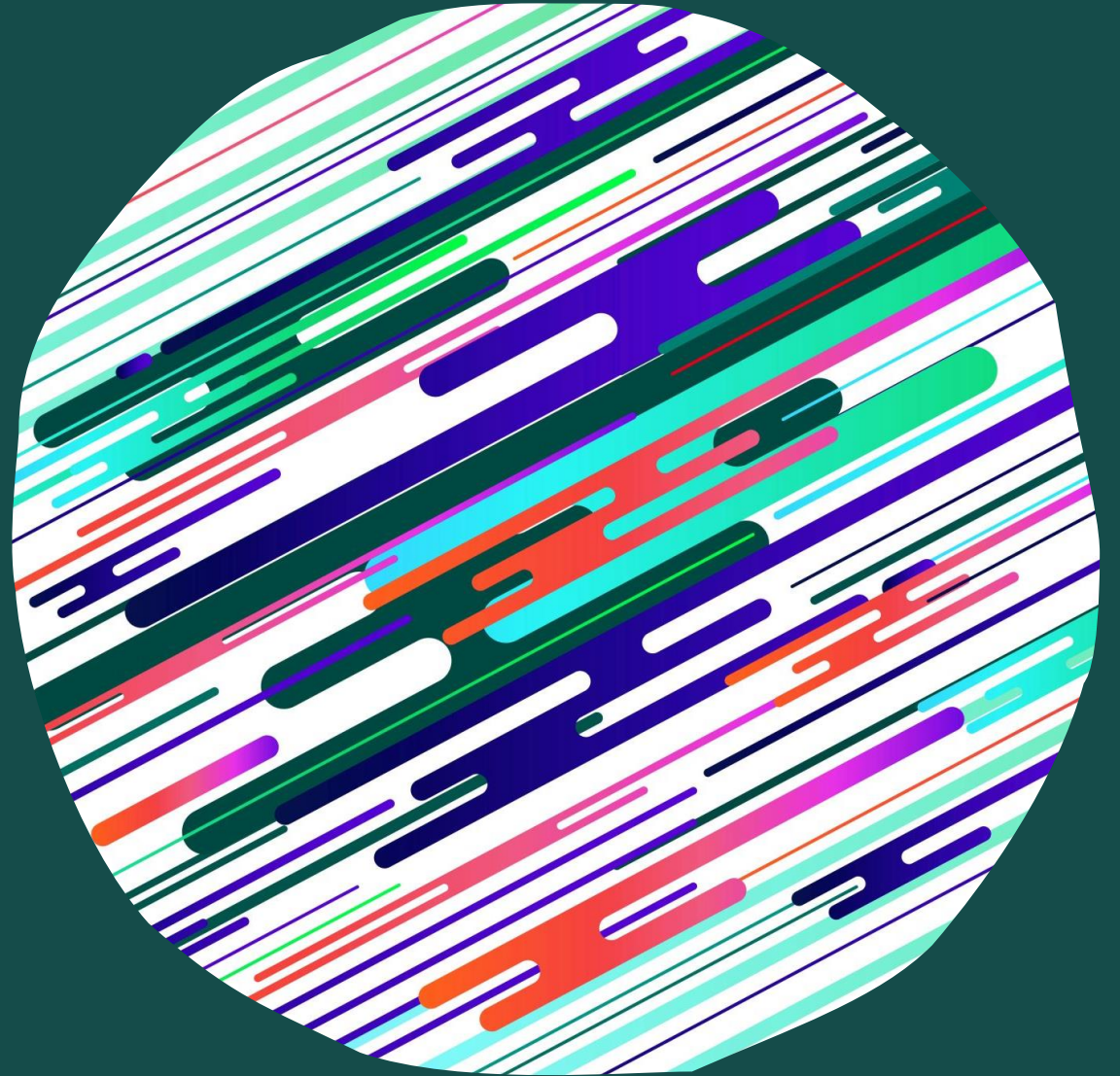


SW-I  
PHP - COMANDOS  
DE REPETIÇÃO

Prof. Anderson Vanin



# COMANDOS DE REPETIÇÃO

Utilizados para que um bloco de instruções seja executado por um número determinado de vezes, ou até que uma condição seja atingida

# COMANDOS DE REPETIÇÃO

- while
- do...while
- for
- for each

# WHILE

- O comando avalia a expressão, e enquanto essa expressão retornar o valor verdadeiro, a execução do conjunto de comandos será repetida. Caso seja falsa o bloco encerra a execução do bloco
- Tomar cuidado para não criar expressões que nunca se tornam falsas pois teríamos um loop infinito.

