Exam: *Tutorial Python for Linguists*

Teachers:
Emiel van Miltenburg & Marten Postma

December 15, 2015

This exam marks the end of the first part of the *Tutorial Python for Linguists*. It consists of 18 questions, worth 30 points in total. Each question is worth one point, unless stated otherwise. Your grade will be computed by dividing the number of points by 3. For example: 27 points corresponds to a 9 out of 10. After the Christmas break we will continue with the final assignment.

1 General

1. Please indicate for each of the lines whether they evaluate to `True` or `False`. (3 points)

```
not (1 and 0 != 1)
not (10 == 1 or ’0’)
not (1 != 10 or 3 == 4)
("testing" == "testing") and []
1 == 1 and (’False’ or 1 == 0))
3 == 3 and (not ("testing" == "testing" or "Python" == "Fun"))
```

2. Please explain for each of the problem cases below why they produce an error. (5 points)

```
# Problem 1
l = [’a’,’b’,’hello’,[’a’,’b’,’c’]]
l.extend(10)
print(l)

# Problem 2:
’Summer’ + 1

# Problem 3:
d = {’a’: [],
     [1]: ’b’}

# Problem 4:
word = ’laptop’
word[-7]

# Problem 5:
def multiply_and_square(n, factor = 5):
    def square(n):
        return n * n
    result = square(factor * n)
    return result

square(5)
```

3. Which built-in function would you use to check whether at least one of the boolean expressions in a list evaluates to `True`?

4. What is the difference between the built-in function `sorted()` and list-method `sort()`?

5. Write out the dictionary generated by: `dict(enumerate(range(2), start=1))`. Use the curly brace notation, e.g. `{'bacon': 4, 'eggs': 2, 'spam': 9}`.

6. What does the result of the following list comprehension look like? `[i for i in range(5,9,2)]`

7. What are the values of `result_1` and `result_2` after executing the code below? (2 points)

```
a = {1, 2, 3}
b = {6, 4, 3}
result_1 = a.union(b)
result_2 = a.intersection(b)
```

8. What is the value of `result` after executing the code below?

```
a_list = [4,10,5,7,2,12]
result = a_list.append(4)
```

The exam is continued on page 3!

2 A larger bit of code

Read the code below, then answer the following questions:

9. What type of function is `annotations_for_line`?
10. What does the `continue` statement do?
11. What would happen if you replaced `continue` with `break`?
12. What is the **type** and **value** of `description` after line 11 is first executed?
13. What is the **type** and **value** of `category` after line 11 is first executed?
14. What is the **type** and **value** of `x` after running the code below?
15. How could you shorten line 8?
16. What would happen if line 3 were removed?
17. What would happen if line 14 were removed?

```
 1  def annotations_for_line(line):
 2      parsing = False
 3      current_annotation = []
 4      for char in line:
 5          if char == ’[’:
 6              parsing = True
 7              continue
 8          if parsing == True:
 9              if char == ’]’:
10                  annotation = ’’.join(current_annotation)
11                  category, *description = annotation.split()
12                  description = ’ ’.join(description).lower()
13                  yield category, description
14                  current_annotation = []
15                  parsing = False
16              else:
17                  current_annotation.append(char)
18
19  line = "[/EN#6/people A man] in [/EN#7/clothing a blue shirt]."
20  annotations = annotations_for_line(line)
21  x = next(annotations)
```

3 Writing code

18. Write a function called `overlap` that takes two filenames as its input, and that returns a `list` or a `set` (you can choose) containing the lines that are shared by both files. (6 points)

    - Be sure to document your function!
    - We will only check whether your program works. It doesn't have to be perfectly optimized.
    - Use clear variable names. So not `x` or `smurf`.

End of your exam, happy holidays!