Training Assignment 02

NUMA01: Computational Programming with Python

Malin Christersson, Robert Klöfkorn

The purpose of this training assignment is to work with `list` and `for` loops.
You should start working on these training assignment during the scheduled computer lab hours and complete them on your own at home.
This assignment has 14 tasks.

Warming-up Exercises

Task 1

Read the `debugging` documentation for spyder: https://docs.spyder-ide.org/5/panes/debugging.html. While there check for other interesting topics that you want to know about your editor. If you are using a different editor then find out how to do debugging in your editor.

Task 2

Implement the following for loop:

```
x=0.5 # x_0
for i in range(200):
    x=x**2  # x_i
print(f'The result after {i+1} iterations is {x}')
```

and inspect each iteration with the `debugger`.
Note: For $x_0 < 1$ the above iteration is called a **fixed point** iteration (see below), because for all starting values $x_0 < 1$ this will converge to 0. For $x_0 > 1$ it will diverge. What about $x_0 = 1$?

Exercises

Task 3

Recall the `if` statement from the lecture and check whether the expression

$$x^2 + 0.25x - 5$$

is zero for $x = 2.3$.

Task 4

Try out this code segment from the lecture

```
L = [1, 2]
L3 = 3*L
```

and test the following commands:

```
L3[0]
```

```
L3[-1]
```

```
L3[10]
```

Task 5

What does the following command do?

```
L4 = [k**2 for k in L3]
```

Task 6

Concatenate `L3` and `L4` to a new list `L5`.

Task 7

Use the command `range` and a list comprehension to generate a list with 100 equidistantly spaced values between (and including) 0 and 1.

Task 8

What does this code do?

```
s = 0
for i in range(0, 500):
    s = s + i
print(s)
```

and this one

```
ss = [0]
for i in range(1, 500):
    ss.append(ss[i-1] + i)
```

Compare the last element in the list `ss` with `s`. What is the value of `i` after the execution of the `for` loop?

Task 9

Again, like in Task 6, set up a list with the name `xplot` which contains 100 equidistant values between 0 and 1. But this time, use a `for`-loop with a counter instead of a list comprehension. Pay attention to the `append`-method for lists. You might need to generate an empty list first.

Task 10

Set up a list `yplot` which contains the values $\arctan(x)$ for all the x in `xplot`.

Task 11

Make a plot of `yplot` versus `xplot`. Use for this the command `plot`. Depending on your environment, you might also need to run `show()` to show the plot.

Task 12

Compute

$$\sum_{i=1}^{200} \frac{1}{\sqrt{i}}.$$

Task 13

Read Chapter 1 up to (not including) *Encapsulating code with functions* and Chapter 3.1 (Lists) of the course book to repeat the things that you have used so far.

Extra Task 14 (in case you want an extra challenge)

Now we use lists for recursions. Consider the recursion formula:

$$u_{n+3} = u_{n+2} + ha\left(\frac{23}{12}u_{n+2} - \frac{4}{3}u_{n+1} + \frac{5}{12}u_n\right)$$

with $n = 0, \ldots, 1000$, $h = 1/1000$ and $a = -0.5$.

1. Create a list `u`. Store in its first three elements e^0, e^{ha} and e^{2ha}. These represent u_0, u_1, and u_2 in the above formula. Build up the complete list from the recursion formula.
2. Construct a second list `td` in which you store the values nh, with $n = 0, \ldots, 1000$. Plot `td` versus `u`. Make a second plot in which you plot the difference, i.e. $|e^{at_n} - u_n|$, where t_n represents the values inside the vector `td`. The `abs` function computes the absolute value of a number. Set axis labels and a title (investigate the commands `xlabel`, `ylabel` and `title`).

If you have time over, plot both the approximation and the exact solution in the same figure and compare the result. The approximation is quite good, so you might want to plot the difference between the approximation and the exact solution instead.

The mathematical background to this problem will be taught in later courses when you became familiar with differential equations.

Good luck!