TEMA 04-Estrutura de controle

4.1 Introdução e conceito

Estrutura de controle em qualquer linguagem de programação é o fluxo de controle dos dados de entrada e saída, vejamos alguns exemplos práticos na sintaxe do PHP.

4.2 IF e ELSE simples e composto

Dentro da programação vamos ter muitas situações de decisão, aí entra o fluxo de controle conhecido como condicional. Caso essa decisão ou expressão seja avaliada como verdadeiro ou falso a mesma vai partir para uma sequência de comando que será logo executada. Vamos falar agora da estrutura if e depois do else, uma das estruturas mais usadas dentro de qualquer linguagem de programação.

IF

Caso a expressão seja verdadeira (true), a programação vai seguir um caminho.

IF E ELSE

Já a expressão IF com ELSE, temos agora o outro lado da questão: caso seja falso, determinamos uma sequência lógica.

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemplo de IF em PHP Interativo</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f4f4f4;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    .container {
      background-color: #fff;
      padding: 20px;
      border-radius: 8px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    }

    h2 {
      color: #333;
      margin-bottom: 20px;
      text-align: center;
    }

    label {
```

```
        display: block;
        margin-bottom: 5px;
        color: #333;
      }

      input[type="number"] {
        width: 100%;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
      }

      button {
        display: block;
        width: 100%;
        padding: 10px;
        background-color: #007bff;
        color: #fff;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        transition: background-color 0.3s;
      }

      button:hover {
        background-color: #0056b3;
      }

      .message {
        margin-top: 20px;
        padding: 10px;
        border-radius: 4px;
        text-align: center;
      }

      .success {
        color: green;
      }

      .error {
        color: red;
      }
    </style>
</head>
<body>

<?php
// Inicializa a variável de idade
$idade = '';

// Verifica se o formulário foi enviado
```

```php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtém a idade fornecida pelo usuário do formulário
    $idade = $_POST['idade'];

    // Verifica a idade e define a mensagem correspondente
    if ($idade >= 18) {
        $mensagem = "Você é maior de idade!";
        $classe = "success"; // Classe CSS para mensagem de sucesso
    } else {
        $mensagem = "Você é menor de idade!";
        $classe = "error"; // Classe CSS para mensagem de erro
    }
}
?>

<div class="container">
    <h2>Verificador de Idade</h2>
    <!-- Formulário para inserir a idade -->
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
        <label for="idade">Insira sua idade:</label>
        <input type="number" id="idade" name="idade" value="<?php echo $idade; ?>" required>
        <button type="submit">Verificar</button>
    </form>

    <!-- Exibe a mensagem na página se o formulário foi enviado -->
    <?php if ($_SERVER["REQUEST_METHOD"] == "POST"): ?>
        <p class="message <?php echo $classe; ?>"><?php echo $mensagem; ?></p>
    <?php endif; ?>
</div>

</body>
</html>
```

ELSEIF

Agora vamos tratar de uma combinação entre ELSE e IF, colocado anteriormente neste tema. Agora vamos colocar if e else um dentro do outro chamado de composto. Estrutura de fluxo uma dentro da outra.

```php
<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemplo de ElseIf em PHP Interativo</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f2f2;
        }
```

```
        .container {
            max-width: 500px;
            margin: 0 auto;
            padding: 20px;
            background-color: #fff;
            border-radius: 8px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }

        h2 {
            color: #333;
            text-align: center;
        }

        .form-group {
            margin-bottom: 20px;
        }

        label {
            font-weight: bold;
            display: block;
        }

        select {
            width: 100%;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
        }

        .message {
            margin-top: 20px;
            padding: 10px;
            border-radius: 4px;
            text-align: center;
        }

