Técnico em Desenvolvimento de Sistemas

Programação WEB 2

Linguagem PHP com banco de dados MySQL

Prof. Laércio Silva

<u>laercio.silva31@etec.sp.gov.br</u>

Indsilva@Hotmail.com

Comandos de Repetição

As estruturas de repetição oferecem uma maneira simples para realizar a repetição de uma ou várias tarefas de acordo com uma condição. Também são conhecidos como estruturas de repetição, iterações ou loops, mantendo a execução de blocos de comandos, até que seu argumento seja falso.

Expressão while

A expressão **while** especifica uma condição no início do comando, onde será executado o bloco de comando, somente se esta condição for verdadeira, ou seja, só executa as instruções se a expressão for verdadeira.

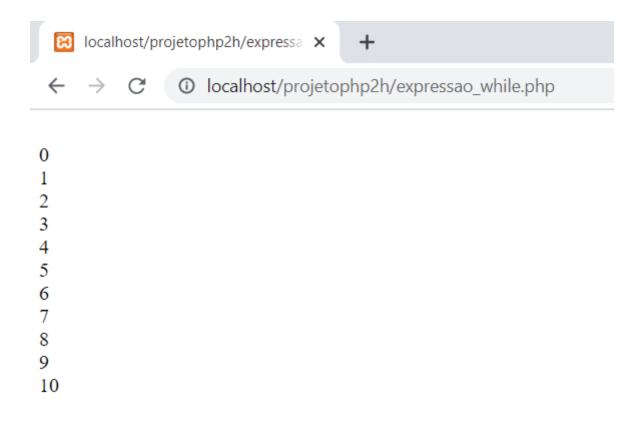
Sintaxe:

```
while (expressão){
comandos(s)
}
```

Expressão while

No exemplo a seguir, será exibido no navegador todos os números de 0 a 10 em ordem crescente. Se o valor inicial de *n* for um valor maior que 10, nenhum resultado será exibido no navegador, pois a condição será falsa.

```
<?php
    n = 0;
    while ($n <= 10) {
        echo "<br/> ".$n;
        $n++;
 ?>
```

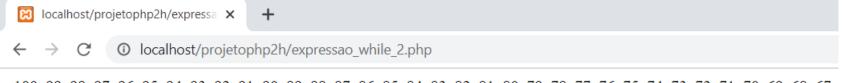


Expressão while

No exemplo a seguir, será exibido no navegador todos os números de 1 a 100 em ordem decrescente. Se o valor inicial de *n* for um valor menor que 0, nenhum resultado será exibido no navegador, pois a condição será falsa.

```
<?php
   n = 100;
   while ($n > 0) {
       echo "  ".$n;
       $n--;
 ?>
```

Resultado



100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67

Expressão do..while

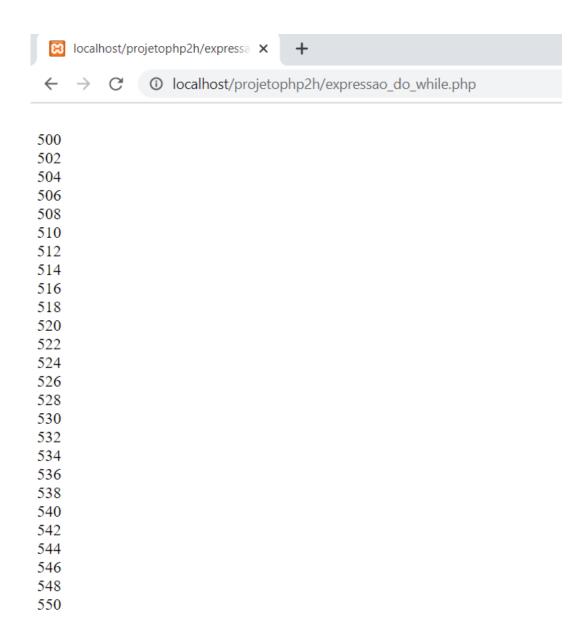
A expressão **do..while** executa pelo menos uma vez os comandos do bloco de repetição e realiza o teste da condição no final do bloco, ou seja, realiza as instruções no mínimo uma vez e enquanto a expressão for verdadeira.

