

---

# **pymods Documentation**

*Release 2.0.10*

**Matthew Miguez**

**Oct 13, 2021**



---

## Contents:

---

<b>1</b>	<b>Installing</b>	<b>3</b>
<b>2</b>	<b>Using</b>	<b>5</b>
2.1	Basics . . . . .	5
2.2	Methods . . . . .	6
<b>3</b>	<b>Examples</b>	<b>7</b>
3.1	Simple tasks . . . . .	7
3.2	More complex tasks . . . . .	8
3.2.1	pymods.reader Module . . . . .	8
3.2.2	pymods.record Module . . . . .	9
3.2.3	pymods.writer Module . . . . .	9
<b>4</b>	<b>Indices and tables</b>	<b>11</b>



pymods is utility module for working with the Library of Congress's MODS XML standard: Metadata Description Schema (MODS). It is a utility wrapper for the lxml module specific to deserializing data out of MODSXML into python data types.

If you need a module to serialize data into MODSXML, see the other [pymods](#) by [Matt Cordial](#).



# CHAPTER 1

---

## Installing

---

Recommended:

```
pip install pymods
```





## 2.1 Basics

XML is parsed using the MODSReader class:

```
mods_records = pymods.MODSReader('some_file.xml')
```

Individual records are stored as an iterator of the MODSRecord object:

```
In [5]: for record in mods_records:
....:     print(record)
....:
<Element {http://www.loc.gov/mods/v3}mods at 0x47a69f8>
<Element {http://www.loc.gov/mods/v3}mods at 0x47fd908>
<Element {http://www.loc.gov/mods/v3}mods at 0x47fda48>
```

MODSReader will work with `mods:modsCollection` documents, outputs from OAI-PMH feeds, or individual MODSXML documents with `mods:mods` as the root element.

### **pymods.MODSRecord**

The MODSReader class parses each `mods:mods` element into a `pymods.MODSRecord` object. `pymods.MODSRecord` is a custom wrapper class for the `lxml.ElementBase` class. All children of `pymods.Record` inherit the `lxml._Element` and `lxml.ElementBase` methods.

```
In [6]: record = next(pymods.MODSReader('example.xml'))
In [7]: print(record.nsmmap)
{'dcterms': 'http://purl.org/dc/terms/', 'xsi': 'http://www.w3.org/2001/XMLSchema-
↪instance', None: 'http://www.loc.gov/mods/v3', 'flvc': 'info:flvc/manifest/v1',
↪'xlink': 'http://www.w3.org/1999/xlink', 'mods': 'http://www.loc.gov/mods/v3'}
```

```
In [8]: for child in record.iterdescendants():
....:     print(child.tag)

{http://www.loc.gov/mods/v3}identifier
```

(continues on next page)

(continued from previous page)

```
{http://www.loc.gov/mods/v3}extension
{info:flvc/manifest/v1}flvc
{info:flvc/manifest/v1}owningInstitution
{info:flvc/manifest/v1}submittingInstitution
{http://www.loc.gov/mods/v3}titleInfo
{http://www.loc.gov/mods/v3}title
{http://www.loc.gov/mods/v3}name
{http://www.loc.gov/mods/v3}namePart
{http://www.loc.gov/mods/v3}role
{http://www.loc.gov/mods/v3}roleTerm
{http://www.loc.gov/mods/v3}roleTerm
{http://www.loc.gov/mods/v3}typeOfResource
{http://www.loc.gov/mods/v3}genre
...
```

## 2.2 Methods

All functions return data either as a string, list, list of named tuples. See the appropriate docstring for details.

```
>>> record.genre?
Type:          property
String form: <property object at 0x000000004812C78>
Docstring:
Accesses mods:genre element.
:return: A list containing Genre elements with term, authority,
        authorityURI, and valueURI attributes.
```

### Importing

```
from pymods import MODSReader, MODSRecord
```

### Parsing a file

```
In [10]: mods = MODSReader('example.xml')
In [11]: for record in mods:
....:     print(record.dates)
....:
[Date(text='1966-12-08', type='{http://www.loc.gov/mods/v3}dateCreated')]
None
[Date(text='1987-02', type='{http://www.loc.gov/mods/v3}dateIssued')]
```

## 3.1 Simple tasks

### Generating a title list

```
In [14]: for record in mods:
....:     print(record.titles)
....:
['Fire Line System']
 ['$93,668.90. One Mill Tax Apportioned by Various Ways Proposed']
 ['Broward NOW News: National Organization for Women, February 1987']
```

### Creating a subject list

```
In [17]: for record in mods:
....:     for subject in record.subjects:
....:         print(subject.text)
....:
Concert halls
```

(continues on next page)

(continued from previous page)

```

Architecture
Architectural drawings
Structural systems
Structural systems drawings
Structural drawings
Safety equipment
Construction
Mechanics
Structural optimization
Architectural design
Fire prevention--Safety measures
Taxes
Tax payers
Tax collection
Organizations
Feminism
Sex discrimination against women
Women's rights
Equal rights amendments
Women--Societies and clubs
National Organization for Women

```

## 3.2 More complex tasks

Creating a list of subject URI's only for LCSH subjects

```

In [18]: for record in mods:
....:     for subject in record.subjects:
....:         if 'lcsch' == subject.authority:
....:             print (subject.uri)
....:
http://id.loc.gov/authorities/subjects/sh85082767
http://id.loc.gov/authorities/subjects/sh88004614
http://id.loc.gov/authorities/subjects/sh85132810
http://id.loc.gov/authorities/subjects/sh85147343

```

Get URLs for objects using a No Copyright US rightsstatement.org URI

```

In [23]: for record in mods:
....:     for rights_elem in record.rights
....:         if rights_elem.uri == 'http://rightsstatements.org/vocab/NoC-US/1.0/
↳ ':
....:             print (record.purl)
....:
http://purl.flvc.org/fsu/fd/FSU_MSS0204_B01_F10_09
http://purl.flvc.org/fsu/fd/FSU_MSS2008003_B18_F01_004

```

### 3.2.1 pymods.reader Module

Custom pymods lxml parsers.

### 3.2.2 pymods.record Module

### 3.2.3 pymods.writer Module



## CHAPTER 4

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`