Lab 01

AUTHOR
Christian Osendorfer

IT@HAW Landshut

All of the following should work (on your local machine) for you:

- xmail
  - Setup a mail client (Outlook, Evolution (Linux), Apple Mail (Apple)).
    - Add your semester schedule to the respective calendar.
    - Overall, not a high priority, but you should have a least one way to send email (i.e. through the web mailer).
- moodle
  - Can you log in? Do you see all your courses?
- Check out (only briefly) syncandshare.
  - *Cloud* option from LRZ for sharing (large) files.
  - Provides a free office suite (onlyoffice).
  - Read more details.
- Eduroam (only on your laptop/mobile device at HAW Landshut!)
  - See here.
  - Works also under Linux/Mac, see link
- VPN (Virtual Private Network, only on your own machine)
  - See here.
  - If you are not getting it to work after 10 minutes, leave it like so, not important.
- Can you login at studilab?
- How do you manage your passwords?
  - bitwarden (open source), 1Password, KeePass, …

Computer Setup

Choose the parts that are relevant for you (and follow the steps sequentially).

Windows

- Setup WSL.
  - Restart your computer after this step.
- Setup vscode with WSL extension.
  - Restart your computer.
- Start vscode and connect to wsl from vscode
  - Open vscode command palette and find correct command for wsl.
  - If you see an error regarding missing wsl installation, restart your computer.
- If you need to redo everything, first uninstall WSL.

Mac/Linux

- Install term2 (only for Mac).
- Install vscode from vscode website (Mac) or through your package manager (Linux):
  - E.g. in Ubuntu, `sudo apt-get install code` (assumes you are `sudo`-er).
  - Start vscode

All Platforms

- Install vscode extensions (python, pylance, jupyter, better comments).
  - Click on the flying square icon on the left side and search for these extensions, then click `install`. VScode may need to restart.
- Create new terminal through vscode command palette, and do all of the next steps in this terminal.
  - Check unix version: `uname`.
  - Check git: `git --version`.
  - If missing on Mac, install using the homebrew option.
  - Check ssh: `ssh yourhawusername@ssh.haw-landshut.de` , use your standard password.
    - You are now logged into your folder at the ssh server at HAW Landshut, which also represents `zds.haw-landshut.de`. Data in this folder is stored for you here at HAW.
    - Type `exit` to log out again.
    - You should be on your local computer, in the terminal of VS Code, again.
  - Download miniforge as described.
  - Run downloaded shell script as in interactive install.
  - Create a new python environment that is named `p1`:
    - `conda create --name p1 python=3.8`.
    - What is `conda`? Try to find out a bit yourself. For example you can read through this.
  - Then type `conda activate p1` into your terminal, then start `python`. What does it say?
- Open a new (empty) python file and save (empty) as `my_first.py` using *Save* shortcut.
  - Use vscode command palette to open a file.
    - Find out about the *command palette* if you don't know what it its.
  - Select python interpreter `p1` for this python file (cf. command palette).

- Type your first python program (see the slides `01.pdf`) in the editor and save.
  - Run `python my_first.py` in terminal.
  - Run `python -i my_first.py` in terminal.
- Create jupyter notebook through vscode command palette.
  - Execute a cell (and connect to a python interpreter) and play around.
  - If this does not work after some trying (5 minutes), skip it.
- Work through this how-to.
- Have a quick glance at this (but don't install anything they mention), in order to see how to configure vscode for yourself. No need to do anything what they mention, just check it out! Maybe you want to see how you can change the editor font to something different, e.g. to Fira Code, or find a better theme.

Your second python program

Type in everything (also the comments!) exactly as is. In order to indent ("Einrücken") use the `TAB` key (or vscode does it for you automatically).

```
import operator # we don't know yet what this line does1

def another_add(x, y):
  '''
  Comments over multiple lines are done
  like so (or with """). Usually, every
  function signature is followed by a
  comment block, before the function body.

  The next few lines are so called 'doc tests'
  in python (https:realpython.com/pythondoctest/).
  These look like the interactive shell in python
  and resemble test cases that demonstrate correct
  useage and correct results for the current function.

  >>> another_add(3, 4) # should be 3+4
  7
  >>> another_add(4, 3) # should be 4+3
  7
  '''
  # we don't know the following conditional statement yet
  # but it might be easy to understand its semantics.
  if x ⩽ y:
    result = operator.add(x, y)
  else:
    # hmm, is this an add function?
    result = x - y
  return result
```

```
print_result = print(another_add(another_add(3, 4), another_add(4, 5)))
print(print_result)
```

- Run `python -m doctest my_second.py`
  - `-m doctest` instructs the interpreter to also run every doctest in the given file (after running the normal program).
  - Understand the various pieces of the output. Interpret your findings.