

Estruturas de Controle

Aula 03

Como dito na aula anterior, o PHP é uma linguagem “C-Like” por isso utiliza a mesma sintaxe da linguagem-irmã.

As estruturas de controle de PHP não poderiam deixar de ser as nossas velhas conhecidas:

- Estruturas Condicionais
- Estruturas Iterativas

Estruturas de Controle

As estruturas condicionais suportadas pelo PHP são:

- `if`
- `switch`

OBS: Lembre-se sempre que o PHP é case-sensitive, isto faz com que as instruções: `IF`, `If`, `SWITCH` ou `Switch` sejam consideradas **inválidas**.

Estruturas Condicionais

Comando **if** (instrução simples):

Para uma única instrução

```
if (<expressão lógica>)  
    <instrução>;
```

Para mais de uma instrução (bloco)

```
if (<expressão lógica>)  
{  
    <bloco>;  
}
```

Estruturas Condicionais

Comando **if** (Exemplo):

```
$num = 3;  
if ($num % 2 == 1)  
    $num++;  
echo $num;
```

Note que a estrutura **if** não possui **then**;

O operador **%** calcula o resto da divisão de **\$num** por 2;

Só existe uma instrução a ser executada caso a expressão seja V;

A instrução **echo \$num**; sempre será executada, pois está fora do **if**;

A estrutura apresentada poderia ser substituída pela instrução ternária:

```
$num = ($num % 2)?$num+1;
```

O resultado exibido será 4;

Estruturas Condicionais

Comando **if** (instrução estendida):

```
if ( <expressão lógica> )  
    <instrução V>;  
else  
    <instrução F>;
```

Estruturas Condicionais

Comando **if** (Exemplo):

```
$n1=6; $n2=4;  
$m=($n1+$n2)/2;  
if ($m >= 7)  
    $sit = "Aprovado";  
else  
    $sit = "Reprovado";  
echo $sit;
```

Só existe uma instrução a ser executada caso **\$m >= 7** e caso **\$m < 7**;

As duas instruções internas deverão terminar com ponto-e-vírgula;

A estrutura apresentada poderia ser substituída pela instrução ternária:

```
$sit = ($m >= 7)? "Aprovado": "Reprovado";
```

O resultado exibido será **Reprovado** já que **\$m = 5**;

Estruturas Condicionais

Comando **if** (instruções aninhadas):

```
if (<expressão lógica>
{
    <bloco A>;
}
elseif (<expressão lógica>)
{
    <bloco B>;
}
else
{
    <bloco C>;
}
```

Estruturas Condicionais

Comando **if** (Exemplo):

```
$n1=6; $n2=4;  
$ap=0; $rp=0; $rc=0;  
$m=($n1+$n2)/2;  
if ($m >= 7)  
{  
    $sit = "Aprovado";  
    $ap++;  
}  
elseif ($m >= 5)  
{  
    $sit = "Em recuperação";  
    $rc++;  
}  
else  
{  
    $sit = "Reprovado";  
    $rp++;  
}  
echo $sit;
```

Os blocos foram necessários;

Podemos ter várias cláusulas **elseif** em uma mesma estrutura **if**;

A instrução **echo \$sit**; sempre será executada, pois está fora dos blocos da estrutura **if**;

O resultado exibido será **Em recuperação**, já que **\$m=5**.

Estruturas Condicionais

As estruturas iterativas (de repetição) permitem que um bloco de comandos possam ser executados mais de uma vez.

O PHP possui três estruturas de repetição:

- `while`
- `do..while`
- `for`

OBS: Lembre-se sempre que o PHP é case-sensitive, isto faz com que as instruções: `While`, `WHILE` ou `While` sejam consideradas **inválidas**.

Estruturas Iterativas

```
while (<expressão lógica>)  
{  
    <bloco de instruções>  
}
```

Instrução while

Exemplo

```
<? $n = 0;  
$a = 1;  
while ($n <= 10)  
{  
    echo "Valor de N: $n" ;  
    $n += 5;  
    $a++;  
}  
echo "Valor de A: $a" ;  
?>
```

\$n	\$a	(\$n <= 10)?	Na Tela
0	1	True	Valor de N: 0
5	2	True	Valor de N: 5
10	3	True	Valor de N: 10
15	4	False	Valor de A: 4

Instrução while

17) No código abaixo:

```
<? $x=2;  
while ($x<10)  
{  
    $x+=2;  
}  
echo $x; ?>
```

Qual será o resultado exibido na tela?

- a. 10
- b. 8
- c. 2
- d. 0

Exercício

```
do {  
    <bloco de instruções>  
} while ( <expressão lógica> );
```

Instrução do..while

Comando **do..while** (Exemplo):

```
$a=0;  
do {  
    $echo "Repetição $a <br>";  
    $a++;  
} while ($a < 5);
```

O resultado exibido será:

Repetição 0
Repetição 1
Repetição 2
Repetição 3
Repetição 4

A variável **\$a** é numérica e fará parte do teste lógico;

Leremos a segunda linha e a quinta linha como **“faça...enquanto \$a for menor que cinco”**;

Instrução **do..while**

**Qual a diferença entre
while e do..while?**

Tem diferença?

Então, qual a diferença?

```
$x=10;  
while ($x < 10)  
{  
    $echo "O valor é $x";  
}
```

No primeiro caso, nada será exibido!

```
$x=10;  
do {  
    $echo "O valor é $x";  
} while ($x < 10);
```

No segundo caso, seria exibido a mensagem **O valor é 10**.

Diferença entre while e do..while

```
for (<inicialização>;<condição>;<incremento>)  
{  
    <bloco de instruções>  
}
```

Exemplo:

```
for ($c=1;$c<20;c+=5)  
{  
    $echo "Valor: $c<br>";  
}
```

O resultado exibido será:

Valor: 1

Valor: 5

Valor: 10

Valor: 15

A variável **\$c** é numérica e fará parte do teste lógico;

Leremos a segunda linha como “**\$c começa com o valor 1, e enquanto for menor que 20, vai aumentando AUTOMATICAMENTE de 5 em 5**”;

Instrução for

18) Quantas vezes será exibida na tela a mensagem “Olá!”, se executarmos o código abaixo?

```
for ($i=0;$i<=6;$i++)  
{  
    echo “Olá!”;  
}
```

- a. 5 vezes
- b. 6 vezes
- c. 7 vezes
- d. Nenhuma vez

Exercícios

19) Qual das estruturas abaixo possui incremento automático por loop?

- a. while
- b. do..while
- c. For
- d. nenhuma delas

20) Qual das opções abaixo causará um loop com exatamente **10 iterações**?

- a. for (\$i=0;\$i<=10;\$i++)
- b. for (\$i=1;\$i<10;\$i++)
- c. for (\$i=10;\$i<=100;\$i+=10)
- d. for (\$i=10;\$i<=10;\$i++)

Exercícios

21) O que será exibido no navegador após a execução do seguinte programa PHP?

```
<?php
$ast=1;
for ($li=1;$li<=3;$li++)
{
    for ($c=1;$c<=$ast;$c++)
    {
        echo "*";
    }
    echo "<br>";
    $ast+=2;
} ?>
```

A)

```
*
* * *
* * * * *
```

B)

```
      *
    * * *
  * * * * *
```

C)

```
          *
        * * *
      * * * * *
```

Exercícios

Próxima Aula: Formulários HTML