

# Curso de PHP

PHP

FATEC - Jundiaí

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Matrizes Multidimensionais

A diferença entre as matrizes e as arrays é que acrescentamos mais um índice para identificar a posição na matriz. O número de índices será diretamente proporcional ao tamanho da matriz.

```
<?php
Exemplos de array: # exemplos de uma dimensão
$c = "PHP"; $d = 4;
$aula[3] = $c;
$aula["duração"] = $d;
```

Exemplos de matrizes de duas dimensões:

```
$f = "teste";
$matr[1][0] = $f;
$matr["pos1"][2] = $f;
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Matrizes Multidimensionais

```
<?php
    $matr[1][2] = 'Teste';

    echo "Isto não vai funcionar: $matr[1][2]";
    // retorna Array[0]

    echo "Isto agora vai funcionar:" . $matr[1][2];
    // Isto agora vai funcionar: Teste

    echo "Isto vai funcionar: {$matr[1][2]}";
    // Isso vai funcionar: Teste
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Matrizes Multidimensionais

Matrizes de mais de duas dimensões:

```
<?php
    $matr[1][2] = "Teste"; // duas dimensões
    echo $matr[1][2]; // retorna Teste

    $matr[1][2][0] = "Aula"; // três dimensões
    echo $matr[1][2][0]; // retorna Aula

    // o que será apresentado na Matriz matr na posição 1 - 2?
    echo $matr[1][2];
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Matrizes Multidimensionais

Para preencher uma matriz multidimensional, podemos usar o comando `array()` como no seguinte exemplo:

```
<?php
$a = array("maçã" => array( "cor" => "vermelho",
                           "sabor" => "doce",
                           "forma" => "redondo"
                         ),
          "melão" => array( "cor" => "amarelo",
                           "sabor" => "doce",
                           "forma" => "redondo"
                         ),
          "limão" => array( "cor" => "verde",
                           "sabor" => "azedo",
                           "forma" => "redondo"
                         )
);
echo "A maçã está " . $a["maçã"]["sabor"] . "<br>";
echo "A cor do limão é {$a["limão"]["cor"]} ";
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Matrizes Multidimensionais

Para cada índice de uma coluna, foi criada uma linha para ele, gerando a multidimensionalidade. A matriz gerada pelo código anterior foi:

Índice	Cor	Sabor	Forma
Maçã	Vermelho	Doce	Redondo
Melão	Amarelo	Doce	Redondo
Limão	Verde	Azedo	Redondo

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Estruturas de Controle

Basicamente, as estruturas de controle do PHP são as mesmas que para outro tipo de linguagem estruturada. O que podemos diferenciar é a sintaxe com que cada uma delas é descrita.

São usadas para realizar decisões lógicas, testar se determinada expressão é verdadeira, repetir um bloco de comandos por um certo número de vezes ou até que uma condição seja atingida.

Comandos condicionais:

IF                      SWITCH

Comandos de repetição:

WHILE      DO..WHILE      FOR      FOREACH

Obs.: Veremos também o comandos break e continue, que controlam o fluxo dentro das repetições (loops).

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Comandos Condicionais

São os responsáveis pela execução dos comandos ou blocos, baseando-se em testes (seleções) feitos em tempo real.

#### IF

É um dos mais importantes comandos condicionais. Executa o comando relacionado caso o resultado seja "verdadeiro" (*true*) e ignora caso seja "falso" (*false*).

Sintaxe:      *if (expressão)*  
                  *Comando;*

*ou*

*if (expressão)*  
                  {  
                  *Comando1;*  
                  *Comando2;*  
                  *(...)*  
                  *Comandox;*  
                  }

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Comandos Condicionais

#### IF

Exemplo do comando IF:

```
<?php
$hora_atual=date('H');
settype($hora_atual, "integer");
if (($hora_atual >= 0) and ($hora_atual < 18)) {
    echo "Bom-dia! ";
}
if ($hora_atual >= 18) {
    echo "Boa-noite!";
}
?>
```

Obs.: A função "date" recebe a data e hora do sistema atual, em formato de string, e por isso usamos a função "settype" para converter a hora atual em inteiro.

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### ELSE

É o complemento opcional para o IF. Tem por objetivo testar a próxima expressão caso o comando IF retorne um valor "falso".

