

# Recursion

## Recursion

Recursion is an algorithmic approach where a program calls itself to solve some problem. It has the advantage of often simplifying the algorithm, but at the cost of added memory and runtime.

## Recursive Requirements

There are three requirements of any recursive program:

- 1) It has a base case that causes it to terminate
- 2) It calls itself
- 3) When it calls itself, the arguments move toward the base case.

## Summation example

A common example to demonstrate recursion is the summation of the positive integers. If you consider them as:

1  
1 + 2  
1 + 2 + 3  
1 + 2 + 3 + 4

Then it is obvious that each row is simply the sum of the current value plus the previous row. This can be written iteratively as:

```
def summer(value, sum):  
    # iteration terminate case  
    while value > 0:  
        # computation  
        sum += value  
        # iteration increment  
        value = value - 1  
    #return result  
    return sum
```

And recursively as:

```
def rec_summer(value, sum):  
    #base case, terminates recursion  
    if value <= 1:  
        return sum  
    # computation  
    sum += value  
    #recursive call with new value for sum  
    return rec_summer(value-1, sum)
```

The first line of the program – **if value <= 1** – is the **base case**. The purpose of having a base case is to guarantee that the program will eventually terminate and not continue calling itself forever.

Note that in the last line of the program – **return rec\_summer(value-1, sum)** – the program is calling itself and that when it calls itself, the argument is such that it approaches the base case.

## Tail Recursion

Recursive functions like this are said to be tail recursive, since the recursive call is at the end of the function and the computational result accumulates in the calls and is returned from the final call on the stack.

As shown above, it is a simple matter to convert an iterative function based on a while loop to tail recursion. The base case is the iteration termination expression, the same computation is used, and the recursion is called with a parameter changing in the same way as the in the loop.