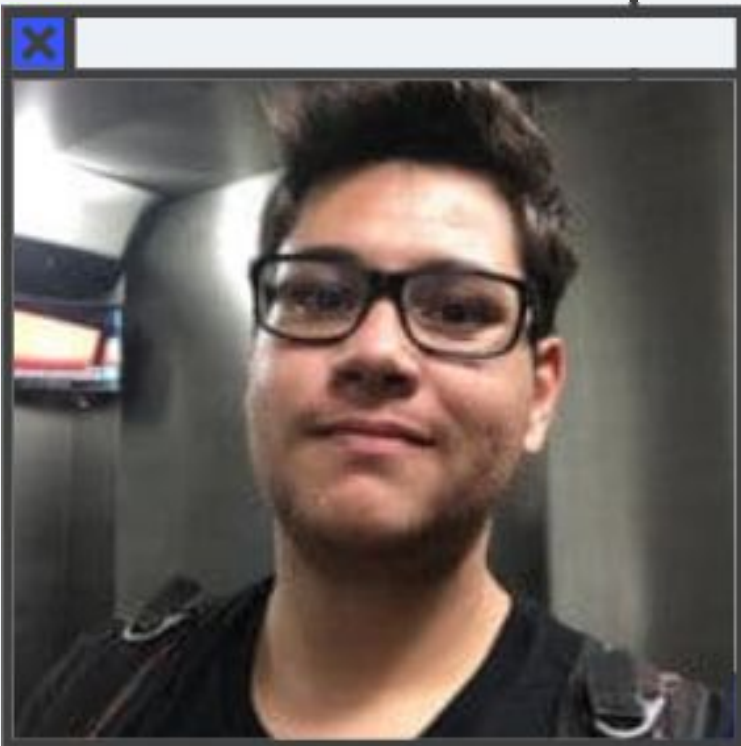


Estruturas de decisão e looping

Aula 04





Wagner Souza

Arquiteto de software
`wagner.souza@uni9.pro.br`

PROGRAMAÇÃO WEB

Resumo da aula

Programação Web - PHP

Tópicos Abordados

- Estruturas de decisão
- Estruturas de looping
- Desafios

- if
- if else
- if elseif else
- switch

Estruturas de Decisão

Estruturas de decisão

O objetivo é identificar o conteúdo de uma condição e direcionar o fluxo do script para uma determinada rotina, desvio, função, etc.

Primeiramente vamos entender o que é uma expressão condicional. Esta é uma expressão que sempre retorna um valor booleano, ou seja, verdadeiro ou falso.

IF

If é uma estrutura de decisão que tem o objetivo de testar se uma condição é verdadeira, a sintaxe do if é relativamente simples.

if - Exemplo

```
if(1 > 3) {  
    return true;  
}
```

if else

A instrução `if...else` é muito semelhante a instrução `if`, porém se a condição verificada for falsa, damos uma opção de saída para nosso código.



if else - Exemplo

```
if(1 > 2) {  
    return true;  
} else {  
    return false;  
}
```

if elseif else

Assim como a instrução IF a instrução ELSEIF também contém uma declaração condicional, todavia, esta instrução deve obrigatoriamente ser precedida por uma instrução IF. Você não pode ter uma instrução ELSEIF sem ter primeiro uma instrução IF.



if elseif else

Quando o PHP avalia uma instrução IF...ELSEIF, a verificação acontece no primeiro if, caso a condição seja falsa aí então será checada a segunda condição, a condicional do elseif.



if elseif else - Exemplo

```
if(1 > 3) {  
    return false;  
} else if(1 < 2) {  
    return false;  
} else {  
    return true;  
}
```

switch

Com o uso da instrução switch/case você pode verificar todas essas condições de uma só vez, e a grande vantagem é que ele faz toda essa verificação de forma muito mais eficiente de que as declarações if...else / if...elseif.



switch - Exemplo

```
switch(1){  
    case 1:  
        return true;  
        break;  
    case 2:  
        return true;  
        break;  
    default:  
        return false;  
}
```

- for
- for each
- while
- do while

Estruturas de looping

Estruturas de looping

As estruturas de repetição são úteis, para executar tarefas repetitivas, que sempre são cansativas, podemos, por exemplo, repetir elementos de uma lista ou de uma tabela de dados, ou simplesmente para executar um mesmo processamento até que uma certa condição seja satisfeita ou até atingir um determinado número de vezes que o bloco de instrução tenha sido executado.

for

De forma básica, utilizamos a instrução for quando já sabemos, previamente, a quantidade de vezes que iremos executar nossa estrutura de repetição



for

Podemos perceber que a instrução for é dividida em três expressões, a saber:

expressao1: é a variável de controle inicial;

expressao2: é a condição a ser testada, sempre que a condição for verdadeira a repetição é executada;

expressao3: é o incremento.



