

Hands-on Python – Exercises – Day 4

Franziska Kranz

12.09.2017 – 15.09.2017

Exercise 1

Write a program that:

- Asks the user for a string, then creates the following dictionary. The values are the letters in the string, with the corresponding key being the place in the string.
- Replaces the entry whose key is the integer 3, with the value "Pie".
- Asks the user for a string of digits, then prints out the values corresponding to those digits.

Exercise 2

- Use a list comprehension to create ['ab', 'ac', 'ad', 'bb', 'bc', 'bd']
- Use the slice operator to extract ['ab', 'ad', 'bc']
- Use the slice operator to extract ['bd', 'bb', 'ac']
- Copy the above list and add 'ca' into the copied list such that the original is still missing it.

Exercise 3

- Define a program, that asks about input strings and checks if they are a valid email address
- Define a program, that reads in a fasta file (download the sample set from StudOn) and finds all occurrences of a subsequence, that starts with an A, then has a C or T, followed by up to 3 arbitrary letters, then has an optional T and ending with a G It should print the number of occurrences

Exercise 4

Create a `sleep_decorator` that makes sure, that before a function call we have to wait for 2 seconds. Write a simple printing function and decorate it with the `sleep_decorator`. Call this function several times (for example in a loop) and test if the waiting works.

Exercise 5 - Numpy

We have to do this exercise inside a Jupyter notebook, as numpy is not installed here. Go to try.jupyter.org and start a new Python3 notebook.

Slicing

- Create a 8×5 matrix with the numbers from 0 to 39
- Use slicing to get the following subparts of the matrix:

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39

Linear Algebra

- Have a look here: <http://docs.scipy.org/doc/numpy/reference/index.html>, especially the linear algebra part
- Create a zero vector of size 10 (use numpy functions!)
- Create a 3x3 matrix ranging from 0 to 8
- Write a function for solving a linear system $Ax = b$. You get A and b as input and should return x - Compare your result with the result of `numpy.linalg.solve(M,b)`

Exercice 6 - Matplotlib

We have to do this exercise inside a Jupyter notebook, as numpy is not installed here. Go to try.jupyter.org and start a new Python3 notebook. To see the plotting results inside the notebook, use a cell with the command `%matplotlib inline`.

Simple function plotting

- Plot the functions x^2 and x^3 in the interval $I = [-5, 5]$ (use `numpy.linspace`).
- Use different colors and line styles for the two functions.
- Add a title, axes labels and a legend.

Image manipulation

In the end of the lecture we saw, that an image in matplotlib is nothing else than a 2D numpy array. So we can use e.g the slicing operator to manipulate the image.

- Load the image like in the lecture, but the colored version
- Have a look at the information stored for one pixel, e.g. `face[0,0]`
- Create a 20 pixel wide colored frame, by setting the pixels on the rim.

Exercise 7

Solve some exercises from <http://codingbat.com/python>.

Exercise 8

Have some fun with my `hangman.py`, maybe you have also ideas for implementing a minigame :)