



Construção. web sites

Prof: José

Select

Os campos select permitem tratar uma variedade de opções, onde o usuário pode selecionar apenas uma opção ou múltiplas opções. Quando você permite múltiplas seleções, deve adicionar "[]" no final do nome, para o PHP interpretar como array.

```
<b>Qual seu processador?</b><br>
<select name=processador>
  <option value=i3>i3</option>
  <option value=i5>i5</option>
  <option value=i7>i7</option>
  <option value=AMD selected>AMD</option>
</select>
```

**Livros que deseja comprar?
**

Obs: segure "CTRL" para selecionar mais de um.


```
<select name="livros[]" multiple>
    <option value="Biblia do PHP">Biblia do PHP</option>
    <option value="PHP Professional">PHP Professional</option>
    <option value="Iniciando em PHP">Iniciando em PHP</option>
    <option value="Biblia do MySQL">Biblia do MySQL</option>
</select>
```

```
<?php
echo "Seu processador é: " . $_POST["processador"] . "<BR>";
// Verifica se usuário escolheu algum livro
if(isset($_POST["livros"])) {
    echo "O(s) livro(s) que você deseja comprar:<br>";
    // Faz loop para os livros
    foreach($_POST["livros"] as $livro){
        echo "- " . $livro . "<br>";
    }
}
else{
    echo "Você não escolheu nenhum livro!";
}
?>
```

Arrays

Arrays, também muito conhecidos como vetores, são variáveis que servem para guardar vários valores de forma uniforme na memória. Por exemplo, se tivemos que utilizar 20 variáveis para guardar o mesmo tipo de informação, nós não faríamos -> \$var1, \$var2, \$var3, \$var4, \$var5, ... ao invés disso, criaríamos apenas uma variável de array para guardar todas as 20 informações de uma vez.

Primeiramente, vamos aprender como construir um array. Podemos construir um array usando a função de mesmo nome do PHP, array().

```
<?php
    $vetor = array();
?>
```

No exemplo acima, a variável *\$vetor* já é um array! Simples assim.

Se quisermos, também podemos inicializar nosso array contendo alguns valores. Por exemplo, vamos inicializar nosso array com três nomes.

```
<?php
    $vetor = array("Fulano", "Ciclano", "Beltrano");
?>
```

Acessar Uma Posição no Array

Como um único array pode possuir vários valores, cada valor fica em uma posição específica da variável.

Em PHP, uma das formas de acessar tais valores é utilizando o índice (index) do array que sempre se inicia em 0.

Ou seja, todo índice é definido pela *posição - 1*. Isso quer dizer que, se quisermos acessar o quarto elemento de um array devemos usar o índice da posição 3, porque $4 - 1 = 3$.

Os índices são indicados por colchetes ([]) que vem logo após o nome da variável. Portanto, se quisermos modificar ou acessar algum dado da variável faríamos:

```
<?php
    $vetor = array();
    $vetor[1] = 540;
    $vetor[3] = 8456;
?>
```

Arrays Associativos

Uma característica bem interessante dos arrays em PHP é a capacidade de fazer associações.

Quando nos referimos a associações, queremos dizer que, ao invés de usarmos números como índices, podemos usar strings (nomes). Dessa forma, fica muito mais simples e intuitivo obter um valor de um array, pois nomes fazem mais sentido do que números.

Para usarmos o array associativo basta apenas substituir o número do índice por uma string. Veja o exemplo abaixo.

```
<?php
$doc = array();
$doc['rg'] = "00.000.000-X";
$doc['cpf'] = "000.000.000-00";
$doc['cartao de credito'] = 12345;
?>
```

Caso queiramos inicializar o array diretamente usando associações, devemos usar seta larga para vincular um valor a uma chave.

A seta larga é composta pelo sinal de igual mais o sinal de maior (=>). Veja o mesmo código usando as setas largas diretamente na associação.

```
<?php
    $doc = array("rg" => "00.000.00-X",
                 "cpf" => "000.000.000-00",
                 "cartao de credito" => 12345);

    echo ("Os documentos de Fulano são:" .
    "<BR>RG: ". $doc['rg'] .
    "<BR>CPF: ". $doc['cpf'] .
    "<BR>Cartão de Crédito: ". $doc['cartao de credito']);

?>
```

Laços de repetição

Quando for necessário efetuar a repetição de um trecho de um programa um determinado número de vezes o que você faria? Escreveria de novo? Copiaria e colaria? Afinal estamos utilizando de máquinas que são feitas para trabalhar para nós e não o contrário. Quando encontrarmos um problema como o apresentado anteriormente temos os laços de repetição

While

Esta instrução é usada quando não sabemos quantas vezes um determinado bloco de instruções precisa ser repetido. Com ele, a execução das instruções vai continuar até que uma condição seja verdadeira. A condição a ser analisada para a execução do laço de repetição deverá retornar um valor booleano.

```
var $contador = 0;
while($contador < 10)
{
    echo $contador;
    $contador++;
}
```

FOR

A estrutura de repetição **FOR** é utilizada para se executar um conjunto de comandos por um número definido de vezes. Para esse operador, são passados uma situação inicial, uma condição e uma ação a ser executada a cada repetição.

Em geral informamos uma variável que serve como contador de repetições, com seu valor inicial, uma condição a ser atendida para que cada repetição seja executada e um incremento ao contador.

```
<?php
for($i =1; $i < 20; $i++){
    echo ("Upload ".$i.".": <input type='text' name='".$i."'><br/>");
}
?>
```

FOREACH

O FOREACH é uma simplificação do operador FOR para se trabalhar em coleções de dados, ou seja, vetores e matrizes. Ele permite acessar cada elemento individualmente iterando sobre toda a coleção e sem a necessidade de informação de índices.

Por exemplo, supondo que fosse preciso percorrer um vetor com alguns elementos com o FOR. Seria necessário utilizar um contador para servir também de índice para acessar cada elemento, conforme a listagem a seguir.

```
$vetor = array(1, 2, 3, 4, 5);  
for($i = 0; $i < 5; $i++){  
    $item = $vetor[$i];  
    echo $item;  
}  
foreach($vetor as $item){  
    echo $item;  
}
```

Exercício

- Crie uma pagina que deve cadastrar os dados em um array contendo(nome, cpf, rg, email, telefone) e deve mostrar estes dados
- Crie uma pagina que ira pedir o nome e a idade a idade deve ser mostrada utilizando um select que terá do ano de 1960 ate 2017(utilize-se do laço for)

Exercício

- Crie uma página que peça a cidade de uma pessoa em um select (utilizar do array do link [ArrayPHP](#)) quando clicar em uma cidade deve mostrar o estado que ela pertence

- Efetue um algoritmo em PHP que receba um valor qualquer e imprima os valores de 0 até o valor recebido, exemplo:

Valor recebido = 9

Impressão do programa – 0 1 2 3 4 5 6 7 8 9

Exercício

- Efetue um algoritmo PHP que receba um valor qualquer e calcule o seu fatorial (!), sabendo que fatorial de um número é:

$$7! = 7*6*5*4*3*2*1$$

$$4! = 4*3*2*1$$

- Efetue um algoritmo em PHP que receba um valor qualquer e imprima os valores de 0 até o valor recebido, exemplo:

Valor recebido = 9

Impressão do programa – 0 1 2 3 4 5 6 7 8 9