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

1 Software

1.1 Python

Python is a high-level, interpreted, general-purpose programming language. Chosen here for the free arability, widespread use and high number of scientific tools.

https://www.python.org/

1.2 Jupyter

Project Jupyter is a community run project that is developing and supporting the interactive computing products Jupyter Notebook, JupyterHub, and JupyterLab. supporting Julia, Python and R.

https://jupyter.org/

- For configuration see: Link

1.3 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment.

https://www.anaconda.com/products/distribution

1.4 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
pyxll	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

1.5 Excel integration

Using Excel and python the tool pyxll 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 pyxll

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

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

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

pythonpath =

./examples

localPath/pyxll → here you put your connection code in

Debugging: https://www.pyxll.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.

2 empir19nrm02 short introduction

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

2.1 Data sets

To make simulations, we need to work with Data-sets. Therfore 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.

2.1.1 *SPDs*

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

2.1.2 *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)
- PhotoLED White LED data from the PhotoLED EMPIR Project

2.2 Functions

- f1Pime 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.

2.3 Jupyter notesbooks

The following Jupyter notebooks are used to test the functions and to implement demonstrations (found in the folder: empir19nrm02\Jupyter):

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 metrolopy.
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.

2.4 Future plans

…

3 How to contribute code

3.1 Requirements

For effectively contributing to empir19nrm02 the following tools and procedures should be followed

3.1.1 *Git*

Git from https://git-scm.com/ "…is a free and open source distributed version control system designed to handle everything from small to very large projects…"

This is the software that is used to track and version the code.

3.1.2 *GitHub access*

- Install GitHub (create your own account) at www.github.com

- 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.

4 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."