PHP&MYSQL

desenvolvimento web no lado do servidor

JON DUCKETT

ALTA BOOKS
GRUPO EDITORIAL
Rio de Janeiro, 2024

SUMÁRIO

Download do código: http://phpandmysql.com

1

VARIÁVEIS, EXPRESSÕES E OPERADORES

Este capítulo mostra como as variáveis armazenam dados que podem mudar sempre que uma página PHP é solicitada e como expressões e operadores trabalham com valores nas variáveis.

As variáveis usam um nome para representar um valor que pode mudar sempre que uma página PHP é solicitada:

- **Nome** descreve o tipo de dado que a variável contém.
- **Valor** é o que a variável deve armazenar cada vez que a página for solicitada.

Assim que a página termina de ser executada e o HTML retorna para o navegador, o interpretador PHP esquece a variável (portanto, na próxima vez que a página for executada, poderá ter um valor diferente).

O PHP distingue os diferentes tipos de valor que você pode armazenar em uma variável (como texto e números); isso é conhecido como **tipos de dados**:

- Uma parte de um texto se chama **string**.
- Um número sem parte decimal se chama **inteiro**.
- Um número fracionário é representando com um **ponto flutuante**.
- Um valor `true` ou `false` se chama **booleano** ou **lógico**.
- Uma série de nomes e valores afins pode ser armazenada em uma **matriz** (ou **array**, em inglês).

Assim que você aprender sobre as variáveis, verá como as **expressões** podem usar diversos valores para criar outro valor. Por exemplo, o texto em duas variáveis pode ser combinado para formar uma sentença, ou um número armazenado em uma variável pode ser multiplicado por um número em outra.

As expressões contam com **operadores** para criar um valor. Por exemplo, o operador + é usado para somar dois valores e o operador – é para subtrair um valor de outro.


Safari File Edit View History Bookmarks Develop Window Help
Thu 6 Jun 9:05 am
phpandmysql.com
The Candy Store
The Candy Store
MULTI-BUY OFFER
Hello, Ivy.
Buy 5 packs of Chocolate for $20
(usual price $25)
Save $5

VARIÁVEIS

As variáveis armazenam dados que podem mudar (ou variar) sempre que uma página PHP é solicitada. Elas usam um **nome** para representar um **valor** que pode mudar.

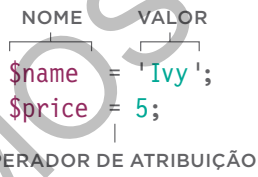
Para criar uma variável e armazenar um valor nela, você precisa de:

- Um **nome** da variável que deve iniciar com cifrão, seguido de um ou mais caracteres alfanuméricos descrevendo o tipo de informação que a variável pode manter.
- Um **sinal de igual** conhecido como **operador de atribuição** porque atribui um valor ao nome da variável.
- O **valor** que você deseja que a variável tenha.

Se a variável mantém texto, o texto é escrito entre aspas. Você pode usar aspas simples ou duplas, mas elas devem combinar (por exemplo, não inicie com aspas simples e termine com aspas duplas).

Se a variável armazena um número ou um valor booleano (`true` ou `false`), não o coloque entre aspas.

Quando uma variável é criada, os programadores chamam isso de **declarar** uma variável. Quando recebe um valor, dizem que um valor está sendo **atribuído** a ela.



Assim que uma variável foi declarada e um valor foi atribuído a ela, o nome da variável poderá ser usado no código PHP sempre que você desejar usar o valor que a variável mantém atualmente.

Quando o interpretador PHP encontra um nome de variável, ele substitui o nome pelo valor mantido. Abaixo, o comando `echo` é usado para exibir o valor armazenado na variável `$name` mostrada acima.

CRIANDO E ACESSANDO VARIÁVEIS

PHP section _ a/c01/variables.php

```
<?php
① $name  = 'Ivy';
② $price = 5;
?>
<!DOCTYPE html>
<html>
  <head>
    <title> Variables </title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <h1>The Candy Store</h1>
③   <h2>Welcome <?php echo $name; ?></h2>
    <p>The cost of your candy is
④     $<?php echo $price; ?> per pack.</p>
  </body>
</html>
```

RESULTADO

Neste capítulo, os valores são atribuídos às variáveis no código PHP. Nos capítulos posteriores, os valores atribuídos às variáveis virão de formulários HTML que os visitantes enviam, de dados em URLs e de bancos de dados.

Neste exemplo, duas variáveis são criadas e os valores são atribuídos no topo da página:

1. `$name` mantém o nome do visitante atual no site. É texto, portanto é escrito entre aspas.
2. `$price` mantém o preço de um pacote de doce. É um número, portanto o valor não fica entre aspas.