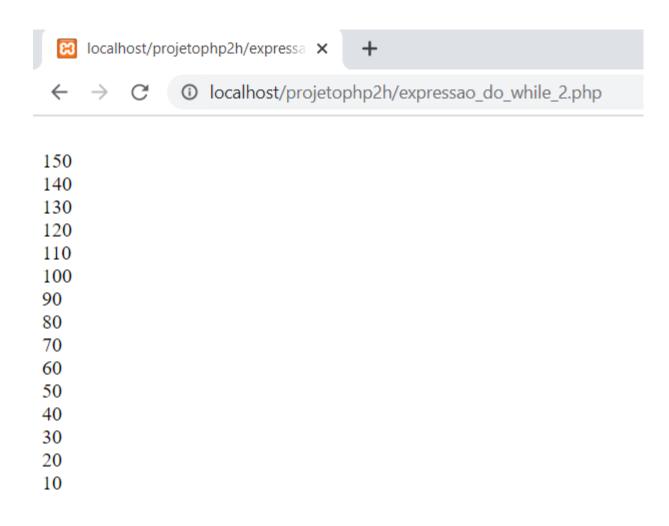
Sintaxe:

```
do {
comando(s)
} while (expressão);
```

Neste exemplo a seguir, serão exibidos os números pares de 500 até 550 em ordem crescente. Se alterarmos o valor inicial de *n* para 800, teremos como resultado de saída apenas o valor 500, pois executará pelo menos uma vez o bloco de instruções.

```
<?php
    n = 500;
    do{
         echo "<br/>".$n;
         n+=2;
    }while ($n <= 550);</pre>
 <?
```





Expressão for

A expressão **for** oferece um mecanismo de repetição mais completo, indicando a expressão inicial, a expressão condicional e expressão que determinada a operação do looping.

Uma das grandes vantagens desta estrutura é que podemos determinar em uma única linha de instruções, a inicialização, a condição de parada e o incremente ou decremento, diminuindo as chances de looping infinito.

Sintaxe:

```
for (expressão 1; expressão 2; expressão 3){
  comando(s);
}
```

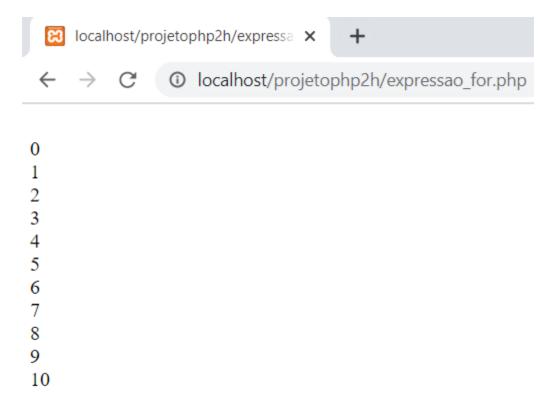
Descrevendo as expressões do comando for:

- Expressão 1: determina a inicialização da variável
- Expressão 2: determina a condição de parada do bloco de instruções
- Expressão 3: determina o incremento ou decremento da variável

Neste exemplo, será exibido no navegador todos os valores de 0 a 10 em ordem crescente.

```
<?php

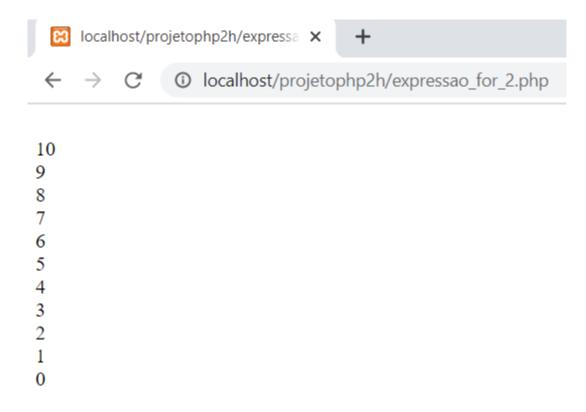
for ($x=0; $x<=10; $x++) {
    echo "<br/>}
}
```



Neste próximo exemplo, será exibido no navegador todos os valores de 0 a 10 em ordem decrescente.

```
for ($x=10; $x >=0; $x--) {
    echo "<br/>}

?>
```



Expressão foreach

É uma simplificação do comando *for* para se trabalhar em coleções de dados, ou seja, vetores e matrizes. É extremamente útil, pois com ele você executa um grupo de comandos para cada elemento de um array.

