

Python installation for very first steps

Attention: If you plan to use the tool pyxll (coupling python and Excel), you have to adapt your python installation to the installed Office (same bit width of the installation 32Bit/64Bit).

Contents

Python installation for very first steps.....	1
1 Anaconda	1
2 Packages.....	1
3 Excel.....	2
4 empir19nrm02 short introduction	3
4.1 Data sets.....	3
4.1.1 SPDs.....	3
4.1.2 RES.....	3
4.2 Functions	3
4.3 Jupyter	3
4.4 Future plans.....	4
5 Contribute Code	4

1 Anaconda

(python, jupyter): [Link](#)

- For configuration see: [Link](#)

2 Packages

You will find the packages on [GitHub](#). But usually an installation using “`pip install`” from the Anaconda command line is sufficient if there is not further contribution debugging is required.

Name	Description
empir19nrm02	Package implementing calculations relating to photometry and spectroradiometry the EMIR Project RevStdLED 19NRM02. https://www.ptb.de/empir2020/revstdled/home/
luxpy	Current version (V1.7) with f1Prime and SMCF f1PrimeDetector.ipynb
sigfig	Formatting numbers with significant figures (very helpful to state results from MU caculations)
MetroloPy	Simple tool for MU calculations from NRC IBudget.ipynb

METAS_uncLib	METAS UncLib (see SPE for METAS_uncLib) IBudgetMETAS.ipynb uncLibSlope.ipynb
jupyter-matplotlib (pip install ipympl)	Using matplotlib in jupyter
pyxls	Excel extension using python function. This package is necessary for the work of empir19nrm02 even if you do not use the excel<->python interface.

Example:

- Start the Anaconda command line (Start | Anaconda PowerShell Prompt (Anaconda3))
- With: `pip install sigfig` you install the sigfig package, which can be used using `import sigfig as sf` in your python script

3 Excel

Using Excel and python the tool pyxls is very helpful. (But is about 300\$ a year ☹). But the functions are really impressive as one can call specific python code direct from Excel.

Example:

```
@xl_func("numpy_array<float>    wl,    numpy_array<float>    rMatrixSPD:    numpy_array<float>",
auto_resize=True)
```

```
def py_CCT( wl, rMatrixSPD):
```

```
    CCT = lx.xyz_to_cct(py_XYZ(wl, rMatrixSPD), cieobs='1931_2')
```

```
    return CCT
```

With this code you have a new XLS-Function py_CCT (callable direct in cells) you get the CCT from a SPD (or an array of SPDs).

Installation

```
pip install pyxls
```

```
pyxls install (from the anaconda command line prompt)
```

```
pyxls-jupyter (Connection jupyter, pyxls und Excel → great to develop new functions) (pip
install pyxls-jupyter)
```

Config in: *User\AppData\Local\Programs\PyXLL*

```
pythonpath =
```

```
    ./examples
```

```
    localPath/pyxls → here you put your connection code in
```

Debugging: <https://www.pyxls.com/blog/debugging-your-python-excel-add-in/>

(Using additional JetBrains python IDE, if you need to develop new code or inspect the current one.)

Attention:

- The package pyxll (pip install pyxll) you need to compile and run the empir19nrm02 package is free of charge.
- The pyxll software you need to run Excel together with python you need to purchase if you want to use the software.

4 empir19nrm02 short introduction

This package is planned to implement the python functions necessary to show the principle implementation

4.1 Data sets

To make simulations, we need to work with Data-sets. Therefore the package provides a couple of data sets for SPDs (spectral power distribution of light sources) and RES (spectral responsivity data). All standard functions from the CIE the package will use from luxpy.

4.1.1 SPDs

- VL Detectors (a current collection, planned to be extended during the project and the compilation from the CIES025)
- Detectors with artificial noise and shifted VL Detectors for simulation and sensitivity analysis only.

4.1.2 RES

A couple of collections for LED based light sources.

- BB, RGB White, PT White LEDs, OSRAM Data sets, Monochromatic LEDs
- CIES025 All data sets for CIES025 calculations (RGB White, PTLED)
- PhotLED White LED data from the PhotoLED EMPIR Project

4.2 Functions

- f1Prime General implementation
- functions to support data plotting (e.g. based on seaborn)
- functions to support the evaluation of the MC simulation data
- McSpectrum

Functions to support MC simulations for spectral data.

4.3 Jupyter

The following Jupyter notebooks are used to test the functions and to implement demonstrations:

Notebook	Functions
f1PrimeDetector	Calculations to evaluate the properties of the different f1Prime definitions
OpexTest	What about f1PrimePrime
SPE	
Ibudget	Simple example for a measurement budget using metrology.
IBudgetMETAS	Simple example for a measurement budget using METAS unclib.

uncLibSlope	Slope caculation using METAS uncLib.
MCTest	Test for MC Simulations with spectral data (very beginning state)
EvalTest	Testframe It is planned to provide the datasets with reference data. The Jupyter notebook is planned to check the reference values for the test data. At the moment this is demonstrated on behalf of some simple tests only.
SPD_Test	Implement some tests for the SPD data sets.

4.4 Future plans

...

5 Contribute Code

- Install GitHub (create your own account)
- Get a copy of the code to your local computer
- Use “`pip install -e .`” from the anaconda command line (and the local directory of the package you will work with)
- Change/Debug code and contribute via pull requests.

6 Acknowledgements

This project (19NRM01 RevStdLED) has received funding from the EMPIR programme co-financed by the Participating States and from the European Union's Horizon 2020 research and innovation programme."