Em seguida, veja o HTML retornado para o navegador do visitante. No HTML:

3. O nome do visitante é escrito na página usando o comando `echo`.
4. O custo do doce é escrito na página.

EXPERIMENTE: na Etapa 1, mude o valor da variável `$name` para manter seu nome. Salve o arquivo e atualize a página no navegador. Seu nome será exibido.

EXPERIMENTE: na Etapa 2, mude o preço para 2. Salve o arquivo e atualize a página. O novo preço será mostrado.

COMO NOMEAR AS VARIÁVEIS

O nome de uma variável deve descrever os dados que ela armazena. Use as seguintes regras para criar um nome de variável.

1

Comece com um cifrão ($).

```
$greeting
greeting
```

2

Siga com uma letra ou um sublinhado (não um número).

```
$greeting
$2_greeting
```

3

Então use uma combinação de letras A-z (maiúsculas e minúsculas), números e sublinhados (traços ou pontos não são permitidos).

```
$greeting_2
$greeting-2
$greeting.2
```

Nota: $this tem um significado especial. Não use como um nome de variável.

```
$this
```

Usar nomes de variável que descrevem os dados que sua variável armazena facilita que o código seja entendido e seguido.

Se você usar mais de uma palavra para descrever os dados que uma variável mantém, é comum separar cada palavra com um sublinhado.

Os nomes levam em conta letras maiúsculas e minúsculas, portanto `$Score` e `$score` seriam nomes diferentes. Mas, em geral, evite criar duas variáveis que usam a mesma palavra e combinações de letras maiúsculas/minúsculas diferentes, pois provavelmente confundirão as outras pessoas que lerem seu código.

Tecnicamente, você pode usar caracteres de diferentes conjuntos (como caracteres chineses e cirílicos), mas é considerada uma boa prática usar apenas letras A-z, números e sublinhados (pois há complicações no suporte de outros caracteres).

TIPOS DE DADOS ESCALARES (BÁSICOS)

O PHP diferencia três **tipos de dados escalares** que mantêm texto, números e valores booleanos.

TIPO DE DADO DE STRING

Os programadores chamam uma parte de texto de **string**. O tipo de dados `string` pode consistir em letras, números e outros caracteres, mas é usado para representar texto.

```
$name = 'Ivy';
```

As strings ficam sempre entre aspas simples ou duplas. A aspa de abertura deve corresponder à de fechamento.

```
$name = 'Ivy';
$name = "Ivy";
$name = "Ivy';
$name = 'Ivy";
```

TIPO DE DADO NUMÉRICO

Os tipos de dados numéricos permitem realizar operações matemáticas com valores que eles mantêm, como adição ou multiplicação.

```
$price = 5;
```

Os números não são escritos entre aspas. Se você os colocar entre aspas, eles poderão ser tratados como strings, em vez de números.

O PHP tem dois tipos de dados numéricos:

`int` representa inteiros (ex.: 275).

`float` mantém números de ponto flutuante, que representam frações (ex: 2,75).

TIPO DE DADO BOOLEANO

O tipo de dado booleano pode ter apenas um destes dois valores: `true` ou `false`. Esses valores são comuns na maioria das linguagens de programação.

```
$logged_in = true;
```

`true` e `false` devem ser escritos com letra minúscula e não ficam entre aspas. De início, os valores booleanos parecem abstratos, mas muitas coisas podem ser representadas com true ou false, como:

- Um visitante fez login?
- Ele concordou com os termos e as condições?
- Um produto se qualifica para o envio gratuito?

TIPO DE DADO NULL

O PHP também tem um tipo de dado chamado `null`, que pode ter apenas o valor `null`. Isso indica que o valor não foi especificado para uma variável.

MANIPULAÇÃO DE TIPOS

Nas páginas 60-61, veremos como o interpretador PHP pode converter um valor de um tipo de dado em outro (ex: uma string em um número).

ATUALIZANDO UM VALOR EM UMA VARIÁVEL

É possível mudar ou sobregravar o valor armazenado em uma variável atribuindo-lhe um novo valor. Isso é feito do mesmo modo como um valor é atribuído a uma variável quando ela é criada.

1. A variável `$name` é **inicializada**. Isso significa que ela é declarada e atribuída a um valor inicial, que será usado se a variável não for atualizada posteriormente na página.

O valor inicial é `Guest` e deve ser escrito entre aspas, pois é texto.

2. A variável `$name` é atribuída a um novo valor `Ivy`.

3. A variável `$price` mantém o preço de um pacote de doce.

Em seguida, você pode ver o HTML que será retornado para o navegador do visitante. No HTML:

4. O nome é escrito na página com o comando `echo`. Ele mostra o valor atualizado que foi atribuído à variável `$name` na Etapa 2.

