Factorial using recursion

May 18, 2020

1 Factorial using recursion

The **factorial** function is a mathematical function that multiplies a given number, n, and all of the whole numbers from n down to 1.

For example, if n is 4 then we will get:

$4 * 3 * 2 * 1 = 24$

This is often notated using an exclamation point, as in 4! (which would be read as "four factorial").

So $4! = 4 * 3 * 2 * 1 = 24$

More generally, we can say that for any input n:

$n! = n * (n-1) * (n-2)...1$

If you look at this more closely, you will find that the factorial of any number is the product of that number and the factorial of the next smallest number. In other words:

$n! = n * (n-1)!$

Notice that this is recursive, meaning that we can solve for the factorial of any given number by first solving for the factorial of the next smallest number, and the next smallest number, and the next smallest number, and so on, until we reach 1.

If you were to write a Python function called `factorial` to calculate the factorial of a number, then we could restate what we said above as follows:

`factorial(n) = n * factorial(n-1)`

So that is the goal of this exercise: To use recursion to write a function that will take a number and return the factorial of that number.

For example, you should be able to call your function with `factorial(4)` and get back `24`.

Note: By definition, $0! = 1$

```
In [5]: # Code

        def factorial(n):
            """
            Calculate n!

            Args:
               n(int): factorial to be computed
            Returns:
               n!
            """
```