

Aula 9 – Laços de Repetição em php

Estruturas de Repetição

While...{

A estrutura While possui a seguinte sintaxe:

```
While <condição> {  
    <comando a>;
```

No php, as instruções são separadas umas das outras pelo ponto-e-vírgula (;) e a estrutura While entende como sendo o seu fim a primeira chave (}) que encontrar.

Exemplos:

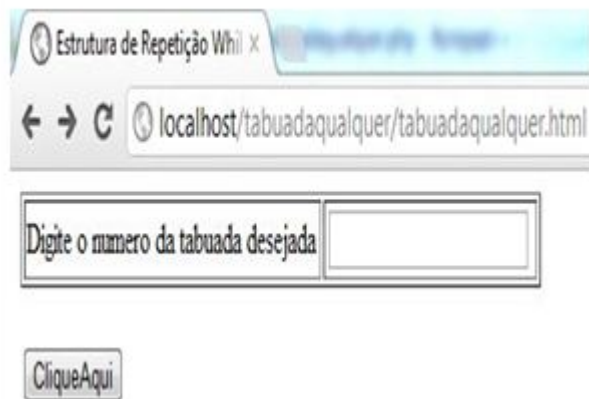
1) Mostrar os valores da tabuada do dois de 0 a 10 no ListBox:



Código da página tabuada.php

```
<html>  
<title> página tabuada.php </title>  
<body>  
<?php  
    $i = 0;  
    while( $i <= 10 ){  
        $stab = 2 * $i;  
        echo '2' . ' X ' . $i . ' = ' . $stab . '<br />';  
        // $i++;  
    }  
>  
</body>  
</html>
```

2) Mostrar a tabuada de um número qualquer.



Código da página
tabuadaqualquer.php

```
<html>
<head>
<title> página tabuadaqualquer.php </title>
</head>
<body>
<?php
    $i = 0;
    $num = $_POST['txtnum'];
    while( $i <= 10 ){
        $tab = $num * $i;
        echo $num . ' X ' . $i . ' = ' . $tab .
'<br />';
        $i++;
    }
?>
</body>
</html>
```

Estrutura Do...While

Repete um bloco de instruções até que uma certa condição seja satisfeita. Sintaxe:

```
Do{
    Comando_1;
    Comando_2;
    Comando_3;
    ....
} While (expressão_lógica);
```

Neste caso, todos os comandos entre as palavras reservadas **DO** e **WHILE** serão executadas, até que a expressão lógica seja verdadeira (FALSE), obviamente, devemos ter o cuidado para que ela venha a ser FALSE em determinado momento, pois caso contrário, teremos um LOOP INFINITO, (o programa fica preso dentro da estrutura DO - While!).

Exemplo: Vamos usar o mesmo exemplo da tabuada visto na estrutura While {:

```

2 X 0 = 0
2 X 1 = 2
2 X 2 = 4
2 X 3 = 6
2 X 4 = 8
2 X 5 = 10
2 X 6 = 12
2 X 7 = 14
2 X 8 = 16
2 X 9 = 18
2 X 10 = 20

```

Implementação da página tabuadadois.php

```

<HTML>
<HEAD>
<TITLE>Página que calcula tabuada do dois</TITLE>
</HEAD>
<BODY>
<?php
    $i = 0;
    do{
        $tab = 2 * $i;
        echo '2' . ' X ' . $i . ' = ' . $tab .
    '<br />';
        $i++;
    } while( $i <= 10 );
    ?>
</body>
</html>

```

Estrutura For{.

O comando FOR identifica uma variável de controle para o loop e indica uma faixa de valores (\$ValorInicial ; \$ValorFinal) que a variável receberá durante as iterações do loop. Além disso, o For utiliza-se de uma variável contadora que controla o número de vezes que o loop será executado. Esse contador não é um contador simples, pois, não **conta obrigatoriamente apenas de uma em uma unidade**, somando (vcontadora++) ou subtraindo (vcontadora--) uma unidade da variável contadora.

Funciona dessa forma:

No início da execução, a variável de controle recebe o valor do ValorInicial, e a primeira iteração é executada.

Antes de cada iteração subsequente, a variável de controle recebe o próximo valor da faixa de ValorInicial até ValorFinal.

O loop termina após a iteração correspondente ao ValorFinal.

```

FOR (variável= ValorInicial; variável<= ValorFinal;i++){
..... comandos;
}

```

Num loop FOR cuja intenção do programador é incrementar, o valor ordinal do ValorInicial deve ser menor que o valor ordinal do ValorFinal. (Senão o loop não terá ação nenhuma). Alternativamente, você pode usar o operador aritmético -- para projetar um loop FOR que seja decrementado dentro de uma faixa ordinal:

<pre> for(\$i = 1; \$i <= 5; \$i++){ echo \$i . '
'; } </pre>	<p>Será apresentado na página:</p> <pre> 1 2 3 </pre>
---	--

}	4
	5

FOR (variável= ValorInicial; variável>= ValorFinal;i--)
 comandos;
 }

<pre>for(\$i = 5; \$i >=1; \$i--){ echo \$i . '
'; }</pre>	Será apresentado na tela: 5 4 3 2 1
--	---

DICAS: A variável do For{ é incrementada ou decrementada automaticamente, portando você nunca deve alterar o valor desta variável dentro do laço do For{.

Exemplo: (Nunca faça !)

FOR (\$Contador = 50; \$Contador>=1;i--){

i-=1; {Você terá um decremento de 2 em 2 e não de um em um como estipulado no for - Contador é alterado no laço! }

- Caso você deseje alterar o valor que será somado ou subtraído a cada execução do for, faça conforme os exemplos abaixo:
- Decremento de 2 em 2: for(\$i = 50; \$i >=1; \$i-=2){ comandos}
- Incremento de 2 em 2: for(\$i = 1; \$i <=50; \$i+=2){ comandos}
- Certifique-se de que a variável contadora vai realmente chegar até o valor final indicado no For{, pois caso no contrário teremos um laço infinito.
- Quando você tem várias instruções a serem executadas dentro do laço For, você deve usar o par de chaves {} (da mesma forma como ocorre com os outros comandos do PHP, como por exemplo no IF/THEN/ELSE).
- Usando um Laço For{ você sabe exatamente o número de execuções que seu programa fará.