Recursion

Announcements

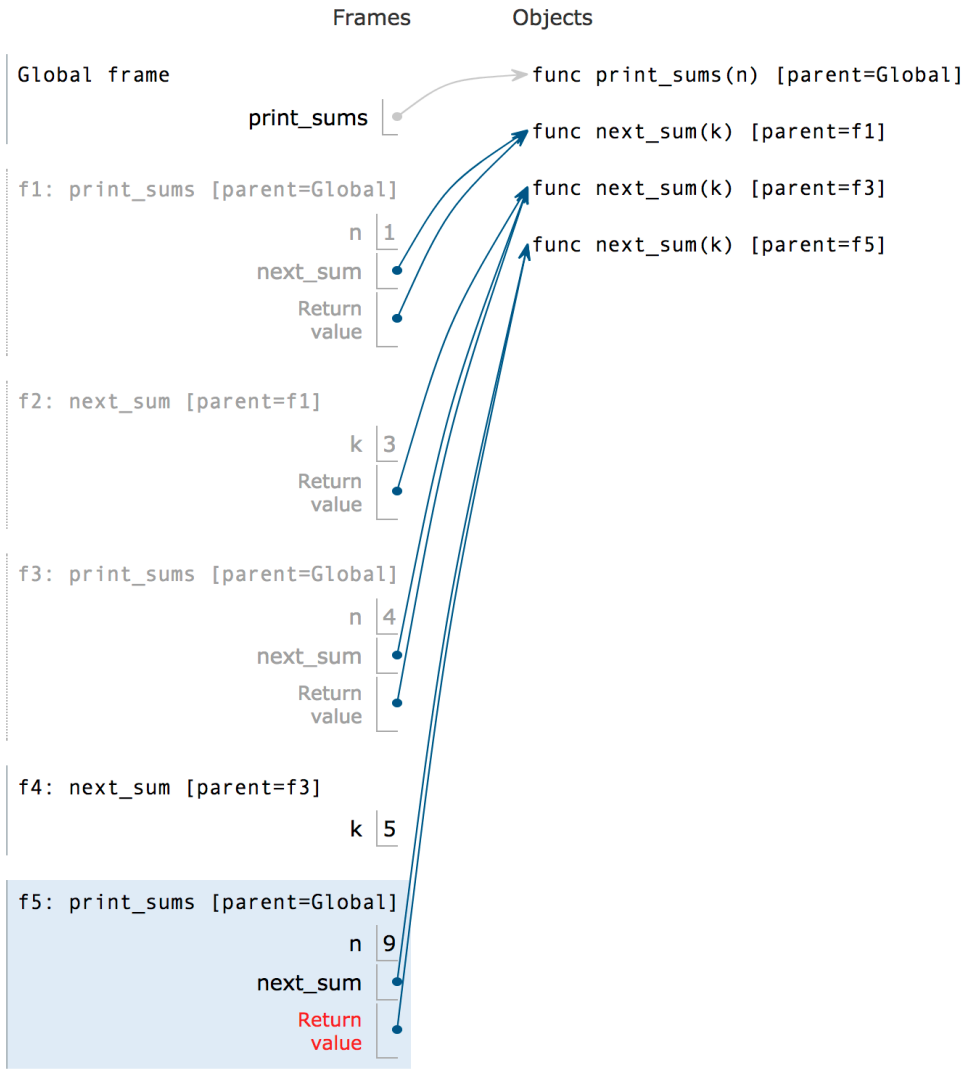
Self-Reference

Returning a Function Using Its Own Name

```
1  def print_sums(n):
2      print(n)
3      def next_sum(k):
4          return print_sums(n+k)
5      return next_sum
6
7  print_sums(1)(3)(5)
```

```
print_sums(1)(3)(5) prints:
1
4  (1 + 3)
9  (1 + 3 + 5)

print_sums(3)(4)(5)(6) prints:
3
7  (3 + 4)
12 (3 + 4 + 5)
18 (3 + 4 + 5 + 6)
```



http://pythontutor.com/composingprograms.html#code=def%20print_all%28k%29%3A%0A%20%20%20%20print%28k%29%0A%20%20%20%20return%20print_all%0A%20%20%20%20%0Aprint_all%281%29%283%29%285%29&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

http://pythontutor.com/composingprograms.html#code=def%20print_sums%28n%29%3A%0A%20%20%20%20print%28n%29%0A%20%20%20%20def%20next_sum%28k%29%3A%0A%20%20%20%20%20%20%20%20return%20print_sums%28n%2Bk%29%0A%20%20%20%20return%20next_sum%0A%0Aprint_sums%281%29%283%29%285%29&cumulative=true&curInstr=0&mode=display&origin=composingprograms.js&py=3&rawInputLstJSON=%5B%5D

Reminder: Twenty-One Rules

Two players alternate turns, on which they can add 1, 2, or 3 to the current total

The total starts at 0

The game end whenever the total is 21 or more

The last player to add to the total loses

(Demo)

Recursive Functions

(Demo)

Discussion Question: Factorial Two Ways

```
Rewrite fact(n) so that the result of fact(5) is computed using the following steps:
    5 (1 * 5)
   20 (1 * 5 * 4)
   60 (1 * 5 * 4 * 3)
  120 (1 * 5 * 4 * 3 * 2)
```

```
def fact(n):
    """Compute n factorial.

    >>> fact(5)
    120
    >>> fact(0)
    1
    """
    if n == 0 or n == 1:
        return 1
    else:
        return fact(n-1) * n
```

Discussion Question: Play Twenty-One

Rewrite play as a recursive function without a while statement.

- Do you need to define a new inner function? Why or why not? If so, what are its arguments?
- What is the base case and what is returned for the base case?

```
def play(strategy0, strategy1, announce=print_result, goal=21):
    "Play twenty-one and return the index of the winner."
    n = 0
    who, who_strat = 0, strategy0
    while n < goal:
        n = n + who_strat(n)
        announce = announce(who, n)
        if who == 0:
            who, who_strat = 1, strategy1
        elif who == 1:
            who, who_strat = 0, strategy0
    return who
```