Sintaxe:

<i>if (expressão)</i>	<i>if (expressão)</i>
<i>Comando;</i>	{
<i>else</i>	<i>Comando1;</i>
<i>Comando;</i>	<i>Comando2;</i>
	<i>(...)</i>
	<i>Comandon;</i>
	}
<i>ou</i>	<i>else</i>
	{
	<i>Comando1;</i>
	<i>Comando2;</i>
	<i>(...)</i>
	<i>Comandox;</i>
	}

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### ELSE

Exemplo do comando ELSE:

```
<?php
$hora_atual=date('H');
settype($hora_atual, "integer");
if (($hora_atual >= 0) and ($hora_atual < 18)) {
    echo "Bom-dia! ";
}
else {
    echo "Boa-noite!";
}
?>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### ELSEIF

Também podemos executar condições dentro do ELSE, incluindo sub-blocos de comandos.

Sintaxe:

<pre>if (expressão1)     Comando1; elseif (expressão2)     Comando2; else     Comando3;</pre>	<pre>if (expressão) {     Comando1;     Comando2; } elseif (expressão2) {     Comando1;     Comando2; } else {     Comando;</pre>
---	---

ou

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### ELSEIF

Exemplo do comando ELSEIF:

```
<?php
$hora_atual=date('H');
$hora_completa=date('H:i:s');
settype($hora_atual, "integer");
if (($hora_atual >= 0) and ($hora_atual < 12)) {
    echo "Bom-dia! <br>";
}
elseif ($hora_atual < 18) {
    echo "Boa-tarde! <br>";
}
else {
    echo "Boa-noite! <br>";
}
echo " Hora atual: $hora_completa";
?>
```

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### SWITCH

Este comando pode substituir uma série de comandos *IF*. Isto pode ser útil caso desejarmos comparar uma variável com diversos valores e executar um código diferente para cada resultado obtido.

Sintaxe:

```
Switch (variável) {
    Case valor1:
        Comando1;
        Break;
    Case valor2:
        Comando2;
        Break;
}
```

Obs.: O comando SWITCH do PHP é similar ao comando CASE de outras linguagens como o Pascal, Visual DataFlex, dentre outras mais.

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos Condicionais

#### SWITCH

Exemplo com comando Switch:

```
<?php
Switch($numero)
{
    Case 0:
        echo "valor 0";
        break; // quebra o fluxo
    Case 1:
        echo "valor 1";
        break;
    default:
        echo "nenhum dos valores";
        break;
}
?>
```

```
<? Php
if ($numero == 0) {
    echo "Valor 0";
}
elseif ($numero == 1) {
    echo "Valor 1";
}
else {
    echo "Nenhum dos valores";
}
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Comandos de Repetição

São comandos utilizados para que um conjunto de instruções seja executado repetidamente por um número determinado de vezes, ou até que determinada condição seja atingida.

#### WHILE

O comando while é composto por uma expressão e por um bloco de comandos. O comando avalia a expressão, e enquanto essa expressão retornar o valor verdadeiro, a execução do bloco de comando em questão será repetida. Quando o valor retornado for falso, encerra-se o loop, a execução é transferida para o fim do comando while.

Sintaxe:

```
While (exp)
{
    comandos
}
```

*Prof. Cláudio Farias Rossoni*



## PHP – Aula 4

### WHILE

Exemplo do comando while:

```
<?php
$cont = 1;
while ($cont < 3)
{
    echo "Valor do contador: $cont<br>";
    $cont++;
}
?>
```

Qual será o resultado na tela do terminal?

Valor do contador: 1  
Valor do contador: 2

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Comandos de Repetição

#### DO..WHILE

A diferença entre o comando while e o comando do..while é que o comando while avalia a expressão no início do loop, e o do..while avalia a expressão no final do loop.

O comando do..while será executado pelo menos uma vez, independente da condição em while ser verdadeira ou falsa, agora quando executamos o comando while, e a condição for falsa, ele nem executa.

Sintaxe:

```
do
{
    comandos
} while (exp);
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### DO..WHILE

Exemplo do comando while:

```
<?php
$cont = 3;
do
{
    echo "Valor atual do contador: $cont<br>";
    $cont--;
} while ($cont > 1)
?>
```

Qual será o resultado na tela do terminal?

Valor atual do contador: 3

Valor atual do contador: 2

E se a variável \$cont iniciar com 0 (zero) ?

Valor atual do contador: 0

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### Comandos de Repetição

#### FOR

Utilizamos o comando for quando queremos executar um conjunto de instruções um número determinado de vezes.

Sintaxe:

```
for (inicialização; condição; operador)
{
    comandos;
}
```

onde temos:

Inicialização - é realizado antes de termos iniciado o laço, atribuindo um valor inicial à variável.

Condição - sempre é do tipo booleana (*true* ou *false*), é responsável pela execução dos comandos. Enquanto for verdadeira, os comandos do laço serão executados.

Incremento - é o comando executado ao final do laço.

Prof. Cláudio Farias Rossoni

## PHP – Aula 4

### FOR

Exemplo do comando for:

```
<?php
for ($cont=0; $cont <3; $cont++)
{
    echo "A variável \$cont vale: $cont<br>";
}
?>
```

Qual será o resultado na tela do terminal?

A variável \$cont vale: 0  
A variável \$cont vale: 1  
A variável \$cont vale: 2

Obs.: O que vemos neste exemplo, é que a variável "\$cont" é iniciada com valor 0, e enquanto seu valor for menor que 3 (\$cont<3), o bloco deste laço é executado. Depois, o valor de cont é incrementado (\$cont++) e a condição é novamente testada. Quando a condição for falsa, o controle do programa sai do laço FOR.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### FOR

Exemplo do comando for para apresentar conteúdo de uma array:

```
<?php
$matr[0][0] = "Linha Zero – Coluna Zero";
...
$matr[2][1] = "Linha Dois – Coluna Um";
for ($lin=0; $lin <3; $lin++)
{
    for ($col=0; $col<2; $col++)
    {
        echo "O elemento da posição $lin, $col vale: ";
        echo $matr[$lin][$col] . "<br>";
    }
}
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### FOR

Podemos iniciar mais de uma variável no parâmetro de inicialização do comando for:

```
<?php
for ($vai=0, $volta=4; $vai < 4; $vai++, $volta--)
{
    echo "\$vai vale $vai e \$volta vale $volta ";
    if ($vai == $volta)
    { echo " (os valores são iguais!) "; }
    echo "<br>";
}
?>
```

Qual será o resultado apresentado?

\$vai vale 0 e \$volta vale 4

\$vai vale 1 e \$volta vale 3

\$vai vale 2 e \$volta vale 2 (os resultados são iguais!)

\$vai vale 3 e \$volta vale 1

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### FOREACH

O comando foreach nos oferece uma maneira mais fácil de navegar entre os elementos de um array.

Sintaxe:

```
Foreach ($nome_array as $elemento)
{
    Comandos
}
```

ou

```
Foreach ($nome_array as $chave=>$valor)
{
    Comandos
}
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### FOREACH

Exemplo do comando foreach:

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

O programa irá mostrar na tela todos os valores do array \$vetor, e depois através do segundo foreach apresentará:

```
$a[um] = 1.
$a[dois] = 2.
$a[tres] = 3.
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Controlando o fluxo de execução

Existem comandos que podem ser usados em conjunto com essas estruturas de controle que acabamos de ver.

Esses comandos são:

- BREAK

e o

- CONTINUE

Utilizamos estes tipos de comandos no interior dos laços DO, FOR e WHILE para interromper estes em um determinado instante.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Break

O break termina a execução do comando atual, que pode ser um if, for, while ou switch. O fluxo de execução passa para o primeiro comando após o término dessa estrutura. O comando break aceita um argumento numérico opcional que diz o número de estruturas que devem ser finalizadas.

```
<?php
$i=0;
while ($i<20)
{
    if ($i == 5) {
        echo "Encerrando o if...";
        break;
        echo "Essa linha não vai ser impressa!";
    }
    elseif ($i==15) { break 2; }
    $i++;
}
?>
```

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

### Continue

Comporta-se de maneira semelhante ao BREAK, com a diferença de que ao invés de interromper o laço, o comando de incremento é executado e o fluxo do laço volta ao início do bloco.

```
<?php
for ($i = 1; $i <= 10; $i++)
{
    if ($i == 5)
        continue; // pula a execução quando $i = 5
    echo "$i ";
}
echo "Saí do laço";
?>
```

Obs.: Neste exemplo, quando "\$i" for 5, a execução voltará ao início do bloco, não mostrando no browser este valor.

*Prof. Cláudio Farias Rossoni*

## PHP – Aula 4

PHP

Termino da aula

*Prof. Cláudio Farias Rossoni*