

Programa de Formação Web Developers PHP

Agosto/2021



CONSTRUSITE BRASIL
O SEU SUCESSO ONLINE

GALAXPAY 
SEU NEGÓCIO RECORRENTE



Dia 03 – 03/09/2021

01

IF, Else

Condicionais

02

while, for, foreach

Funções de loop

03

Arrays

Vetores, armazenamento de grandes valores.

04

Exercícios

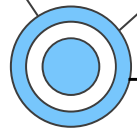
Exercícios bacanas para fixar conteúdo.



GALAXPAY\$
SEU NEGÓCIO RECORRENTE

IF , ELSE





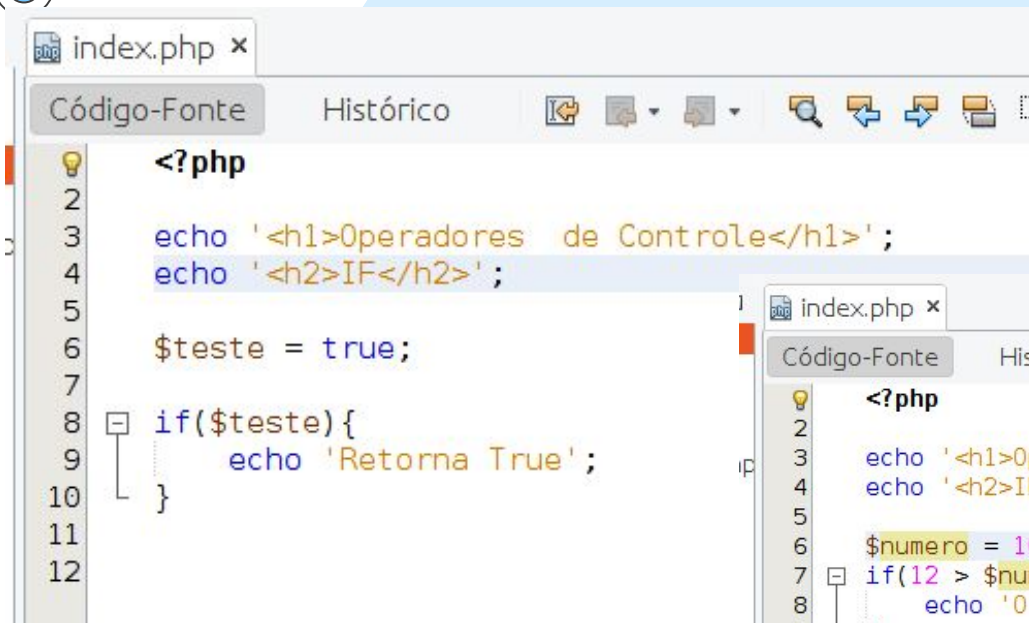
Construtor IF é um dos recursos mais importantes das linguagens.

- Permite a execução condicional de fragmentos de código
- O PHP entrará no if se o resultado for um Booleano **true** caso retorne um false, a expressão será ignorada.

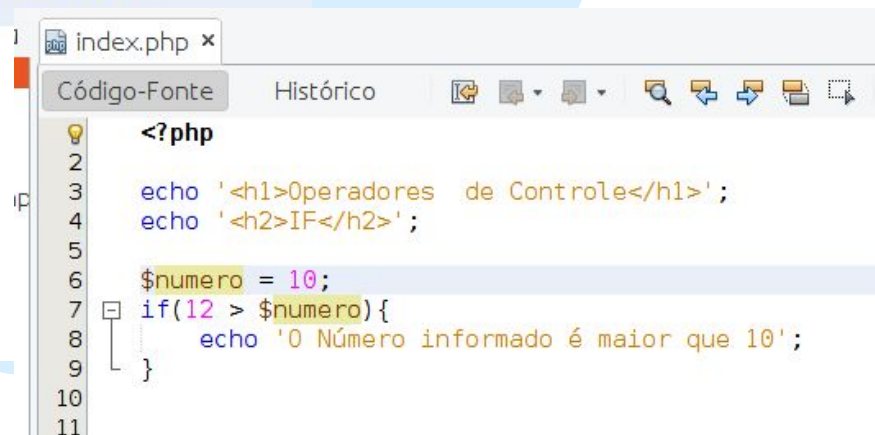


- IF, ELSE :

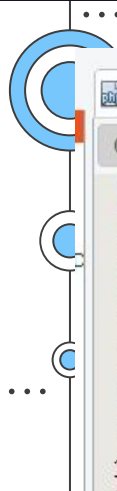
- Executar uma determinada ação caso o valor seja verdadeiro.
- Caso não seja, entra no else e finaliza a aplicação.



```
<?php
2
3 echo '<h1>Operadores de Controle</h1>';
4 echo '<h2>IF</h2>';
5
6 $teste = true;
7
8 if($teste){
9     echo 'Retorna True';
10 }
11
12
```



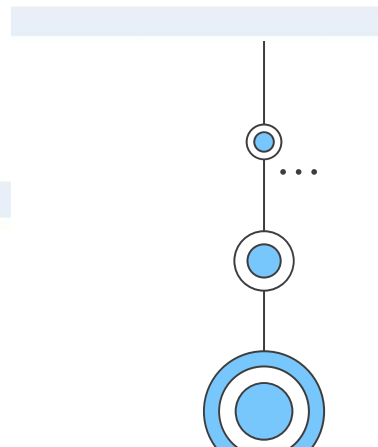
```
index.php x
Código-Fonte Histórico
<?php
2
3 echo '<h1>Operadores de Controle</h1>';
4 echo '<h2>IF</h2>';
5
6 $numero = 10;
7 if(12 > $numero){
8     echo 'O Número informado é maior que 10';
9 }
10
11
```



```
index.php x
Código-Fonte  Histórico
1  <?php
2  |
3  echo '<h1>Operadores de Controle</h1>';
4  echo '<h2>IF</h2>';
5
6  $teste = false;
7
8  if ($teste) {
9      echo 'Retorna True';
10 } else {
11     echo 'Retorna False';
12 }
13
```

```
index.php x
Código-Fonte  Histórico
1  <?php
2
3  echo '<h1>Operadores de Controle</h1>';
4  echo '<h2>IF</h2>';
5
6  $numero = 10;
7  if (9 > $numero) {
8      echo 'O Número informado é maior que 10';
9  } else {
10     echo 'O Número informado NÃO é maior que 10';
11 }
12
```

```
5
6
7  $number = 10;
8
9  if ($number % 2 == 0) {
10     echo "O número informado é PAR";
11 } else {
12     echo "O número informado é IMPAR";
13 }
14
15
16
```





- Else If()
 - Testar condições em cascata.

The image shows three overlapping code editor windows, each displaying a different version of a PHP script. The windows are titled 'index.php x'.

Top-left window: Shows a basic script with two echo statements and two variables set to 22.

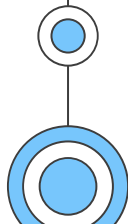
```
<?php
2
3 echo '<h1>Operadores de Controle</h1>';
4 echo '<h2>IF</h2>';
5
6 $idadeVacinaçao = 22;
7 $minhaIdade = 22;
```

Top-right window: Shows the same script as the top-left window.

```
<?php
2
3 echo '<h1>Operadores de Controle</h1>';
4 echo '<h2>IF</h2>';
```

Bottom window: Shows a more complex script with an if-else-if structure. The first condition is highlighted in red.

```
<?php
2
3 echo '<h1>Operadores de Controle</h1>';
4 echo '<h2>IF</h2>';
5
6 $idadeVacinaçao = 25;
7 $minhaIdade = 22;
8
9 if ($idadeVacinaçao < $minhaIdade) {
10     echo 'Vish, já passamos sua idade de vacinação!';
11 } else if ($idadeVacinaçao == $minhaIdade) {
12     echo 'Eba! Estamos na sua idade, corra para o posto!';
13 } else {
14     echo 'Calma! Em breve chegamos na sua idade de vacinação.';
15 }
```



- Exemplo de cenário.
 - Validação simples e fictícia de login.

