

Работа с XML документами. Добавление нового источника валют.

№ урока: 5 **Курс:** Python Practice

Средства обучения: Интерпретатор Python, virtualenv, текстовый редактор.

Обзор, цель и назначение урока

Познакомиться с особенностями формата XML, с основными принципами работы с XML документами. Ознакомиться с python библиотеками для работы с XML с примерами работы. Выбрать библиотеку для работы в проекте golden-eye. Изучить стандартную библиотеку xml. Реализовать новый модуль для получения курса USD => RUB по API ЦБР. Понять важность процесса рефакторинга проекта.

Изучив материал данного занятия, учащийся сможет:

- Иметь представление о принципах построения XML документов.
- Уметь работать с XML документами, производить парсинг и получение необходимой информации.
- Закрепить навыки работы с библиотекой requests.
- Производить рефакторинг кода.

Содержание урока

1. Краткое резюме того, что мы уже имеем на данный момент. Обзор дальнейших действий.
2. Что такое XML формат, XML документ. Из каких частей состоит XML документ.
3. Обзор библиотек для работы с XML документами в Python.
4. Примеры парсинга XML документов с помощью стандартной библиотеки xml и сторонних библиотек - xmltodict и beautifulsoap.
5. Реализация модуля cbr с использованием библиотеки xml и запуск теста для проверки его работы.
6. Обзор полученной структуры проекта.
7. Что такое рефакторинг, почему он нужен. Идеи для рефакторинга.

Резюме

- XML (Extensible Markup Language) - расширяемый язык разметки. Предназначен для передачи и хранения информации. Состоит из элементов, каждый элемент состоит из тегов (открывающийся и закрывающийся) и содержимого. У также у элементов могут быть атрибуты. Содержимым элемента могут быть другие элементы, которые называются – подэлементы (дочерние элементы). Соответственно, начальный элемент называется - родительским.
- Корневой элемент - это родительский элемент для всех других элементов.
- Библиотеки для работы с XML документами в Python: стандартная xml, простая xmltodict, универсальная BeautifulSoup.
- Стандартная библиотека позволяет работать с XML документами как с объектами в терминах XML. В результате парсинга с помощью метода fromstring получаем объект класса Element (xml.etree.ElementTree.Element) - корневой элемент, у которого (как и у его дочерних элементов) есть методы:
 - get(key) для получения значения атрибута с именем key;

- `find(tag_name)` и `findall(tag_name)` для получения первого всех дочерних элементов с именем `tag_name`. В случае `findall` вернется список элементов.
- Библиотека `xmltodict` позволяет работать с разобранным XML документом как со словарем в Python (`dict`). Для получения доступа к атрибуту элемента необходимо в качестве ключа указать: символ `@` + имя атрибута.
- Рефакторинг - процесс внесения изменений в структуру написанного кода с целью оптимизации, улучшения читабельности кода. При реализации нового проекта разработчик должен часто и критично оглядывать свой код, анализировать его на предмет необходимости рефакторинга. После реализации проекта, при дальнейшей его поддержке, также необходимо проводить рефакторинг, но очень осторожно.

Закрепление материала

- Что такое XML? Из каких частей состоит XML документ?
- Что такое элементы, теги, атрибуты? Что такое родительский, дочерний, корневой элементы XML документа?
- Какие библиотеки для парсинга XML документов вы знаете?
- Каковы особенности работы с библиотекой `xml` и `xmltodict`? В чем основное отличие при работе с этими библиотеками
- Что такое рефакторинг, какие цели он преследует?
- На примере XML документа, как получить доступ к элементу после парсинга с помощью библиотеки `xml` и `xmltodict` (небольшой python код)? К атрибуту?

Дополнительное задание

Задание

Добавить проверку ответа от API - что структура полученного XML документа соответствует ожидаемым в скрипте.

Самостоятельная деятельность учащегося

Задание 1

Реализовать парсинг XML с помощью библиотеки `xmltodict` или `BeautifulSoup`. Переписать проверку ответа от API.

Задание 2

Изучив библиотеки `xmltodict` и `xml`, напишите скрипт, который будет формировать xml. Сравните код, полученный при использовании разных библиотек.

```
<today date="2018-07-15" me="Kseniia">
  <task number="1">Get awake</task>
  <task number="2">Make breakfast</task>
  <task number="3">Learn Python</task>
</today>
```

Рекомендуемые ресурсы

<http://xml.readthedocs.io/xml-intro.html>

<https://docs.python.org/3.6/library/xml.etree.elementtree.html>

<https://pypi.org/project/xmltodict/>

<http://wiki.python.su/%D0%94%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D0%B0%D1%86%D0%B8%D0%B8/BeautifulSoup>