PythonTip 03 - Recursion

February 23, 2022

1 Recursion

A recursive algorithm is an algorithm that calls itself. You need a base case so you don't get stuck in an infinite loop.

Example: suppose we want to calculate the quantity $n! = n(n-1)(n-2)\cdots 3 \cdot 2 \cdot 1$.

$5! = 5 * 4 * 3 * 2 * 1 = 120$

We'll use the fact that $n! = n \cdot (n-1)!$.

[1]:
```
# What's wrong with this function?
def factorial(n):
    return n * factorial(n-1)
```

```
factorial(3)
 -> 3 * factorial(2)
 -> 3 * 2 * factorial(1)
 -> 3 * 2 * 1 * factorial(0)
 -> 3 * 2 * 1 * 0 * factorial(-1)
 -> infinite recursion
```

[3]:
```
factorial(3)
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
Input In [3], in <module>
----> 1 factorial(3)

Input In [1], in factorial(n)
      2 def factorial(n):
----> 3     return n * factorial(n-1)

Input In [1], in factorial(n)
      2 def factorial(n):
----> 3     return n * factorial(n-1)

    [… skipping similar frames: factorial at line 3 (2970 times)]

Input In [1], in factorial(n)
```

```
      2 def factorial(n):
----> 3     return n * factorial(n-1)

RecursionError: maximum recursion depth exceeded
```

What calls are happening?

[4]:
```
# We need a base case!
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n-1)
```

[5]:
```
factorial(5)
```

[5]: 120

```
factorial(5)
5 * factorial(4)
5 * (4 * factorial(3))
5 * (4 * (3 * factorial(2)))
5 * (4 * (3 * (2 * factorial(1))))
5 * (4 * (3 * (2 * 1)))
```

[6]:
```
factorial(1000)
```

[6]:
```
402387260077093773543702433923003985719374864210714632543799910429938512398629020592044208486969404800479988610197196058631666872994808558901323829669944590997424504087073759918823627727188732519779505950995276120874975462497043601418278094646496291056393887437886487337119181045825783647849977012476632889835955735432513185323958463075557409114262417474349347553428646576611667797396668820291207379143853719588249808126867838374559731746136085379534524221586593201928090878297308431392844403281231558611036976801357304216168747609675871348312025478589320767169132448426236131412508780208000261683151027341827977704784635868170164365024153691398281264810213092761244896359928705114964975419909342221566832572080821333186116811553615836546984046708975602900950537616475847728421889679646244945160765353408198901385442487984959953319101723355556602139450399736280750137837615307127761926849034352625200015888535147331611702103968175921510907788019393178114194545257223865541461062892187960223838971476088506276862967146674697562911234082439208160153780889893964518263243671616762179168909779911903754031274622289988005195444414282012187361745992642956581746628302955570299024324153181617210465832036786906117260158783520751516284225540265170483304226143974286933061690897968482590125458327168226458066526769958652682272807075781391858178889652208164348344825993266043367660176999612831860788386150279465955131156552036093988180612138558600301435694527224206344631797460594682573103790084024432438465657245014402821885252470935190620929023136493273497565513958720559654228749774011413346962715422845862 3
```

```
77387538230483865688976461927383814900140767310446640259899490222221765904339901
88601856652648506179970235619389701786004081188972991831102117122984590164192106
88843871218556461249607987229085192968193723886426148396573822911231250241866493
53143970137428531926649875337218940694281434118520158014123344828015051399694290
15348307764456909907315243327828826986460278986432113908350621709500259738986355
42771967428222487575867657523442202075736305694988250879689281627538488633969099
59826280956121450994871701244516461260379029309120889086942028510640182154399457
15680594187274899809425474217358240106367740459574178516082923013535808184009699
63725242305608559037006242712434169090041536901059339838357779394109700277534720
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000
```

[ ]: