CSCA08 Fall 2016

Week 7 - MEMORY MODEL & QUIZ 3

Bo(Kenny) Zhao

University of Toronto Scarborough

October 26, 2016

LEARNING OBJECTIVES

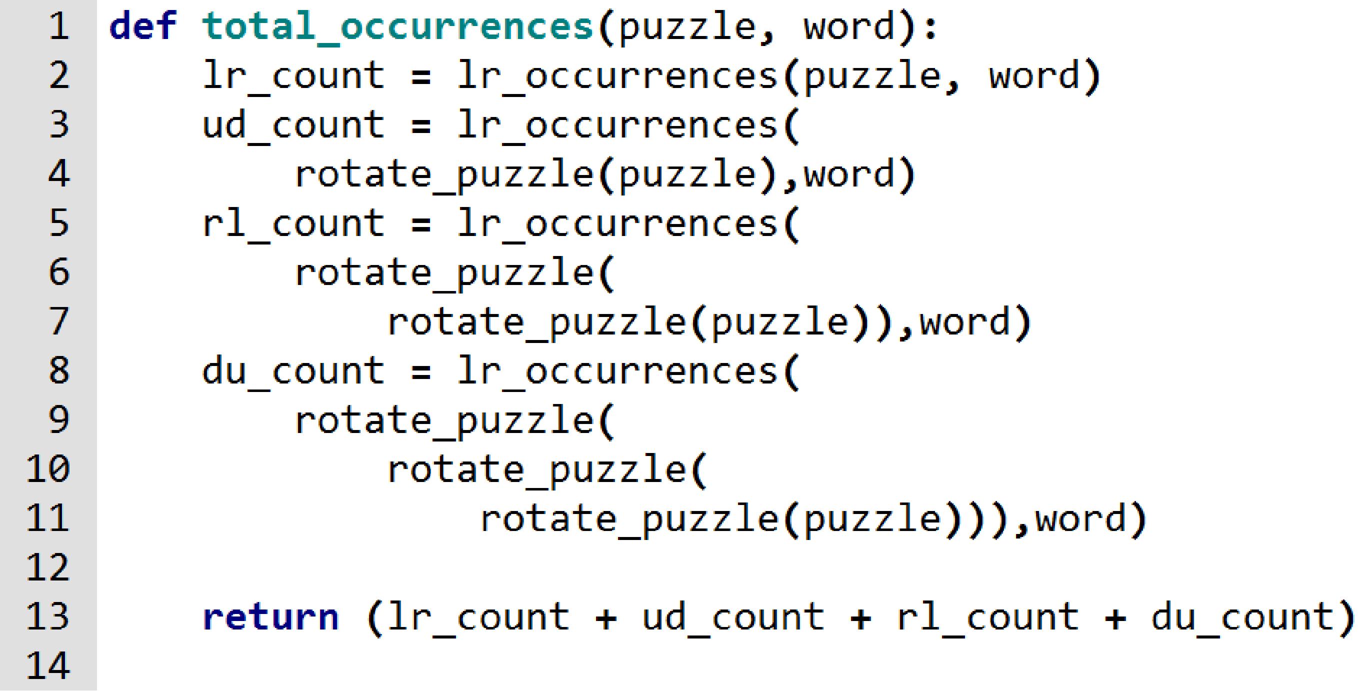
- At the end of the tutorial, you will be able to …

1. Get a full mark on Quiz 3 (~20 mins)
    - the quiz is very hard ☹
    - the way we do it will make it a freebie ☺

2. Work with lists and strings in the memory model (~20 mins)
    - a few examples
    - introduction to python tutor

QUIZ 3

- Function 1 (original version)

