

Программирование и разработка веб- приложений

Использование Python для работы с XML

Вид XML файла

```
<?xml version="1.0"?>
<books>
  <book name="Айболит">
    <number>1</number>
    <year>2005</year>
    <pages>100</pages>
    <person name="Айболит" gender="M"/>
    <person name="Два" gender="M"/>
    <person name="Чи-чи" gender="F"/>
  </book>
  <book name="Бармадей">
    <number>4</number>
    <year>2015</year>
    <pages>200</pages>
    <person name="Манечка" gender="F"/>
    <person name="Ванечка" gender="M"/>
  </book>
  <book name="Крокодил">
    <number>68</number>
    <year>2017</year>
    <pages>50</pages>
    <person name="Крокодил" gender="M"/>
    <person name="Медведь" gender="M"/>
  </book>
</books>
```

Основные модули для парсинга

```
from xml.dom import minidom
```

```
from xml.etree import ElementTree
```

```
#import xml.etree.ElementTree as ET
```

Работа с файлом

```
XML_FILE='sample3.xml'  
tree = ET.ElementTree(file=XML_FILE)
```

```
tree=ET.parse('sample3.xml')
```

```
fileo1=open('books.xml','r')  
tree=ElementTree.parse(fileo1)
```

```
xmldoc=minidom.parse('books.xml')
```

Парсинг XML файла

Использование xml.dom

```
from xml.dom import minidom
xmldoc=minidom.parse('books.xml')
a_list=xmldoc.getElementsByTagName('person')
print(len(a_list))
print(a_list[0].attributes['name'].value)
for i in a_list:
    print(i.attributes['name'].value)
```

```
G:\XML_JSON_DB>python xml_01.py
7
Айболит
Айболит
Авва
Чи-чи
Манечка
Ванечка
Крокодил
Медведь
```

```
<person name="Айболит" gender="M"/>
<person name="Авва" gender="M"/>
<person name="Чи-чи" gender="F"/>
```

Распарсить строку

Использование xml.dom

```
from xml.dom import minidom
f = minidom.parseString("""
<Books>
  <Book name='Aibloit' author='Chukovskiy' pers="Dobriy Doctor">
</Book>
  <Book name='Barmalei' author='Chukovskiy' pers="Zloy Razboinik">
</Book>
  <Book name='12 Mesyacev' author='Marshak' pers="Padcherica">
</Book>
</Books>
""")
```

```
a = f.getElementsByTagName('Book') #взятие элементов по тегу
name=[]
author=[]
pers=[]
for i in a:
    name.append(i.getAttribute('name'))
    #добавление атрибутов в список
    author.append(i.getAttribute('author'))
    pers.append(i.getAttribute('pers'))
```

```
['Dobriy Doctor', 'Zloy Razboinik', 'Padcherica']
```

Просмотр корня дерева

ElementTree

```
from xml.etree import ElementTree
fileo1=open('books.xml','r')
tree=ElementTree.parse(fileo1)
print(tree)
root=tree.getroot()
print(root)
print(root.tag)
print(root.attrib)
print(root.text)
```

```
<xml.etree.ElementTree.ElementTree object at 0x00941E70>
<Element 'books' at 0x01EC7A50>
books
<>
```


Просмотр тегов и атрибутов

```
for i in root:  
    print(i.tag, i.attrib)  
    print(root[o][o].text)
```

```
for i in root:  
    for j in i:  
        print(j.tag, j.attrib)
```

```
book {'name': 'Айболит'}  
book {'name': 'Барналей'}  
book {'name': 'Крокодил'}  
1  
number {}  
year {}  
pages {}  
person {'gender': 'M', 'name': 'Айболит'}  
person {'gender': 'M', 'name': 'Авва'}  
person {'gender': 'F', 'name': 'Чи-чи'}  
number {}  
year {}  
pages {}  
person {'gender': 'F', 'name': 'Манечка'}  
person {'gender': 'M', 'name': 'Ванечка'}  
number {}  
year {}  
pages {}  
person {'gender': 'M', 'name': 'Крокодил'}  
person {'gender': 'M', 'name': 'Медведь'}
```

Если не работает - смотрите кодировку xml

Просмотр тегов корня

```
from xml.etree import ElementTree
file01=open('books.xml','r')
tree=ElementTree.parse(file01)
#tree=ElementTree.ElementTree(file='books.xml')
root=tree.getroot()
print(root)
print(root.tag)
print(root.attrib)
print(root.text)
for i in root.iterfind('.'):
    print(i.tag)
```

```
<Element 'books' at 0x00818A20>
books
<
books
```

Проход по свойствам корня

```
root=tree.getroot()
for i in root:
    print(i.tag, i.attrib)
```

```
book {'name': 'Айболит'}
book {'name': 'Бармалей'}
book {'name': 'Крокодил'}
```

