



AULA 1 – VEM COMIGO PHP

Seja bem-vindo ao Vem Comigo PHP, um curso voltado para iniciantes que desejam aprender PHP de maneira prática e dinâmica! Aqui, você vai adquirir conhecimentos fundamentais sobre a linguagem de programação PHP, amplamente usada no desenvolvimento web.

PHP (Hypertext Preprocessor) é uma linguagem de script do lado do servidor, usada principalmente para o desenvolvimento de sites dinâmicos e aplicativos web. A sua simplicidade, flexibilidade e integração com diversos bancos de dados tornam o PHP uma das linguagens mais populares para web development.

No PHP, as **tags** são os delimitadores usados para abrir e fechar blocos de código PHP dentro de um arquivo. Elas indicam ao servidor onde o PHP começa e onde termina, permitindo que você insira scripts PHP dentro de arquivos HTML ou outros tipos de documentos. Existem diferentes tipos de tags no PHP, e cada uma tem seu uso específico. Vamos entender as principais:

1. Tag padrão: `<?php ?>`

Essa é a forma mais comum e recomendada para escrever código PHP. Sempre que você quiser executar um script PHP, deve utilizar essa tag. Veja um exemplo:

php

Copiar código

```
<?php
```

```
    echo "Olá, mundo!";
```

```
?>
```

- **Abertura:** `<?php`
- **Fechamento:** `?>`

O código PHP entre essas tags será processado pelo servidor, e o que estiver fora será tratado como HTML comum.

2. Tag curta: `<? ?>`

Essa é uma forma abreviada da tag padrão, permitindo abrir o bloco de código PHP com apenas `<?` ao invés de `<?php`. Exemplo:

php

Copiar código



HOT CODE

```
<?
```

```
    echo "Olá, mundo!";
```

```
?>
```

No entanto, **essa tag curta pode não estar habilitada por padrão em algumas configurações de servidor**, o que pode causar problemas de compatibilidade. Por esse motivo, é **recomendado evitar o uso de tags curtas** e sempre optar pela tag padrão.

3. Tag de impressão: <?= ?>

Essa tag é uma versão simplificada e abreviada da tag padrão, destinada apenas para imprimir valores. Ela equivale ao uso de <?php echo ... ?>, mas com menos escrita. Exemplo:

```
php
```

Copiar código

```
<?= "Olá, mundo!" ?>
```

O código acima é equivalente a:

```
php
```

Copiar código

```
<?php echo "Olá, mundo!"; ?>
```

Essa tag é amplamente utilizada para facilitar a exibição de dados e está habilitada por padrão nas versões mais recentes do PHP.

4. Tag de script HTML

Além de código PHP, é comum que um arquivo misture HTML com PHP. Veja como o PHP pode ser inserido em um contexto HTML:

```
html
```

Copiar código

```
<!DOCTYPE html>
```

```
<html>
```

```
    <head>
```

```
        <title>Meu site em PHP</title>
```

```
    </head>
```



HOT CODE

```
<body>

    <h1>Bem-vindo ao meu site!</h1>

    <p>Hoje é: <?php echo date("d/m/Y"); ?></p>

</body>

</html>
```

Aqui, o PHP é usado dentro de um arquivo HTML para gerar conteúdo dinâmico, como a data atual.

5. Tag de estilo XML: <?xml ... ?>

Ao trabalhar com documentos XML dentro de arquivos PHP, o cabeçalho XML (<?xml ... ?>) pode ser confundido com as tags PHP. Para evitar que o PHP interprete isso como um bloco de código PHP, é recomendável usar a função echo para imprimir o cabeçalho XML ou configurar a saída adequadamente.

Conclusão

As tags PHP são fundamentais para delimitar onde o código PHP deve ser processado dentro de um arquivo. Embora existam diferentes formas de abrir e fechar blocos de PHP, a tag **padrão (<?php ?>)** e a **tag de impressão (<?= ?>)** são as mais recomendadas e amplamente utilizadas para garantir compatibilidade e boas práticas no desenvolvimento.

Métodos GET e POST

Os métodos GET e POST são usados para enviar dados de um formulário (ou por um link, no caso do GET) HTML para um servidor em PHP, mas eles funcionam de maneiras diferentes:

Método GET

- Envio de Dados: Os dados são enviados anexados à URL, visíveis na barra de endereços do navegador.
- Limite de Tamanho: Há um limite de caracteres (geralmente até 2048 caracteres).
- Segurança: Menos seguro, pois os dados são visíveis na URL. Não é recomendado para enviar informações sensíveis.
- Uso Comum: Ideal para consultas rápidas e envio de dados não sensíveis, como parâmetros de pesquisa.

