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).

# Anaconda

(python, jupyter): [Link](https://www.anaconda.com/products/individual-d)

* For configuration see: [Link](https://stackoverflow.com/questions/35254852/how-to-change-the-jupyter-start-up-folder)

# Packages

You will find the packages on [GitHub](https://github.com/). 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 all the stuff for the EMIR Project. |
| 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](https://nrc-cnrc.github.io/MetroloPy/_build/html/index.html)  IBudget.ipynb |
| METAS\_uncLib | METAS [UncLib](https://www.metas.ch/metas/en/home/fabe/hochfrequenz/unclib.html) (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

# Excel

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.

# empir19nrm02 short introduction

This package is planed to implement the pyton functions necessary to show the principle implementation

## Data sets

To make simulations, we need to work with Data-sets. Therfore the package provides a couple of data stes 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.

### 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.

### 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

## 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.

## 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 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. |

## Future plans

…

# 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.