Просмотр, используя keys и items

for i in root:

```
print(i.tag,i.keys(),i.items())
```

```
book ['name'] [(<'name', 'Айболит'>)]  
book ['name'] [(<'name', 'Бармалей'>)]  
book ['name'] [(<'name', 'Крокодил'>)]
```

keys и items с итератором по root

```
for i in root.iter():
```

```
    print(i.tag, i.keys(),i.items(),i.text)
```

```
books [] []
book ['name'] [(<'name', 'Айболит')>]
number [] [] 1
year [] [] 2005
pages [] [] 100
person ['name', 'gender'] [(<'name', 'Айболит'), (<'gender', 'М')>] None
person ['name', 'gender'] [(<'name', 'Авва'), (<'gender', 'М')>] None
person ['name', 'gender'] [(<'name', 'Чи-чи'), (<'gender', 'F')>] None
book ['name'] [(<'name', 'Бармалей')>]

number [] [] 4
year [] [] 2015
pages [] [] 200
person ['name', 'gender'] [(<'name', 'Манечка'), (<'gender', 'F')>] None
person ['name', 'gender'] [(<'name', 'Ванечка'), (<'gender', 'М')>] None
book ['name'] [(<'name', 'Крокодил')>]

number [] [] 68
year [] [] 2017
pages [] [] 50
person ['name', 'gender'] [(<'name', 'Крокодил'), (<'gender', 'М')>] None
person ['name', 'gender'] [(<'name', 'Медведь'), (<'gender', 'М')>] None
```

Использование итератора для просмотра сведений

```
for i in root.iter('book'):
    print(i.tag, i.keys(), i.items(), i.text)
```

```
book ['name'] [(<'name', 'Айболит'>)]
book ['name'] [(<'name', 'Бармалей'>)]
book ['name'] [(<'name', 'Крокодил'>)]
```

Использование итератора поиска

```
for i in root.iterfind('.'):
    print(i.tag)
```

```
G:\XML_JSON_DB>python xml_dop02.py
books
```

```
for i in root.iterfind('.//'):
    print(i.tag)
```

```
book
number
year
pages
person
person
person
book
number
year
pages
person
person
book
number
year
pages
person
person
```



```
for i in root.iterfind('./book//'):
    print(i.tag)
```

```
number
year
pages
person
person
person
number
year
pages
person
person
number
year
pages
person
person
```

Использование итератора для просмотра сведений

```
for i in tree.iter('person'):  
    print(i.attrib)
```

```
<'name': 'Айболит', 'gender': 'M'>  
<'name': 'Авва', 'gender': 'M'>  
<'name': 'Чи-чи', 'gender': 'F'>  
<'name': 'Манечка', 'gender': 'F'>  
<'name': 'Ванечка', 'gender': 'M'>  
<'name': 'Крокодил', 'gender': 'M'>  
<'name': 'Медведь', 'gender': 'M'>
```

Одиночный поиск и поиск всех информации

```
for i in root.findall('book'):  
    bookpages=i.find('pages').text  
    bookpersons=i.findall('person')  
    name=i.get('name')  
    print(name)  
    print(bookpages)  
    print(bookpersons)
```

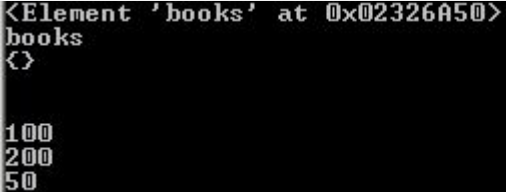
```
Айболит  
100  
[<Element 'person' at 0x022B4ED0>, <Element 'person' at 0x022B4F00>, <Element 'p  
erson' at 0x022B4F30>]  
Бармалей  
200  
[<Element 'person' at 0x022BA060>, <Element 'person' at 0x022BA090>]  
Крокодил  
50  
[<Element 'person' at 0x022BA180>, <Element 'person' at 0x022BA1B0>]
```

Запись

```
#iter ищет среди потомков number
for i in root.iter('number'):
    new_number=int(i.text)+1
    i.text=str(new_number)
    i.set('updated','2017')
    tree.write('books2.xml')
```

Поиск и удаление

```
for i in root.findall('book'):  
    pages=int(i.find('pages').text)  
    print(pages)  
    if pages >100:  
        root.remove(i)
```



```
<Element 'books' at 0x02326A50>  
books  
< >  
100  
200  
50
```

Проход по дереву и обновление значений

- `book=ElementTree.Element('book')`
- `number=ElementTree.SubElement(book,'number')`
- `year=ElementTree.SubElement(book,'year').set('year','2017')`
- `pages=ElementTree.SubElement(book,'pages').set('pages','100')`
- `number.set('pages','50')`

```
<Element 'books' at 0x01FB7A20>  
books  
└─┬─┘
```

```
<book><number pages="50" /><year year="2017" /><pages pages="100" /></book>
```

```
book=ElementTree.Element('book')
number=ElementTree.SubElement(book,'number')
year=ElementTree.SubElement(book,'year').set('year','2017')
pages=ElementTree.SubElement(book,'pages').set('pages','100')
number.set('pages','50')
root.append(book)
```

```
<book><number pages="50" /><year year="2017" /><pages pages="100" /></book></books>
```

Если файл доступен по ссылке

```
from urllib.request import urlopen
import xml.etree.ElementTree as ET

z=
urlopen("https://courses.openedu.ru/asset-v1:ITMOUniversity+PWADEV+spring_2017+type@asset+block@sample3.xml", timeout=10)
tree=ET.parse(z)
```