

1 Variables en PHP

1.1 Sumario

- 1 Conceptos básicos
- 2 Variables predefinidas
- 3 Existencia das variables
- 4 Obter información sobre o tipo dunha variable
- 5 Modificar as variables
- 6 Ámbito das variables

1.2 Conceptos básicos

En PHP todas as variables comezan co símbolo do dólar \$ e non é necesario definir unha variable antes de usala.

Ademais, saber que, o nome da variable é sensible a minúsculas e maiúsculas.

Un nome de variable válido ten que comezar cunha letra ou raia (underscore), seguido de calquera número de letras, números e raías.

Tampouco teñen tipos, é dicir que unha mesma variable pode conter un número e logo pode conter caracteres.

Vexamos un exemplo:

```
<html>
<head>
  <title>Exemplo de PHP</title>
</head>
<body>
<?php
  $a = 1;
  $b = 3.34;
  $c = "Ola Mundo";
  echo $a, "<br>", $b, "<br>", $c;
?>
</body>
</html>
```

Neste exemplo definimos tres variables, \$a , \$b e \$c e coa instrución echo imprimimos o valor que contiñan, inserindo un salto de liña entre elas.

Existen dous tipos de variables, as variables locais que só poden ser usadas dentro de funcións e as variables globais que teñen o seu ámbito de uso fóra das funcións, podemos acceder a unha variable global dende unha función coa instrución global nome_variable;

1.3 Variables predefinidas

PHP proporciona unha serie de variables predefinidas contendo información do servidor web, o entorno e as entradas do usuario.

Estas variables predefinidas son globais por defecto.

A lista completa podémola atopar no enlace [Variables Predefinidas Reservadas](#). Algunhas destas variables son as seguintes:

- **\$_GLOBALS** : Contén unha referencia a cada variable existente no script.
- **\$_SERVER** : é unha matriz que contén información tal como cabeceiras, rutas e ubicacións de scripts. As entradas desta matriz son creadas polo servidor web.
- **\$_GET** : Unha matriz asociativa de variables pasadas ao script actual a través do método HTTP GET.
- **\$_POST** : Unha matriz asociativa de variables pasadas ao script actual a través do método HTTP POST.
- **\$_FILES** : Unha variable tipo array asociativo de elementos cargados ao script actual a través do método POST.
- **\$_COOKIE** : Unha variable tipo array asociativo de variables pasadas ao script actual a través de Cookies HTTP.

1.4 Existencia das variables

Se queremos comprobar se unha variable existe, é dicir, está xa definida ou só ten un valor nulo, para iso temos as seguintes funcións:

- **empty()** : Determina se unha variable está baleira.
- **isset()** : Determina se unha variable está definida.
- **is_null()** : Determina se unha variable ten valores nulos.

URL con información e explicación máis detallada das diferencias entre estas tres funcións:

http://www.virendrachandak.com/techtalk/php-isset-vs-empty-vs-is_null/

1.5 Obter información sobre o tipo dunha variable

Moitas veces precísase de que tipo son as variables. PHP dispón dun conxunto de funcións que comezan por `is_` e devolven verdadeiro se o tipo de variable é o que se indica.

Así temos as seguintes funcións:

- `is_array()`
- `is_bool()`
- `is_float()`
- `is_int()`
- `is_object()`
- `is_resource()`
- `is_scalar()`
- `is_string()`
- `is_null()`
- `is_numeric()`

Por último, temos a función **gettype()**, que devolve directamente o tipo da variable que se lle pasa como argumento. Os tipos posibles son integer, double, string, array, object e unknown (descoñecido).

1.6 Modificar as variables

Para asignar un tipo concreto a unha variable PHP ten a función `settype()`.

Así, por exemplo, se queremos que a variable `$cantidade` sexa do tipo integer debemos escribir: `settype($cantidade,"integer")`.

Moitas veces, unha variable de cadea pode conter letras e números.

PHP dispón de varias funcións que permiten separar os números do texto e asignalos ás variables correspondentes do seu tipo, estas son:

- **floatval()** - Obtén o valor flotante dunha variable.
- **intval()** - Obtén o valor enteiro dunha variable.
- **strval()** - Obtén o valor de cadea dunha variable.

Se vemos que dentro dun script php se consumen innecesariamente moitos recursos por variables definidas que xa non se usan, podemos empregar a **función unset()**, que elimina unha variable definida previamente.

Por exemplo, se queremos que a variable `$nome` xa non estea definida podemos escribir **unset(\$nome)**.

1.7 Ámbito das variables

O ámbito das variables en PHP é o contexto onde foi definida dita variable. A maior parte das variables soamente teñen ámbito simple. Este ámbito abarca os arquivos incluídos e requeridos.

Dentro das funcións as variables declaradas terán o ámbito local da función.

Si queremos acceder a unha variable global dentro da función teremos que especificalo claramente empregando **global**.

ATENCIÓN: En PHP as variables globales por defecto non son accesibles dentro das funcións a non ser que especifiquemos claramente global antes da variable.

Por exemplo: **global \$nome;** // Desde xeito poderíamos acceder á variable global **\$nome** dentro da función.

[Máis información sobre ámbito das variables.](#)