

CSCA08 FALL 2015

WEEK 7 – MEMORY MODEL & QUIZ 3

Bo(Kenny) Zhao

University of Toronto Scarborough

October 27, 2015



LEARNING OBJECTIVES

- At the end of the tutorial, you will be able to ...
 1. Work with lists and strings in the memory model (~20 mins)
 - a few examples
 - introduction to python tutor
 2. Get a full mark on Quiz 3 (~25 mins)
 - the quiz is very hard ☹️
 - the way we do it will make it a freebie 😊

MEMORY MODEL

- Example 1

1. `x = 1`
2. `y = x`
3. `y = 2`
4. `print(x, y)`

- Output

MEMORY MODEL

- Example 2
 1. `x = [1, 2, 3]`
 2. `y = x`
 3. `x[0] = 99`
 4. `print(x, y)`

- Output

[illegible]

MEMORY MODEL

- Example 3

1. `x = "Hello"`
2. `y = x`
3. `x = "?ello"`
4. `print(x, y)`

- Output

[illegible]

MEMORY MODEL

- Example 4
 1. `x = [1, 2, 3]`
 2. `y = [1, 2, 3]`
 3. `x[0] = 99`
 4. `y[0] = 98`
 5. `print(x, y)`

- Output

[illegible]

MEMORY MODEL

- Example 5

1. `x = [1, 2, 3]`
2. `y = x[:]`
3. `x[0] = 99`
4. `y[0] = 98`
5. `print(x, y)`

- Output

[illegible]

MEMORY MODEL

- Example 6

1. `x = [[1, 2], 3]`
2. `y = x[:]`
3. `x[0][0] = 99`
4. `y[0][1] = 98`
5. `print(x, y)`

- Output

[illegible]

MEMORY MODEL

• Example 7

```
1. def mutator1(x):
2.     x[0] = "MUTATED"

3. def mutator2(x):
4.     x[0][0] = "MUTATED"
5.
6. def cloner1(x):
7.     clone = x[:]
8.     clone[0] = "COPIED"
9.     return clone

10. def cloner2(x):
11.     clone = x[:]
12.     clone[0][0] = "COPIED"
13.     return clone

14. x = [['A', 'B'], 'C']
15. y = mutator1(x)
16. print(x, y)

17. x = [['A', 'B'], 'C']
18. y = cloner1(x)
19. print(x, y)

20. x = [['A', 'B'], 'C']
21. y = mutator2(x)
22. print(x, y)

23. x = [['A', 'B'], 'C']
24. y = cloner2(x)
25. print(x, y)
```

MEMORY MODEL

- Want more practice?
course website -> practicals -> Week 3 & 7
past term tests
- Get stuck?
use <http://www.pythontutor.com/>
settings:
python 3.3
render all objects on the heap
use text labels for references
- Warning:
You will **not** have python tutor on the term test 2.

QUIZ 3

• Function 1 (original version)

```
1 def total_occurrences(puzzle, word):
2     lr_count = lr_occurrences(puzzle, word)
3     ud_count = lr_occurrences(
4         rotate_puzzle(puzzle),word)
5     rl_count = lr_occurrences(
6         rotate_puzzle(
7             rotate_puzzle(puzzle)),word)
8     du_count = lr_occurrences(
9         rotate_puzzle(
10            rotate_puzzle(
11                rotate_puzzle(puzzle))),word)
12
13     return (lr_count + ud_count + rl_count + du_count)
14
```

Problem: too many rotations

QUIZ 3

• Function 1 (improved version)

```
1 def total_occurrences(puzzle, word):
2     # count occurrences from left to right
3     lr_count = lr_occurrences(puzzle, word)
4     # rotate puzzle 90 degrees to the left
5     puzzle_rotated_once = rotate_puzzle(puzzle)
6     ud_count = lr_occurrences(puzzle_rotated_once, word)
7     # rotate the puzzle 90 degrees to the left again
8     # in other words, it's 180 degrees to the left
9     puzzle_rotated_twice = rotate_puzzle(puzzle_rotated_once)
10    rl_count = lr_occurrences(puzzle_rotated_twice, word)
11    # rotate the puzzle 90 degrees to the left again
12    # in other words, it's 270 degrees to the left
13    puzzle_rotated_thrice = rotate_puzzle(puzzle_rotated_twice)
14    du_count = lr_occurrences(puzzle_rotated_thrice, word)
15    # sum up occurrences in all 4 directions
16    return (lr_count + ud_count + rl_count + du_count)
17
```

QUIZ 3

- Function 2 (original version)

```
1 def in_puzzle(puzzle, word):
2     lr_count = lr_occurrences(puzzle, word)
3     ud_count = lr_occurrences(
4         rotate_puzzle(puzzle),word)
5     rl_count = lr_occurrences(
6         rotate_puzzle(
7             rotate_puzzle(puzzle)),word)
8     du_count = lr_occurrences(
9         rotate_puzzle(
10            rotate_puzzle(
11                rotate_puzzle(puzzle))),word)
12
13     return (lr_count or ud_count or
14            rl_count or du_count)
15
```

Problem: too many duplicated code

QUIZ 3

- Function 2 (improved version)

```
1 def in_puzzle(puzzle, word):  
2     # check if a word appears at least once in puzzle  
3     return (total_occurrences(puzzle, word) > 0)  
4
```

Don't forget to use existing function(s)

QUIZ 3

• Function 3 (original version)

```
1 def in_puzzle_horizontal(puzzle, word):
2     lr_count = lr_occurrences(puzzle, word)
3     rl_count = lr_occurrences(
4         rotate_puzzle(
5             rotate_puzzle(puzzle)), word)
6     in_horiz = (lr_count > 0) or (rl_count > 0)
7     if(in_horiz == True):
8         return True
9     else:
10        return False
11
```

(improved version)

```
return in_horiz
```

Problem: using if-statement

in_horiz is already a Boolean, so return it directly

Before you leave ...

1. Hand back the freebie Quiz 3
2. Collect your Quiz 2
 - note: a full mark doesn't mean you got everything correct
 - a. a meaningful function name
 - b. parameter names: num_pizzas, pizza_size
 - c. type contract: (int, str) -> int
 - d. REQ: num_pizzas > 0 or num_pizzas >= 0
 - e. both are acceptable
 - f. description: mention everything in your own words
 - g. examples: at least one example for each pizza size