



Week 7

Introduction to Programming and Numerical Analysis

Ebener, Luca

UNIVERSITY OF COPENHAGEN



Overview

- About me
- Exercise classes
- Programs
- DataCamp
- Takeaways week 7

About me

Headers

- Last semester of master programme "Research in Economics and Finance" at University of Vienna
- Spent third semester on exchange at KU
- Interested in monetary and computational economics

Contact

- wmn553@econ.ku.dk

Exercise Classes

- I present some key points of the week or hints how to tackle the problems
- You work on the problem sets alone or in groups and I go around answering questions
- Happy to answer your questions via mail
- You can send me topics you like to repeat at the beginning of the class in advance
- Open for suggestions

Rest of the Semester

- First two classes: DataCamp and problem set 0
- Seven classes with problem sets
- Five classes dedicated to your projects
- Exam consists of three assignments during the semester and a final exam assignment

Programs

Installation

- Follow the guide on the course webpage

Programs

- Anaconda: Comprehensive Python distribution with pre-installed packages and environments management for streamlined development.
- VS Code: Versatile and lightweight code editor with extensive Python support, including debugging capabilities and extensions for Jupyter notebooks integration.
- Git: Distributed version control system enabling collaborative development, tracking changes, and managing code repositories efficiently, crucial for team projects and versioning notebooks.

DataCamp

Exercises

- Introduction to Data Science in Python
- Intermediate Python
- Python Data Science Toolbox (Part 1)
- Python Data Science Toolbox (Part 2)

General

- Deadline is the 25 of February
- Do it with time as the estimated length of the courses is 15 hours
- Try to understand not only how to complete the problems, but why all the code works

Atomic types

```
1 x = 1 # x references an integer type object with a value of 1
2
3 type(x) # type of x
4 id(x) # address of x
5 sys.getsizeof(x) # amount of memory used by x
6
7 str(x) # parses into string
8 float(x) # parses into float
9 int(x) # parses into integer
10 bool(x) # parses into boolean
11
12 # arithmetic operators are e.g. +,-,*,/,**,%
13 # augmentation operators are e.g. +=,-=,*=,/=
14 # logical operators are e.g. ==,!=,<,<=,>,>=
```


Containers

```
1 x = [0,1,2,3] # x references a list
2 x[1:3] # gives out the second to third element of the list
3 x[: -2] # gives out the first to second to last element of the list
4 # if you change the data of a list that one variable refers to, you
   change all variables referring to that list
5 y is x # true if they contain the same reference
6
7 y = copy(x) # y refers to a copy of list x not the same reference
8 y = deepcopy(x) # necessary, when the list contains mutable objects
9
10 z = (1,2,3) # z is a tuple which is an immutable list that cannot
    be changed
11
12 q = {'one': 1, 'two': 2, 'three': 3} # q is a dictionary
```

Conditions and loops (1)

```
1 if bool1:
2     print('bool1_is_true')
3 elif bool2:
4     print('bool2_is_true')
5 else:
6     print('neither_bool1_nor_bool2_is_true')
7
8 for x in xlist: # iterates over elements in xlist
9     pass
10
11 [x*2 for x in xlist] # returns list where each element of xlist is
    doubled
12
13 while bool3: # executes until bool3 is false
14     print('watch_out_for_endless_loops')
```

Conditions and loops (2)

```
1 for x in range(6): # iterates over list from 0 to 5
2     pass
3
4 for i,x in enumerate(xlist): # x is the i'th element in xlist
5     pass
6
7 for x,y in zip(xlist,ylist): # x is the element of xlist and y the
8     element of ylist for every iteration
9     pass
10
11 for key in my_dict.keys(): # iterates over the keys
12     for val in my_dict.values(): # values
13         for key,val in my_dict.items(): # key-value pairs
14             pass
```