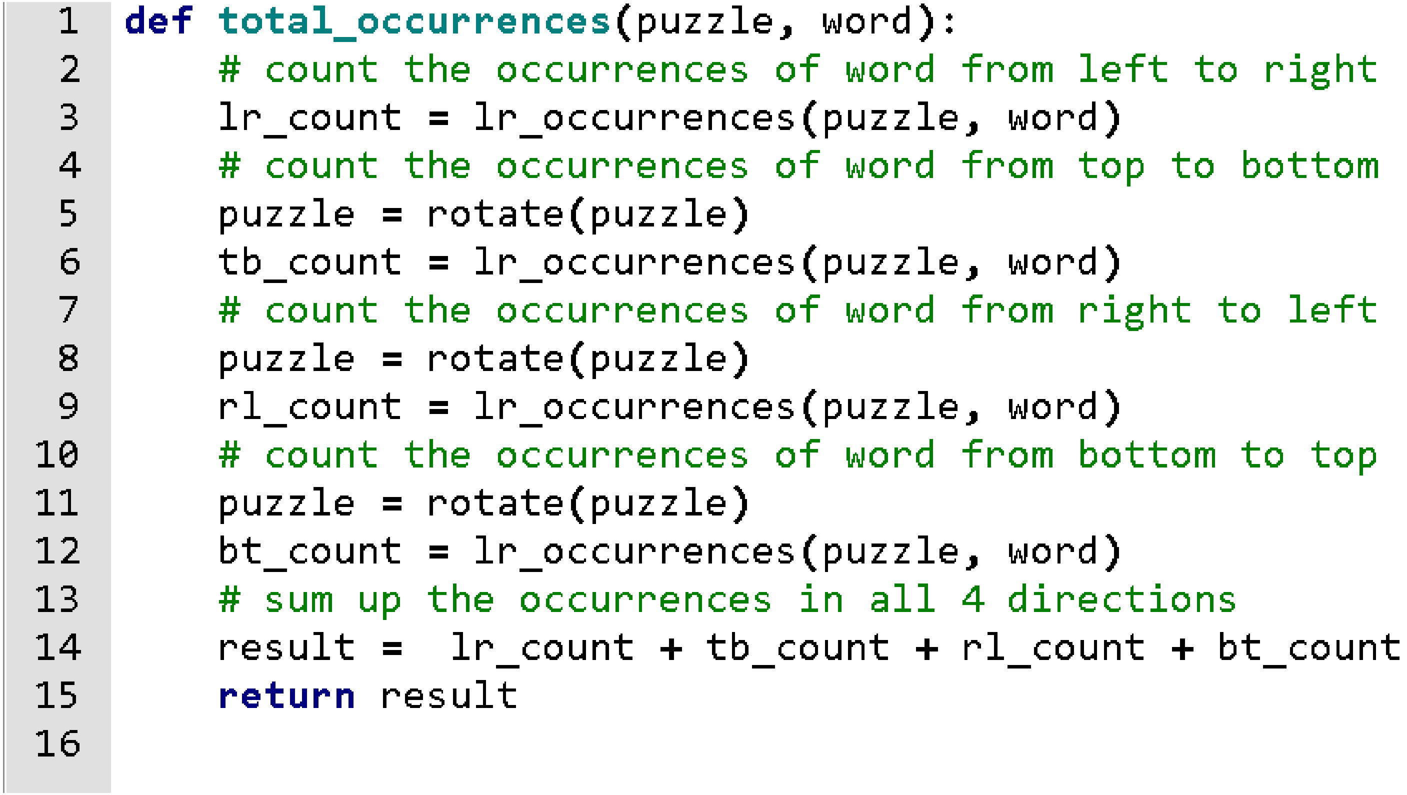


```
def total_occurrences(puzzle, word):
    lr_count = lr_occurrences(puzzle, word)
    ud_count = lr_occurrences(
        rotate_puzzle(puzzle),word)
    rl_count = lr_occurrences(
        rotate_puzzle(
            rotate_puzzle(puzzle)),word)
    du_count = lr_occurrences(
        rotate_puzzle(
            rotate_puzzle(
                rotate_puzzle(puzzle))),word)

    return (lr_count + ud_count + rl_count + du_count)