5. O custo do doce é escrito na página.

section _ a/c01/updating-variables.php — PHP

```
<?php
① $name  = 'Guest';
② $name  = 'Ivy';
③ $price = 5;
?>
<!DOCTYPE html>
<html>
  <head>
    <title>Updating Variables</title>
    <link rel="stylesheet" href="css/styles.css">
  </head>
  <body>
    <h1>The Candy Store</h1>
④   <h2>Welcome <?php echo $name; ?></h2>
    <p>The cost of your candy is
⑤      $<?php echo $price; ?> per pack.</p>
  </body>
</html>
```

RESULTADO

EXPERIMENTE: na Etapa 2, mude o valor da variável `$name` para manter seu nome. Salve o arquivo e atualize a página no navegador. Seu nome será exibido.

EXPERIMENTE: adicione uma nova linha após a Etapa 2 e defina a variável `$name` para manter outro nome. Salve o arquivo e atualize a página. O novo nome será mostrado.

Uma variável também pode manter um **array**, que armazena uma série de valores afins. Os arrays são conhecidos como um **tipo de dado composto** porque podem armazenar mais de um valor.

Um array é como um contêiner que mantém um conjunto de variáveis afins. Cada item no array se chama **elemento**. Do mesmo modo como uma variável usa um nome para representar um valor, cada elemento em um array tem:

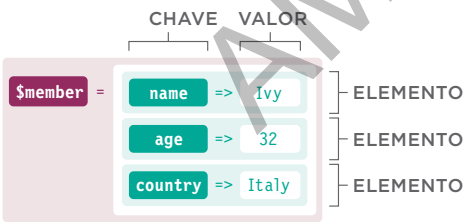
- Uma **chave**, que age como o nome da variável.
- Um **valor**, que é o dado que o nome representa.

O PHP tem dois tipos de array:

- Nos **arrays associativos**, a chave de cada elemento é um nome que descreve os dados que ele representa.
- Nos **arrays indexados**, a chave de cada elemento é um número conhecido como **número do índice**.

ARRAY ASSOCIATIVO

O array abaixo é para manter dados que representam um membro do site. Sempre que o array for usado, os nomes utilizados nas chaves (que descrevem os dados armazenados em cada elemento do array) continuarão iguais.

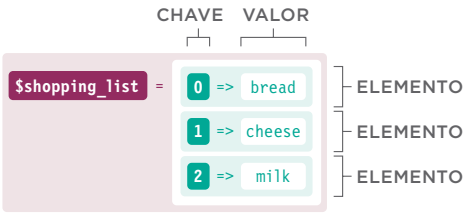


Nos dois exemplos, cada valor armazenado no array é um tipo de dado escalar (uma parte individual do dado).

Na página 44, veja exemplos de arrays em que um elemento do array mantém outro array.

ARRAY INDEXADO

O array abaixo é para manter uma lista de compras. Listas assim podem manter um número diferente de elementos sempre que são usadas. A chave não usa um nome para descrever cada item na lista, mas um número do índice (que é um inteiro e inicia em 0).



NOTA: os números do índice iniciam em 0, não em 1. O primeiro elemento na lista tem um número de índice 0. O segundo elemento é identificado pelo número de índice 1, e daí por diante. O número de índice costuma ser usado para descrever a ordem dos itens na lista.

ARRAYS ASSOCIATIVOS

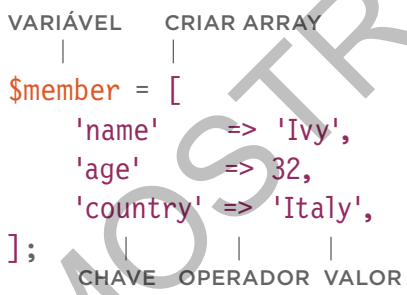
Para criar um array associativo, dê a cada elemento (ou item) no array uma **chave** que descreve os dados que ele mantém.

Para armazenar um array associativo, use:

- Um nome de variável que descreve o conjunto de valores que o array manterá.
- O operador de atribuição.
- Colchetes para criar o array.

Entre chaves ou parênteses, use:

- O nome da chave entre aspas.
- O operador de seta dupla =>.
- O valor desse elemento (pode strings entre aspas; números e booleanos, não pode).
- Uma vírgula após cada elemento.



Um array associativo também pode ser criado usando a sintaxe mostrada abaixo, com a palavra `array` seguida de parênteses (em vez de colchetes).

```
$member = array(
  'name'    => 'Ivy',
  'age'     => 32,
  'country' => 'Italy',
);
```

Para acessar um elemento no array associativo, use:

- O nome da variável que mantém o array.
- Seguido por colchetes e aspas.
- A chave do elemento que você deseja recuperar.

