PHP: Declaração e atribuição de arrays em PHP

Nesta documentação você encontrará o conteúdo que precisa para declarar, atribuir e acessar valores em arrays na linguagem PHP.

Array é um tipo de dado, assim como integer, float, string ou boolean. Contudo, **um array pode armazenar mais de um valor**, relacionando-o a uma chave. No **PHP**, um mesmo array pode conter diferentes tipos de dados, incluindo novos arrays.

Neste documento apresentamos **como trabalhar com arrays em PHP**.

Declaração de `array php`

Para **declarar um array em PHP** utilizamos o construtor de linguagem `array()`, para o qual podemos passar por parâmetro os valores que desejamos armazenar, separados por vírgula, como mostra o exemplo a seguir:

```
$array = array(1, 2, 3);
```

Também é possível iniciar o array utilizando colchetes, como pode ser observado abaixo, pois para o PHP essas duas formas de inicialização são equivalentes:

```
$array = [1, 2, 3];
```

O PHP permite ainda a declaração de arrays associativos. Para esse fim, o construtor `array()` pode receber quais serão as chaves às quais os valores estão associados como parâmetro. Um exemplo dessa sintaxe pode ser visto a seguir:

```
$array = array(
 “chave1” => 1,
 “chave2” => “PHP”,
 “chave3” => false
 );
```

Acessando os índices de um array

Após declarar a variável que contém os valores, podemos acessá-los utilizando sua posição, como mostra o exemplo a seguir:

```
echo $array[0];
echo $array[1];
echo $array[2];
```

O exemplo acima imprime os três **valores contidos no array**. Uma vez que a contagem se inicia em zero, devemos acessar a primeira posição através do índice `$array[0]`, a segunda pelo índice `$array[1]`, e assim em diante.

É possível acessar um valor diretamente através da chave a qual ele está relacionado. O trecho de código a seguir imprime o valor `dois`, contido na segunda posição do array para o qual a chave é `chave2`:

```
echo $array[“chave2”];
```

Também podemos sobrescrever o valor presente em uma posição específica do array utilizando a chave a ele associada. Ao executar o código abaixo, `chave2` vai apontar para o valor numérico `2`, em lugar da palavra `dois`:

```
$array[“chave2”] = 2;
```

Caso a chave seja omitida no **momento da declaração do array**, os valores informados serão associados a índices numéricos sequenciais, como vemos no exemplo a seguir:

```
$array = array(1, “PHP”, false);
```

Sendo assim, para imprimir o valor na segunda posição do array devemos informar a chave `1`, uma vez que a contagem dos índices se inicia em zero:

```
echo $array[1];
```

Percorrendo os índices de um array

Por meio de uma estrutura de repetição, como o `foreach`, podemos percorrer os **dados em um array**. O exemplo a seguir imprime todas as chaves do array `$array`, bem como o valor associado a cada uma delas:

```
foreach($array as $chave => $valor){
    echo “{$chave}: {$valor}\n“;
}
```

Arrays multidimensionais

É possível que um array seja utilizado como valor para outro array. Sendo esse o caso, dizemos que este é um array multidimensional. Abaixo podemos conferir como criar um **array multidimensional em PHP**:

```
$linguagens = array(
    array("PHP", "PHP: Hypertext Preprocessor"),
    array("SQL", "Structured Query Language")
);
```

Então, para acessar os valores presentes nesse array precisamos de dois índices, visto que se trata de um array de duas posições:

```
echo $linguagens[0][1];
```

A partir do primeiro índice `0`, estamos acessando o array na primeira posição de `$linguagens`. Na sequência, acessamos a segunda posição neste array a partir do índice `1`. Ao final de sua execução, o código acima imprime o valor

Exemplo prático

No código a seguir temos um array com os dados de funcionários:

```
$funcionarios = array(
   array("nome" => "Alex", "idade" => 21, "salario" => 1285.27, "ativo" => true),
   array("nome" => "Emerson", "idade" => 35, "salario" => 3885.27, "ativo" => false),
   array("nome" => "Osvaldo", "idade" => 54, "salario" => 5285.27, "ativo" => true),
);

$bonificacao = 10;

foreach($funcionarios as $funcionario){
    if($funcionario["ativo"]){
     $funcionario["salario"] += $funcionario["salario"] * ($bonificacao/100);

     echo "Funcionario: {$funcionario['nome']} - {$funcionario['salario']}\n";
    } else {
     echo "Funcionario: {$funcionario['nome']} - INATIVO\n";
    }
}
```

Linhas 1 a 5: Temos o nosso array com os seguintes dados dos funcionários: nome, idade e se ele se encontra ou não ativo;

Linha 07: Declaramos uma variável chamada bonificação, que é utilizada para acrescentar 10% do valor do salário dos funcionários ativos;

Linhas 09 a 17: Para simplificar o processo de modificação desses valores, percorremos a lista de funcionários em um `foreach`, imprimindo ao final de cada iteração o resultado do cálculo, quando aplicável.