

XML: библиотека ElementTree

Практикум по программированию

XML: Формат

```
<?xml version="1.0"?>
```

```
<data>
```

```
<country name="Liechtenstein">
```

```
<rank>1</rank>
```

элемент

```
<year>2008</year>
```

```
<gdppc>141100</gdppc>
```

```
<neighbor name="Austria" direction="E">
```

атрибуты

```
<neighbor name="Switzerland" direction="W"/>
```

```
</country>
```

```
<country name="Singapore">
```

```
<rank>4</rank>
```

содержимое

```
<year>2011</year>
```

```
<gdppc>59900</gdppc>
```

```
<neighbor name="Malaysia" direction="N"/>
```

```
</country>
```

```
<country name="Panama">
```

```
<rank>68</rank>
```

```
<year>2011</year>
```

```
<gdppc>13600</gdppc>
```

```
<neighbor name="Costa Rica" direction="W"/>
```

```
<neighbor name="Colombia" direction="E"/>
```

теги

```
</country>
```

```
</data>
```

xml.etree.ElementTree

```
import xml.etree.ElementTree as etree
tree = etree.parse('file.xml') #получаем представление всего документа
root = tree.getroot() #получаем корневой элемент
root.tag
len(root) #"Длина" корневого элемента есть количество дочерних элементов root
for child in root:
    print(child)
root[3].attrib #возвращает словарь атрибутов элемента
```

xml.etree.ElementTree: поиск узлов

```
for neighbor in root.iter('neighbor'): #рекурсивно обходим дерево
    print neighbor.attrib

for country in root.findall('country'): #находим элементы с заданным тегом,
    |                                     | #которые являются прямыми наследниками элемента
    |                                     |
    |                                     |
    |                                     |
    |                                     |
    |                                     |
    rank = country.find('rank').text #находим первого наследника с заданным тегом
    name = country.get('name') #получаем значения атрибутов
    print name, rank

tree.write('output.xml') #записываем дерево в файл
```