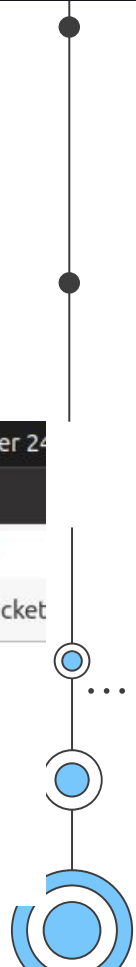


The screenshot displays a web browser window with the address bar showing `192.168.2.101/Formação`. The page title is `Google Chrome`. The browser's address bar shows a warning icon and the text `Não seguro | http://192.168.2.101/Formação%20Web%20Developers`. Below the address bar, there are navigation buttons: `Apps`, `Intranet Const...`, `Informações d...`, `Logins autom...`, and `Pr...`. The browser's tab bar shows two tabs: `index.php x` and `Teste.php x`. The browser's toolbar includes buttons for `lgo-Fonte` and `Histórico`. The browser's status bar shows `Atividades` and `Google Chrome`. The browser's page content shows a PHP code snippet: `<?php`, `include 'c`, `echo '<h1>`, `//App\Test`, `if (App\Te`, `var_du`, `die;`, `var_dump('`, `string(27) "Usuário não está logado!"`, `{`, `al";`. The browser's page content also shows a `loggedIn()` function call.

```
<?php
2
3 include 'c
4
5 echo '<h1>
6 //App\Test
7
8 if (App\Te
9     var_du
10     die;
11 }
12
13 var_dump('
14 string(27) "Usuário não está logado!"
15
16 {
17     al";
```

`loggedIn()`



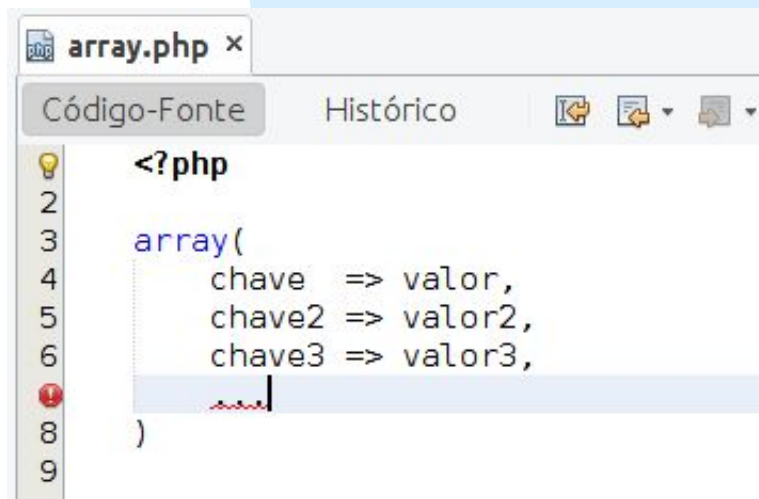


```
string(47) "Usuário está logado! Seu Usuário é: Gabriel"
```

Arrays



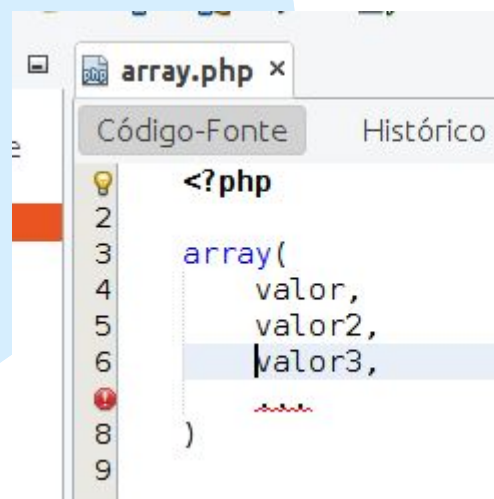
- **Array**
- Mapa ordenado.
 - Um mapa é um tipo que relaciona *chaves* e *valores*
- Pode ser tratado como um vetor, uma lista de itens, pilha e vários outros



The screenshot shows a code editor window titled 'array.php'. The 'Código-Fonte' tab is active. The code is as follows:

```
<?php
2
3 array(
4     chave => valor,
5     chave2 => valor2,
6     chave3 => valor3,
7
8 )
9
```

A light blue semi-transparent shape is overlaid on the right side of the code editor.