        .beginner {
            color: blue;
        }

        .intermediate {
            color: orange;
        }

        .advanced {
            color: green;
        }
    </style>
</head>
<body>
```

```
<div class="container">
  <h2>Selecione seu Nível de Habilidade</h2>
  <div class="form-group">
    <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
      <label for="nivel">Nível de Habilidade:</label>
      <select id="nivel" name="nivel">
        <option value="iniciante">Iniciante</option>
        <option value="intermediario">Intermediário</option>
        <option value="avancado">Avançado</option>
      </select>
      <button type="submit">Verificar</button>
    </form>
  </div>

  <?php
  if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $nivel_habilidade = $_POST['nivel'];

    if ($nivel_habilidade == "iniciante") {
      $mensagem = "Você é um jogador iniciante. Continue praticando!";
      $classe = "beginner";
    } elseif ($nivel_habilidade == "intermediario") {
      $mensagem = "Você está melhorando! Continue assim!";
      $classe = "intermediate";
    } elseif ($nivel_habilidade == "avancado") {
      $mensagem = "Você é um jogador avançado. Parabéns!";
      $classe = "advanced";
    } else {
      $mensagem = "Seu nível de habilidade não foi reconhecido.";
      $classe = "error";
    }
    ?>

    <div class="message <?php echo $classe; ?>">
      <?php echo $mensagem; ?>
    </div>
  <?php } ?>
</div>

</body>
</html>
```

4.3 SWITCH

O comando condicional Switch no PHP ou em qualquer linguagem de programação segue a mesma sintaxe, e com a mesma similaridade do If e else, a diferença entre ambos é que o switch usa somente a cláusula case no caso somente se for verdadeiro.

```
<!DOCTYPE html>
<html lang="pt-br">
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Exemplo de ElseIf em PHP Interativo</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f2f2f2;
    }

    .container {
        max-width: 500px;
        margin: 0 auto;
        padding: 20px;
        background-color: #fff;
        border-radius: 8px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }

    h2 {
        color: #333;
        text-align: center;
    }

    .form-group {
        margin-bottom: 20px;
    }

    label {
        font-weight: bold;
        display: block;
    }

    select {
        width: 100%;
        padding: 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
    }

    .message {
        margin-top: 20px;
        padding: 10px;
        border-radius: 4px;
        text-align: center;
    }

    .beginner {
        color: blue;
    }

    .intermediate {
```

```
        color: orange;
      }

      .advanced {
        color: green;
      }
    </style>
</head>
<body>

<div class="container">
    <h2>Selecione seu Nível de Habilidade</h2>
    <div class="form-group">
      <form method="post" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
        <label for="nivel">Nível de Habilidade:</label>
        <select id="nivel" name="nivel">
          <option value="iniciante">Iniciante</option>
          <option value="intermediario">Intermediário</option>
          <option value="avancado">Avançado</option>
        </select>
        <button type="submit">Verificar</button>
      </form>
    </div>

    <?php
    if ($_SERVER["REQUEST_METHOD"] == "POST") {
      $nivel_habilidade = $_POST['nivel'];
      switch ($nivel_habilidade) {
        case 'iniciante':
          $mensagem = "Você é um jogador iniciante. Continue praticando!";
          $classe = "beginner";
          break;
        case 'intermediario':
          $mensagem = "Você está melhorando! Continue assim!";
          $classe = "intermediate";
          break;
        case 'avancado':
          $mensagem = "Você é um jogador avançado. Parabéns!";
          $classe = "advanced";
          break;
        default:
          $mensagem = "Seu nível de habilidade não foi reconhecido.";
          $classe = "error";
          break;
      }
      ?>

      <div class="message <?php echo $classe; ?>">
        <?php echo $mensagem; ?>
      </div>
    <?php } ?>
```

```
</div>

</body>
</html>
```

4.4 Comando de repetição

Comando de repetição tem a função de determinar os números de vezes que um bloco uma sequência de instruções pode ocorrer, uma repetição colocada pelo o programador, ajuda a diminuir uma codificação dentro da sintaxe. Vamos falar nesse capítulo dos principais comandos de repetição usados no PHP, sendo while, do .. while, for e foreach.

WHILE

No laço while, a condição é verificada antes da execução do bloco de código. Se a condição for falsa desde o início, o código dentro do laço nunca será executado.

Exemplo:

