

---

# **pyxylookup Documentation**

***Release 0.1***

**Samuel Bosch**

**Nov 03, 2020**



---

## Table of contents:

---

<b>1</b>	<b>Getting help</b>	<b>3</b>
<b>2</b>	<b>Getting started</b>	<b>5</b>
2.1	Usage . . . . .	5
2.2	LICENSE . . . . .	6



Python client for the OBIS xylookup API

Source on GitHub at [iobis/pyxylookup](https://github.com/iobis/pyxylookup)

Other OBIS xylookup clients:

- R: *obistools*, [iobis/obistools](https://github.com/iobis/obistools)



# CHAPTER 1

---

## Getting help

---

Having trouble? Or want to know how to get started?

- Looking for specific information? Try the [genindex](#)
- Report bugs with pyxylookup in our [issue tracker](#).





## CHAPTER 2

---

### Getting started

---

The main function in this package is the lookup function:

```
import pyxylookup as xy

xy.lookup([[120,0], [-170,1]])
```

*Usage* More examples

*LICENSE* The pyxylookup license.

### 2.1 Usage

The *lookup* function supports 3 different inputs for the points parameter. A nested list of longitude/latitude values, a 2D numpy array or a pandas DataFrame with 2 columns.

```
# nested list of longitude/latitude
import pyxylookup as xy
xy.lookup([[120,0], [-170,1]])

# numpy 2d array
import numpy as np
points = np.asarray([[120,0], [-170,1]])
xy.lookup(points)

## pandas DataFrame
import pandas as pd
points = pd.DataFrame({'x': [120,-170], 'y': [0,1]})
## retrieve results as a pandas DataFrame
xy.lookup(points, asdataframe = True)
```

Additional parameters allow you to specify the data returned by the service:

```
import pyxylookup as xy
points = [[120,0], [-170,1]]
# only distance to the shore
xy.lookup(points, shoredistance=True, grids=False, areas=False, asdataframe=False)
# only grids (bathymetry, sea surface temperature, ...)
xy.lookup(points, shoredistance=False, grids=True, areas=False, asdataframe=False)
# all data
xy.lookup(points, shoredistance=True, grids=True, areas=True, asdataframe=False)
```

For those using the pandas package, a pandas.DataFrame can be used for the points parameter and can be returned by the *lookup* function:

```
import pandas as pd
points = pd.DataFrame({'x': [120,-170], 'y': [0,1]})
## retrieve results as a pandas DataFrame
xy.lookup(points, shoredistance=True, grids=True, areas=True, asdataframe = True)
```

Please report bugs, questions and suggestions in our [issue tracker](#).

## 2.2 LICENSE

Copyright (C) 2018 OBIS

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.