

CHS PYTHON library Project

What's the aim?

CHS PYTHON library Project

What's the aim?

What is already there?

How to checkout the complete Python lib?

How to contribute to the Python lib?

The goal is to provide a library of Python functions.

What is already there?

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 docstrings.

Getting, for example, help on [fread.py](#) for reading ascii files:

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

How to checkout the complete Python lib?

To 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/
```

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 *
```

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 precision.
.
.
.
Miscellaneous
-----
around      Round to the passed power of ten.
...

```