Exemplo de uso:

PHP



HOT CODE

```
<form method="GET" action="processa.php">  
  <input type="text" name="nome">  
  <input type="submit" value="Enviar">  
</form>
```

Método POST

- Envio de Dados: Os dados são enviados no corpo da requisição HTTP, não visíveis na URL.
- Limite de Tamanho: Praticamente sem limite de tamanho.
- Segurança: Mais seguro que o GET, pois os dados não são visíveis na URL.
- Uso Comum: Ideal para enviar dados sensíveis ou grandes quantidades de dados, como formulários de login e upload de arquivos.

Exemplo de uso:

PHP

```
<form method="POST" action="processa.php">  
  <input type="text" name="nome">  
  <input type="submit" value="Enviar">  
</form>
```

A função include do PHP é uma das formas mais utilizadas para inserir o conteúdo de um arquivo externo dentro de um script PHP. Isso permite que você reaproveite código e mantenha a organização dos seus projetos, facilitando a manutenção e a modularização do sistema.

Como funciona o include

A função include pega o conteúdo de um arquivo e o insere onde ela for chamada no código. Esse arquivo pode conter qualquer tipo de código válido em PHP, HTML, CSS ou até mesmo JavaScript. Quando o arquivo é incluído, o PHP processa o conteúdo como se ele fizesse parte do código original.

Exemplo básico:

php

Copiar código

// arquivo principal.php

<?php



```
include 'cabecalho.php'; // Inclui o arquivo cabecalho.php

echo "Conteúdo principal do site.";

?>
```

Nesse exemplo, o arquivo `cabecalho.php` será incluído no ponto onde a função `include` foi chamada. Se o `cabecalho.php` contiver um menu de navegação ou um cabeçalho HTML, ele será renderizado juntamente com o conteúdo do arquivo principal `principal.php`.

Vantagens do `include`

1. **Reaproveitamento de código:** Um dos maiores benefícios do `include` é a possibilidade de reutilizar o mesmo código em diferentes partes do site. Por exemplo, você pode criar um arquivo `menu.php` e incluí-lo em todas as páginas do seu site, mantendo o menu sempre igual em todas elas.
2. **Organização:** A função `include` ajuda a organizar melhor o código, dividindo-o em arquivos menores e mais focados. Isso facilita a leitura e a manutenção.
3. **Modularidade:** Com `include`, é possível criar módulos independentes para diferentes partes de uma aplicação. Por exemplo, um arquivo para conexão com o banco de dados (`conexao.php`) pode ser incluído sempre que necessário, sem a necessidade de duplicar esse código em várias páginas.

O que acontece se o arquivo não for encontrado?

Se o arquivo especificado não for encontrado, o PHP emitirá um aviso (warning), mas continuará executando o script. Isso pode ser útil quando a ausência de um arquivo não deve interromper a execução completa do sistema. Exemplo:

php

Copiar código

```
<?php

include 'arquivo_que_nao_existe.php';

echo "Ainda assim, o script continua rodando.";

?>
```

Mesmo que o arquivo `arquivo_que_nao_existe.php` não esteja disponível, a mensagem "Ainda assim, o script continua rodando." será exibida.

Diferença entre `include` e `require`



Além do `include`, o PHP também tem a função `require`, que é muito semelhante. A principal diferença está no comportamento quando o arquivo a ser incluído não é encontrado:

- `include`: Gera um aviso (warning) e continua a execução do script.
- `require`: Gera um erro fatal e interrompe a execução do script.

Por isso, o `require` é geralmente usado quando a ausência do arquivo é crítica para o funcionamento do sistema, como ao incluir um arquivo de configuração essencial.

`include_once`

Outra variante do `include` é o `include_once`. Essa função funciona da mesma forma que o `include`, com a diferença de que garante que o arquivo será incluído somente uma vez no script, mesmo que você tente incluí-lo várias vezes. Isso é útil para evitar a inclusão duplicada de arquivos, o que pode causar erros, especialmente em casos de definições de funções ou classes.

Exemplo de `include_once`:

php

Copiar código

```
<?php
    include_once 'config.php';

    include_once 'config.php'; // Esse arquivo será incluído apenas uma vez

?>
```

Conclusão

A função `include` do PHP é uma poderosa ferramenta para melhorar a modularidade, a organização e o reaproveitamento de código dentro de projetos. Ela é amplamente usada para incorporar arquivos externos em scripts PHP, como cabeçalhos, rodapés, menus, e conexões de banco de dados. Embora similar ao `require`, sua flexibilidade em continuar o script mesmo que o arquivo não seja encontrado a torna uma escolha mais adequada em situações onde a ausência do arquivo não é crítica.