Last modified on 05/24/14 16:05:24

CHS PYTHON library Project

What is the aim of the project

The goal is to provide a library of Python functions.

How to install the library

Checkout the library in a local directory also called PYTHON_chs_lib:

```
svn checkout https://svn.ufz.de/svn/chs-svn/PYTHON_chs_lib/
```

To checkout into a local folder with the local name "local_name", which will be created if it does not exist yet:

```
svn checkout https://svn.ufz.de/svn/chs-svn/PYTHON_chs_lib/ local_name/
```

The library has to be in your Python path. For example in bash:

```
export PYTHONPATH=/path/to/the/ufz/library
```

It can also be installed with the usual setup.py commands using distutils:

```
python setup.py install
```

If one wants to use the development capabilities of setuptools, you can use something like

```
python -c "import setuptools; execfile('setup.py')" develop
```

This basically creates an .egg-link file and updates an easy-install.pth file so that the project is on sys.path by default.

Distutils also allows to make Windows installers with

```
python setup.py bdist_wininst
```

What is in the library

See the docstring of ufz.py which functions are available.
On the Python prompt:

```
>>> import ufz
>>> help(ufz)
```

The individual functions also provide their help as doctrings.
Getting, for example, help on fread.py for reading ascii files:

```
>>> import ufz
>>> help(ufz.fread)
```

How to contribute to the Python lib

Here we give an example to add the function around.py:

1. Write the function:

    ```
    def around(num, powten, ceil=False, floor=False):
        # Check input
        if (ceil and floor):
    ```

```
     ...
    return out
```

2. Add documentation as a docstring just after the function definition:

```
def around(num, powten, ceil=False, floor=False):
    """
        Round to the passed power of ten.

        Definition
        ----------
        def around(num, powten=None, ceil=False, floor=False):


        Input
        -----
        num        number array
        .
        .
        .
    """
```

3. In the docstring provide examples with outputs for all options:

```
def around(num, powten, ceil=False, floor=False):
    """
        .
        .
        .
        Examples
        --------
        >>> around(np.array([3.5967,345.5967]), -3)
        array([   3.597,  345.597])
        >>> around(np.array([1994344,345.5967]), [3,-3])
        array([  1.99400000e+06,   3.45597000e+02])
        >>> around(np.array([1994344,345.5967]), [3,-3], ceil=True)
        array([  1.99500000e+06,   3.45597000e+02])
        >>> around(np.array([1994344,345.5967]), [3,-3], floor=True)
        array([  1.99400000e+06,   3.45596000e+02])
        >>> around(np.array([3.5967,345.5967]), 3)
        array([ 0.,  0.])
        >>> around(np.array([3.5967,345.5967]), 3, ceil=True)
        array([ 1000.,  1000.])
    ...
```

4. The end of the file should provide a call to doctest, which tests all the examples in the docstring:

```
if __name__ == '__main__':
    import doctest
    doctest.testmod()
```

5. The routine is then tested by doctest when called stand-alone:

```
python around.py
```

6. Add the routine to the Python library:
   a. Import the function in ufz.py:

   ```
   from around      import around
   ```

   b. then add the function with a short description in the docstring of ufz.py. Add it in the alphabetical section and in the section per category:

```
    ...
    Provided functions (alphabetic)
    -------------------------------
    around          Round to the passed power of ten.
    autostring      Format number (array) with given decimal precisio
    .
    .
    .
    Miscellaneous
    -------------
    around          Round to the passed power of ten.
    ...
```

License

Not all files in the library are free software. The license is given in the 'License' section of the docstring of each routine.

There are 3 possibilities:

1. The routine is not yet released under the GNU Lesser General Public License.

   This is marked by a text such as

   > This file is part of the UFZ Python library.
   >
   > It is NOT released under the GNU Lesser General Public License, yet.
   >
   > If you use this routine, please contact Matthias Cuntz.
   >
   > Copyright 2012-2013 Matthias Cuntz

   If you want to use this routine for publication or similar, please contact the author for possible co-authorship.

2. The routine is already released under the GNU Lesser General Public License

   but if you use the routine in a publication or similar, you have to cite the respective publication, e.g.

   > If you use this routine in your work, you should cite the following reference
   > Goehler M, J Mai, and M Cuntz (2013)
   >
   > > Use of eigendecomposition in a parameter sensitivity analysis
   > > of the Community Land Model,
   > > J Geophys Res 188, 904-921, doi:10.1002/jgrg.20072

3. The routine is released under the GNU Lesser General Public License. The following applies:

   The UFZ Python library is free software: you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

   The UFZ Python library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

   You should have received a copy of the GNU Lesser General Public License along with the UFZ makefile project (cf. gpl.txt and lgpl.txt). If not, see <http://www.gnu.org/licenses/>.

Copyright 2009-2014 Matthias Cuntz, Arndt Piayda, Matthias Zink, Tino Rau, Maren Goehler, Stephan Thober, Juliane Mai

Goto MainPage