The screenshot shows a code editor window titled 'array.php'. The 'Código-Fonte' tab is active. The code is as follows:

```
<?php
2
3 array(
4     valor,
5     valor2,
6     valor3,
7
8 )
9
```

```
array.php x
Código-Fonte Histórico
<?php
$aulas = array(
    'Apresentação Equipe',
    'Frontend x BackEnd x FullStack',
    'Inspeccionar do Chrome',
    'Introdução ao PHP',
    'Sintaxe Básica',
    'Variáveis em PHP',
    'PSRs',
    'Operadores matemáticos e concatenação de strings',
    'While, for, foreach',
    'arrays',
);
echo '<pre>';var_dump($aulas);
```

Atividades Google Chrome

192.168.2. x Dia 03 - Go x Aula 03 - If x PHP: Arra x

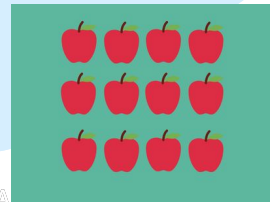
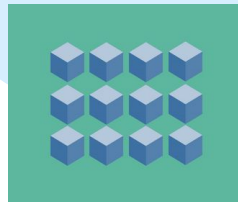
Não seguro | http://192.168.2.101/Formação%20Web%20Develc

Apps Intranet Const... Informações d... Logins autom...

```
array(10) {
    [0]=>
    string(21) "Apresentação Equipe"
    [1]=>
    string(30) "Frontend x BackEnd x FullStack"
    [2]=>
    string(21) "Inspeccionar do Chrome"
    [3]=>
    string(19) "Introdução ao PHP"
    [4]=>
    string(15) "Sintaxe Básica"
    [5]=>
    string(17) "Variáveis em PHP"
    [6]=>
    string(4) "PSRs"
    [7]=>
    string(51) "Operadores matemáticos e concatenação de strings"
    [8]=>
    string(19) "While, for, foreach"
    [9]=>
    string(6) "arrays"
}
```

- **Array**

- Tipo de dado, assim como integer, float, string ou boolean, porém ele armazena mais um valor relacionado com suas chaves.
- Muito poderosos e oferece muitas possibilidades.
- Usado para consulta de dados no banco, retorno de requisições ajax , loop de blocos, popular telas.
- Também devemos citar os arrays multidimensionais, conhecidos também como Multi Arrays, que são arrays compostos por outros array, academicamente, conceito de **Matriz**.



```
array.php x
Código-Fonte Histórico
<?php
$aulas = array(
    'Apresentação Equipe',
    'Frontend x BackEnd x FullStack',
    'Inspeccionar do Chrome',
    'Introdução ao PHP',
    'Sintaxe Básica',
    'Variáveis em PHP',
    'PSRs',
    'Operadores matemáticos e concatenação de strings',
    'While, for, foreach',
    'arrays',
);
echo '<pre>';var_dump($aulas);
```

Atividades Google Chrome

192.168.2. x Dia 03 - Go x Aula 03 - If x PHP: Arra x

Não seguro | http://192.168.2.101/Formação%20Web%20Develc

Apps Intranet Const... Informações d... Logins autom...

```
array(10) {
    [0]=>
    string(21) "Apresentação Equipe"
    [1]=>
    string(30) "Frontend x BackEnd x FullStack"
    [2]=>
    string(21) "Inspeccionar do Chrome"
    [3]=>
    string(19) "Introdução ao PHP"
    [4]=>
    string(15) "Sintaxe Básica"
    [5]=>
    string(17) "Variáveis em PHP"
    [6]=>
    string(4) "PSRs"
    [7]=>
    string(51) "Operadores matemáticos e concatenação de strings"
    [8]=>
    string(19) "While, for, foreach"
    [9]=>
    string(6) "arrays"
}
```

