Workshop

**Python Programming for Linguists**

**Exercises**

2020, Ingo Kleiber

There are multiple ways you can approach these exercises. However, it is best if you actually try to write some code! You can do this on ***Google Colab*** (in any notebook or in an empty one, e.g., “playground”) or in your **own development environment** (see Video “*Setting Up Your Development Environment”).* If you do not have the time or resources, I want to encourage you to think about these problems, even without writing out some code.

Please be aware that some of these exercises are very challenging for beginners. Please do **not feel disheartened** by them! You can always look at the **provided solutions** and use them as a starting point for your own exploration.

### Working with Files, Texts, and Regular Expressions

### Exercise 6 – Slicing and Modifying

Take the string below and print the third word (can) in uppercase without actually typing the word into the print function.

String: ‘Python programming can be fun.’

Expected Result: ‘CAN’

Avoid print(‘CAN’)!

### Exercise 7 – Counting Tokens

Write a function that takes a path to a file and returns the number of tokens in that file.

You can use /data/tokenize/simple.txt to test your solution. For this file, your function should return 6 or 7 tokens, depending on whether you count punctuation marks as tokens.

If you want to keep experimenting, try to write a tokenizer that manages to tokenize /data/tokenize/challenge.txt. While there is no clear solution to this task, many state-of-the-art tokenizers end up with 16 tokens (13, if we are not counting punctuation marks).