# 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.
- 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.
- 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.

- Setup vscode with WSL extension.
- Start vscode and connect to wsl from vscode
    - Open vscode command palette and find correct command to connect to wsl.

## Mac/Linux

- Install iterm2 (only for Mac).
- Install vscode from vscode website (Mac) or through your package manager (Linux)
    - E.g. in Ubuntu, `sudo apt-get install code`.
    - 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.
    - Check unix version: `uname`.
    - Check git: `git --version`.
        - If missing on Mac, install using the homebrew option.
    - Check ssh: `ssh your-haw-username@ssh.haw-landshut.de`, use your standard password.
    - Download miniforge as described.
    - Run downloaded shell script as in interactive install.
    - Create a new python environment `p1`: `conda create --name p1 python=3.8`
        - Then `conda activate p1`, then start `python`. What does it say?
- In vscode, *open* a new (empty) python file and save (empty) as `my_first.py` using *Save* shortcut.
    - Use vscode command palette to open a file.
    - Select python interpreter `p1` for this python file (cf command palette).
- Type your first python program (cf. slides) in the editor and save it.
    - Run `python my_first.py` in vscode terminal.
    - Run `python -i my_first.py` in vscode terminal.
- Create jupyter notebook through vscode command palette.
    - Execute a cell (and connect to a python interpreter) and play around.
- 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

```python
import operator # we don't know yet what this line does ...

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/python-doctest/).
  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.