

# **Programação para WEB**

## **PHP Básico**

**Prof. Bruno Camargo Ribeiro**

### **Estruturas de controle**

Assuntos que serão abordados:

1	Estrutura de controle .....	2
1.1	Expressão if .....	2
1.2	Expressão else .....	2
1.3	Expressão elseif .....	3
1.4	xpressão switch .....	3
2	Expressão de Looping .....	4
2.1	Expressão while .....	4
2.2	Expressão do..while .....	5
2.3	Expressão for .....	6
2.4	Expressão foreach .....	7
2.5	Expressão break e goto .....	7
2.6	Expressão continue .....	8

## 1 Estrutura de controle

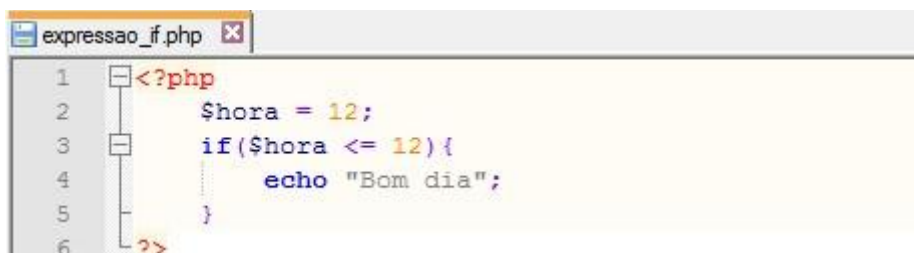
Estruturas de controle determinam o fluxo de códigos dentro de uma aplicação.

### 1.1 Expressão if

É uma das construções mais comuns de qualquer linguagem de programação, oferecendo um meio conveniente para execução de código condicional.

```
if(expressão){  
    comando(s)  
}
```

Exemplo:



```
1 <?php  
2     $hora = 12;  
3     if($hora <= 12){  
4         echo "Bom dia";  
5     }  
6 ?>
```

### 1.2 Expressão else

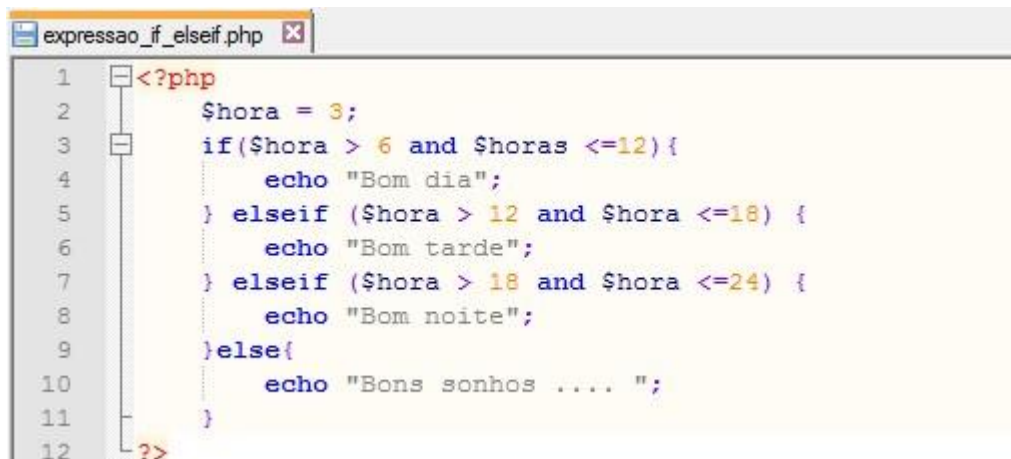
Permite indicar alternativa, caso a primeira situação não seja atendida.



```
1 <?php  
2     $hora = 15;  
3     if($hora <= 12){  
4         echo "Bom dia";  
5     } else {  
6         echo "Bom tarde";  
7     }  
8 ?>
```