Atividades

- 1) Faça um script, que receba um valor, e retorne se o valor informado é par ou ímpar.
- 2) Dado o seguinte array:
`$nomes = array("Gabriel", "Izabella", "Pedro", "Davi", "Arthur", "Sobral", "Diego", "João", "Santos", "Henrique", "Roberto", "Júlio", "Márcio");`
- Verifique:
 - Se um determinado nome informado, está presente nesta lista ou não.
 - Caso positivo, print uma frase com o nome localizado.
 - Caso não esteja, adicione este nome no array, e retorne o array.
 - Conte a quantidade de itens do array antes, e depois da validação.
 - Informe se a quantidade foi alterada, caso sim, mostre o valor adicionado.



Algumas ~ Funções...





- Funções

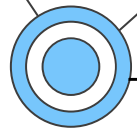
- Temos inúmeras funções , podemos destacar:
- **in_array** - Verifica se existe uma string específica dentro do array.
- **count** - Conta a qtd de elementos dentro de um array.
- **array_merge** - Combina um ou mais arrays
- **end** - Pega o último elemento do array => end(\$array);
- **reset** - Aponta o primeiro elemento do array.
- **sort** - Ordena o array, crescente.
- **rsort** - Ordena o array de forma decrescente.
- **array_search** - Procura um valor dentro do array, e retorna sua chave caso encontrado.
- **array_push** - Adiciona mais um elemento no array.
- **isset** - Verifica se existe aquela posição no array.
- Dentre várias: https://www.php.net/manual/pt_BR/book.array.php



**WHILE, FOR,
FOREACH**



GALAXYPAY 
SEU NEGÓCIO RECOMENDADO



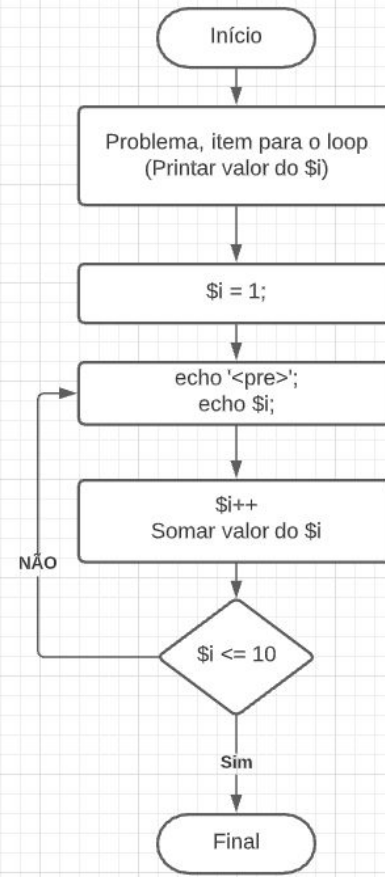
Expressão de LOOP mais simples da linguagem.

- Instrução de código fica em repetição até alcançar a condição verdadeira.

```
1 while.php x
2
3 <?php
4 while (condição) {
5     // bloco de execução de código
6 }
```



- Situação:
 - Printar um número 10 vezes, usando repetição;



- While

```
while.php x
Código-Fonte Histórico
1 <?php
2
3 $i = 1;
4 while ($i <= 10) {
5     echo '<pre>';
6     echo $i;
7     $i++;
8 }
9
10
```

Atividades Google Chrome

192.168.2.101/Formação x +

Não seguro | http://192.168

Apps Intranet Const... Informa

```
1
2
3
4
5
6
7
8
9
10
```

- **do-while**

- Os laços do-while é muito similar aos laços while, com exceção que a expressão de avaliação é verificada ao final de cada iteração.
- A maior diferença para o laço while é que a primeira iteração do laço do-while sempre é executada (a validação é executada somente no final da iteração).
- Considerando que no laço while não é necessariamente executada (a expressão de avaliação é executada no começo de cada iteração, se avaliada como false logo no começo, a execução do laço será abortada imediatamente).
- O que isso representa na prática?



