

1 Curso POO PHP Variables e tipos de datos

1.1 Variables e tipos de datos

As variables en PHP sempre deben comezar polo signo "\$". Os nomes das variables deben comezar por letras ou polo carácter "_", e poden conter tamén números. Non obstante, ao contrario que en moitas outras linguaxes, en PHP non é necesario declarar unha variable nin lle especificar un tipo (enteiro, cadea,?) concreto. Para empezar a usar unha variable, simplemente asígnalle un valor utilizando o operador =.

Dependendo do valor que se lle asigne, á variable aplícase internamente un tipo de datos, que pode cambiar se cambia o seu contido. Isto é, o tipo da variable decídese en función do contexto en que se empregue.

Os tipos de datos en PHP son:

- **booleano** (boolean). Os seus posibles valores son true e false. Ademais, calquera número enteiro se considera como true, salvo o 0 que é false.
- **enteiro** (integer). Calquera número sen decimais. Pódense representar en formato decimal, octal (comezando por un 0), ou hexadecimal (comezando por 0x).
- **real** (float). Calquera número con decimais. Pódense representar tamén en notación científica.
- **cadea** (string). Conxuntos de caracteres delimitados por comiñas simples ou dobres.
- **NULL**. É un tipo de datos especial, que se usa para indicar que a variable non ten valor.
- **array**. É un grupo de valores que se poden identificar por un índice (un número comezando en 0 que indica a súa posición) ou un nome que o identifique (array asociativo).
- **obxecto**. PHP soporta POO (Programación Orientada a Obxectos). Os obxectos en PHP instáncianse a partir da clase empregando a palabra clave new.
- **recurso**. Empréganse para facer referencia a recursos como poden ser as conexións a bases de datos. PHP mantén unha táboa coas referencias abertas a cada recurso, de xeito que cando se borra a última, se liberan de xeito automático as estruturas reservadas para o recurso.

Se realizas unha operación con variables de distintos tipos, ambas as dúas convértense primeiro a un tipo común. Por exemplo, se sumas un enteiro cun real, o enteiro se converte a real antes de realizar a suma. Estas conversións de tipo, que normalmente se levan a cabo de forma automática, tamén se poden **realizar de forma forzada**, empregando os operadores de conversión de tipos:

- **(int)** ou **(integer)** para converter a enteiro.
- **(bool)** ou **(boolean)** para converter a booleano.
- **(float)**, **(double)** ou **(real)** para converter a real.
- **(string)** para converter a cadea.
- **(array)** para converter a array.
- **(object)** para converter a obxecto.

En PHP existen funcións específicas para comprobar e establecer o tipo de datos dunha variable. **gettype** obtén o tipo da variable que se lle pasa como parámetro e devolve unha cadea de texto, que pode ser "array", "boolean", "double", "integer", "object", "string", "null", "resource" ou "unknown type".

Tamén podemos comprobar se a variable é dun tipo concreto utilizando unha das seguintes funcións: **is_array**, **is_bool**, **is_float**, **is_integer**, **is_null**, **is_numeric**, **is_object**, **is_resource**, **is_scalar** e **is_string**. Devolven true se a variable é do tipo indicado.

Analogamente, para establecer o tipo dunha variable utilizamos a función **settype** pasándolle como parámetros a variable a converter, e unha das seguintes cadeas: boolean, integer, float, string, array, object ou null. A función settype devolve true se a conversión se realizou correctamente, ou false en caso contrario.

Se o único que che interesa é saber se unha variable está definida e non é null, podes usar a función **isset**. A función **unset** destrúe a variable ou variables que se lle pasa como parámetro.

Unha variable pode estar definida pero estar baleira (isto é, contén false, 0, unha cadea baleira ou semellante). Podes empregar a función **empty** para comprobar se unha variable está baleira.

1.1.1 Constantes

Existe tamén en PHP unha función, **define**, coa que podes definir constantes, isto é, identificadores aos que se lles asigna un valor que non cambia

durante a execución do programa.

```
define ("OLA", "Ola mundo.");
```

Os identificadores das constantes non van precedidos polo signo "\$" e adoitan escribirse en maiúsculas, aínda que existe un terceiro parámetro opcional, que se vale true fai que se recoñeza o identificador independentemente de se está escrito en maiúsculas ou en minúsculas.

A partires da versión 5.3 da linguaxe, permítese tamén empregar a palabra **const** de xeito xeral para definir constantes.

```
const OLA = "Ola mundo.";
```

Só se permiten os seguintes tipos de valores para as constantes: integer, float, string, boolean e null.

--Víctor Lourido 14:41 25 jun 2013 (CEST)