### 1.3 Expressão elseif

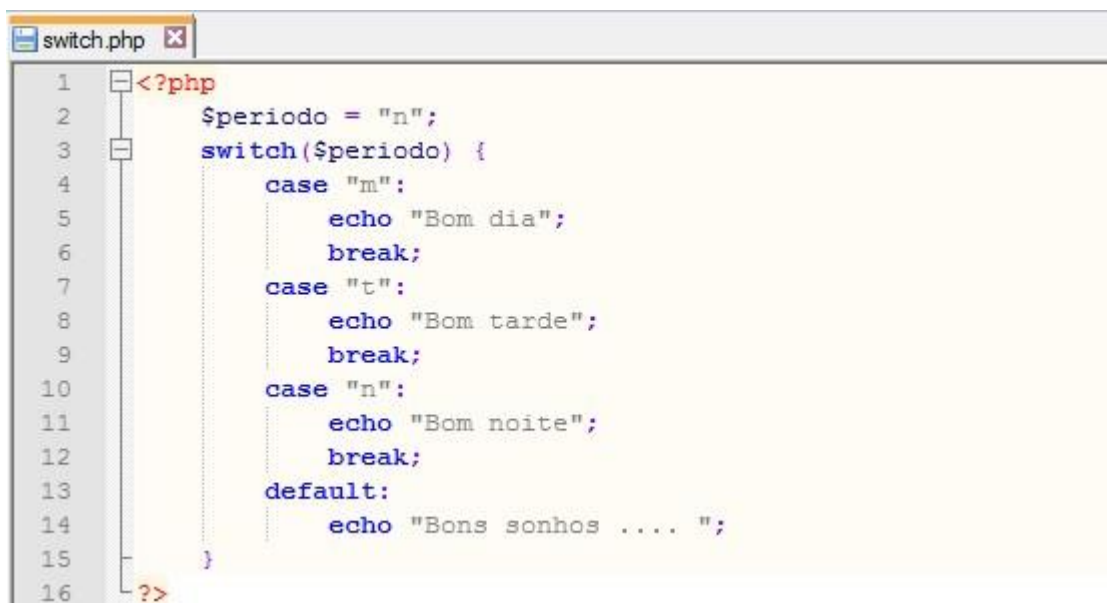
A expressão **if-else** funciona muito bem quando a situação é "um-ou-outro", ou seja, somente dois resultados são permitidos, a expressão **elseif** permite identificar novas possibilidade na sequência de testes.



```
1 <?php
2 $hora = 3;
3 if($hora > 6 and $horas <=12){
4     echo "Bom dia";
5 } elseif ($hora > 12 and $hora <=18) {
6     echo "Bom tarde";
7 } elseif ($hora > 18 and $hora <=24) {
8     echo "Bom noite";
9 } else{
10     echo "Bons sonhos .... ";
11 }
12 ?>
```

### 1.4 Expressão switch

A expressão switch pode ser utilizada como uma variante da combinação **if-elseif-else**, usando quando existe um grande número de valores para serem testados.



```
1 <?php
2 $periodo = "n";
3 switch($periodo) {
4     case "m":
5         echo "Bom dia";
6         break;
7     case "t":
8         echo "Bom tarde";
9         break;
10    case "n":
11        echo "Bom noite";
12        break;
13    default:
14        echo "Bons sonhos .... ";
15 }
16 ?>
```

## 2 Expressão de Looping

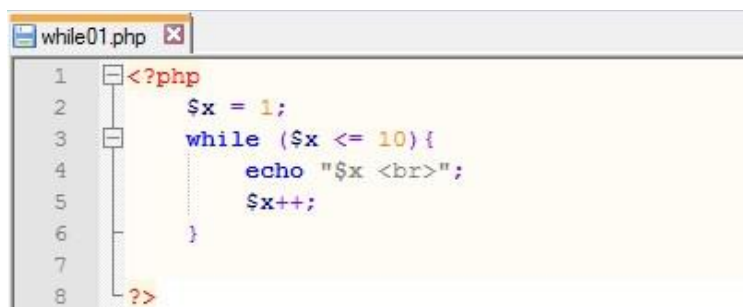
As estruturas de looping oferecem uma maneira simples para realizar a repetição de uma ou várias tarefas de acordo com uma condição.

### 2.1 Expressão while

A expressão **while** especifica uma condição que precisa ser preenchida antes de sua execução.

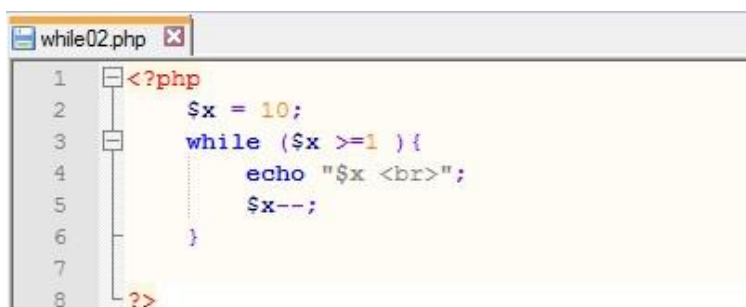
```
while (expressão){  
    comandos(s)  
}
```

Exemplos:



```
1 <?php  
2     $x = 1;  
3     while ($x <= 10){  
4         echo "$x <br>";  
5         $x++;  
6     }  
7  
8 ?>
```

Contador (crescente)



```
1 <?php  
2     $x = 10;  
3     while ($x >= 1){  
4         echo "$x <br>";  
5         $x--;  
6     }  
7  
8 ?>
```

