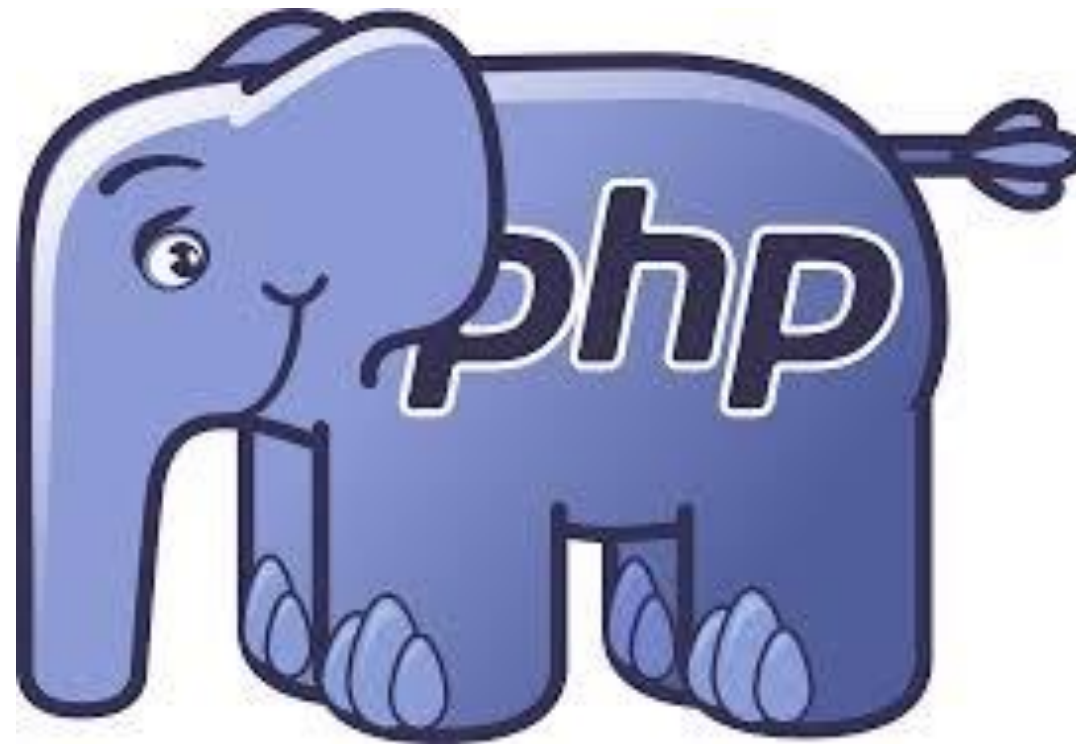




AULA 2

Estruturas de Controle

Condicional Simples

```
if ( $a > $b )  
    { bloco 1 }
```

Condicional Composto

```
if ( $a > $b )  
    { bloco 1 }  
else  
    { bloco 2 }
```

Condicional Aninhado

```
if ( $a > $b )  
    { bloco 1 }  
elseif ( $a == $b )  
    { bloco 2 }  
elseif ( $a != $b )  
    { bloco 3 }  
else  
    { bloco 4 }
```

Condicional (outro modelo)

```
If ($x>$y):  
    echo $x;  
elseif ($x<$y) :  
    echo $y;  
else:  
    echo "iguais";  
endif;
```

```
<?php
    if ($nota == 10)
    {
        echo "Parabéns! <br>";
        echo "Você tirou a nota máxima!";
    }
?>
```

```
<?php
    $prova1 = 7;
    $prova2 = 5;
    $nota = ($prova1+$prova2) / 2;
    if ($nota<3)
        $desempenho = "PÉSSIMO";
    elseif ($nota<5)
        $desempenho = "RUIM";
    elseif ($nota<7)
        $desempenho = "MÉDIO";
    elseif ($nota<9)
        $desempenho = "BOM";
    else
        $desempenho = "EXCELENTE";
    echo "O seu desempenho foi $desempenho";
?>
```

68 – Utilizando a linguagem PHP, analise o código abaixo e marque a opção correta.

```
if ($x == $y)
    comando1;
    comando2;
```

- a) Se \$x for igual a \$y, somente o comando1 será executado.
- b) O comando2 somente será executado se \$x for diferente de \$y.
- c) O comando2 será executado independentemente dos valores de \$x e \$y.
- d) O comando1 nunca será executado.

Condicional múltiplos casos

```
switch ($valor)
{
    case 0:
        bloco 0
        break;
    case 1:
        bloco 1
        break;
    default:
        bloco 2
        break;
}
```



Utilizando switch

 exemplo5_3.php

```
<?php
...
switch ($numero)
{
    case 0:
        echo "número vale 0";
        break;
    case 1:
        echo "número vale 1";
        break;
    case 2:
        echo "número vale 2";
        break;
}
...
?>
```

Enquanto

```
$x = 1;  
while ($x <= 10)  
{  
    bloco  
    $x++;  
}
```

```
$x = 1;  
while ($x <= 10):  
    bloco  
    $x++;  
endwhile;
```

```
<?php  
    $cont = 1;  
    while ($cont<100)  
    {  
        echo "O valor atual do contador é $cont <br>";  
        $cont++;  
    }  
?>
```

