



OpenMath / py-openmath

Unwatch 4

★ Unstar 2

🍴 Fork 1

&lt;&gt; Code

🔔 Issues 3

🔗 Pull requests 0

📁 Projects 0

📖 Wiki

📶 Pulse

📊 Graphs

Branch: master ▾

py-openmath / README.rst

Find file

Copy path

defeo Big improvements to convert module

9d34b8e 9 hours ago

2 contributors

112 lines (81 sloc) 3.08 KB

Raw

Blame

History



## pyopenmath

build failing

Python [OpenMath](#) 2.0 implementation.

## Description

[OpenMath](#) is an extensible standard for representing the semantics of mathematical objects.

## Installation

```
pip install openmath
```

## Usage

This package provides an object implementation of [OpenMath](#), and XML parsing/serialization.

See [py-scscp](#) for an example of use.

## XML Serialization

The modules `encoder` and `decoder` provide XML de-serialization for [OpenMath](#) objects.

```
>>> from openmath import encoder, decoder, openmath as om
>>> xml = encoder.encode_xml(om.OMString('hello world')); xml
<Element {http://www.openmath.org/OpenMath}OMSTR at 0x7fcb3cd82708>
>>> b = encoder.encode_bytes(om.OMString('hello world')); b
b'<OMSTR xmlns="http://www.openmath.org/OpenMath">hello world</OMSTR>'
>>> decoder.decode_xml(xml)
OMString('hello world', id=None)
>>> decoder.decode_bytes(b, snippet=True)
OMString('hello world', id=None)
```

## Conversions between Python and OpenMath

This package provides facilities for easy conversions from Python to [OpenMath](#) and back. The module `convert` contains two functions, `to_python()` and `to_openmath()`, that do the conversion as their names suggest, or raise a `ValueError` if no conversion is known.

This module only implements conversions for basic Python types:

- booleans,

- ints,
- floats,
- complex numbers,
- strings,
- bytes,
- lists (recursively),
- sets (recursively).

Furthermore, any object that defines an `__openmath__(self)` method will have that method called by `to_python`.

Finally, this module contains a mechanism for registering converters.

```
>>> from fractions import Fraction
>>> from openmath import convert, openmath as om
>>> def to_om_rat(obj):
...     return om.OMApplication(om.OMSymbol('rational', cd='nums1'),
...                             list(map(convert.to_openmath, [obj.numerator, obj.denominator])))
...
>>> def to_py_rat(obj):
...     return Fraction(convert.to_python(obj.arguments[0]), convert.to_python(obj.arguments[1]))
...
>>> convert.register(Fraction, to_om_rat, 'nums1', 'rational', to_py_rat)
>>> omobj = convert.to_openmath(Fraction(5, 6)); omobj
OMApplication(OMSymbol('rational', 'nums1', id=None, cdbase=None), [OMInteger(5, id=None), OMInteger(6, id=None)], ic
>>> convert.to_python(omobj)
Fraction(5, 6)
```

---

## Contributing

The source code of this project can be found on [GitHub](#). Please use GitHub issues and pull requests to contribute to this project.

---

## Credits

This work is supported by [OpenDreamKit](#).

---

## License

This work is licensed under the MIT License, for details see the LICENSE file.

