

Cuando tenemos variables que contienen datos de diferentes tipos, y realizamos operaciones con ellos, PHP intenta realizar una conversión entre ellos de la manera más conveniente posible. De esta manera, enteros son convertidos directamente a cadenas al realizar concatenación, etc.

En ocasiones esta conversión automática no es adecuada, y podemos querer forzarla nosotros por nuestra cuenta. Podemos hacerlo anteponiendo a la utilización de la función el tipo entre paréntesis que queremos forzar que sea el resultado.

<http://php.net/manual/es/language.types.type-juggling.php>

```
$foo = 10;           // $foo es un integer
$str = "$foo";      // $str es un string
$fst = (string) $foo; // $fst es tambien un string
```

Dado que PHP convierte automáticamente los tipos siempre que es posible, esto hace que algunas comparaciones puedan dar como resultado que dos variables son iguales, cuando en realidad los datos que contienen son de tipos diferentes.

```
if (10=="10") echo "Parecen iguales";
```

Para esto existen los operadores `===` y `!==`, que no solo verificarán que los contenidos de las variables o expresiones comparadas sean equivalente, sino que además sean del mismo tipo.

```
if (10=== "10") echo "Solo verdadero si son del mismo tipo";
else echo "No tiene el mismo valor y tipo";
```

Respecto a los objetos, con `==` se comparan de forma superficial los atributos de dos objetos, y que son de la misma clase, para considerarlos equivalentes. Con `===` el resultado es verdadero únicamente cuando ambas variables hacen referencia a la misma instancia de objeto de la misma clase.

<http://php.net/manual/es/language.oop5.object-comparison.php>