

Tipos de Dados Suportados por PHP MBADSPHP - Tópico 2 / Aula 4

Objetivo

O objetivo desta aula é apresentar os tipos de dados existentes na linguagem PHP, e como se pode fazer a transformação de tipos de dados.

Conteúdo

A linguagem PHP suporta os tipos de dados inteiro, ponto flutuante, string, *array* e objeto. Como a checagem das variáveis é dinâmica, não é necessário declarar inicialmente o tipo da variável. Isso significa que uma variável pode receber um ponto flutuante e logo em seguida receber um valor booleano.

Inteiro: tipo *int* ou *long*. Para informar ao interpretador que a variável é inteira, basta atribuir um valor inteiro a ela. Ex: `$soma = 1023;`

Ponto flutuante: *double* ou *float*. Para informar ao interpretador que a variável é um *float*, basta atribuir um valor real a ela. Ex: `$resultado = 12.34;`

String: Para informar ao interpretador que a variável é uma string, existem duas formas diferentes: utilizando ' ' ou " ". O primeiro caso o valor da variável é exatamente o que existe dentro das ' '. Ex: `$nome = 'Fulano';` O segundo caso o valor da variável também é exatamente o que existe dentro das " ". Ex: `$sobrenome = "de tal".` A exceção aqui é que caso exista uma variável entre as " ", substitui-se o nome da variável pelo conteúdo da mesma.

Array: vetores. Para informar ao interpretador que a variável é um *array*, basta atribuir um valor qualquer a ela, informando também em que posição do *array* a variável está. Importante observar que pode-se atribuir tipos diferentes no mesmo *array*. Ex: `$vetor[0] = 1023; $vetor[1] = 34.6; $vetor[2] = 'casa'.`

Lista: As listas na linguagem PHP são utilizadas para executar atribuições múltiplas. Assim, com listas é possível atribuir valores que estão num *array* para variáveis. Ex: `list($x, $y, $z) = array(1,2,3);`

Objeto: Para informar ao interpretador que a variável é um objeto, deve-se utilizar o comando `new` juntamente com o nome da classe do objeto. Ex:

```
class exemplo {  
    function testa() {  
        echo "<p>teste";  
    }  
}
```

```
$objeto = new exemplo;
```

```
$objeto -> testa();
```

A **transformação de tipos de dados** pode ser feita de 3 maneiras: utilizando **coerção**, utilizando transformação explícita, ou ainda com a função **settype**. A **coerção** ocorre quando ocorrem determinadas operações ("+", por exemplo) entre dois valores de tipos diferentes, e o PHP converte o valor de um deles automaticamente. Exemplo: Fique atento aos comentários no código.

```
$exemplo = "5";
```

```
$exemplo = $ exemplo + 1; // $ exemplo é o integer 6
```

```
$exemplo = $ exemplo + 3.7; // $ exemplo é o double 9.7
```

```
$exemplo = 6 + 4.5 // $ exemplo é o double 10.5
```

A **transformação explícita** ocorre quando o tipo a ser transformado é colocado entre parênteses durante a atribuição. Exemplo: Fique atento aos comentários no código.

```
$teste = 24.5;
```

```
$teste = (int) $teste; // $teste contém 24
```

Finalmente, com a função **settype**. Exemplo: Fique atento aos comentários no código.

```
$teste = 50.4; // $teste é double
```

```
settype($teste,integer) // $teste é inteiro
```

Exercícios

Faça um programa PHP que teste todos os recursos aprendidos neste Tópico. Sempre apresente o resultado ao usuário com o comando `echo ""`. Lembre-se de colocar o arquivo PHP no diretório `htdocs` do Apache HTTP Server.