CRIANDO E ACESSANDO ARRAYS ASSOCIATIVOS

PHP section _ a/c01/associative-arrays.php

```php
<?php
$nutrition = [
    'fat'   => 16,
    'sugar' => 51,
    'salt'  => 6.3,
];
?>
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body>
    <h1>The Candy Store</h1>
    <h2>Nutrition (per 100g)</h2>
    <p>Fat:   <?php echo $nutrition['fat']; ?>%</p>
    <p>Sugar: <?php echo $nutrition['sugar']; ?>%</p>
    <p>Salt:  <?php echo $nutrition['salt']; ?>%</p>
  </body>
</html>
```

RESULTADO

1. Neste exemplo, um array associativo é criado e armazenado em uma variável chamada `$nutrition`.

O array é criado entre colchetes e tem três elementos (cada um tem um par de chave/valor). O operador => atribui valores a cada uma das chaves.

2. Para exibir os dados armazenados no array, use:
   - O comando `echo` para indicar que o valor a seguir deve ser escrito na página web.
   - Seguido pelo nome da variável que mantém o array.
   - E então por colchetes e aspas, com o nome da chave que você deseja acessar.

Por exemplo, para escrever o conteúdo sugar na página, use:

```
echo $nutrition['sugar'];
```

EXPERIMENTE: na Etapa 1, mude os valores do array. Forneça a chave:

- `fat` com um valor 42.
- `sugar` com um valor 60.
- `salt` com um valor 3.5.

Salve e atualize a página para ver os valores atualizados.

EXPERIMENTE: na Etapa 1, adicione outro elemento ao array. Use a chave `protein` e atribua um valor `2.6` a ela. Então, na Etapa 2, mostre o valor protein na página.

ARRAYS INDEXADOS

Quando um array é criado, se uma chave não for fornecida para cada elemento, o interpretador PHP atribuirá um número chamado **número do índice**. Esses números iniciam em zero (0), não em (1).

Para armazenar um array indexado em uma variável, use:

- Um nome da variável que descreve o conjunto de valores que o array manterá.
- O operador de atribuição.
- Colchetes para criar o array.

Entre colchetes ou parênteses, use:

- A lista de valores que o array deve manter (strings com aspas, números e booleanos sem).
- Uma vírgula após cada valor.

Cada elemento será atribuído a um número do índice.

Acima, bread seria atribuído ao número do índice 0, cheese a 1 e milk a 2. Números do índice são usados para indiciar a ordem dos itens listados no array.

Um array indexado também pode ser criado usando a sintaxe mostrada abaixo, com a palavra `array` seguida por parênteses (não colchetes).

```
$shopping _ list = array('bread',
                         'cheese',
                         'milk');
```

Cada valor adicionado ao array pode estar na mesma linha ou em uma nova linha (como mostrado acima).

Para acessar os itens em um array indexado, use:

- O nome da variável que mantém o array.
- Seguido por colchetes (sem aspas).
- O número de índice do item que você deseja acessar (entre colchetes).

O código abaixo obtém o terceiro item no array, portanto, neste exemplo, obteria o valor `milk`.

CRIANDO E ACESSANDO ARRAYS INDEXADOS

PHP section _ a/c01/indexed-arrays.php

```php
<?php
$best _ sellers = ['Chocolate', 'Mints', 'Fudge',
    'Bubble gum', 'Toffee', 'Jelly beans',];
?>
<!DOCTYPE html>
<html>
  <head> ... </head>
  <body>
    <h1>The Candy Store</h1>
    <h2>Best Sellers</h2>
    <ul>
      <li><?php echo $best_sellers[0]; ?></li>
      <li><?php echo $best_sellers[1]; ?></li>
      <li><?php echo $best_sellers[2]; ?></li>
    </ul>
  </body>
</html>
```

RESULTADO

1. Este exemplo começa criando uma variável chamada `$best _ sellers`. Seu valor é um array que mantém uma lista dos itens mais vendidos no site.

O array é criado usando colchetes e os itens são adicionados ao array entre esses colchetes. Como os itens no array são texto, ficam entre aspas (números e booleanos não ficariam entre aspas). Cada item é seguido por uma vírgula.

2. Os três itens mais vendidos são escritos na página:
   - O comando `echo` indica que o valor seguinte deve ser escrito.
   - Seguido pelo nome da variável que mantém o array.
   - E então os colchetes que mantêm o número de índice do item que você deseja recuperar. Lembre-se de que os números do índice iniciam em 0, não em 1.

EXPERIMENTE: na Etapa 1, adicione Licorice ao array após Fudge. Na Etapa 2, adicione o quarto e o quinto itens ao array.