

Recursion

Now that we are familiar with functions, let's have a review on the most important concept using a function: recursion.

WE'LL COVER THE FOLLOWING ^

- What Is Recursion?
- Parts of a Recursion

What Is Recursion?

Recursion is when a function calls itself again and again until it reaches the base condition.

Parts of a Recursion

There are two parts of recursion: **base condition** and **recursive function**.

For instance, take the example of a Factorial function.

Factorial(!) of a number n is the product of all positive numbers from our chosen number n down to 1.

```
factorial(n) = 0    if n is 0 #base case

factorial (n) = n * factorial(n-1) #recursive functio
n
```

Let's use an example of factorial of 5.

```
5! = 5 × 4!
    = 5 × 4 × 3!
    = 5 × 4 × 3 × 2!
    = 5 × 4 × 3 × 2 × 1
    = 120
```

Below is the call to the function factorial given 5 as a parameter to the function. factorial(5)

It is calculated recursively in the following manner:

<code>factorial(5)=5*factorial(4)</code>	<code>call factorial(4)</code>
--	--------------------------------

1 of 9

<code>factorial(4)=4*factorial(3)</code>	<code>call factorial(3)</code>
--	--------------------------------

<code>factorial(5)=5*factorial(4)</code>
--

2 of 9

<code>factorial(3)=3*factorial(2)</code>
<code>factorial(4)=4*factorial(3)</code>
<code>factorial(5)=5*factorial(4)</code>

call factorial(2)

3 of 9

<code>factorial(2)=2*factorial(1)</code>
<code>factorial(3)=3*factorial(2)</code>
<code>factorial(4)=4*factorial(3)</code>
<code>factorial(5)=5*factorial(4)</code>

call factorial(1)

4 of 9

<code>factorial(2)=2*factorial(1)</code>	<code>return 2*1</code>
<code>factorial(3)=3*factorial(2)</code>	
<code>factorial(4)=4*factorial(3)</code>	
<code>factorial(5)=5*factorial(4)</code>	

5 of 9

<code>factorial(3)=3*factorial(2)</code>	<code>return 3*2</code>
<code>factorial(4)=4*factorial(3)</code>	
<code>factorial(5)=5*factorial(4)</code>	

6 of 9

<code>factorial(4)=4*factorial(3)</code>	<code>return 4*6</code>
<code>factorial(5)=5*factorial(4)</code>	

7 of 9

<code>factorial(5)=5*factorial(4)</code>	<code>return 5*24</code>
--	--------------------------

8 of 9

factorial(5)=120

9 of 9

—

[]

Basically, the factorial of 5 is 5 times the factorial of 4, etc. Finally, the factorial of 1 (or of zero) is 1 which breaks the recursion. In Python, we could write the following recursive function:

```
def factorial(n):  
    if(n <= 1):  
        return 1  
    else:  
        return(n * factorial(n-1))  
print("Factorial:")  
print(factorial(5))
```



The trick with recursive functions is that there must be a “base” case where the recursion must end with a recursive case that iterates towards the base case. In the case of factorial, we know that the factorial of zero is one, and the factorial of a number greater than zero will depend on the factorial of the previous number until it reaches zero.

Now that the concept of ‘Recursion’ in python is clear, let’s check your knowledge in the upcoming exercises before moving on to the next chapter —‘Iteration and Loops’.