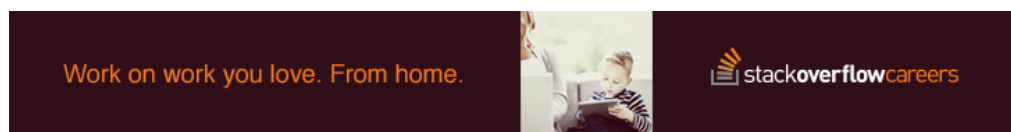




Converting to and from CENTS



So I know there have been multiple questions regarding Money and converting to and from cents. Heck I have even asked another one, but I want to make a slightly different question so I hope there are no duplicates out there.

So I have created a function that takes a Dollar Value and sends it to CENTS. But I think I have a slight problem with my code and hoping I can get it tweaked a little.

```
$money4 = "10.0001";
// Converted to cents, as you can see it's slightly off.
$money41 = "1001";
// So when "1001", get's put in the database, and then I return it back as a Money
variable.
// We get, "$10.01"... but what I have now is a Leak in my amounts... as it rounded up to
the second point.
```

So to do what I have done, I have used to functions I made to do this.

```
// This essentially gets a DOLLAR figure, or the CENT's Figure if requested.
function stripMoney($value, $position = 0, $returnAs = "")
{
    // Does it even have a decimal?
    if(isset($value) && strpos($value, ".")) {

        // Strip out everything but numbers, decimals and negative
        $value = preg_replace("/([^\0-9\.\-])/i", "", $value);
        $decimals = explode(".", $value);

        // Return Dollars as default
        return ($returnAs == "int" ? (int)$decimals[$position] : $decimals[$position]);

    } elseif(isset($value)) {
        // If no decimals, Lets just return a solid number
        $value = preg_replace("/([^\0-9\.\-])/i", "", $value);
        return ($returnAs == "int" ? (int)$value : $value);
    }
}
```

The next function I use is to generate the CENTS or return it back as dollars.

```
function convertCents($money, $cents = NULL, $toCents = TRUE)
{
    if(isset($money)) {
        if($toCents == TRUE) {
            // Convert dollars to cents
            $totalCents = $money * 100;
            // If we have any cents, Lets add them on as well
            if(isset($cents)) {
                $centsCount = strlen($cents);
                // In case someone inputs, $1.1
                // We add a zero to the end of the var to make it accurate
                if($centsCount < 2) {
                    $cents = "{$cents}0";
                }
                // Add the cents together
                $totalCents = $totalCents + $cents;
            }
            // Return total cents
            return $totalCents;
        } else {
            // Convert cents to dollars
            $totalDollars = $money / 100;
            return $totalDollars;
        }
    }
}
```

And the final function that puts everything together. So we just use 1 function to merge the 2 functions together basically.

```
function convertMoney($value, $toCents = TRUE) {
    if(isset($value) && strpos($value, ".")) {
        return convertCents(stripMoney($value, 0), stripMoney($value, 1), $toCents);
    }
}
```

```
} elseif(!empty($value)) {  
    return convertCents(stripMoney($value, 0), NULL, $toCents);  
}  
}
```

What I have done might be overkill, But I think it's fairly solid, other than this 1 detail, that I can see.

can anyone help me with these adjustments?

[php](#) [numbers](#) [money](#) [money-format](#)

asked Jul 9 '11 at 20:46



Justin

796 3 15 50

2 Answers

Do not use floating point arithmetic if you need exact answers. This applies to almost all languages, not just PHP. Read the big warning in the [PHP manual](#).

Instead check out [BC Math](#) or the [GMP extension](#). The latter only works with integer numbers so you are probably most interested in BC Math.

answered Jul 9 '11 at 20:50



Emil Vikström

44.5k 8 60 103

1 your right, I was way overcomplicating it...all I really need to do is remove everything but, decimal points, store that in MySQL with DECIMAL field, and then use BCMath to figure out the rest. — [Justin](#) Jul 9 '11 at 21:01

That sounds like the right approach! — [Emil Vikström](#) Jul 9 '11 at 21:05

I think [money_format](#) is the function you were looking for...

```
<?php  
  
$number = 1234.56;  
  
// Let's print the international format for the en_US locale  
setlocale(LC_MONETARY, 'en_US');  
echo money_format('%i', $number) . "\n";  
// USD 1,234.56  
  
// Italian national format with 2 decimals`  
setlocale(LC_MONETARY, 'it_IT');  
echo money_format('%.2n', $number) . "\n";  
// Eu 1.234,56  
  
?>
```

answered Jul 9 '11 at 20:51



diEcho

23.4k 14 59 126

yeah I can format my numbers alright, i'm just having a hard time getting my 4 decimal locations... — [Justin](#) Jul 9 '11 at 20:55