Karlsruher Institut für Technologie
Institut für Theorie der Kondensierten Materie
Dr. Andreas Poenicke
http://comp.physik.kit.edu/Lehre/ERA/

SS 2022

11.04.2022

era@physik.kit.edu

Exercise Sheet 4 for "Einführung in das Rechnergestützte Arbeiten"

The exercise are supposed to be done on the Jupyter server of the faculty. Prerequisite is only a browser and an account for the pool.
The jupyter server is available under `https://jupytermachine.etp.kit.edu`.

1 Jupyter

Before starting with Python it is useful to get to know the working environment.

- Open the link to the Jupyter server with a web browser and login in with your KIT credentials. The first time you do this, you will be asked about `Server Options`. Choose `Python` and start Jupyter.
- Ignore the main window with the `Launcher` for now. Next to it should be the `File Browser`, if not open it from the menu on the left. Create a new folder with the name `ERA` open it.
- You can find a Jupyter notebook containing a Python introduction on the lecture homepage. Transfer this file with `Upload Files` into the `ERA` folder. Opening this file create a new tab with the introduction. You can use this as a reference while working on this sheet.

  (Activating the `Table of Contents` on the left causes the `File Browser` to be replaced with a table of contents.)
- Create a new notebook from the launcher. This is automatically named `Untitled.ipynb` and saved. Rename it to `sheet4.ipynb`.

  (Keep in mind that the right mouse button often opens helpful context menus in the Jupyter webinterface.)
- Add a markdown cell with the `heading` "Sheet 4" and a subheading "Task 2". (Help regarding the markdown syntax can be found under `Help` → `Markdown Reference`).

2 Python

Now we are getting started with Python. The cells have to be of the type `Code` for that.

- In one cell, assign the variable `a` an integer value and then use `print()` to output the string "`->`" and then the result of `5*a`.

  Repeat this for a floating point number and a string.
- Use the next cell for inputs. Print the string "`A number:`", get an input and assign it to the variable `x` . Which type does `x` have after the input?
- Convert `x` now to the type `float` and check the variable type again.

3 A first Python script

Create a new notebook and rename it to "exercise3.ipynb".

1. Import the module `math` and test by printing the variable `math.pi` if it worked.
2. Let the user enter a number "Summation limit:" and assign the input to the variable `n_max`. Convert the value of the input to the data type `int` directly before the assignment. Afterwards print the content of the variable.

   What happens if you enter "Hallo" instead of a number?
3. Now comes a first conditional command: Test if `n_max > 12` holds and if so, print the message ``` ``n_max may not be greater than 12!'' ```.

`math` has a function for calculating $\sin(x)$, but here it should be implemented manually. For this the Taylor series should be used:

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

4. First, write a `for`-loop, in which n goes from 0 to (including) n_{max} and print n in every step.
5. Calculate $y = \sin(\pi/2)$ by setting $x = \pi/2$ and $y = 0$ before the loop and then in the loop add the terms of the Taylor series to y one after another. Also print the value of y at every step of the loop. (The function `math.factorial()` calculates the factorial.)
6. Repeat, but let the loop run backwards, i.e. from n_{max} to 0. Does the result change? *Hint:* Step sizes can also be negative.

4 Custom functions

To build this calculation into a program like this is cumbersome.

Define a function `my_sin(x)` that calculates and returns $\sin(x)$ in this way. Set n_{max} to the fixed value 12 inside the function. Test the function by calculating `my_sin(x)` for different values of x.

Custom functions are typically defined at the beginning of a script after the `import` statements. Move the cell with the function definition to the top of the notebook.