- do-while

```
while.php x
Código-Fonte  Histórico
<?php
2
3  $i = 1;
4  do {
5      echo '<pre>';
6      echo $i;
7      $i++;
8  } while ($i < 10);
9
10 echo '<hr>';
11
12 /* ----- */
13
14 $j = 1;
15 while ($j <= 10) {
16     echo '<pre>';
17     echo $j;
18     $j++;
19 }
20
```

1

2

3

4

5

6

7

8

9

1

2

3

4

5

6

7

8

9

10

while_vs_dowhile.php x

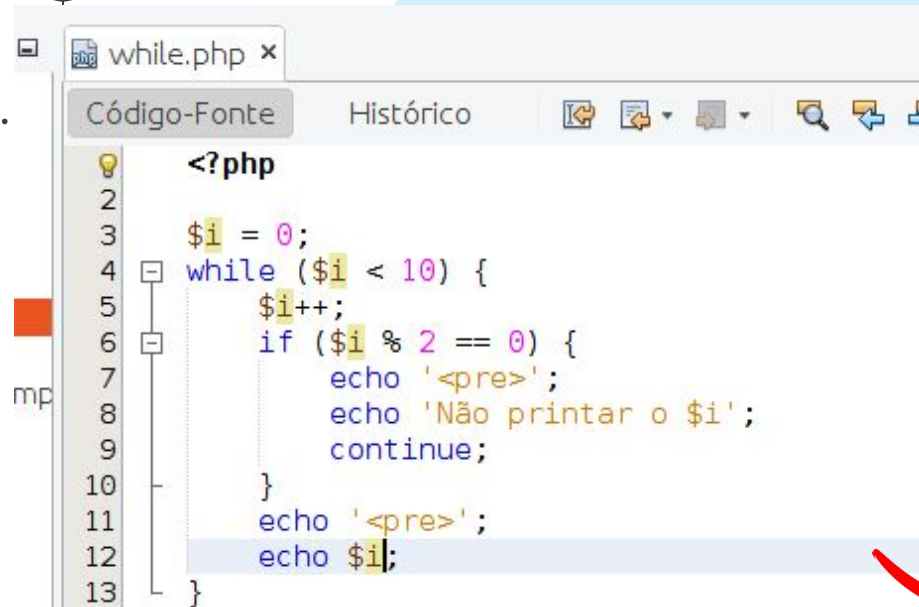
webdevs > aula03 > while_vs_dowhile.php > ...

```
1  <?php
2
3  echo '<pre>';
4
5  $i = 0;
6  do {
7      var_dump($i);
8  } while ($i > 1);
9
10 echo '<hr>';
11
12 $i = 0;
13 while ($i > 1) {
14     var_dump($i);
15 }
16
17 echo '<hr>';
18 echo 'terminou';
19
```

/var/www/html/webdevs/aula03/while_vs_dowhile.php:7:int 0

terminou

- **Continue**
- Continue ele pula exibição quando atingido



```
<?php
2
3  $i = 0;
4  while ($i < 10) {
5      $i++;
6      if ($i % 2 == 0) {
7          echo '<pre>';
8          echo 'Não printar o $i';
9          continue;
10     }
11     echo '<pre>';
12     echo $i;
13 }
```



Atividades Google Chrome

192.168.2.101/Formação \ x +

Não seguro | http://192.168.2.

Apps Intranet Const... Informaçõ

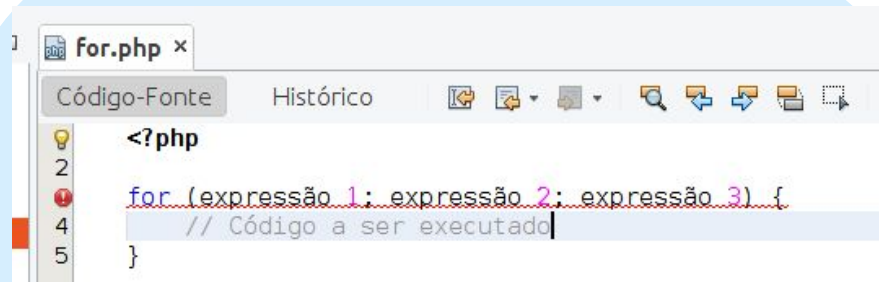
```
1
Não printar o $i
3
Não printar o $i
5
Não printar o $i
7
Não printar o $i
9
Não printar o $i
```

FOR



GALAXYPAY 
SEU NEGÓCIO RECOMENDADO

- Utilizamos o FOR quando sabemos a quantidade exata de repetições que devem ser executadas.

