
Subject: Recursion

Posted by [akrabat](#) on Wed, 16 Jul 2008 19:57:42 GMT

[View Forum Message](#) <> [Reply to Message](#)

Recursion is the term used when an operation is repeated on the results of the same operation. That's a little confusing, but what it means to a programmer is that you have a function that calls itself.

Consider the process of adding up numbers in an array such as:

```
<?php$array = array(10, 21, 4);?>
```

All we need to do is iterate over the array and add each number to a variable as shown here:

```
<?phpfunction sumArray($array){ $total = 0; foreach ($array as $element) { $total += $element; } return $total;}$array = array(10, 21, 4);$result = sumArray($array);echo "result = $result\n";?>
```

As you would expect, the output of the script is:

```
result = 35
```

But, what if the array is nested? For example, an array like this:

```
<?php$array = array(10, 20, 5, array(5, 2, 3) );?>
```

As we iterate over this new array, we will come to an element that is itself an array. We need to sum up the elements within this sub-array and, fortunately, we have just written a function that does just that (sumArray() !), so let's call it within the foreach() loop:

```
<?phpfunction sumArray($array){ $total = 0; foreach ($array as $element) { if(is_array($element)) { $total += sumArray($element); } else { $total += $element; } } return $total;}?>
```

This is recursion as we have called the sumArray() function from within the sumArray() function itself. That is, sumArray() is a recursive function.

We can now use our improved function to add up our nested array:

```
<?php$array = array(10, 20, 5, array(5, 2, 3));$result = sumArray($array);echo "result = $result\n";?>
```

which will now output:

```
result = 45
```

As sumArray() is a recursive function, it will happily handle an array that is many nested levels deep. For example, consider an array that is nested 5 levels deep:

```
<?php$array = array(10, 20, 5, array(5, 2, 3, array(5, 3, array(2, 10, array(19, 1)), 3), 2, 7), 3);$result = sumArray($array);echo "result = $result\n";?>
```

When this code is run, the output is:

```
result = 100
```

Clearly recursion is a powerful and flexible technique for solving problems involving nested data such as reading through XML structures or handling a tree within a database table (usually implemented with a parent_id column). It is also helpful when writing a sorting algorithm, but most PHP programmers don't need to do that! Recursive solutions also tend to be fairly compact and easy to debug as you only need to do it once!

Obviously, as we're programming, there are trade-offs involved! Recursive solutions are inefficient in terms of performance, so consider caching the result. Also, it's possible to get into situation where the recursion never stops. Always make sure that your function will end! Related to that, when you have deeply nested recursion, you can run out of "stack space" (this area reserved for the list of functions that are currently being called. In other words - make sure you test thoroughly :)

Go on.. write a recursive function today!

Edited to include more notes on the trade offs as noted by mvriel. Thanks!

Subject: Re: Recursion

Posted by [mvriel](#) on Wed, 16 Jul 2008 20:10:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

I agree with the last paragraph though I would have added some warnings about the use of recursion in a best practice.

Always double check or triple check that your code stop recursing at a moment in time. Because if it should happen to not have a final statement, it will start running in circles causing an infinite loop.

Those are bad

Subject: Re: Recursion

Posted by [coche](#) on Thu, 07 Aug 2008 15:16:23 GMT

[View Forum Message](#) <> [Reply to Message](#)

Great article thanks!

Subject: Re: Recursion

Posted by [KathyReid](#) on Sun, 10 Aug 2008 00:01:11 GMT

[View Forum Message](#) <> [Reply to Message](#)

This article got me thinking - I remember in high school when we were taught Pascal (yes, that's how old I am) with functional decomposition - and that Pascal had a maximum recursion level of 256 - go above that and you'd crash the stack.

This lead to the thought

"What is the maximum level of recursion in PHP?"

Many other operations in PHP are only limited by memory - such as the size of arrays or strings.

Therefore my first thought was that the level of recursion would be bounded by available memory.

Looking at the PHP Manual, I'm still not sure;

Quote:It is possible to call recursive functions in PHP. However avoid recursive function/method calls with over 100-200 recursion levels as it can smash the stack and cause a termination of the current script.

So, this still leaves some grey areas;

Obviously the amount of recursion is bounded by memory if greater than 100-200 recursions can crash a stack. But is there any way to know before doing a recursive operation whether there is sufficient memory available to do so?

Anyone found a good solution to this?

Cheers,
Kathy

Subject: Re: Recursion
Posted by [Anonymous Coward](#) on Sun, 10 Aug 2008 19:06:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Very interesting your article!

Subject: Re: Recursion
Posted by [kevdev](#) on Wed, 10 Sep 2008 09:43:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Why not just let PHP do the recursion for you with iterators as follows..

```
<?php
function arraySumRecursive($array)
{
    $total = 0;
    foreach(new recursiveIteratorIterator( new recursiveArrayIterator($array)) as $num)
    {
        $total += $num;
    }
    return $total;
}
```

}
?>

SPL iterators are far more efficient than foreach as foreach creates a copy of the array and iterates over that. SPL Iterators know only the current element and so save on resources. The recursion is done internally at a lower level which again saves on resources.

This function is available at
<http://phpro.org/examples/Array-Sum-Recursive.html>
and more info on SPL Iterators can be found at
<http://phpro.org/tutorials/Introduction-to-SPL.html>

Kind regards
Kev

Subject: Re: Recursion
Posted by [agentile](#) on Fri, 28 Aug 2009 13:36:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

A great book on recursion is "The Little Schemer" by Daniel P. Friedman. Very interesting approach for learning about the topic.
