**Тема «xml»**

Одним из используемых расширений файлов является расширение .xml. XML – это расширяемый язык разметки, чем-то напоминающий html. XML удобен для создания и обработки документов программами и удобен для чтения и создания человеком.

**API**

**Введение**

API (application programming interface) – это описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой. Часто API реализуется отдельно программой или сервисов операционной системы.

Часть веб-ресурсов имеют свое API для взаимодействия с ним. Часть этих API может быть использовано для загрузки данных.

**API и requests**

Рассмотрим задачу получения информацию об ip-адресе: требуется получить геолокацию по заданному ip-адресу.

Для этого рассмотрим ip-api.com.

Для выполнения исходной задачи можно совершить один запрос и получить ответ, не содержащий лишних строк.

|  |
| --- |
| xml\_url = 'http://ip-api.com/xml/24.48.0.1'  response = requests.get(url)  print(response.text) |

В данного кода вывод будет следующим:

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <query>  <status>**success**</status>  <country>**Canada**</country>  <countryCode>**CA**</countryCode>  <region>**QC**</region>  <regionName>**Quebec**</regionName>  <city>**Montreal**</city>  <zip>**H1S**</zip>  <lat>**45.5808**</lat>  <lon>**-73.5825**</lon>  <timezone>**America/Toronto**</timezone>  <isp>**Le Groupe Videotron Ltee**</isp>  <org>**Videotron Ltee**</org>  <as>**AS5769 Videotron Telecom Ltee**</as>  <query>**24.48.0.1**</query>  </query> |

Таким образом, API вернул только необходимую информацию. Данное API позволяет получать информацию также и в формате JSON.

Каждое API чем-то уникально и практически всегда имеет документацию по использованию

**XML**

**Установка xml**

Библиотека xml позволяет обрабатывать разметку XML в Python, Кроме того, в ней присутствует возможность разложения элементов документа в дерево. Установка библиотеки производится через pip

|  |
| --- |
| pip install xml |

**Чтение xml**

Для работы с XML-файлами используется в основном два модуля из библиотеки xml: etree и minidom. Разница между ними не существенная, но будем использовать etree.

|  |
| --- |
| **with** open(xml\_file, 'r', encoding='UTF-8') **as** file:  tree = ET.parse(file) |

**Обработка xml**

Существует несколько способов получения элементов xml-файла.

1. Получение всех элементов, используя «дерево». Заметьте, что каждый элемент иерархической структуры хранит значение в атрибуте text.

|  |
| --- |
| root = tree.getroot()  **for** child **in** root:  **for** tag **in** child:  print(tag.tag, tag.text, sep='-->') |

1. Получения элемента по индексу

|  |
| --- |
| root = xml\_tree.getroot()  print(root[**0**][**0**].text) |

1. Получения элемента с помощью методов.

|  |
| --- |
| countries = {}  **for** country **in** xml\_tree.iter('COUNTRY'):  **if** country.text **in** countries:  countries[country.text] += **1**  **else**:  countries[country.text] = **1**  print(countries) |

**Использование xpath**

Следует заметить, что xml поддерживает xpath.

|  |
| --- |
| countries = xml\_tree.findall('COUNTRY')  countries\_xpath = xml\_tree.findall('.//COUNTRY')  print(len(countries), len(countries\_xpath))  >>> 0 58219 |

**Ссылки**

1. XML <https://docs.python.org/2/library/xml.etree.elementtree.html>