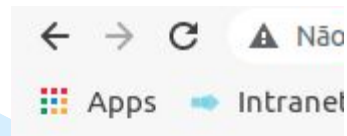


The screenshot shows a code editor window titled 'for.php'. It has tabs for 'Código-Fonte' (Source Code) and 'Histórico' (History). The source code is as follows:

```
1 <?php
2
3 for (expressão 1: expressão 2: expressão 3) {
4     // Código a ser executado
5 }
```

- Expressão 1 = Usamos para declarar o valor inicial da nossa variável que irá controlar o loop.
- Expressão 2 = Expressão booleana, validada antes da iteração do loop se for verdadeira o loop será executado.
- Expressão 3 = Executada ao final de cada iteração, usamos para sobrescrever o valor da variável.

- Exemplo:



0

1

2

3

4

5

6

7

8

9

for.php x

Código-Fonte

Histórico

<?php

```
2
3 for ($i = 0; $i < 10; $i++) {
4     echo '<pre>';
5     echo $i;
6 }
```

Emp



FOREACH



GALAXPAY 
SEU NEGÓCIO RECOMENDADO

- Simplificação do For.
- Voltado para o trabalho com coleção de dados, vetores, matrizes.
- Permite o acesso a cada elemento individual percorrido sem a necessidade de informação de um índice.
- Exemplo:

FOR

FOREACH

```
foreach.php x
Código-Fonte  Histórico
1  <?php
2
3  $vetorComFor = array(1, 2, 3, 4, 5);
4  for ($i = 0; $i < 5; $i++) {
5      $item = $vetorComFor[$i];
6      echo '<pre>';
7      echo $item;
8  }
9
```

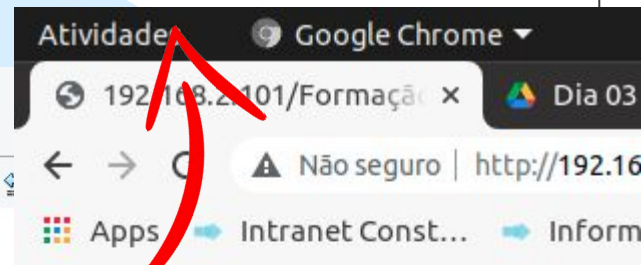
```
foreach.php x
Código-Fonte  Histórico
1  <?php
2
3  $vetorComForeach = array(1, 2, 3, 4, 5);
4  foreach ($vetorComForeach as $item) {
5      echo '<pre>';
6      echo $item;
7  }
```



- **\$key**
- Loop pegando a chave do array.

```
1 <?php
2
3 $array = array("Aprender", "Php", "é", "muito", "massa");
4 foreach ($array as $key => $value) {
5     echo '<pre>';|
6     var_dump($key, $value);
7 }
```

\$key (Chave)



int(0)
string(8) "Aprender"

int(1)
string(3) "Php"

int(2)
string(2) "é"

int(3)
string(5) "muito"

