura: " . $altura . "\n";
echo "Aprovado: " . $aprovado . "\n";
echo "Frutas: " . implode(", ", $frutas) . "\n";
echo "Valor: " . $valor . "\n";

?>
```


2: Manipulação de Inputs

Aqui, iremos explorar a manipulação de inputs em PHP. Abordaremos o processamento de formulários HTML, desde a criação até a submissão dos dados para processamento em um script PHP. Você aprenderá a recuperar os dados do formulário e a utilizá-los em seu código.

2: Manipulação de Inputs

- Os formulários HTML são usados para coletar dados dos usuários, como nome, e-mail, etc.
- Ao criar um formulário, é necessário definir o método HTTP (geralmente POST ou GET) e o atributo action, que indica para qual URL os dados serão enviados.

```
<!-- Exemplo de formulário HTML -->  
<form method="POST" action="processar.php">  
  <label for="nome">Nome:</label>  
  <input type="text" name="nome" id="nome">  
  
  <input type="submit" value="Enviar">  
</form>
```


2: Manipulação de Inputs

- No lado do servidor, os dados do formulário são acessados por meio da superglobal `$_POST`, usando o nome dos campos como índices.
- Os dados recebidos podem ser utilizados em um script PHP para executar ações, como salvar em um banco de dados ou enviar e-mails.

```
// processar.php - Exemplo de script PHP para manipular o input
```

```
    $nome = $_POST['nome'];
```

```
    echo "Olá," . $nome . "!";
```

```
// Verifica se o formulário foi submetidoif ($_SERVER['REQUEST_METHOD'] === 'POST')
```

```
{
```

```
    // Obtém o valor do input 'nome' e remove espaços em branco $nome =
```

```
    trim($_POST['nome']);
```

```
}
```

3: Validação de Inputs

A validação de inputs é uma etapa crucial para garantir a integridade dos dados recebidos. Neste capítulo, mostraremos como validar os dados inseridos pelos usuários, utilizando funções e recursos específicos do PHP. Veremos como garantir que os inputs atendam a critérios específicos, como formato de e-mail, comprimento mínimo ou máximo, entre outros.

```
// Verifica se o campo 'email' está preenchido
if (!empty($email)) {
    // Sanitiza o valor do input 'email' removendo caracteres inválidos    $email =
    filter_var($email, FILTER_SANITIZE_EMAIL);
```

3: Validação de Inputs

- No exemplo, utilizamos a função `filter_var()` com o filtro `FILTER_SANITIZE_STRING` para remover tags HTML e caracteres especiais.
- Outros filtros podem ser usados, como `FILTER_SANITIZE_EMAIL` para remover caracteres inválidos de um endereço de e-mail.

```
if (!empty($nome)) {  
    // Sanitiza o valor do input 'nome' removendo tags HTML    $nome = filter_var($nome,  
        FILTER_SANITIZE_STRING);  
    // Exibe a mensagem com o nome inserido    echo "Olá, $nome! Seu nome foi enviado  
        com sucesso.";  
    } else {  
        // Exibe uma mensagem de erro se o campo 'nome' estiver vazio    echo "Por favor,  
        preencha o campo nome.";}  
}
```