Function and Subroutine Length - Excel Homework Help

If you got stuck doing your [Excel homework](#), this is the right place to find reliable assistance. Only experienced programmers and fast delivery.

Without doubt one of the most common issues I come across when reviewing code is the excessive length of function or subroutines: the classic doIt() function spanning hundreds of lines. So in this post I'm going to talk about the need to keep functions small (and the term 'function' covers subroutines too).

Functions should be small encapsulated units. Typically, never longer than just a few lines and I'm talking fewer than 10. Fewer than five is even better. Now, for those that write hundreds of lines, this may seem a tall order, but it is possible and something that you should strive for.

So, why so short? Let's start with a really simple point. It should easily fit on a screen or in the editor window. From a readability view, you don't want to have scroll back and forth to understand what a function does. When I started coding, screens only had 25 lines, and the same rule applied back then - hence at least limiting functions to 25 lines. Now we have the ability to put much more on screen, but that's no reason to increase the limit. Remember you are trying to make functions as short as possible.

Continuing the readability theme, once you start getting loops (for, while, do) or conditional if statements where the body of the loop or conditional spans many lines it gets hard to see which part of the statement the code belongs to. At most your functions should resemble something like the example below where the comment lines should themselves be calls to other functions, rather than lots of individual statements.

```
If calculationType = CALC_TYPE_NUMERIC Then
    ' call a function
Else
    ' call a different function
End If
```

or

```
For i = 1 To LIMIT
    ' call function
Next i
```

Functions should only do one thing, not a multitude of tasks and therefore can be kept small. It is so important to use meaningful names for functions. Don't be afraid of using longer names that accurately describe what the function does. Functions called 'doIt' or similar aren't helpful and are almost always a sign of a function that does multiple things.

Another reason to have keep functions short is that it is much simpler to debug and reason about what is going one when things aren't going to plan. It also makes it considerably easier to test. Testing is the subject of a future post, but here I'm talking about writing code to test your code - not running your code endless times to check it doesn't fall over. When I come on to it, you will see the huge benefits of keeping functions small.

In summary, try and minimise your function or subroutine length. If they grow to be more than a few lines, you should start to think about breaking them down. It will be hard to start with, but the benefits are great.