Exercise Sheet 08 – Practical Python

Sebastian Höffner Aline Vilks

Deadline: Mon, 29 May 2017 08:00 +0200

Submission

By the end of this sheet you will have a number of different files to submit. In Stud.IP you will have a directory for your own group, please upload them there. It is easier for you if you just archive (preferably zip) all files and upload your archive, but it is okay if you upload them one by one.

Exercise 1: There's always more than one way to solve a problem!

Can you write code which performs the following tasks by using

a) for loops with accumulators,
b) lambdas, map and filter (you should take a look at `itertools` for 3 and 4, don't use explicit loops),
c) list comprehensions?

Name your file `manyways.py`.

1. Convert the list `string.ascii_lowercase` into a list of its ascii values (use `ord(x)`). `[97, 98, 99, ...]`.
2. Create a list which contains all numbers except for those divisible by 3 or 5, up to 100. `[1, 2, 4, 7, 8, 11, ..., 98]`.
3. Create a list which contains all pair-wise permutations of the numbers 1, 2, and 3. `[(1, 1), (1, 2), (1, 3), (2, 1), ..., (3, 3)]`.
4. Create a list which contains the sums of all pair-wise permutations of 1, 2, and 3. `[2, 3, 4, 3, ..., 6]`.

Note: To use `string.ascii_lowercase`, you need to import `string`.

Exercise 2: Passing functions

In the previous exercise you already passed functions to e.g. `map`. There are several other functions in the Python library which expect functions, for example the `sorted` function.

Inside the module `carsorter`, write a function *outside the scope of the class* `Car` which allows the `sorted` function to sort the list of cars by comfort.

1. Download the Car Evaluation Data Set[1] from the UCI Machine Learning Repository[2].
2. The code to read it is already there. No need to do anything.
3. Write a function `comfort_evaluation(car)` which calculates a measure of comfort.
4. Sort the cars using the `comfort_evaluation` and the `sorted` function.

Note that the comfort values are somewhat arbitrary:

```
doors: 2, 3, 4, 5more.
persons: 2, 4, more.
lug_boot: small, med, big.
```

As a simplification, convert them to numerical values and just take the sum. E.g. a car with three doors, four seats (= persons), and a small luggage boot would have a value of `2 + 2 + 1`.

[1] https://archive.ics.uci.edu/ml/machine-learning-databases/car/car.data
[2] https://archive.ics.uci.edu/ml/datasets/car+evaluation