```
<?php
$x = 5;

while ($x < 5) {
   echo "O valor de x é: $x <br>";
   $x++;
}
?>
```

Explicação:

O valor inicial de $x é 5.
A condição ($x < 5) é falsa já no início, então o código dentro do laço nunca será executado.
Saída: Nenhuma saída, porque o laço não é executado.

DO ... WHILE

No laço do...while, o bloco de código é sempre executado pelo menos uma vez, independentemente da condição, porque a condição é verificada após a execução.

Exemplo:

```
<?php
$x = 5;

do {
   echo "O valor de x é: $x <br>";
   $x++;
} while ($x < 5);
?>
```

Explicação:

O valor inicial de $x é 5.

O código dentro do laço é executado uma vez, mesmo que a condição ($x < 5) seja falsa.
Após executar o bloco, a condição é verificada e o laço termina.
Saída: O código imprime uma vez: O valor de x é: 5, e só depois a condição é checada.

Diferença chave:

while: Verifica a condição antes de executar o bloco. Se a condição for falsa, o código nunca será executado.

do...while: Executa o bloco pelo menos uma vez, independentemente da condição, e depois verifica a condição para continuar ou parar.

Esses exemplos mostram que o do...while garante ao menos uma execução, enquanto o while pode não executar o bloco de código se a condição for falsa logo no início.

FOR

Este comando é o loop que repete o início e fim específico determinado pelo programador, diferente do while que só sabemos o início.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Loop For</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
    }

    .number-list {
      list-style: none;
      padding: 0;
    }

    .number-item {
      margin-bottom: 5px;
      padding: 10px;
      background-color: #f0f0f0;
      border-radius: 5px;
    }
  </style>
</head>
<body>
  <h2>Números de 1 a 10:</h2>
  <ul class="number-list">
    <?php
    for ($i = 1; $i <= 10; $i++) {
      echo "<li class='number-item'>$i</li>";
```

```
    }
    ?>
  </ul>
</body>
</html>
```

FOREACH

O comando foreach é um laço de repetição usado para Array, sendo mais fácil a interação dentro de um conjunto usando matriz e vetor, também usado para interagir com uma lista de dados vindo de uma base de dados. Na prática vamos ver ele quando entrarmos no tema de banco de dados.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Loop Foreach</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
    }

    .item-list {
      list-style: none;
      padding: 0;
    }

    .item {
      margin-bottom: 5px;
      padding: 10px;
      background-color: #f0f0f0;
      border-radius: 5px;
    }
  </style>
</head>
<body>
  <h2>Lista de Cores:</h2>
  <ul class="item-list">
    <?php
    $colors = array("Red", "Green", "Blue", "Yellow", "Orange");