for - Exemplo

```
for(expressao1; expressao2;expressao3) {  
    // instruções a serem executadas repetidamente  
}
```

for - Exemplo

```
for(expressao1; expressao2;expressao3) {  
    // instruções a serem executadas repetidamente  
    for(expressao4; expressao5;expressao6){  
        // instruções a serem executadas repetidamente  
    }  
}
```

for - Desafio 1

Tabuada do 1

for - Desafio 2

Tabuada do 1 ao 10

for each

Esta estrutura de repetição é bem semelhante a instrução for, porém, indicado quando temos uma coleção de dados armazenada numa variável, ou seja, num vetor ou em uma matriz. Na instrução foreach não precisamos verificar o limite do looping como na instrução for, nos preocupamos somente em manipular o objeto atual. Para simplificar, pensemos da seguinte forma: Para cada elemento contido na coleção, faça algo.



for each - Exemplo

```
$expressao_array=array(valor1,valor2,...valorN);  
foreach ($expressao_array as $valor){  
    // instruções executadas repetidamente;  
}
```


for each - Exemplo

```
$nomes=array('Thiago','Maria','Joao');  
foreach ($nomes as $valor){  
    echo $valor . "<br>"  
}
```

for each

Na segunda forma fazemos praticamente a mesma coisa da primeira opção, exceto pelo fato de que a posição do elemento atual será atribuído à variável `$indice` em cada iteração e o conteúdo desse índice será atribuído a variável `$valor`.



for each - Exemplo

```
$nomes = array('Thiago', 'João', 'Ricardo', 'Paula');  
foreach ($nomes as $indice => $valor) {  
    echo $indice . " - " . $valor . "<br>";  
}
```

while

A instrução while irá repetir a execução de um determinado bloco de instrução enquanto a condição testada seja verdadeira, quando essa condição se tornar falsa, essa repetição é suspensa.

No caso de scripts que utilizem a estrutura de repetição while, a presença de um contador se torna obrigatória, pois, caso contrário, esta estrutura de repetição cairá num loop infinito.



while - Exemplo

```
$i=1; // inicialização do nosso contador;  
while (condição) {  
    // instruções a serem executadas repetidamente;  
    $i++; // incremento do contador  
}
```

while - Exemplo

```
$i=1; // inicialização do nosso contador;  
while ($i <= 10) {  
    echo $i . " ";  
    $i++;  
}
```

do while

As instruções do tipo `do...while` são semelhantes aos loops `while` com a diferença de que os loops `do-while` primeiro executa o bloco de instruções e depois verifica se a expressão é verdadeira. Ou seja, neste tipo de estrutura o bloco de repetição será executado ao menos uma vez, para somente depois a condição ser verificada, caso esta continue verdadeira o bloco de instruções será repetido novamente, caso contrário, o bloco de instruções `do...while` será encerrado.



do while - Exemplo

```
$i=1; // inicialização do nosso contador;  
do {  
    // instruções a serem executadas repetidamente;  
    $i++; // incremento do contador  
} while (condição);
```



do while - Exemplo

```
$i=1; // inicialização do nosso contador;  
do {  
    echo $i . " ";  
    $i++; // incremento do contador  
} while ($i < 10);
```

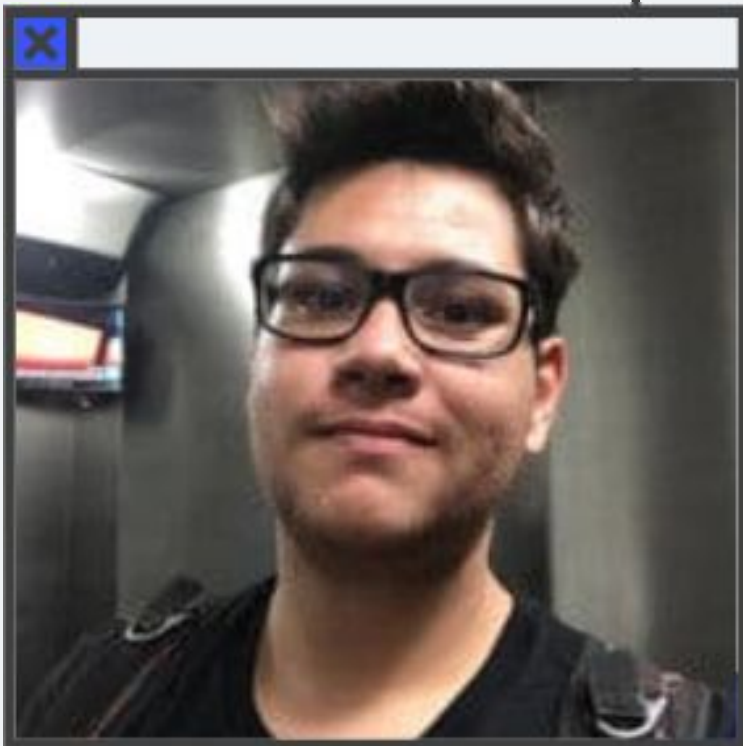


Diferenças entre blocos de repetições

Na instrução `for`, a condição sempre será numérica inteira, neste tipo de instrução não são aceitos valores numéricos do tipo `float` ou do tipo `string`.

Na instrução `for`, a quantidade de vezes que o loop será executado já conhecido anteriormente, a estrutura do tipo `while` será executada enquanto a condição for verdadeira.

A estrutura do tipo `do...while`, será executada ao menos uma vez, diferentemente do `while` que somente será executada caso a condição seja verdadeira.



Obrigado Wagner Souza

Arquiteto de software
`wagner.souza@uni9.pro.br`

Programação Web