CS 152: Recursion

CS 152: Python for STEM

Colorado State University
Computer Science Department
Slides Originally Created by Albert Lionelle and Updated by Marcia Moraes

Colorado State University

Weekly Announcements!

TODO Reminders:

- Reading 14 (zybooks) – you should have already done that ☺
- Lab 09
- Reading 15 (zybooks) – you should have already done that ☺
- Lab 10
- Reading 16 (zybooks)

Recall Activity

- Write a Python dictionary to represent products and prices.
- Use your creativity to build that dictionary ☺.
- Write your answer in our today's attendance assignment.

Factorial Function

```
def factorial(num):
    if(num == 0 or num == 1):
        return 1
    fact = num
    for i in range(num-1, 1, -1):
        fact = fact * i
    return fact

print(factorial(4))
print(factorial(0))
```

What is the original formula?

0! = 1
1! = 1
num! = num * num-1!

Recursion!
Function calling itself until reach a base case

Recursion

0! = 1
1! = 1
num! = num * num-1!

- Always have a base case
  - No recursive call
- Recursive call
  - Need to change the parameter so it will reach the base case and stop calling the function recursively

```
def factorial(num):
    if(num == 0 or num == 1):
        return 1
    fact = num
    for i in range(num-1, 1, -1):
        fact = fact * i
    return fact

print(factorial(4))
print(factorial(0))
```

What is the base case?

```
if(num == 0 or num == 1):
        return 1
```

What is the recursive call?

```
return num * factorialRecursive(num-1)
```

Factorial – Two Versions

```
def factorial(num):
    if(num == 0 or num == 1):
        return 1
    fact = num
    for i in range(num-1, 1, -1):
        fact = fact * i
    return fact

print(factorial(4))
print(factorial(0))
```

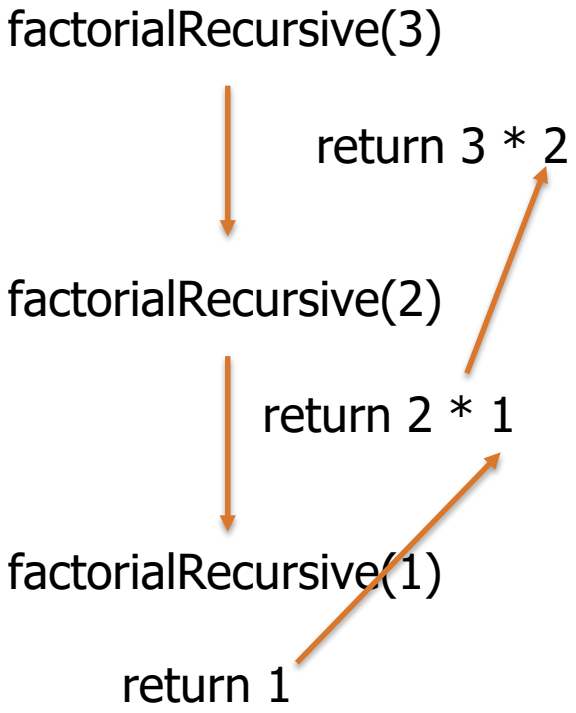
```
def factorialRecursive(num):
    if(num == 0 or num == 1):
        return 1
    return num * factorialRecursive(num-1)

print(factorialRecursive(4))
print(factorialRecursive(0))
```

Factorial – Recursive Version

```
def factorialRecursive(num):
    if(num == 0 or num == 1):
        return 1
    return num * factorialRecursive(num-1)

print(factorialRecursive(3))
```



Coding Along

- Write a Python recursive function that reverse a string.
- Thinking process
  - Get the last element in the string, use index -1
  - Call the method again, now passing a string that does not have the last element
  - Stop calling the function recursively when the string is empty
    - How do you know if a string is empty?

Pair Coding

- Write a Python recursive function that sum the elements in a list.
- Tip:
  - You need to go through the entire list.
  - How do you know when you reach the end of a list?
  - Base case: when you reach the end of the list – what should you return?
  - How many parameters your method should receive?
  - Recursive case: remember that you need to change the parameter so it will reach the base case