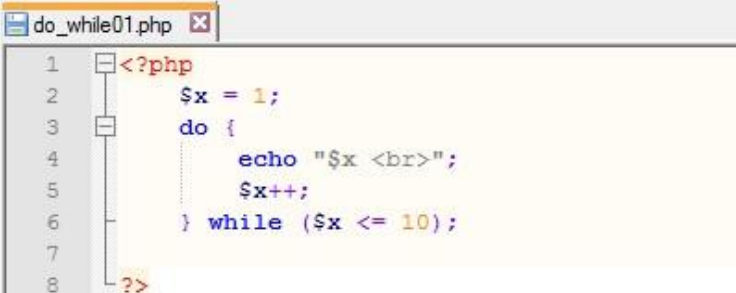
Contador (decrecente)

## 2.2 Expressão do..while

A expressão **do..while** executa no mínimo uma vez a tarefa e depois realiza a verificação da condição.

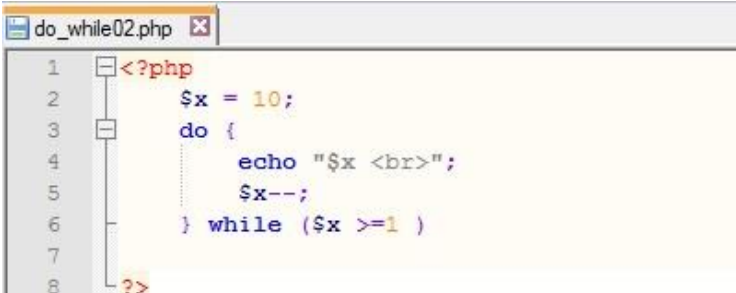
```
do {  
comando(s)  
} while (expressão);
```

Exemplos:



```
do_while01.php  
1 <?php  
2 $x = 1;  
3 do {  
4     echo "$x <br>";  
5     $x++;  
6 } while ($x <= 10);  
7  
8 ?>
```

Contador (crescente)



```
do_while02.php  
1 <?php  
2 $x = 10;  
3 do {  
4     echo "$x <br>";  
5     $x--;  
6 } while ($x >= 1 )  
7  
8 ?>
```

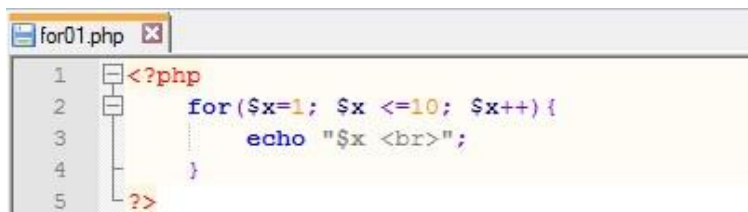
Contador (decrescente)

### 2.3 Expressão for

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

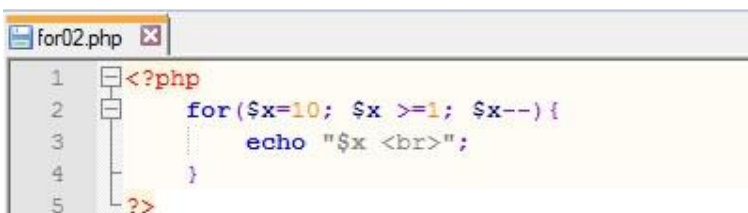
```
for (expressão 1; expressão 2; expressão 3){  
    comando(s);  
}
```

Exemplos:



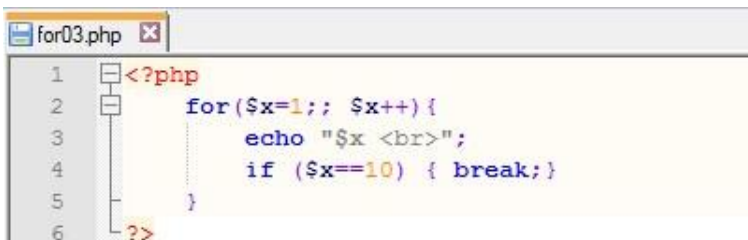
```
1 <?php  
2 for($x=1; $x <=10; $x++){  
3     echo "$x <br>";  
4 }  
5 ?>
```

Contador (crescente)



```
1 <?php  
2 for($x=10; $x >=1; $x--){  
3     echo "$x <br>";  
4 }  
5 ?>
```

Contador (decrecente)



```
1 <?php  
2 for($x=1;; $x++){  
3     echo "$x <br>";  
4     if ($x==10) { break;}  
5 }  
6 ?>
```

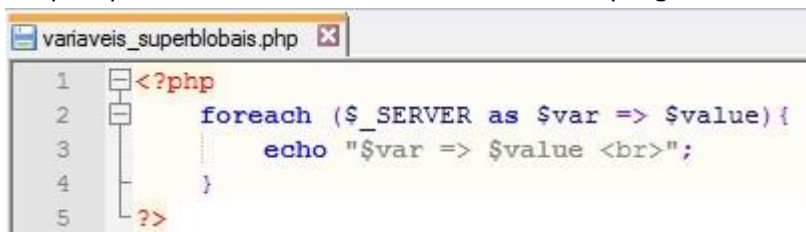
Realizando o teste da expressão condicional dentro do fluxo do looping.

## 2.4 Expressão foreach

A construção do loop **foreach** é especialista em **array**, retirando cada valor do **array** até que todos os itens terminem, ou até que uma condição encerre a sua execução.

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

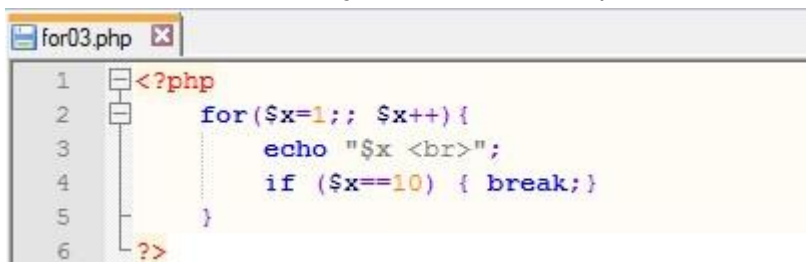
Usamos um exemplo quando listamos todas as variáveis superglobais do PHP.



```
1 <?php  
2 foreach ($_SERVER as $var => $value){  
3     echo "$var => $value <br>";  
4 }  
5 ?>
```

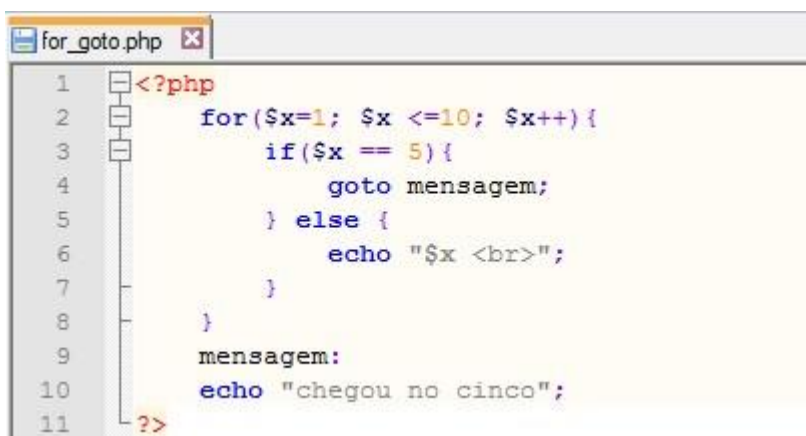
## 2.5 Expressão break e goto

A expressão **break** irá finalizar a execução do bloco do loop.



```
1 <?php  
2 for($x=1;; $x++){  
3     echo "$x <br>";  
4     if ($x==10) { break;}  
5 }  
6 ?>
```

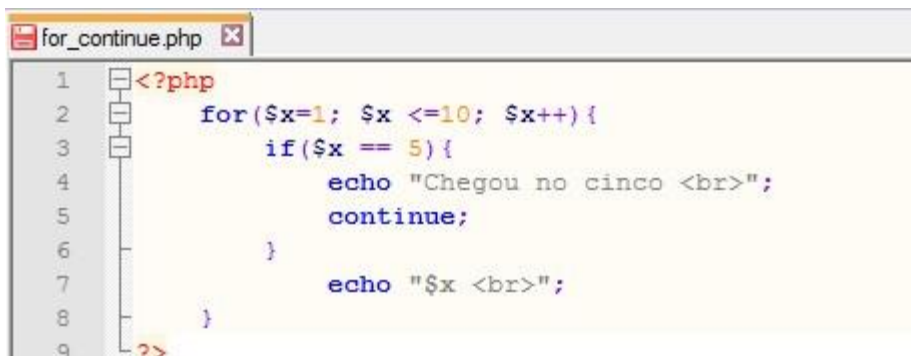
A expressão **goto**, estendida para o PHP 6, permite pular para uma localização específica fora do looping.



```
1 <?php  
2 for($x=1; $x <=10; $x++){  
3     if($x == 5){  
4         goto mensagem;  
5     } else {  
6         echo "$x <br>";  
7     }  
8 }  
9 mensagem:  
10 echo "chegou no cinco";  
11 ?>
```

## 2.6 Expressão continue

A expressão **continue** faz com que a execução da repetição do loop atual termine e inicie na próxima repetição.



```
1 <?php
2 for($x=1; $x <=10; $x++){
3     if($x == 5){
4         echo "Chegou no cinco <br>";
5         continue;
6     }
7     echo "$x <br>";
8 }
9 ?>
```