**What is a base condition in recursion?**

In a recursive function, the solution to the **base case** is provided and the solution of the bigger problem is expressed in terms of smaller problems.

The role of the base condition is to stop a recursive function from executing endlessly – once a pre-specified base condition is met, the function knows it’s time to exit.

|  |
| --- |
|  |
| Factorial calculation using recursion |

|  |
| --- |
| class Factorial {  public static int fact(int n)  {  if (n <= 1) // base case  return 1;  else  return n \* fact(n-1);  }  public static void main( String args[] ) {  System.out.println(fact(5));  }  } |

In the example above, we can compute the factorial of n if we know the factorial of (n-1). The base case for this factorial function would be n = 0. We return 1 when n = 0, the base case, is reached.

Every recursive program must have a base case to make sure that the function will terminate at some point. Missing the base case results in unexpected behavior.

Simply put, the idea behind recursion is to represent a larger problem in terms of one or more smaller problems and add one or more base conditions that stop the recursion from looping endlessly.