Exercise 11: GUI Programming (1)

The objective of this exercise is to create a dialog to control the simulation of the movable pendulum (picture: see last exercise). The parameters of the model (mass, rope length) should be adjustable in the graphical user interface. as well as the simulation duration and the simulation step size. The execution of the simulation shoud be triggered via a button.
Also, the program should be able to save and load settings and parameters via the module `configparser`.

Exercise 11.1: GUI for Controlling the Simulation (simgui.py)

1. Open the file simgui.py. Create labels and input fields for the parameters `l` (pendulum length), `m1` and `m2` (masses for carriage and pendulum). Assign reasonable values to the fields as defaults.

2. Do the same for simulation step size and simulation duration.

3. Create two buttons for simulating and exiting the program.

4. Insert all widgets into a (7x2) grid layout (see: example-code/gui-example4.py resp. doc) and right-justify the buttons. Try the GUI application now.

5. Give each `LineEdit` a validator for numeric values.

6. Set the text alignment in the LineEdits to right-aligned.

7. Connect the simulation button with the `simulate` function; all of the following tasks are to be processed within the `simulate` function

8. Get all values from `LineEdits` and convert them to floating point values (`float(...)`).

9. Create a time axis with `arange` (from scipy) (use parameters from GUI!).

10. Simulate the system with odeint:
    `res = odeint( rhs, y0, t, args=(m1, m2, l) )`
    The initial values `y0` (corresponding to $[x(0), \varphi(0), \dot{x}(0)\dot{\varphi}(0)]$) can be freely chosen.

11. Create two subplots on <fig> and plot the results for trolley and load in one subplot each. Pay attention to axis labeling, legend, auxiliary lines etc.

Exercise 11.2: Saving and Loading the Configuration

1. Create two more buttons for loading and saving.

2. Connect them with the functions `openFile` and `saveFile`.

3. Complete both functions (start with saveFile!) with calls to the appropriate dialogs for file names and write/read the data with the `configparser` module:
   Write:

   ```
   c = configparser.SafeConfigParser()
   c.add_section("Parameter")o
   c.set("Parameter", "m2", str(mass1Edit.text()))
   ```

   Read:

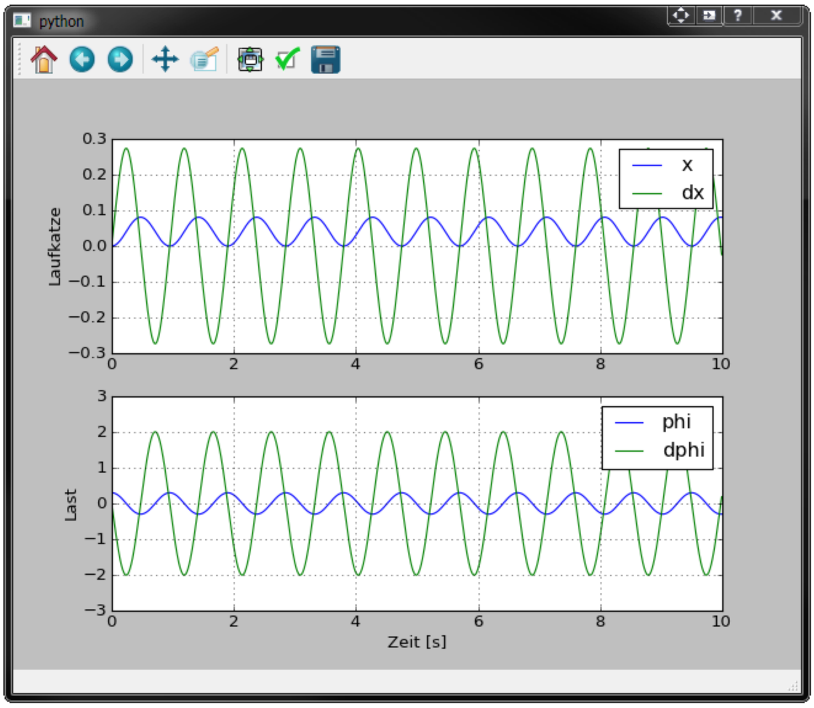
   ```
   mass1Edit.setText(c.get("Parameter", "m1"))
   ```

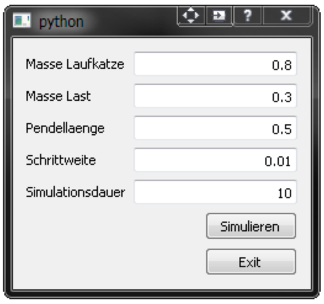
o Result and example configuration file see Fig. 2.

Exercise 11.3 (optional)

- Extend the program to enter the initial values of the simulation.

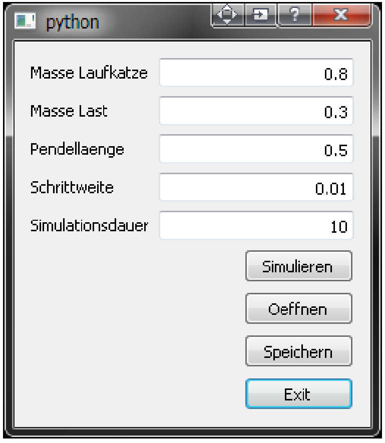
for exercise 11.1



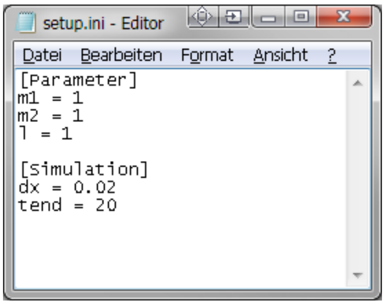


Masse Laufkatze	0.8
Masse Last	0.3
Pendellaenge	0.5
Schrittweite	0.01
Simulationsdauer	10

for exercise 11.2



Masse Laufkatze	0.8
Masse Last	0.3
Pendellaenge	0.5
Schrittweite	0.01
Simulationsdauer	10



```
[Parameter]
m1 = 1
m2 = 1
l = 1

[Simulation]
dx = 0.02
tend = 20
```