# WHILE - EXEMPLO

 exemplo\_while.php

```
1  <?php
2      $cont = 0;
3      while($cont<100){
4          echo "O valor atual do contador é $cont <br>";
5          $cont++;
6      }
7  ?>
```

O que acontece se ao invés de **\$cont<100** tivéssemos **\$cont!=0** ?

# DO...WHILE

- A única diferença entre o **while** e o **do...while** é que o **while** avalia a expressão no início do laço e o **do...while** ao final
- **Vai ser executado ao menos uma vez** e caso usasse o **while** não seria executado nenhuma vez

# DO...WHILE - EXEMPLO

🐞 exemplo\_do\_while.php

```
1  <?php
2      $numero = 1;
3      do{
4          echo "O valor atual do contador é $numero <br>";
5          $numero++;
6      }
7      while($numero<15);
8  ?>
```

# FOR

- Usado quando queremos executar um conjunto de instruções por quantidade especifica de vezes.
- Pode ser usado para imprimir os elementos de um **array** ou todos os resultados de uma consulta no banco de dados.



# FOR - EXEMPLO

🐘 exemplo\_for.php

```
1  <?php
2      echo "Contagem Progressiva <br> <br>";
3      for($cont=0;$cont<10;$cont++){
4          echo "A variável \$cont vale $cont <br>";
5      }
6      echo "<br> Contagem Regressiva <br> <br>";
7      for($cont=13;$cont>0;$cont--){
8          echo "A variável \$cont vale $cont <br>";
9      }
10  ?>
```

# FOR

- Também é possível fazer loops aninhados
- Útil para um **array** bidimensional

# FOR - EXEMPLO

🐞 exemplo\_for\_aninhado.php

```
1  <?php
2      $vetor[0][0] = "elemento00";
3      $vetor[0][1] = "elemento01";
4      $vetor[1][0] = "elemento10";
5      $vetor[1][1] = "elemento11";
6
7      for($contI=0;$contI<2;$contI++){
8          for($contJ=0;$contJ<2;$contJ++){
9              echo "O valor do vetor é " . $vetor[$contI][$contJ];
10             echo "<br>";
11         }
12     }
13  ?>
```

# FOREACH

- Oferece uma maneira mais fácil de “navegar” entre os elementos de um **array**

```
foreach($nome_array as $elemento)  
{  
    <comandos>  
}
```

- Todos os itens de **\$nome\_array** serão visitados. A cada iteração o item da vez será armazenado em **\$elemento**. Assim é possível trabalhar todos os elementos usando somente uma variável.

# FOREACH

- Essa segunda sintaxe funciona da mesma forma porém enquanto o elemento é adicionado **\$valor**, o índice atual é atribuído a **\$chave**

```
foreach($nome_array as $chave => $valor)  
{  
    <comandos>  
}
```

# FOREACH - EXEMPLO

 exemplo\_foreach.php

```
1  <?php
2      $vetor = array(1,2,3,4,5);
3      foreach($vetor as $v){
4          print "O valor atual do vetor é $v. <br>";
5      }
6      $a = array("um"=>1, "dois"=>2, "tres"=>3);
7      foreach($a as $chave => $valor){
8          print("\$a[$chave] => $valor.<br>");
9      }
10  ?>
```

# EXERCÍCIOS

1. Elabore um programa em PHP que permita mostrar os números entre o intervalo do número digitado pelo usuário até o cubo do mesmo.
2. Elabore um programa em PHP que permita contar o intervalo do número digitado pelo usuário até o cubo do mesmo, em seguida exibir somente os números ímpares.
3. Escreva um programa que pergunte um número ao usuário, e mostre sua tabuada (de 1 até 10).
4. Elabore uma aplicação em PHP que permita ao usuário entrar com um número, calcular o seu fatorial.