

# Programação Dinâmica para Web

Prof. Daniel Brandão

[www.DanielBrandao.com.br](http://www.DanielBrandao.com.br)

## Objetivos

- Apresentar as estruturas condicionais da linguagem PHP
- Formulários HTML e PHP
- Funções POST, GET e REQUEST

# 1. PHP - Estruturas Condicionais

# PHP - Estruturas Condicionais

- Usadas para executar diferentes ações baseadas em diferentes condições;
- Em PHP temos as seguintes estruturas condicionais:
  - if (SE)
  - if...else (SE...SENÃO)
  - if...elseif...else (SE...SENÃO SE...SENÃO)
  - switch (ESCOLHA)

# PHP - Estruturas Condicionais

- A estrutura **if**
  - Delimita uma lista de instruções que será executada apenas se uma condição específica for verdadeira;
  - Sintaxe:

```
if (condição)
{
    // região de código
}
```

# PHP - Estruturas Condicionais

- A estrutura **if**
  - Exemplo:

```
<?php
    $d = 2;
    if ($d > 0)
    {
        echo "Número positivo";
    }
?>
```

# PHP - Estruturas Condicionais

- A estrutura **if...else**

- Delimita duas listas de instruções. Caso a condição definida for verdadeira, a primeira lista será executada. Caso contrário, a segunda lista será executada;
  - Sintaxe:

```
if (condição)
{
    // região de código 1
}
else
{
    // região de código 2
}
```

# PHP - Estruturas Condicionais

- A estrutura **if...else**
  - Exemplo:

```
<?php
    $d = 2;
    if ($d < 0)
        echo "Número negativo";
    else
        echo "Número positivo";
?>
```

# PHP - Estruturas Condicionais

- A estrutura **if...elseif...else**
  - Delimita várias regiões de código de acordo com determinadas condições;

- Sintaxe:

```
if (condição)
    // região de código
elseif (condição)
    // região de código
else
    // região de código
```

# PHP - Estruturas Condicionais

- A estrutura **if...elseif...else**
  - Exemplo:

```
<?php
    $d = 2;
    if ($d < 0)
        echo "Menor que zero";
    elseif ($d > 0)
        echo "Maior que zero";
    else
        echo "Igual a zero";
?>
```

# PHP - Estruturas Condicionais

- A declaração switch

- Compara um dado valor a constantes, desviando o fluxo de código para a constante que seja igual ao valor comparado;
- Sintaxe:

```
switch ($variavel)
{
    case valor_1:
        // Lista de instruções
        break;
    case valor_2:
        // Lista de instruções
        break;
    default:
        // Lista de instruções
}
```

# PHP - Estruturas Condicionais

- Exemplo:

```
<?php
    switch ($x)
    {
        case 1:
            echo "Número 1";
            break;
        case 2:
            echo "Número 2";
            break;
        default:
            echo "Não é 1 ou 2";
    }
?>
```

## 2. Formulários HTML e PHP

# Tratando dados de *forms* HTML

- A partir de formulários HTML, podemos fazer com que usuários submetam dados para scripts PHP;
- Elementos contidos em inputs HTML (caixa de texto, caixa de seleção, botões de seleção, etc.) poderão ser disponibilizados automaticamente para scripts PHP;

# Tratando dados de *forms* HTML

- Exemplo: Formulário HTML que contem dois campos de texto e um botão de submissão.

```
<html>
<body>
    <form action="teste.php" method="POST">
        Nome: <input type="text" name="nome" /><br/>
        Idade: <input type="text" name="idade" /><br/>
        <input type="Submit" value="Enviar" />
    </form>
</body>
</html>
```

# Tratando dados de *forms* HTML

- Quando clicamos no botão de submissão em um formulário HTML, os dados são **submetidos** para o arquivo indicado no atributo **action** da tag **form**;
- No exemplo anterior, os dados contidos no formulário (nome e idade) serão enviados para o script **teste.php**;
- Portanto, precisamos criar o script que irá tratar os dados enviados pelo formulário HTML;