    foreach ($colors as $color) {
      echo "<li class='item'>$color</li>";
    }
    ?>
  </ul>
</body>
</html>
```

Neste exemplo:

Um array $colors é definido com algumas cores.
Um loop foreach é usado para iterar sobre cada elemento do array $colors.
Dentro do loop, cada cor é envolvida em um elemento <li> para criar uma lista.
CSS é usado para estilizar a lista e cada item da lista para torná-los visualmente agradáveis.

A estrutura foreach é frequentemente utilizada em conjunto com bancos de dados devido à sua capacidade de iterar facilmente sobre os resultados de uma consulta SQL. Quando você executa uma consulta SQL em um banco de dados, ela geralmente retorna múltiplas linhas de dados (resultados). O foreach é ideal para iterar sobre esses resultados e processá-los conforme necessário.

Aqui estão algumas razões pelas quais o foreach é comumente usado com bancos de dados:

Facilidade de uso: O foreach é simples e fácil de entender. Ele permite iterar sobre cada resultado de uma consulta de forma direta e sem a necessidade de acompanhar índices ou contadores.

Flexibilidade: O foreach pode ser usado com diferentes tipos de dados retornados por uma consulta SQL, como arrays associativos ou objetos.

Clareza de código: O uso do foreach torna o código mais legível e conciso em comparação com outras formas de iteração, como loops for ou while.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Leitura de Registros</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
    }

    table {
      width: 100%;
      border-collapse: collapse;
    }

    th, td {
      border: 1px solid #ddd;
      padding: 8px;
      text-align: left;
    }

    th {
      background-color: #f2f2f2;
    }
```

```
    </style>
</head>
<body>
    <h2>Registros de Setores</h2>
    <table>
        <tr>
            <th>Setor</th>
            <th>Local</th>
            <th>Lotação</th>
        </tr>
        <?php
        // Array associativo com os registros simulados
        $registros = array(
            array("Setor" => "Compras", "Local" => "Bloco A1", "Lotacao" => 4),
            array("Setor" => "RH", "Local" => "Bloco A2", "Lotacao" => 6),
            array("Setor" => "Finanças", "Local" => "Bloco B1", "Lotacao" => 8),
            array("Setor" => "Atendimento", "Local" => "Bloco B2", "Lotacao" => 14)
        );

        // Loop foreach para percorrer os registros e exibi-los na tabela
        foreach ($registros as $registro) {
            echo "<tr>";
            echo "<td>{$registro['Setor']}</td>";
            echo "<td>{$registro['Local']}</td>";
            echo "<td>{$registro['Lotacao']}</td>";
            echo "</tr>";
        }
        ?>
    </table>
</body>
</html>
```

Neste exemplo:

Definimos um array associativo chamado $registros, onde cada elemento representa um registro com informações de Setor, Local e Lotação.
Utilizamos um loop foreach para percorrer cada registro no array $registros e exibir suas informações em uma tabela HTML.
O HTML e o CSS são usados para estruturar e estilizar a tabela de forma adequada.

4.5 BREAK

O comando ou também conhecido como um elemento dentro da sintaxe do PHP, será usado ou chamada para parar uma instrução, podendo ser de loop laço de repetição ou até de uma condição.

4.6 CONTINUE

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Exemplo de Break e Continue</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 20px;
        }

        .number-list {
            list-style: none;
            padding: 0;
        }

        .number-item {
            margin-bottom: 5px;
            padding: 10px;
            background-color: #f0f0f0;
            border-radius: 5px;
        }

        .highlight {
            background-color: yellow;
        }
    </style>
</head>
<body>
    <h2>Números</h2>
    <ul class="number-list">
        <?php
        $numbers = [2, 4, 6, 8, 10, 7, 12, 3, 14, 16];

        foreach ($numbers as $number) {
            if ($number > 13) {
                echo "<li class='number-item highlight'>$number (Break)</li>";
                break;
            }

            if ($number % 2 != 0) {
                echo "<li class='number-item'>$number (Ímpar, Continue)</li>";
                continue;
            }