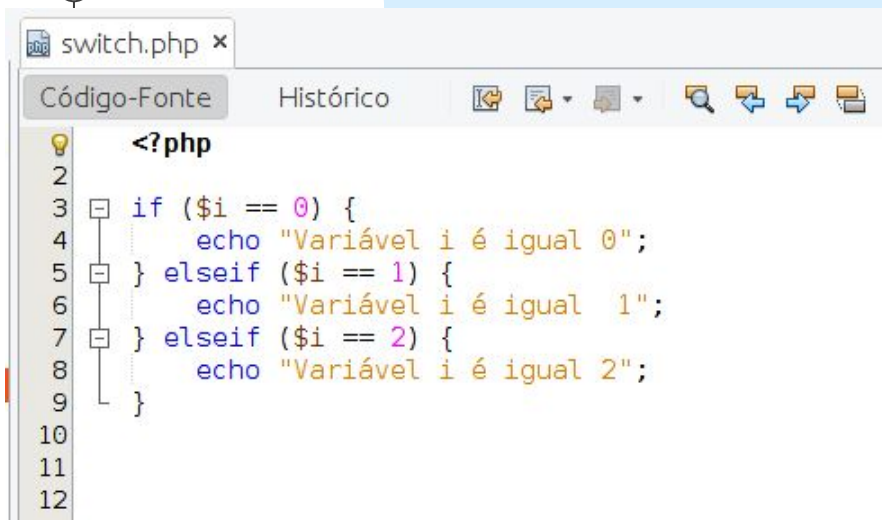
int(4)
string(5) "massa"

Valor

Switch case



- **Switch case**
- Similar a uma série de declarações IF na mesma expressão.
- Em muitos casos, se deseja comparar as mesmas variáveis (ou expressões), com diferentes valores, e executar pedaços diferentes de código dependendo de qual valor ela é igual.



```
switch.php x
Código-Fonte  Histórico
<?php
2
3 if ($i == 0) {
4     echo "Variável i é igual 0";
5 } elseif ($i == 1) {
6     echo "Variável i é igual 1";
7 } elseif ($i == 2) {
8     echo "Variável i é igual 2";
9 }
10
11
12
```



```
switch.php x
Código-Fonte  Histórico
<?php
2
3 $i = 2;
4
5 switch ($i) {
6     case 0:
7         echo "Variável i é igual 0";
8         break;
9     case 1:
10        echo "Variável i é igual 1";
11        break;
12    case 2:
13        echo "Variável i é igual 2";
14        break;
15 }
```

Atividades

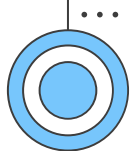


GALAXYPAY 
SEU NEGÓCIO RECORRENTE

Atividades

- 3) Faça um script, que receberá um valor X
Caso o valor seja menor que 20, exiba a frase: "O valor é menor que 20"
E vá incrementando esse número X recebido, até o número atingir o valor solicitado.
- 4) Faça um script, que irá calcular um valor de troco, dado 2 valores informados.
 - Receber 2 valores: \$pago, \$totalConta
 - Verificar se o valor pago é suficiente para pagar a conta, se não retornar direto.
 - Se for suficiente, retornar o valor que deve ser retornado para o usuário, em notas e moedas.
 - Podemos usar os seguintes arrays:
`$nota = array('100', '50', '10', '5', '1');`
`$centavos = array('50', '10', '5', '1');`





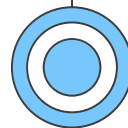
Conteúdo extra: Switch/case vs. Array

Em PHP vamos dar preferência a utilizar arrays ao invés de switch/case, pra deixar o código mais limpo.

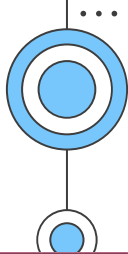
Veja um exemplo:



...



...



webdevs > aula03 > switch_vs_array.php > ...

```
1 <?php
2
3 function getTextUsingSwitchCase($index)
4 {
5     switch ($index) {
6         case 1:
7             return 'numero um';
8             break;
9         case 2:
10            return 'numero dois';
11            break;
12        default:
13            return '??????';
14            break;
15    }
16 }
17
18 getTextUsingSwitchCase(1);
19
```

switch_vs_array.php × while_vs_dowhile.php

webdevs > aula03 > switch_vs_array.php > ...

```
19
20 echo '<br><br>';
21 |
22 function getTextUsingArray($index)
23 {
24     $numbers = [
25         1 => 'numero um',
26         2 => 'numero dois',
27     ];
28     return isset($numbers[$index]) ? $numbers[$index] : '??????';
29 }
30
31 echo getTextUsingArray(1);
32
```

