

Tipos de variables

Tipos de variables

En PHP **no es necesaria** una definición previa del tipo de variables.

Según los valores que se les vayan asignando, las variables podrán cambiar de tipo –de modo automático– y se irán adaptando a los valores que contengan en cada momento.

Las variables en PHP pueden ser de tres tipos:

- **Enteras** (tipo *Integer*)
- **De coma flotante** (tipo *Double*)
- **Cadenas** (tipo *String*)

Cualquier número entero cuyo valor esté comprendido entre $\pm 2^{31}$ será interpretado por PHP como de tipo **Integer**.

Si el valor de una variable es un *número decimal* o –siendo entero– *desborda* el intervalo anterior, bien por asignación directa o como resultado de una operación aritmética, PHP la convierte a tipo **Double**.

Cualquier variable a la que se le asigne como valor el contenido de una cadena de caracteres (letras y/o números delimitados por comillas) es interpretada por PHP como tipo **String**.

Determinación de tipos de variables

PHP dispone de la función:

gettype(variable)

que *devuelve una cadena de caracteres* indicando el tipo de la variable que contiene.

La cadena devuelta por esta función puede ser: **Integer**, **double** o **string**.

Forzado de tipos

PHP permite *forzar* los tipos de las variables. Eso quiere decir que se puede **obligar** a PHP a asignar *un tipo determinado a una variable determinada*, siempre que los valores que contenga **estén dentro del rango** del nuevo tipo de variable.

En el cuadro siguiente podemos ver los tres tipos de variables que utiliza PHP.

Las variables en PHP				
Tipo	Ejemplo	Valor máximo	Valor mínimo	Observaciones
Integer	\$a=1234	2147483647	-2147483647	Cualquier valor numérico entero (dentro de este intervalo) que se asigne a una variable será convertido a este tipo
Double	\$a=1.23	Cualquier valor numérico decimal, o entero fuera del intervalo anterior, que se asigne a una variable la convertirá a este tipo		
String	\$a="123"	Cualquier valor <i>entrecomillado</i> (sean números o letras) que se asigne a una variable la convertirá a este tipo		

Determinación del tipo de variable utilizada

Dado que PHP gestiona las variables de forma automática y modifica los tipos de acuerdo con los valores que va tomando durante la ejecución del *script*, se puede recurrir a la función **gettype(nombre de la variable)** para determinar el **tipo de la variable actual**.

En la tabla siguiente tienes algunos ejemplos de aplicación de esa función.

Podemos observar –en la columna *Sintaxis*– que para *visualizar* el resultado anteponemos **echo** a **gettype**. Es decir, le indicamos a PHP que *muestre* el resultado obtenido al determinar el tipo de variable.

Ejemplos de determinación del tipo de una variable		
Variable	Sintaxis	Devuelve
\$a1=347	echo gettype(\$a1)	integer
\$a2=2147483647	echo gettype(\$a2)	integer
\$a3=-2147483647	echo gettype(\$a3)	integer
\$a4=23.7678	echo gettype(\$a4)	double
\$a5=3.1416	echo gettype(\$a5)	double
\$a6="347"	echo gettype(\$a6)	string
\$a7="3.1416"	echo gettype(\$a7)	string
\$a8="Solo literal"	echo gettype(\$a8)	string
\$a9="12.3 Literal con número"	echo gettype(\$a9)	string
\$a10=""	echo gettype(\$a10)	string

Forzado de tipos

Aquí tienes algunos ejemplos de *forzado de tipos*. Te sugerimos que *eches un vistazo* a las advertencias que hemos puesto después de esta tabla.

Forzado de tipos		
Variable	Sintaxis	Devuelve
\$a1=347	echo gettype((real)\$a1)	double
\$a2=2147483647	echo gettype((double)\$a2)	double
\$a3=-2147483647	echo gettype((float)\$a3)	double
\$a4=23.7678	echo gettype((int)\$a4)	integer
\$a5=3.1416	echo gettype((integer)\$a5)	integer
\$a6="347"	echo gettype((double)\$a6)	double
\$a7="3.1416"	echo gettype((int)\$a7)	integer
\$a7="3.1416"	echo gettype((string)\$a7)	string

<code>\$a8="Solo literal"</code>	<code>echo gettype((double)\$a8)</code>	double
<code>\$a9="12.3 Literal con número"</code>	<code>echo gettype((int)\$a9)</code>	integer

¡Cuidado!