# Tratando dados de *forms* HTML

- **teste.php** poderia ser editado da seguinte maneira:

```
<?php  
    $nome = $_POST['nome'];  
    $idade = $_POST['idade'];  
  
    echo "Nome do usuário: " . $nome;  
    echo "Sua idade: " . $idade;  
?>
```

### 3. A função `$_POST`

# A função `$_POST`

- A função PHP pré-definida `$_POST` é usada para receber valores de formulário que foram submetidas através do método **POST**;
- Tal método é indicado no atributo **method** da tag HTML **form**;
- Informações enviadas via formulários através do método POST são invisíveis para os usuários;
- Não há limites em relação a quantidade de dados a serem enviados;

# A função `$_POST`

- Ex.:
  - Formulário HTML:

```
<html>  
<body>  
<form action='teste2.php' method='post'>  
Seu time: <input type='text' name='time' />  
<input type='submit' value='Enviar' />  
</form>  
</body>  
</html>
```

# A função `$_POST`

- Ex.:
  - Script PHP (teste2.php):

```
<?php
```

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

```
echo 'Sou torcedor do ' . $time;
```

```
?>
```

# 3. A função `$_GET`

# A função `$_GET`

- A função PHP pré-definida `$_GET` é usada para receber valores de formulário que foram submetidas através do método **GET**;
- Esse método é indicado no atributo **method** da tag HTML **form**;
- Informações enviadas via formulários através do método GET são visíveis para os usuários (visíveis na barra de endereços do navegador);
- Há limites em relação a quantidade de dados a serem enviados;

# A função \$\_GET

- Ex.:
  - Formulário HTML:

```
<html>
<body>
<form action='teste2.php' method='get'>
Seu time: <input type='text' name='time' />
<input type='submit' value='Enviar' />
</form>
</body>
</html>
```

# A função `$_GET`

- Ex.:
  - Script PHP (teste2.php):

```
<?php  
  
$time = $_GET['time'];  
  
echo 'Sou torcedor do ' . $time;
```

```
?>
```

# 4. A função `$_REQUEST`

# A função `$_REQUEST`

- A função PHP pré-definida `$_REQUEST` é usada para receber valores de formulários que foram submetidas através do método **GET** ou **POST**;
- Não importa qual método de envio foi definido no formulário HTML. A função `$_REQUEST` obterá os dados enviados, reconhecendo automaticamente o padrão de envio de dados;

# A função `$_REQUEST`

- Ex.:
  - Formulário HTML:

```
<html>  
<body>  
<form action='teste2.php' method='get'>  
Seu time: <input type='text' name='time' />  
<input type='submit' value='Enviar' />  
</form>  
</body>  
</html>
```

# A função `$_REQUEST`

- Ex.:
  - Script PHP (teste2.php):

```
<?php
```

```
$time = $_REQUEST['time'];
```

```
echo 'Sou torcedor do ' . $time;
```

```
?>
```

# Exercícios

# Exercícios

- 1) Faça um documento HTML que permita ao usuário digitar um número;
- Faça um script PHP que receba os dados submetidos a partir do documento HTML anterior. O script deverá informar se o número digitado é positivo ou negativo.

# Exercícios

2) Crie um documento HTML que permita ao usuário digitar o nome de um aluno e suas duas notas na disciplina de Programação Dinâmica para Web no semestre 2016.2;

Faça um script PHP que receba os dados submetidos a partir do documento HTML anterior, calcula a média do aluno e exiba a seguinte mensagem: “O aluno XXXXX ficou com XX,X de média”;

# Exercícios

3) Um funcionário recebe um salário fixo mais 4% de comissão sobre as vendas. Faça uma aplicação Web, seguindo os seguintes requisitos:

- Faça um documento HTML que permita ao usuário digitar o salário fixo de um funcionário e o valor total apurado em vendas;
- Faça um script PHP que receba os dados submetidos pelo documento HTML anterior. O script deverá calcular e mostrar a comissão e o salário final do funcionário”;

# Exercícios

4) Faça um documento HTML que permita ao usuário digitar a base maior, a base menor e a altura de um trapézio. Em seguida, Faça um script PHP que receba os dados submetidos pelo documento HTML anterior, calcula e exiba sua área;

$$\text{área do trapézio} = \frac{(base\,maior + base\,menor)}{2} * altura$$