The Recursive Leap of Faith

The computer treats recursive functions just like all other functions. It is useful to put the underlying details aside and focus on **a single level of the operation**; **assume** that any recursive call automatically gets the right answer as long as the arguments are in **some sense simpler** than the original.

This psychological strategy—assuming that any simpler recursive call will work correctly—is called **the recursive leap of faith**. Learning to apply this strategy is essential to using recursion in practical applications.

Consider what happens when you call `factorial(4)`; the function must compute the expression `n * factorial(n - 1)`, and, by substituting the current value of `n` into the expression, you know that the result is `4 * factorial(3)`.

Stop right there. Computing `factorial(3)` is **simpler** than computing `factorial(4)`. **Because** it is simpler, the recursive leap of faith allows you to **assume that it works**. Thus, you should assume that the call to `factorial(3)` will correctly compute the value of `3!`, which is `3 × 2 × 1`, or `6`. The result of calling `fact(4)` is therefore `4 × 6`, or `24`.

As you look at the examples in the rest of this chapter, try to **focus on the big picture** instead of the details. Once you have made the recursive decomposition and identified the simple, base cases, be satisfied that the computer can handle the rest.