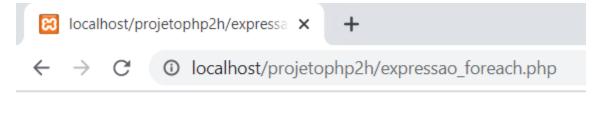
Sintaxe:

```
foreach (array_expressao as $valor){
comando(s)
}
```

Neste exemplo a seguir temos a declaração de um vetor de nomes e a estrutura **foreach** irá percorrer as posições do vetor e atribuir os valores na variável *item*, para exibir na tela os valores.

```
<?php
    $nomes = array('João','José','Paulo','Pedro');
    foreach ($nomes as $item) {
        echo "<br/>'.$item;
```

Resultado



João

José

Paulo

Pedro

Outras expressões

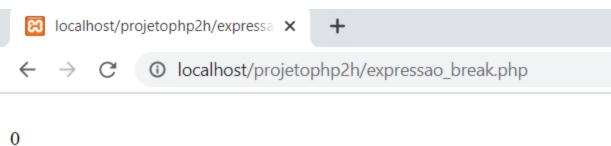
Expressão break

Permite finalizar a execução do bloco de comandos de uma estrutura. Vimos a utilização no comando switch, mas também podemos utilizar em blocos de repetição, para interromper o bloco de comandos quando uma determinada condição for satisfeita.

No exemplo a seguir, foi utilizada a expressão break, para interromper o laco de repetição quando o valor da variável for igual a 50.

```
<?php
    x=0;
    while ($x < 100) {
        echo "<br/>".$x;
        if (\$x == 10)
            break;
            $x++;
```

Resultado

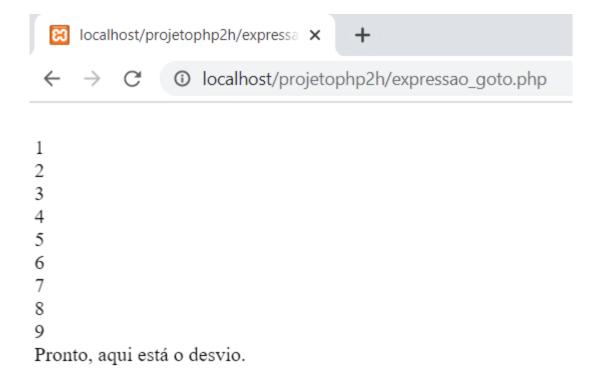


Expressão goto

Permite pular para uma localização específica fora do looping. O ponto de destino é definido por um rótulo seguido de dois pontos, e a instrução é usada como *goto* seguida do rótulo de destino desejado

Neste exemplo o fluxo será desviado, quando o valor da variável x for igual a 10.

```
<?php
    for ($x=1; $x < 100; $x++) {
        if ($x == 10)
           goto desvio;
        echo "<br/>".$x;
    desvio:
    echo "<br/>br/> Pronto, aqui está o desvio.";
 ?>
```

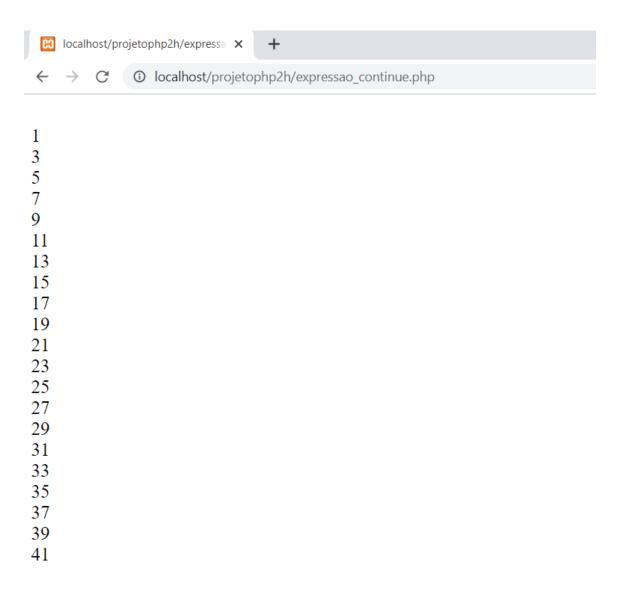


Expressão continue

Permite que a execução da repetição do loop atual termine e inicie na próxima repetição., ou seja, o looping não é encerrado, apenas encerramos a **iteração**, volta do looping, atual e iniciamos a próxima.

No exemplo a seguir, irá aparecer no navegador todos os números ímpares de 1 a 100, pois sempre que a condição testar um número e ele for par, irá pular para a próxima iteração..

```
<?php
    for ($x=1; $x <=100; $x++) {
        if ($x % 2 == 0) {
            continue;
        echo "<br/>".$x;
```



Muito Obrigado

Até a Próxima Aula

Prof. Laércio Silva Email: Indsilva@hotmail.com