



Estruturas de Repetição

Academia Indiana

Estruturas de Repetição



Estruturas de repetição são recursos que permitem executar mais de uma vez trechos de código de acordo com uma condição.

O PHP possui quatro estruturas de repetição: `for`, `foreach`, `while` e `do-while`.

for



O **for** é a estrutura de repetição do PHP que utilizamos quando sabemos a quantidade de repetições que devem ser executadas.

```
for (expressão 1; expressão 2; expressão 3)
{
    // bloco de código
}
```

Expressão 1: Executada somente uma vez, ao iniciar o loop. Normalmente a utilizamos para declarar e inicializar as variáveis que faremos uso para controlar o número de iterações do loop;

Expressão 2: Expressão booleana, validada antes de cada iteração do loop. O loop somente será executado enquanto essa expressão retornar **true**;

Expressão 3: Executada ao final de cada iteração, normalmente a utilizamos para declarar a forma de atualização do valor da variável avaliada na expressão 2.



for (Exemplo prático)

Abaixo um exemplo de uso do **for**, que exibe os números de 0 a 10:

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

Também podemos declarar o **for** utilizando a sintaxe alternativa, com “dois pontos” e **endfor**.

```
for ($i = 0; $i < 10; $i++) :  
    echo $i."<br>";  
endfor;
```



foreach

O **foreach** é uma simplificação do operador FOR para se trabalhar em coleções de dados (vetores e matrizes), permitindo acessar cada elemento individualmente iterando sobre toda a coleção e sem a necessidade de informação de índices.

Digamos que fosse preciso percorrer um vetor com alguns elementos com o FOR. Seria necessário utilizar um contador para servir também de índice para acessar cada elemento. ex.:

```
$vetor = array(1, 2, 3, 4, 5);  
for($i = 0; $i < 5; $i++)  
{  
    $item = $vetor[$i];  
    echo $item;  
}
```



foreach (exemplo prático)

O código que fizemos anteriormente poderia ser muito mais simples usando o **foreach**.

```
$vetor = array(1, 2, 3, 4, 5);  
foreach($vetor as $item)  
{  
    echo $item;  
}
```

Nesse exemplo, a cada iteração (repetição) a variável `$item` representa um elemento do vetor.



break

As estruturas de repetição possuem um comando para finalizar um loop caso seja necessário: o comando `break`. Com ele podemos interromper a execução de um loop a qualquer momento.

```
$pararLoop = 9;  
for($numero = 0; ; $numero++){  
    if($numero == $pararLoop){  
        break;  
    }  
    echo $numero."<br>";  
}
```



continue

Podemos usar o comando `continue` em uma estrutura de repetição para fazer com que os comandos subsequentes do bloco de código sejam desconsiderados e uma nova iteração iniciada.

```
for($i = 0; $i < 10; $i++){  
    if($i % 2 == 0){  
        continue;  
    }  
    echo $i."<br>";  
}
```

O código acima imprime os números ímpares entre 0 e 10. Para não imprimir os números pares, utilizamos o **continue**, que pulará para a próxima iteração do loop toda vez que o resto da divisão do valor de `$i` por 2 for igual a 0.



While

O **while** é a estrutura de repetição mais simples do PHP. Com ele informamos que um bloco de código deve ser repetido enquanto a condição declarada for verdadeira.

A sintaxe do while é a seguinte:

```
while (condição) {  
    // bloco de código  
}
```

A condição é uma expressão que é validada antes de cada iteração do loop, caso resulte em verdadeiro (true) o bloco de código será executado. Se a expressão retornar falso, o loop não é executado.



while (exemplo)

Vamos usar um exemplo parecido com o do comando for anterior, porém agora usando o while:

```
$i = 1;  
while ($i <= 10) {  
    echo $i;  
    $i++;  
}
```

O código acima vai imprimir os números de 1 a 10.



do... while

O loop **do-while** tem comportamento parecido com o **while**, diferenciando-se somente na validação do loop, que é feita no final de cada iteração.

Sintaxe do do-while:

```
do {  
    //bloco de código  
} while (condição);
```

Devido a essa característica, normalmente utilizamos essa estrutura de repetição quando precisamos que o bloco de código declarado no loop seja executado **pelo menos uma vez**.



do... while (exemplo)

Suponha que você precisa apresentar ao usuário os valores inteiros entre dois números. Para fazer isso usando a estrutura de repetição **while**

```
$minimo = 10;  
$maximo = 30;  
$numero = $minimo + 1;  
while ( $numero < $maximo) {  
    echo $numero;  
    $numero++;  
}
```

O código acima irá imprimir os números entre 10 e 30.

Exercícios



1- Termine o script abaixo, imprimindo todos os números entre \$numero1 e \$numero2.

```
<php
```

```
$numero1 = 7;
```

```
$numero2 = 27
```

```
...
```

Exercícios



2- Crie um script em PHP que crie um vetor de 5 elementos inteiros com valores aleatórios, entre 0 e 10. Encontre e mostre o menor elemento e sua posição no vetor.

Exercícios



3- Crie um script em PHP crie dois vetores de 10 posições preenchidos com valores aleatórios. Faça a soma dos elementos de mesmo índice, colocando o resultado em um terceiro vetor. Imprima os três vetores, um após o outro. Obs: Utilize um for

...

Exercícios



4- Seu salário atual é de R\$5.000,00. O do seu colega é de R\$2.500,00. Suponha que você tenha 10% de aumento todo ano e seu colega 20%. Crie um script que calcula em quantos anos seu colega irá passar a ganhar mais que você.

```
<?php
```

```
$sal1 = 5000;
```

```
$sal2 = 2500;
```

```
...
```