

Python Library: lxml

DSCI 55x

Wensheng Wu

Extraction using Python library lxml

- lxml should be installed by default in your Amazon EC2 AMI Linux instance
- If not
 - `sudo pip install lxml` (this is for python 2)
 - `sudo python3 -m pip install lxml` (python 3)
- If you are using Cygwin
 - First install libxml2, libxslt (see next slide)
 - Then "pip install lxml"

Cygwin

Select Packages
Select packages to install







View Search ☐ Keep ☒ Current

Current	New	Bin?	Src?	Categ...	Size	Package
2.9.4-2	<input checked="" type="radio"/> Keep	n/a	<input type="checkbox"/>	Libs	678k	libxml2: GNOME XML library (runtime)
	<input checked="" type="radio"/> Skip	n/a	n/a	Debug	1,925k	libxml2-debuginfo: Debug info for libxml2
2.9.4-2	<input checked="" type="radio"/> Keep	n/a	<input type="checkbox"/>	Libs	112k	libxml2-devel: GNOME XML library (development)
	<input checked="" type="radio"/> Skip	n/a	n/a	Doc	505k	libxml2-doc: GNOME XML library (API documentation)
	<input checked="" type="radio"/> Skip	n/a	n/a	Devel	739k	mingw64-i686-libxml2: GNOME XML library for Win32 toolchain
	<input checked="" type="radio"/> Skip	n/a	n/a	Devel	759k	mingw64-x86_64-libxml2: GNOME XML library for Win64 toolchain
2.9.4-2	<input checked="" type="radio"/> Keep	n/a	<input type="checkbox"/>	Python	156k	python2-libxml2: GNOME XML library (Python bindings)
	<input checked="" type="radio"/> Skip	n/a	n/a	Python	156k	python3-libxml2: GNOME XML library (Python3 bindings)

Install libxml2 and libxml2-devel

Cygwin

View Full Search libxslt Clear Keep Current

Current	New	Bin?	Src?	Categori...	Size	Package
1.1.29-1	 Keep	n/a	<input type="checkbox"/>	Libs	191k	libxslt: GNOME XSLT library (runtime)
	 Skip	n/a	n/a	Debug	519k	libxslt-debuginfo: Debug info for libxslt
1.1.29-1	 Keep	n/a	<input type="checkbox"/>	Libs	33k	libxslt-devel: GNOME XSLT library (development)
	 Skip	n/a	n/a	Libs	173k	libxslt-doc: GNOME XSLT library (API documentation)
	 Skip	n/a	n/a	Devel	214k	mingw64-i686-libxslt: GNOME XSLT library for Win32 toolchain
	 Skip	n/a	n/a	Devel	218k	mingw64-x86_64-libxslt: GNOME XSLT library for Win64 toolchain

Install libxslt and libxslt-devel

```
▼<bib>
  <cd>abc</cd>
  ▼<book>
    <publisher>Addison-Wesley</publisher>
    <author>Serge Abiteboul</author>
    ▼<author>
      <first-name>Rick</first-name>
      <last-name>Hull</last-name>
    </author>
    <author age="20">Victor Vianu</author>
    <title>Foundations of Databases</title>
    <year>1995</year>
    <price>38.8</price>
  </book>
  ▼<book price="55">
    <publisher>Freeman</publisher>
    <author>Jeffrey D. Ullman</author>
    <title>Principles of Database and Knowledge Base Systems</title>
    <year>1998</year>
  </book>
  ▼<book>
    <title>xyz</title>
    <author/>
  </book>
</bib>
```

Example

- `from lxml import etree`
- `f = open('bibs.xml')`
- `tree = etree.parse(f)`
- `print(etree.tostring(tree, pretty_print=True))`

Example

- for element in tree.xpath("//author"):
 print(etree.tostring(element))



```
<author>Serge Abiteboul</author>
```

```
<author><first-name>Rick</first-name><last-name>Hull</last-name></author>
```

```
<author age="20">Victor Vianu</author>
```

```
<author>Jeffrey D. Ullman</author>
```

```
<author/>
```

Example

- for element in tree.xpath("//author"):
 print element.tag, element.text

=>

author Serge Abiteboul
author None
author Victor Vianu
author Jeffrey D. Ullman
author None

Example

- for element in tree.xpath('//author[first-name="Rick"]'):
 print(etree.tostring(element))

=>

```
<author><first-name>Rick</first-name><last-name>Hull</last-name></author>
```

Helper function

```
def printf(elems):  
    if (isinstance(elems, list)):  
        for elem in elems:  
            if isinstance(elem, str):  
                print(elem)  
            else:  
                print(etree.tostring(elem, pretty_print=True))  
    else: # just a single element  
        print(etree.tostring(elems))
```

- `printf(tree.xpath('//author[first-name="Rick"]'))`

Work with HTML document

```
from lxml import html
```

```
myfile = open('Express.html')
```

```
htree = html.parse(myfile)
```

```
▼ <table border="1"> == $0
  ▼ <thead>
    ▼ <tr>
      <td>Account number</td>
      <td>First name</td>
      <td>Last name</td>
      <td>Address</td>
      <td>Balance</td>
    </tr>
  </thead>
  ▼ <tbody>
    ▼ <tr>
      <td>136</td>
      <td>Winnie</td>
      <td>Holland</td>
      <td>198 Mill Lane</td>
      <td>45801</td>
    </tr>
    ► <tr>...</tr>
    ► <tr>...</tr>
    ► <tr>...</tr>
```

Work with HTML document

```
print(html.tostring(htree, pretty_print=True))
```

```
htree.xpath('//tbody/tr[1]/td[1]/text()')
```

```
htree.xpath('//tbody/tr[1]/td[2]/text()')
```

```
htree.xpath('//tbody/tr[1]/td[3]/text()')
```

Resources

- Lxml - XML and HTML with Python
 - <https://lxml.de/>