```

Problem: too many rotations

QUIZ 3

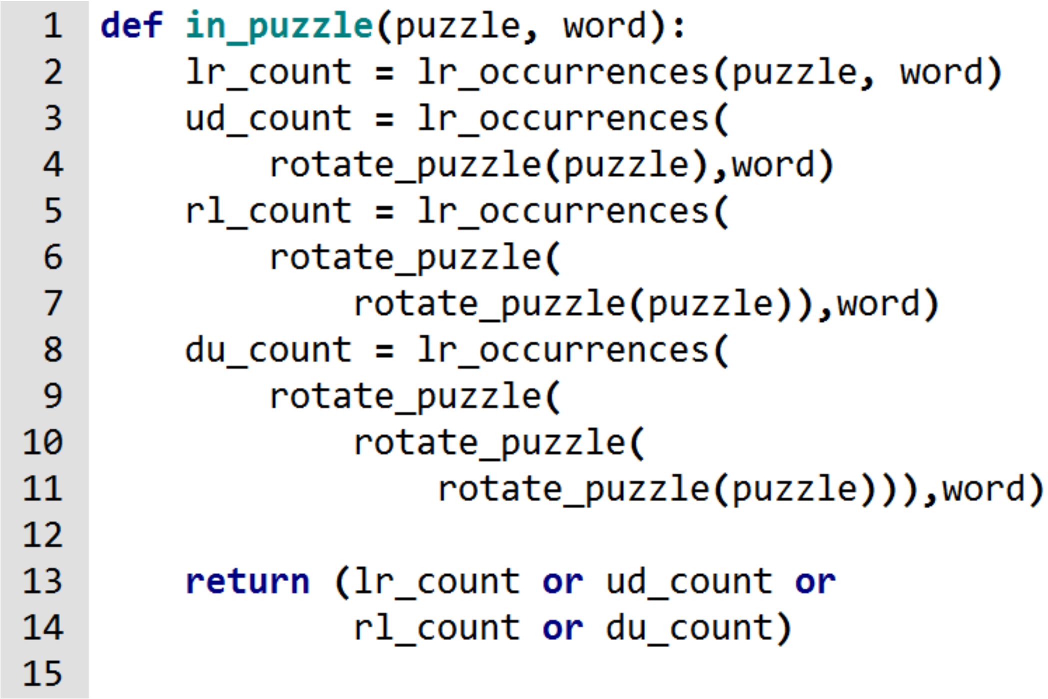
- Function 1 (improved version)



```python
def total_occurrences(puzzle, word):
    # count the occurrences of word from left to right
    lr_count = lr_occurrences(puzzle, word)
    # count the occurrences of word from top to bottom
    puzzle = rotate(puzzle)
    tb_count = lr_occurrences(puzzle, word)
    # count the occurrences of word from right to left
    puzzle = rotate(puzzle)
    rl_count = lr_occurrences(puzzle, word)
    # count the occurrences of word from bottom to top
    puzzle = rotate(puzzle)
    bt_count = lr_occurrences(puzzle, word)
    # sum up the occurrences in all 4 directions
    result =  lr_count + tb_count + rl_count + bt_count
    return result

```

QUIZ 3

- Function 2 (original version)

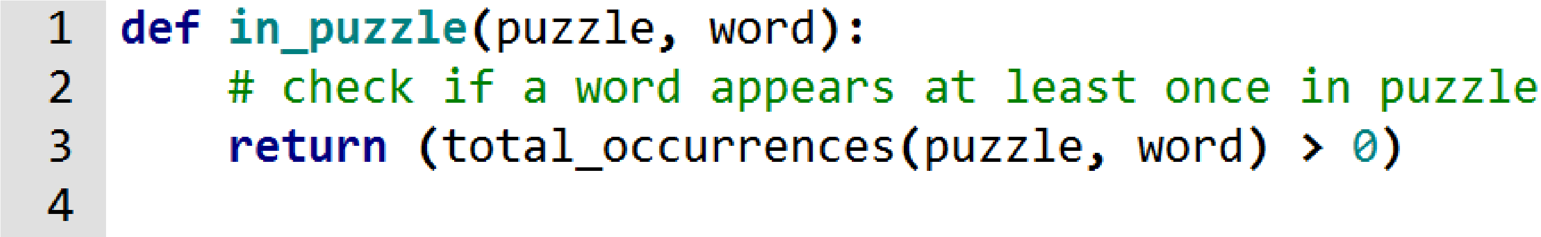


```python
def in_puzzle(puzzle, word):
    lr_count = lr_occurrences(puzzle, word)
    ud_count = lr_occurrences(
        rotate_puzzle(puzzle),word)
    rl_count = lr_occurrences(
        rotate_puzzle(
            rotate_puzzle(puzzle)),word)
    du_count = lr_occurrences(
        rotate_puzzle(
            rotate_puzzle(
                rotate_puzzle(puzzle))),word)

    return (lr_count or ud_count or
            rl_count or du_count)

```

Problem: too many duplicated code

QUIZ 3

- Function 2 (improved version)

