Documentation for the software tool for analysing optical emission spectra

# Organization of Python code in the files

#!/usr/bin/env python3

# -\*- coding: utf-8 -\*-

"""Module documentation goes here

and here

and ...

"""

* **Python interpreter:** The first line is for \*nix users. It will choose the Python interpreter in the user path, so will automatically choose the user preferred interpreter.
* **File encoding:** The second one is the file encoding. Nowadays every file must have a encoding associated. UTF-8 will work everywhere. Just legacy projects would use other encoding.
* **Documentation:** And a very simple documentation. It can fill multiple lines.
* See also: <https://www.python.org/dev/peps/pep-0263/>
* If the file just implements a class, the documentation goes into the class documentation.

(Section above adapted from: https://stackoverflow.com/questions/1523427/what-is-the-common-header-format-of-python-files)

# Style Guide

The default style guide that is followed is the PEP 8. The full description can be found here: <https://pep8.org/>. Because it is a general style guide, there are complements for specific cases in the software:

Classes:

* Each class is defined in a separate file.
* The name of the class is the same as the name of the file (except the ending obviously)
* Methods of the class are named in…
  + … camelCase, if the method has to be recognized by the Qt-backend, e.g. events like dragEnterEvent
  + … snake\_case, if the method is independent of the Qt-backend, like file\_open

User Interface (UI)

* Elements are named in camelCase and follow the convention “type”+”name”, e.g. btnClear where “btn” is the type and “Clear” is the name of the button.
* Groups of elements are named in PascalCase, e.g. BtnParameters are several parameters grouped.
  + Might be important for loop-implementation
* Overview of abbreviations for naming elements
  + btn  button
  + cb  checkbox
  + list  list
  + tin  text input
  + tout  text output
  + act  action
  + menu  menu
  + fout  file output
  + dd  dropdown
  + bar  progress bar
  + lbl  label
  + mpl  plot from matplotlibary
  + layout  horizontal/vertical layout