Base Case

In recursion, there must be a base case (also referred to as *terminating case*). Because recursive methods call themselves, they will never stop unless this base case is reached. Stack overflow from recursion is most likely the result of not having a proper base case. In the base case, there are *no* recursive function calls.

Let’s examine the following function, which prints numbers counting down from n to 0 as an example:

function countDownToZero(n) {

*// base case. Stop at 0*

if (n < 0) {

return; *// stop the function*

} else {

console.log(n);

countDownToZero(n - 1); *// count down 1*

}

}

countDownToZero(12);