From php.net

1.

When you have a deal with money like dollars, you need to display it under this condition:  
-format all number with two digit decimal for cents.  
-divide 1000 by ,  
-round half down for number with more than two decimal  
  
I approach it using round function inside the number\_format function:  
  
number\_format((float)round( 625.371 ,2, PHP\_ROUND\_HALF\_DOWN),2,'.',',')  // 625.37  
number\_format((float)round( 625.379 ,2, PHP\_ROUND\_HALF\_DOWN),2,'.',',')  // 625.38  
number\_format((float)round( 1211.20 ,2, PHP\_ROUND\_HALF\_DOWN),2,'.',',')  // 1,211.20  
number\_format((float)round( 625 ,2, PHP\_ROUND\_HALF\_DOWN),2,'.',',')      // 625.00

2.

I discovered that under some conditions you can get rounding errors with round when converting the number to a string afterwards.  
  
To fix this I swapped round() for number\_format().  
  
Unfortunately i cant give an example (because the number cant be represented as a string !)  
  
essentially I had round(0.688888889,2);  
  
which would stay as 0.68888889 when printed as a string.  
  
But using number\_format it correctly became 0.69.

From stackoverflow.com

3.

I want to round a number and I need a proper integer because I want to use it as an array key. The first "solution" that comes to mind is:

$key = (int)round($number)

However, I am unsure if this will always work. As far as I know (int) just truncates any decimals and since round($number) returns a float with theoretically limited precision, is it possible that round($number) returns something like 7.999999... and then $key is 7 instead of 8?

If this problem actually exists (I don't know how to test for it), how can it be solved? Maybe:

$key = (int)(round($number) + 0.0000000000000000001) // number of zeros chosen arbitrarily

Is there a better solution than this?

Integers stored within floats are always accurate, up to around 2^51, which is much more than can be stored in an int anyway. You are worrying over nothing. – [Niet the Dark Absol](https://stackoverflow.com/users/507674/niet-the-dark-absol) [Aug 3 '16 at 14:22](https://stackoverflow.com/questions/38746283/php-round-to-integer#comment64866303_38746283)

* @NiettheDarkAbsol Oops, I actually knew that (from Javascript) but didn't make the connection. You should make that an answer. BTW, in JS it's 2^53-1 iirc. – [AndreKR](https://stackoverflow.com/users/476074/andrekr) [Aug 3 '16 at 14:23](https://stackoverflow.com/questions/38746283/php-round-to-integer#comment64866330_38746283)
* When you use rounded numbers as key, you run in trouble if you have more then one 7.x numbers with the rounded result of 8. – [u-nik](https://stackoverflow.com/users/3619711/u-nik) [Aug 3 '16 at 14:25](https://stackoverflow.com/questions/38746283/php-round-to-integer#comment64866421_38746283)
* @u-nik It's exactly my intention that 7.4 and 7.6 end up with the same key. :) – [AndreKR](https://stackoverflow.com/users/476074/andrekr) [Aug 3 '16 at 14:27](https://stackoverflow.com/questions/38746283/php-round-to-integer#comment64866560_38746283)
* There are three functions to handle rounding numbers: round, [ceil][] (round up), [floor][] (round down). I'm unsure which one you want to use. [ceil]: [secure.php.net/ceil](https://secure.php.net/ceil) [floor]: [secure.php.net/floor](https://secure.php.net/floor) – [Exagone313](https://stackoverflow.com/users/3046871/exagone313) [Aug 3 '16 at 14:28](https://stackoverflow.com/questions/38746283/php-round-to-integer#comment89240015_38746283)

To round floats properly, you can use:

* [ceil($number)](http://php.net/manual/en/function.ceil.php): round up
* [round($number, 0)](http://php.net/manual/en/function.round.php): round to the nearest integer
* [floor($number)](http://php.net/manual/en/function.floor.php): round down

Those functions return float, but from [Niet the Dark Absol comment](https://stackoverflow.com/questions/38746283/php-round-to-integer): *"Integers stored within floats are always accurate, up to around 2^51, which is much more than can be stored in an int anyway."*

round(), without a precision set always rounds to the nearest whole number. By default, round rounds to zero decimal places.

So:

$int = 8.998988776636;

round($int) //Will always be 9

$int = 8.344473773737377474;

round($int) //will always be 8

So, if your goal is to use this as a key for an array, this should be fine.

You can, of course, use modes and precision to specify exactly how you want round() to behave. See [this](http://php.net/manual/en/function.round.php).

**UPDATE**

You might actually be more interested in [intval](http://php.net/manual/en/function.intval.php):

echo intval(round(4.7)); //returns int 5

echo intval(round(4.3)); // returns int 4

Integers stored within floats are always accurate, up to around 253, which is much more than can be stored in an int anyway. I am worrying over nothing.