Dr. Gunjan Mansingh

Department of Computing

UWI

July 4- 29, 2022

Week 2 - Day 1, Session 2

Take-away Message

All non-trivial programming solutions to a problem require some sort of repetition of a process in order to complete a task. (i.e. looping is unavoidable)

Looping can be achieved through recursion or iteration

- Iteration: Focus on changing state; extract result from state at end.
- Recursion: Focus on result in terms of smaller results

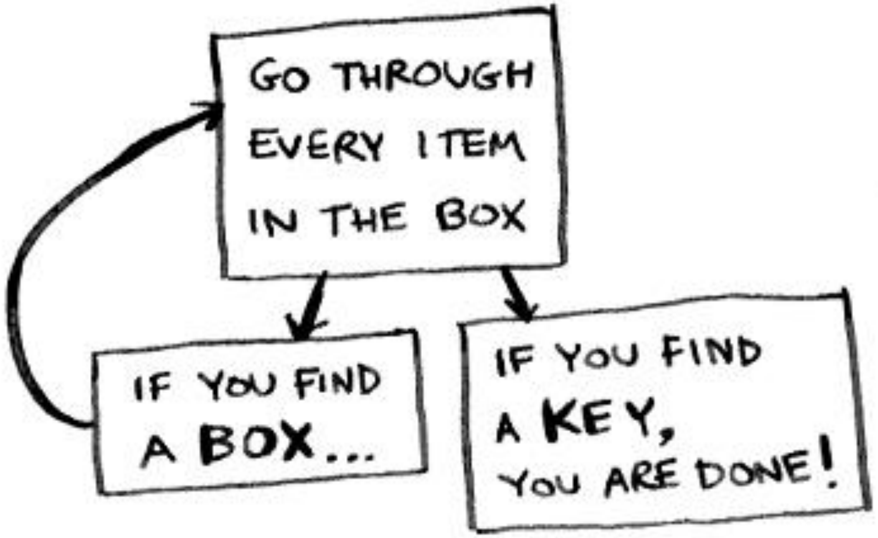
Repetition - Recursion & Iteration

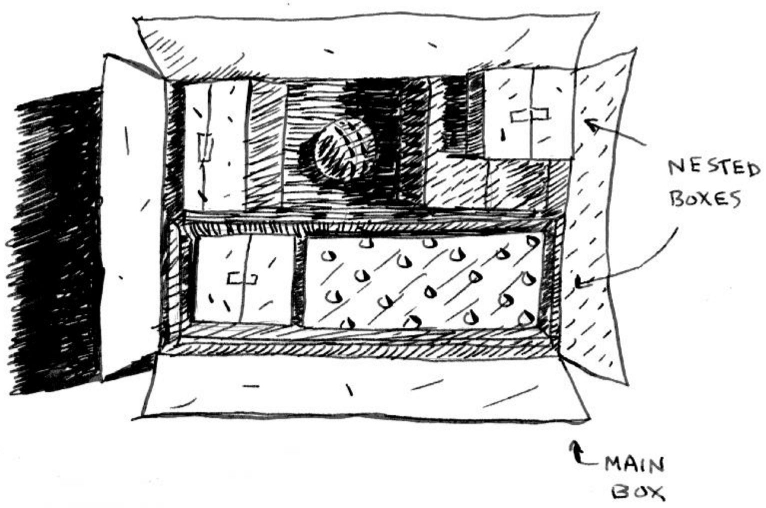
- Iteration
  - The use of looping special forms to create repetition
  - Loops infinitely if condition never evaluates to false
- Recursion
  - The use of function calls to create repetition
  - Loops infinitely if condition never breaks down to base case
  - Repeatedly invokes the mechanism and function
    - Uses more memory
      - Copies of the function’s variables are made
  - Often presents elegant solutions

Recursion

Recursive function calls itself an undetermined number of times before combining the output of all the function calls in one return statement.

Recursive Approach





Recursion Example

What is (4 * 3)?

Can we say this is same as 4 + (4 * 2)

(4 * 2)?

Can we say this is same as 4 + (4 * 1)

(4 * 1)?

Can we say this is same as 4 + (4 * 0)

(4 * 0)?

0

Recursion Example

What is (4 * 3)?

Can we say this is same as 4 + (4 * 2)

(4 * 2)?

Can we say this is same as 4 + (4 * 1)

(4 * 1)?

Can we say this is same as 4 + (4 * 0)

(4 * 0)?

0

Recursion

Function name

2 Parameters

```
def mult(x,y) :
    if y==0:
        return 0
    else:
        return x + mult(x, y-1)
```

BASE CASE or Stopping Condition

RECURSIVE CASE

Recursion Example 1

mult(4,3)
  4 + mult(4,2)
    4 + mult(4,1)
      4 + mult(4,0)
        0
      4 + 0
    4 + 4
  4 + 8
$\boxed{12}$

```python
def mult(x,y):
    if y == 0:
        return 0
    else:
        return x + mult(x, y-1)
```

Recursion Example 2

power(4,3)

4 * power(4,2)

4 * power(4,1)

4 * power(4,0)

1

4 * 1

4 * 4

4 * 16

64

```
def power(x,y):
    if y == 0:
        return 1
    else:
        return x * power(x, y-1)
```