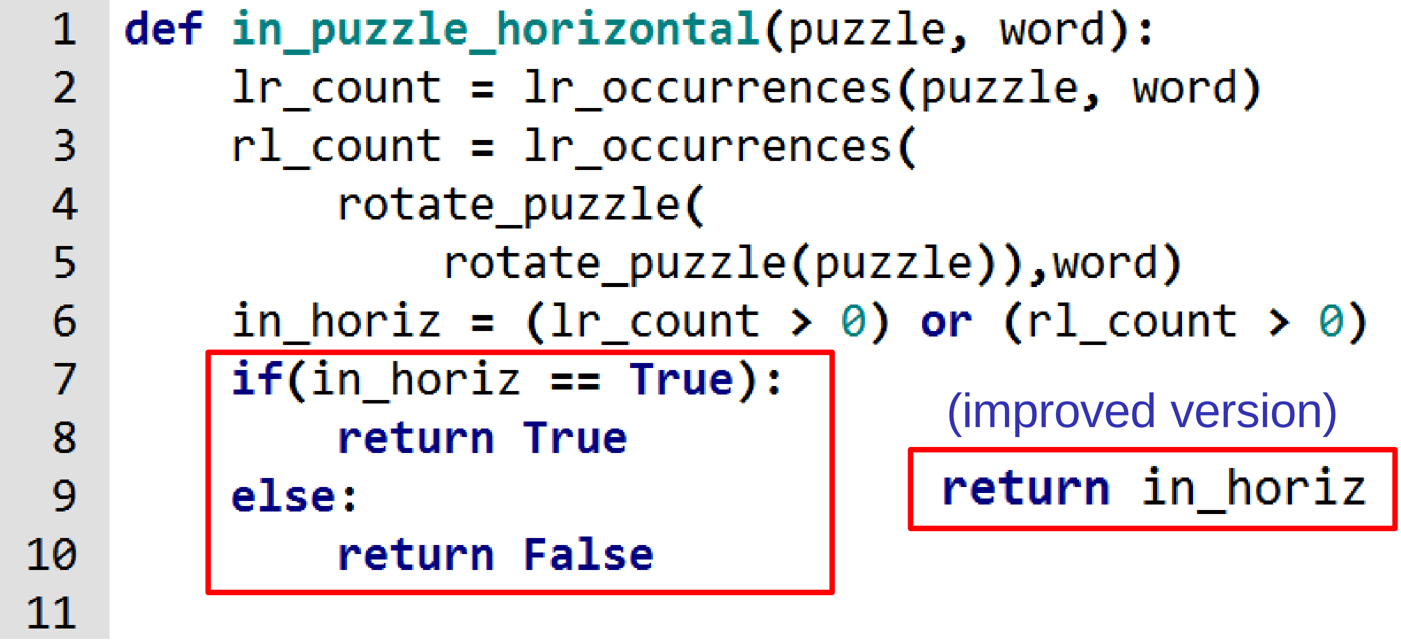


```python
def in_puzzle(puzzle, word):
    # check if a word appears at least once in puzzle
    return (total_occurrences(puzzle, word) > 0)

```

Don’t forget to use existing function(s)

QUIZ 3

- Function 3 (original version)



```
def in_puzzle_horizontal(puzzle, word):
    lr_count = lr_occurrences(puzzle, word)
    rl_count = lr_occurrences(
        rotate_puzzle(
            rotate_puzzle(puzzle)),word)
    in_horiz = (lr_count > 0) or (rl_count > 0)
    if(in_horiz == True):
        return True
    else:
        return False

```

(improved version)

```
return in_horiz
```

Problem: using if-statement

in_horiz is already a Boolean, so return it directly

MEMORY MODEL

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

- Output

```
1 2
```

MEMORY MODEL

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

- Output

```
[99, 2, 3]    [99, 2, 3]
```

MEMORY MODEL

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

- Output

```
?ello Hello
```

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

```
[99, 2, 3]    [98, 2, 3]
```

MEMORY MODEL

- Example 5

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

- Output

```
[99, 2, 3]     [98, 2, 3]
```

MEMORY MODEL

- Example 6

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

- Output

```
[[99, 98], 3]     [[99, 98], 3]
```

MEMORY MODEL

- Example 7

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

def mutator2(x):
    x[0][0] = "MUTATED"

def cloner1(x):
    clone = x[:]
    clone[0] = "COPIED"
    return clone

def cloner2(x):
    clone = x[:]
    clone[0][0] = "COPIED"
    return clone

x = [['A', 'B'], 'C']
y = mutator1(x)
print(x, y)

x = [['A', 'B'], 'C']
y = cloner1(x)
print(x, y)

x = [['A', 'B'], 'C']
y = mutator2(x)
print(x, y)

x = [['A', 'B'], 'C']
y = cloner2(x)
print(x, y)
```

SUMMARY

- 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