            echo "<li class='number-item'>$number</li>";
        }
        ?>
    </ul>
</body>
</html>
```

Neste exemplo:

Criamos um array $numbers com uma série de números.
Usamos um loop foreach para iterar sobre esses números.
Dentro do loop:
Se encontrarmos um número maior que 13, destacamos-o com a classe highlight e usamos break para parar a iteração.
Se encontrarmos um número ímpar, simplesmente o ignoramos usando continue e passamos para a próxima iteração.
Caso contrário, exibimos o número normalmente.
Usamos CSS para estilizar a lista e os itens, e também para destacar os números que encontram as condições especiais.

RESUMO

O comando ou também chamado de elemento dentro da sintaxe do PHP, vai ser usado quando paramos uma instrução no Break, sendo então para a lógica voltar a correr usamos o continue.
No capítulo sobre "Estrutura de Controle" abordamos os conceitos fundamentais relacionados ao fluxo de controle de dados em linguagens de programação, com foco na sintaxe do PHP. Discutimos as estruturas condicionais, com ênfase em IF, ELSE e ELSEIF, que permitem tomar decisões com base na avaliação de expressões como verdadeiras ou falsas. Além disso, introduzimos a estrutura de controle SWITCH, que oferece uma alternativa à condicional IF.
Exploramos os comandos de repetição, incluindo WHILE, DO...WHILE, FOR e FOREACH, destacando suas características e aplicações. O WHILE é útil quando conhecemos o início, mas não o fim de uma repetição, enquanto o DO...WHILE garante que a repetição ocorra pelo menos uma vez, mesmo que a condição seja falsa inicialmente. O FOR permite repetições com início e fim específicos determinados pelo programador, oferecendo mais controle.
Introduzimos os comandos BREAK e CONTINUE, que são elementos importantes na sintaxe do PHP. O BREAK é usado para interromper uma instrução, como um loop ou uma condição, enquanto o CONTINUE permite retomar a execução lógica após uma interrupção.Essas estruturas de controle são essenciais para direcionar o fluxo de um programa, tornando-o mais eficiente e capaz de tomar decisões com base em condições específicas. No próximo capítulo, exploraremos a aplicação dessas estruturas em cenários práticos de desenvolvimento back-end.

ATIVIDADES:

1. Descreva a função das estruturas condicionais IF e ELSE, e dê um exemplo prático de situação em que elas seriam aplicadas no desenvolvimento web.
2. Explique a diferença entre as estruturas condicionais IF e SWITCH. Em que cenários cada uma delas é mais adequada?
3. Suponha que você esteja desenvolvendo um sistema de login para um site. Crie um pseudocódigo que utilize uma estrutura condicional composta (IF ELSE) para verificar se as credenciais de um usuário são válidas ou não.
4. Em que situação você usaria a estrutura condicional ELSE IF em vez de várias estruturas IF independentes? Dê um exemplo prático
5. Descreva as principais características da estrutura de repetição FOR e forneça um exemplo de aplicação dessa estrutura em um contexto de desenvolvimento web.
6. Qual é a finalidade da estrutura de repetição DO...WHILE? Dê um exemplo simples de como ela pode ser usada em um site.
7. Crie um cenário hipotético em que o comando CONTINUE seja útil em um código PHP de uma página da web.

8. Qual é a diferença fundamental entre a estrutura condicional IF e a estrutura condicional SWITCH? Explique usando um exemplo.
9.Qual é a função do laço de repetição na programação?
10. Em um site de e-commerce, como a estrutura de repetição FOR poderia ser empregada para exibir uma lista de produtos em uma página?

RESPOSTAS
1. As estruturas condicionais IF e ELSE são utilizadas para tomar decisões com base na avaliação de expressões como verdadeiras ou falsas. O IF permite executar um bloco de código se uma condição for verdadeira, enquanto o ELSE permite executar um bloco de código alternativo caso a condição do IF seja falsa.

Exemplo prático:
Suponha que você esteja desenvolvendo um sistema de e-commerce e deseja exibir um desconto para produtos que estejam em promoção. Você pode usar uma estrutura IF para verificar se o produto está em promoção e, se estiver, exibir o preço com desconto. Caso contrário, exiba o preço normal do produto utilizando a estrutura ELSE.

2. A diferença entre as estruturas condicionais IF e SWITCH reside principalmente na forma como elas lidam com múltiplas condições. O IF é adequado para situações em que há apenas algumas condições a serem verificadas e cada uma é diferente. O SWITCH, por outro lado, é mais adequado quando há uma grande quantidade de condições a serem verificadas e cada uma leva a um resultado diferente.

O IF é mais flexível e pode lidar com condições mais complexas, enquanto o SWITCH é mais eficiente em termos de desempenho quando há muitas condições a serem avaliadas.

3. Pseudocódigo para verificação de credenciais de usuário:

```php
if (credenciais são válidas) {
   redirecionar para a página de perfil do usuário;
} else {
   exibir mensagem de erro informando que as credenciais são inválidas;
}
```

4. A estrutura condicional ELSE IF é utilizada quando há necessidade de verificar múltiplas condições encadeadas de forma mais eficiente do que usar várias estruturas IF independentes. Ela é mais adequada quando cada condição depende do resultado da condição anterior.

Exemplo prático:
Suponha que você esteja desenvolvendo um sistema de classificação de produtos em um e-commerce. Você pode usar ELSE IF para classificar os produtos em categorias diferentes com base em seu preço. Por exemplo, se o preço for inferior a $10, é uma categoria "Barato", se estiver entre $10 e $50, é uma categoria "Médio", e assim por diante.

5. A estrutura de repetição FOR é caracterizada por um loop controlado por um contador, com início, condição de parada e incremento definidos pelo programador. Ela é especialmente útil quando o número de iterações é conhecido antecipadamente.

Exemplo de aplicação em desenvolvimento web:
Suponha que você precise exibir uma lista de posts em um blog. Você pode usar um loop FOR para percorrer um array de posts e exibir cada um deles em uma página HTML.

```php
<?php
$posts = array("Post 1", "Post 2", "Post 3", "Post 4", "Post 5");

for ($i = 0; $i < count($posts); $i++) {
   echo "<div class='post'>" . $posts[$i] . "</div>";
}
?>
```

6. A estrutura de repetição DO...WHILE é usada quando se deseja que o bloco de código seja executado pelo menos uma vez, mesmo que a condição de teste seja falsa. É útil quando é necessário garantir que o bloco de código seja executado antes de verificar a condição.

Exemplo simples:
Suponha que você esteja desenvolvendo um formulário de feedback em um site. Você pode usar um loop DO...WHILE para garantir que o formulário seja exibido pelo menos uma vez, mesmo que o usuário não forneça nenhum feedback.

```php
<?php
do {
   exibir_formulario_feedback();
} while ($feedback_nao_fornecido);
?>
```

7. O comando CONTINUE é útil quando se deseja interromper a iteração atual de um loop e continuar com a próxima iteração. Um cenário em que isso seria útil é quando se deseja pular a iteração de um loop para determinados casos ou condições.

Exemplo hipotético:
Suponha que você esteja processando uma lista de pedidos em um sistema de e-commerce e deseja pular os pedidos que já foram entregues. Você pode usar CONTINUE para pular esses pedidos e continuar com o processamento dos próximos pedidos.

8. A diferença fundamental entre a estrutura condicional IF e a estrutura condicional SWITCH é a maneira como elas lidam com múltiplas condições. IF é mais flexível e pode lidar com condições mais complexas, enquanto SWITCH é mais eficiente em termos de desempenho e é mais adequado para uma grande quantidade de condições com resultados diferentes.

Exemplo:
Suponha que você esteja desenvolvendo um sistema de autenticação e deseja verificar o nível de acesso de um usuário. Com IF, você pode ter várias condições para verificar diferentes níveis de acesso (por exemplo, admin, usuário regular, convidado). Com SWITCH, você pode comparar o nível de acesso em uma estrutura mais simplificada.

9. O laço de repetição na programação é utilizado para executar um bloco de código várias vezes, com base em uma condição de teste. Ele permite automatizar tarefas repetitivas e processar grandes volumes de dados de forma eficiente.

10. No contexto de um site de e-commerce, a estrutura de repetição FOR poderia ser usada para exibir uma lista de produtos em uma página. Por exemplo, suponha que você tenha um array contendo informações sobre os produtos disponíveis. Você pode usar um loop FOR para percorrer esse array e exibir cada produto em um formato adequado na página, juntamente com seus detalhes, como nome, preço e imagem.

