Advanced Python

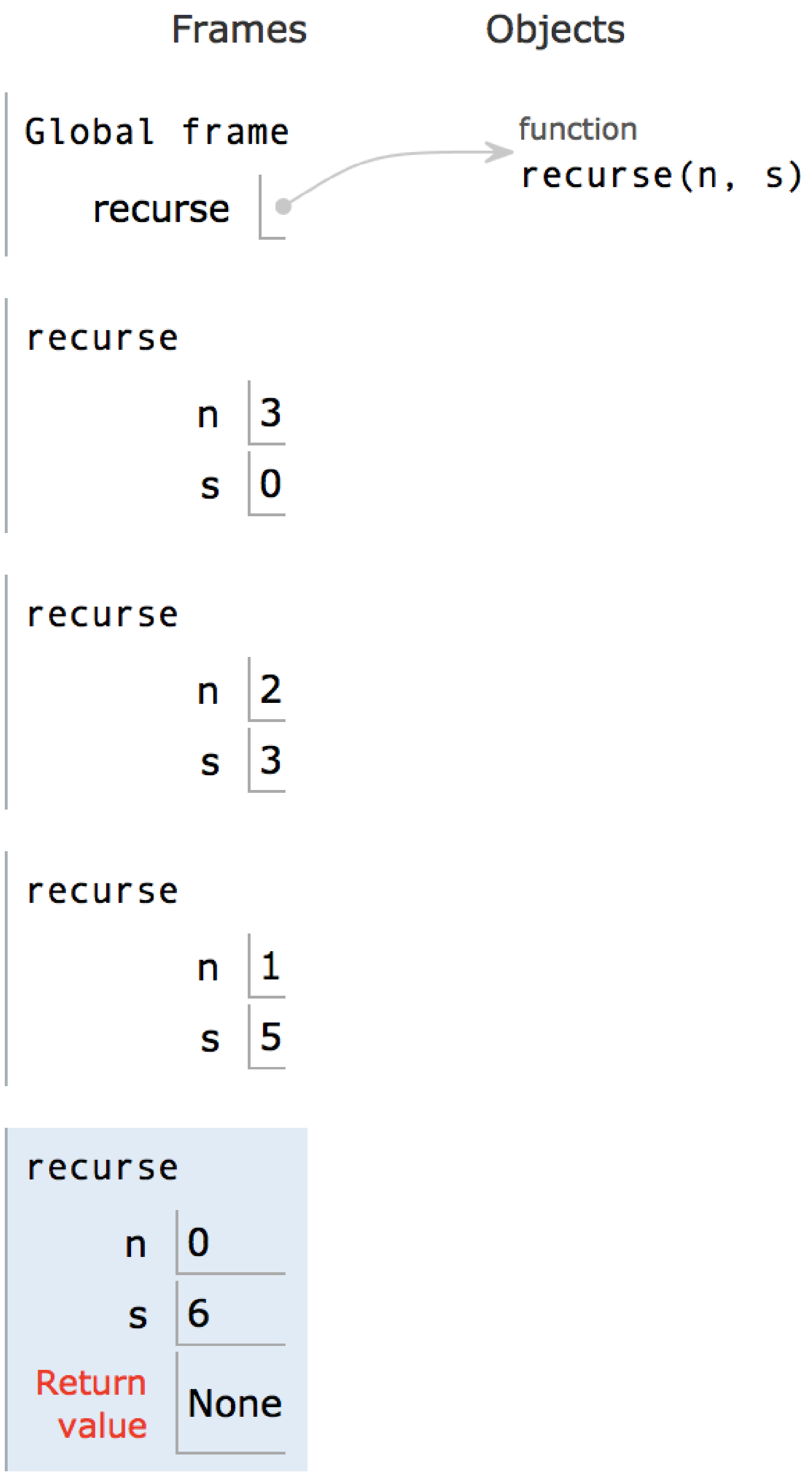
Exercise Answers

1. **[This exercise should be submitted through the Assessments section of Moodle for Tutorial Exercise 6]** Write a recursive function called *is_palindrome* that takes a string named *word* as its parameter and returns *True* if word has length less than or equal to *1*. If the length of *word* is greater than *1*, the function should return *False* if the first character of *word* is different to the last character of *word*. If the length of *word* is greater than *1*, and the first and last characters of the string are identical, the function should return the result of *is_palindrome()* with the parameter of *word* with its first and last characters removed (e.g. *is_palindrome("anna")* should return the result of *is_palindrome("nn")*).
2. Download the file https://raw.githubusercontent.com/AllenDowney/ThinkPython2/master/code/words.txt and use your code from 1) to list all palindromes in the file
3. Draw a stack diagram that shows the state of this program when it prints its result:

```
def recurse(n, s):
  if n == 0:
    print(s)
  else:
    recurse(n - 1, n + s)

recurse(3, 0)
```

This stack diagram was made with http://pythontutor.com - it can be quite useful to visualise how code works.



4. What would happen if you called the *recurse* function from question 2 with the parameters *-1* and *0*?

   The base case of the recursion would never be hit, so the system would probably run out of memory after calling the function numerous times. It might be better to change the check to n <= 0, or use an assertion to ensure it is non-negative.

5. Write a program that searches a directory and all of its subdirectories, recursively, and returns a list of complete paths for all files with a given suffix (e.g. *.txt*). Hint: `os.path` provides functions for manipulating file and path names.

```
import os

def list_files(base, suffix):
  assert os.path.isdir(base)
  for file in os.listdir(base):
    path = os.path.join(base, file)
    if path.endswith(suffix):
      print(path)
    if os.path.isdir(path):
      list_files(path, suffix)
```

6. The following function recursively computes the binomial coefficient. Rewrite the body of the function using nested conditional expressions.

```
def binomial_coeff(n, k):
  """Compute the binomial coefficient "n choose k".
  n: number of trials
  k: number of successes
  returns: int
  """

  if k == 0:
    return 1
  if n == 0:
    return 0

  res = binomial_coeff(n-1, k) + binomial_coeff(n-1, k-1)
  return res
```

```
def binomial_coeff(n, k):
  """Compute the binomial coefficient "n choose k".
  n: number of trials
  k: number of successes
  returns: int
  """

  return 1 if k == 0 else 0 if n == 0 else binomial_coeff(n-1, k) + binomial_coeff(n-1, k-1)
```

Last modified: Thursday, 1 February 2024, 10:47 AM