Há também uma sintaxe alternativa, utilizando os dois-pontos:

```
while (exp):  
    comandos  
endwhile;
```

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

42 – Selecione a alternativa com o resultado exibido na tela caso haja a execução de um programa em PHP com as seguintes linhas de código:

```
<?php
    $ponto = 5;
    while ($ponto < 10)
    {
        print $ponto;
        for($i = 0; $i < 1; $i++)
        {
            $ponto ++;
        }
    }
?>
```

- a) 56789
- b) 456789
- c) 678910
- d) 5678910

94 – Ao analisar o código PHP abaixo, pode-se afirmar que

```
<?php
    $a=2;
    while($a <= 10){
        if ( ( ++$a % 2 ) == 0 ) {
            echo $a;
        }
        $a++;
    }
?>
```

- a) nenhum número será mostrado na tela.
- b) os números 2, 4, 6, e 8 serão mostrados na tela.
- c) os números 4, 6, 8 e 10 serão mostrados na tela.
- d) os números 2, 4, 6, 8 e 10 serão mostrados na tela.



Repita

```
$x = 1;  
do {  
    bloco  
    $x++;  
} while ($x <= 10);
```

```
<?php  
    $numero = 1;  
    do  
    {  
        echo "O valor atual de número é $numero <br>";  
        $numero++;  
    } while ($numero<4);  
?>
```

for

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

Assim como no *if* e no *while*, há uma sintaxe alternativa para o *for*:

```
for ( inicialização ; condição ; operador ):  
    comandos  
endfor;
```

Para

```
for ($x = 1; $x <= 10; $x++)  
{  
    bloco  
}
```

```
for ($x = 1; $x <= 10; $x++):  
    bloco  
endfor;
```

Para Cada

```
$vetor = array range(1,30);  
foreach ($vetor as $valor)  
{  
    echo "O valor atual é $valor <br>";  
}
```

```
$vetor = array ("nome" => "Gustavo", "profissão" =>  
"Professor", "idade" => 36);  
foreach ($vetor as $chave => $valor)  
{  
    echo "Campo $chave contém $valor <br>";  
}
```

exemplo5_7.php

```
<?php
    for($cont=0 ; $cont<10 ; $cont++)
    {
        echo "A variável \$cont vale $cont";
        echo "<br>";
    }
?>
```

```
<html>
<body>
<?php
    echo "Estou fazendo uma contagem regressiva: <br>";
    for($i=15 ; $i>=0 ; $i-)
    {
        echo $i . " , ";
    }
    echo "... FIM!";
?>
</body>
</html>
```



```
<?php
    $vetor[0][0]= "elemento00";
    $vetor[0][1]= "elemento01";
    $vetor[1][0]= "elemento10";
    $vetor[1][1]= "elemento11";
    for($i=0 ; $i<2 ; $i++)
    {
        for($k=0 ; $k<2 ; $k++)
        {
            echo "O elemento da posição $i,$k é ";
            echo $vetor [$i][$k] . "<br>";
        }
    }
}
```

```
<?php
    for( $i=0, $k=10  ;  $i<10  ;  $i++, $k- )
    {
        echo "\$i vale $i e \$k vale $k";
        if ($i==$k)
            { echo " (os valores são iguais!)"; }
        echo "<br>";
    }
?>
```

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 o *break* e o *continue*.

Vamos ver qual é a utilidade de cada um deles:

break

O *break* termina a execução do comando atual, que pode ser um *if*, *for*, *while*, ou *switch*. Quando o *break* é encontrado dentro de algumas dessas estruturas, o fluxo de execução passa para o primeiro comando após o término dessa estrutura. É um recurso que podemos utilizar para forçar a saída de um laço ou de um comando condicional. Acompanhe o exemplo a seguir:

Interrompendo o fluxo

```
$vetor = array (1, 4, 6, 2, -1, 8, 7, 5);  
foreach ($vetor as $valor)  
{  
    if ($valor == -1) {  
        break;  
    }  
    echo "O valor atual é $valor <br>";  
}
```

```
<?php
$i = 0;
$k = 0;
while ($k < 10)
{
    $i++;
    $k++;
    while ($i < 20)
    {
        if($i==10)
        {
            echo "Encerrando o primeiro while...<br>";
            break ;
            echo " Essa linha não vai ser impressa!!!";
        }
        elseif ($i==15)
        {
            echo "Encerrando os dois whiles...";
            break 2;
        }
        $i++;
    }
}
?>
```

exemplo5_13.php

```
<?php  
    $vetor = array (1, 3, 5, 8, 11, 12, 15, 20);  
    for($i=0 ; $i<sizeof($vetor) ; $i++)  
    {  
        if ($vetor[$i] % 2 != 0)    // é ímpar  
        { continue; }  
        $num_par = $vetor[$i];  
        echo "O número $num_par é par. <br>";  
    }  
?>
```

Modificando o fluxo

```
$vetor = array (1, 4, 6, 2, -1, 8, 7, 5);  
for ($i = 0; $i < sizeof($vetor); $i++)  
{  
    if ($vetor[$i] % 2 == 0) {  
        continue;  
    }  
    echo "O valor atual é $valor <br>";  
}
```