Al modificar los tipos de variables pueden modificarse sus valores. Si forzamos a entera una variable que contenga un número decimal se perdería la parte decimal y la variable modificada solo contendría el valor de la parte entera. Si tratamos de convertir a numérica una variable alfanumérica el nuevo valor sería cero.

Aquí tienes algunos ejemplos relacionados con la advertencia anterior

Nuevos valores de la variable		
Valor inicial	Sintaxis	Nuevo valor
<code>\$a1=347</code>	<code>echo ((real)\$a1)</code>	347
<code>\$a2=2147483647</code>	<code>echo ((double)\$a2)</code>	2147483647
<code>\$a3=-2147483647</code>	<code>echo ((float)\$a3)</code>	-2147483647
<code>\$a4=23.7678</code>	<code>echo ((integer)\$a5)</code>	23
<code>\$a5="3.1416"</code>	<code>echo ((double)\$a6)</code>	3.1416
<code>\$a6="347"</code>	<code>echo ((int)\$a7)</code>	347
<code>\$a7="3.1416"</code>	<code>echo ((string)\$a7)</code>	3.1416
<code>\$a8="Solo literal"</code>	<code>echo ((int)\$a8)</code>	0
<code>\$a9="12.3 Literal con número"</code>	<code>echo ((double)\$a9)</code>	12.3
<code>\$a10=""</code>	<code>echo ((int)\$a9)</code>	0

Los tipos se pueden *forzar* tanto en el momento de definir la variable como después de haber sido definida.

Forzado y asignación simultánea de valores

Al asignar un valor a una variable, se puede *forzar* su tipo de la siguiente forma. Si deseamos que la variable pase a ser tipo de **double** basta con **anteponer** a su valor –entre paréntesis– tal como se indica una de las expresiones:

(double), **(real)** o **(float)**.

Por ejemplo:

```
$a=((double)45); o
$a=((float)45); o
$a=((real)45);
```

cualquiera de ellas produciría el mismo efecto: convertir la variable **\$a** a tipo **Double**.

Para *forzar* una variable a tipo **Integer** podemos **anteponer** a su valor una de estas expresiones:

(integer), o **(int)**.

Por ejemplo:

```
$b=((integer)4.5); o
$b=((int)45);
```

producirían el mismo efecto: convertir la variable **\$b** a tipo **Integer**.

Para *forzar* una variable a tipo **String** basta con **anteponer** a su valor (entre paréntesis): **(string)**.

Por ejemplo:

```
$c=((string)4.5);
```

convertiría la variable **\$c** a tipo **String**.

Forzado de tipos en variables ya definidas

La forma más aconsejable de *forzado* de tipos en variables que ya estuvieran definidas previamente, es el uso de la siguiente instrucción:

settype(var,tipo)

donde *var* es el nombre de la variable cuyo tipo pretendemos modificar y *tipo* una expresión que puede contener (**entre comillas**) uno de estos valores: **'double'**, **'integer'**, o **'string'** según se trate de forzar a: *coma flotante*, *entero*, o *cadena*.

Un ejemplo podría ser este:

```
settype($a,'integer')
```

que convertiría a *tipo entero* la variable **\$a**.

Forzado de tipos usando settype()

Aquí tienes algunos ejemplos del uso de esa función. La tabla está organizada en bloques de *tres filas* que corresponden a la ejecución de tres instrucciones y a la visualización del resultado de cada una de ellas.

El resultado de **settype** –primera fila– solo podrá ser **1** ó **0** según la instrucción se haya ejecutado con éxito o no haya podido realizarse.

En la *segunda fila* comprobamos el nuevo tipo de variable obtenida mediante la ejecución de la instrucción anterior y en la *tercera* visualizamos los nuevos valores de la variable, que pueden haber cambiado como consecuencia del cambio de tipo.

