

PHP - aula 4

Conteúdo

Strings

Constantes

Tamanho da string

A função `strlen` retorna o tamanho da string:

```
echo strlen("abacaxi");
```

Tamanho da string

Pensando em senhas (tamanho de senhas) com `strlen`

Quantidade de palavras em string

A função **str_word_count()** conta a quantidade de palavras em uma string:

```
echo str_word_count("Uma frase qualquer");
```

Quantidade de palavras em string

Pensando o nome completo de uma pessoa em um formulário.

String revertida

A função `strrev()` reverte uma string:

```
echo strrev("Uma frase embaralhada");
```

String revertida

Pensando um sistema de perguntas e respostas.

Buscando textos

A função `strpos()` busca um determinado texto em uma string

```
echo strpos("Hello world!", "world");
```

Retorna a posição do primeiro caracter ou `false` se não encontrar.

Buscando textos

Pensando como funciona o Google

Trocar palavras em uma string

A função `str_replace()` um texto por outro dentro de uma string.

```
echo str_replace("world", "Barretos", "Hello world!");
```

Constantes

Em PHP uma constante é um nome para um valor.

O valor não pode ser alterado.

Nomes de constantes não começam com \$

A função `define()` cria uma constante.

Constantes

A função `define()` cria uma constante.

```
define("pi", 3.14, true)
```

```
echo pi;
```

Constantes

A função `define()` cria uma constante.

```
define("pi", 3.14, false) // false é o valor padrão.
```

```
echo pi;
```

Vetores constantes

As constantes também podem ser vetores:

```
define("carros", [  
    "Fusca",  
    "Parati",  
    "Opala"  
]);
```

```
echo carros[1];
```

Constantes são globais

Você pode usar o nome de uma constante no código PHP inteiro.

Operadores aritméticos

+ adição

- subtração

* multiplicação

/ divisão

% módulo

** exponenciação

Operadores de atribuição

$x = y$ x recebe o valor de y

$x += y$ $x = x + y$ x recebe o valor de $x + y$

$x -= y$ $x = x - y$ x recebe o valor de $x - y$

$x *= y$ $x = x * y$ x recebe o valor de $x * y$

$x /= y$ $x = x / y$ x recebe o valor de x / y

$x \% = y$ $x = x \% y$ x recebe o valor de $x \% y$

Operadores de comparação

==	<code>\$x == \$y</code>	true se \$x e \$y iguais, caso contrário false
===	<code>\$x === \$y</code>	true se \$x e \$y são idênticos, mesmo tipo também.
!=	<code>\$x != \$y</code>	true se \$x e \$y são diferentes
<>	<code>\$x <> \$y</code>	true se \$x e \$y são diferentes
!==	<code>\$x !== \$y</code>	true se \$x e \$y não são iguais ou não são do mesmo tipo
>	<code>\$x > \$y</code>	true se \$x é maior que \$y

Operadores de comparação

< $\$x < \y true se $\$x$ é menor que $\$y$

>= $\$x \geq \y true se $\$x$ é maior ou igual a $\$y$

<= $\$x \leq \y true se $\$x$ é menor ou igual a $\$y$

<=> $\$x \lt=> \y retorna -1 se $\$x$ é menor que $\$y$, retorna 0 se $\$x$ é igual a $\$y$, retorna 1 se $\$x$ é maior que $\$y$

Operador de incremento

`++$x` pré-incremento incrementa `$x` de 1 e retorna `$x`

`$x++` pós-incremento retorna o valor de `$x` depois incrementa em 1.

Operador de decremento

--\$x	pré-decremento	decrementa \$x de 1 e retorna \$x
\$x-	pós-decremento	retorna o valor de \$x depois decrementa em 1.

Operadores lógicos

Esses operadores são usados para combinar condições

and	<code>\$x and \$y</code>	true se <code>\$x</code> e <code>\$y</code> são true
or	<code>\$x or \$y</code>	true se <code>\$x</code> ou <code>\$y</code> são true
xor	<code>\$x xor \$y</code>	true se <code>\$x</code> é true ou se <code>\$y</code> é true, mas não ambos
&&	<code>\$x && \$y</code>	true se <code>\$x</code> e <code>\$y</code> são true
	<code>\$x \$y</code>	true se <code>\$x</code> ou <code>\$y</code> são true
!	<code>!\$x</code>	true se <code>\$x</code> é false

Operadores de string

. \$txt1 . \$txt2 concatena \$txt1 e \$txt2

.= \$txt1 .= \$txt2 concatena \$txt1 e \$txt2 e salva em \$txt1

Operadores de vetor

`+` `$x + $y` une `$x` e `$y`

`==` `$x == $y` retorna true se `$x` e `$y` possuem as mesmas
chaves/valores

`===` `$x === $y` retorna true se `$x` e `$y` possuem as mesmas
chaves/valores, na mesma ordem e com mesmos tipos

`!=` `$x != $y` retorna true se `$x` não é igual a `$y`

`<>` `$x <> $y` retorna true se `$x` não é igual a `$y`

`!==` `$x !== $y` retorna true se `$x` não é idêntico a `$y`

Operadores de atribuição condicional

?: \$x = expr1 ? expr2 : expr3 se expr1 é true atribui expr2 à \$x, ou atribui expr3 a \$x, retorna sempre \$x

?? \$x = expr1 ?? expr2 se expr1 é diferente de NULL atribui expr1 a \$x, se expr1 não existir ou for null, o valor de \$x é expr2. retorna o valor de \$x.