```php
<?php
$produtos = array(
   array("nome" => "Camisa", "preco" => 25.00, "imagem" => "camisa.jpg"),
   array("nome" => "Calça", "preco" => 50.00, "imagem" => "calca.jpg"),
   array("nome" => "Tênis", "preco" => 80.00, "imagem" => "tenis.jpg")
);

for ($i = 0; $i < count($produtos); $i++) {
   echo "<div class='produto'>";
   echo "<

img src='" . $produtos[$i]['imagem'] . "' alt='" . $produtos[$i]['nome'] . "'>";
   echo "<p>" . $produtos[$i]['nome'] . "</p>";
   echo "<p>R$" . $produtos[$i]['preco'] . "</p>";
   echo "</div>";
}
?>
```

Espero que essas explicações e exemplos tenham esclarecido as diferenças e aplicações das estruturas condicionais e de repetição na programação.

Desafio

1) Neste desafio, você deve criar uma aplicação simples em PHP que receba o nome e a nota de 3 alunos por meio de um formulário HTML e calcule o seguinte:

a) A média das notas.
b) Informe se cada aluno foi aprovado (nota maior ou igual a 7) ou reprovado (nota menor que 7).
c) Verifique quem teve a maior nota e exiba o nome do aluno com a maior pontuação.

```html
<!DOCTYPE html>
<html lang="pt-br">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Desafio: Sistema de Pontuação de Alunos</title>
</head>
<body>
   <h1>Sistema de Pontuação de Alunos</h1>
```

```
<form action="" method="post">
    <h2>Aluno 1</h2>
    Nome: <input type="text" name="aluno1_nome" required><br>
    Nota: <input type="number" name="aluno1_nota" min="0" max="10" required><br><br>

    <h2>Aluno 2</h2>
    Nome: <input type="text" name="aluno2_nome" required><br>
    Nota: <input type="number" name="aluno2_nota" min="0" max="10" required><br><br>

    <h2>Aluno 3</h2>
    Nome: <input type="text" name="aluno3_nome" required><br>
    Nota: <input type="number" name="aluno3_nota" min="0" max="10" required><br><br>

    <input type="submit" value="Calcular Resultados">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Receber os dados do formulário
    $alunos = [
        ['nome' => $_POST['aluno1_nome'], 'nota' => (float)$_POST['aluno1_nota']],
        ['nome' => $_POST['aluno2_nome'], 'nota' => (float)$_POST['aluno2_nota']],
        ['nome' => $_POST['aluno3_nome'], 'nota' => (float)$_POST['aluno3_nota']]
    ];

    $soma_notas = 0;
    $maior_nota = 0;
    $aluno_maior_nota = "";

    // Percorrer os alunos para calcular a média e verificar maior nota
    foreach ($alunos as $aluno) {
        $soma_notas += $aluno['nota'];

        // Condicional para aprovação
        if ($aluno['nota'] >= 7) {
            echo "<p>{$aluno['nome']} foi <strong>aprovado</strong> com nota
{$aluno['nota']}.</p>";
        } else {
            echo "<p>{$aluno['nome']} foi <strong>reprovado</strong> com nota
{$aluno['nota']}.</p>";
        }

        // Verificar maior nota
        if ($aluno['nota'] > $maior_nota) {
            $maior_nota = $aluno['nota'];
            $aluno_maior_nota = $aluno['nome'];
        }
    }

    // Calcular a média
    $media = _________________ / count($alunos);
```

```
        echo "<p>A média das notas é: <strong>$media</strong>.</p>";

        // Mostrar o aluno com maior nota
        echo "<p>O aluno com a maior nota é: <strong>$aluno_maior_nota</strong> com nota
$maior_nota.</p>";
    }
    ?>
</body>
</html>
```

2) Faça um programa em PHP que mostre todos os números inteiros de 100 a 200 com incremento de 2 em 2.

3) Faça um programa em PHP que apresente todos os valores ímpares no intervalo de 500 a 1000.