Forzado de tipos con settype()		
Variable	Sintaxis	Devuelve
<code>\$a1=347</code>	<code>echo (settype(\$a1,'double'))</code>	1
	<code>echo gettype(\$a1)</code>	double
	<code>echo \$a1</code>	347
<code>\$a2=2147483647</code>	<code>echo (settype(\$a2,'double'))</code>	1
	<code>echo gettype(\$a2)</code>	double
	<code>echo \$a2</code>	2147483647
<code>\$a3=-2147483647</code>	<code>echo settype(\$a3,'double')</code>	1
	<code>echo gettype(\$a3)</code>	double
	<code>echo \$a3</code>	-2147483647
<code>\$a4=23.7678</code>	<code>echo settype(\$a4,'integer')</code>	1
	<code>echo gettype(\$a4)</code>	integer
	<code>echo \$a4</code>	23
<code>\$a5=3.1416</code>	<code>echo settype(\$a5,'integer')</code>	1
	<code>echo gettype(\$a5)</code>	integer
	<code>echo \$a5</code>	3
<code>\$a6="347"</code>	<code>echo settype(\$a6,'double')</code>	1
	<code>echo gettype(\$a6)</code>	double

La ejecución de la instrucción **settype** devuelve (da como resultado) un valor que puede ser: **true** o **false** (1 ó 0) según la conversión se haya realizado con éxito o no haya podido realizarse.

Operaciones con distintos tipos de variables

PHP permite la realización de operaciones aritméticas con cualquiera de los tres tipos de variables y adecúa el resultado al tipo más apropiado.

En la tabla de la derecha puedes ver algunos ejemplos, pero, en resumen, ocurre lo siguiente:

- Al operar con dos enteros, si el resultado está dentro del rango de los enteros, devuelve un entero.
- Si al operar con dos enteros el resultado desborda el rango entero, convierte su valor, de forma automática, al tipo coma flotante
- Al operar un entero con una variable tipo coma flotante el resultado es de coma flotante.
- Al operar con una cadena lo hace como si se tratara de un entero. Si hay caracteres numéricos al comienzo, los extrae (hasta que aparezca un punto o un carácter no numérico) y los opera como un número entero.
- Si una cadena no comienza por un carácter numérico PHP la operará tomando su valor numérico como CERO.

	echo \$a6	347
\$a7="3.1416"	echo settype(\$a7,'integer')	1
	echo gettype(\$a7)	integer
	echo \$a1	3
\$a8="Solo literal"	echo settype(\$a8,'double')	1
	echo gettype(\$a8)	double
	echo \$a8	0
\$a9="12.3 Literal con número"	echo settype(\$a9,'integer')	1
	echo gettype(\$a9)	integer
	echo \$a9	12

Tipos de variable de los operadores y de los resultados

La tabla siguiente contiene –en cada fila– los valores asignados a dos variables (A y B) y el resultado de la suma de ambas. A continuación se recogen los tipos de variable de cada una de ellas y el del resultado. El tipo de este último –generado por PHP– estará condicionado por el valor del resultado de cada una de las operaciones.

Resultados de operaciones y tipos de variables resultantes					
Valores			Tipos de variables		
A	B	A+B	A	B	A+B
12	16	28	integer	integer	integer
12	2147483647	2147483659	integer	integer	double
-12	-2147483640	-2147483652	integer	integer	double
12	1.2456	13.2456	integer	double	double
1.2456	12	13.2456	double	integer	double
1.2456	123.4567	124.7023	double	double	double
12	abc	12	integer	string	integer
1.2456	abc	1.2456	double	string	double
12	12abc	24	integer	string	integer
12	12.34567abc	24.34567	integer	string	double
1.2456	12.34567abc	13.59127	double	string	double
1.2456	12.3e2abc	1231.2456	double	string	double
abc	12abc	12	string	string	integer
abc	12.34567abc	12.34567	string	string	double
12abc	12.34567abc	24.34567	string	string	double