PYTHON SCRIPT FOR MIROC BINARY OUTPUT READING

This script was written in **Python 3.6** and contains a simple function to read a single binary file contained in the repository. Each file is written sequentially and contains a number of records equal to the number of days for the given year: 365 for normal years and 366 for leap years. The output array always contains 366 records to allow for loop-analysis of all 100 members (years) in a standardized manner. The 60th record index (29 February) is filled with NaN values for non-leap years. Member (year) range is 2011 to 2110 (100 members).

There are 10 available geophysical and dynamical quantities with three kinds of gridding:

1) 3D data (pressure levels = 31, latitude = 64, longitude = 128)
    a. ozone field ("xo3_P", volume mixing ratio, ppv)
    b. temperature field ("T_P", kelvin, K)
    c. zonal mean zonal wind ("u_P", m.s^{-1})
    d. air number density ("air_P", m^{-3}, to convert O_3 VMR into number density / partial pressure in DU.km^{-1})
2) Integrated data (latitude = 64, longitude = 128)
    a. Total column ozone ("dobson", DU)
3) Zonal data (pressure levels = 31, latitude = 64)
    a. EP flux divergence ("epfdiv", m.s^{-2})
    b. EP flux meridional component ("epfy", kg.m^{-1}.s^{-2})
    c. EP flux vertical component ("epfz", kg.m^{-1}.s^{-2})
    d. Residual mean meridional wind ("v_sta", m.s-1)
    e. Residual mean vertical wind ("w_sta", m.s-1)

The script reads from a binary file into a **numpy** array. Each record in the binary file has a header for metadata that is also included in the output Python dictionary. Grid points values for the T42 resolution and 31 pressure levels of the CCM output are defined in the script.

How to use:

- Under the Python command line (or iPython prompt, etc.) import the function "CCM_readmodel" from the (same name) script file:

```
from CCM_readmodel import CCM_readmodel
```

- Then call CCM_readmodel with two input parameters: "`datafile`" (filename string; e.g., xo3_P_2021 for the 11th member), and "`datadir`" (data directory string; no ending separator, last subdirectory = parameter file name):

```
out_dic = CCM_readmodel(<datafile>, <datadir>)
```

- or directly call:

```
out_dic = CCM_readmodel.CCM_readmodel(<datafile>, <datadir>)
```

Note that later processing steps, for example conversion of O3 VMR into partial columns in DU.km^{-1} or bulk reading/storing of a full experiment (100 members) for a given quantity are not provided.