нетология

Работа с разными форматами данных

Елена
Никитина

Елена Никитина

Руководитель проектов ГК «Геоскан»

План занятия

1. Введение
2. Формат CSV
3. Формат JSON
4. Формат YAML
5. Формат XML
6. Проблема кодировок

ВВЕДЕНИЕ

CSV	JSON	YAML	XML
плоский	дерево	дерево	дерево

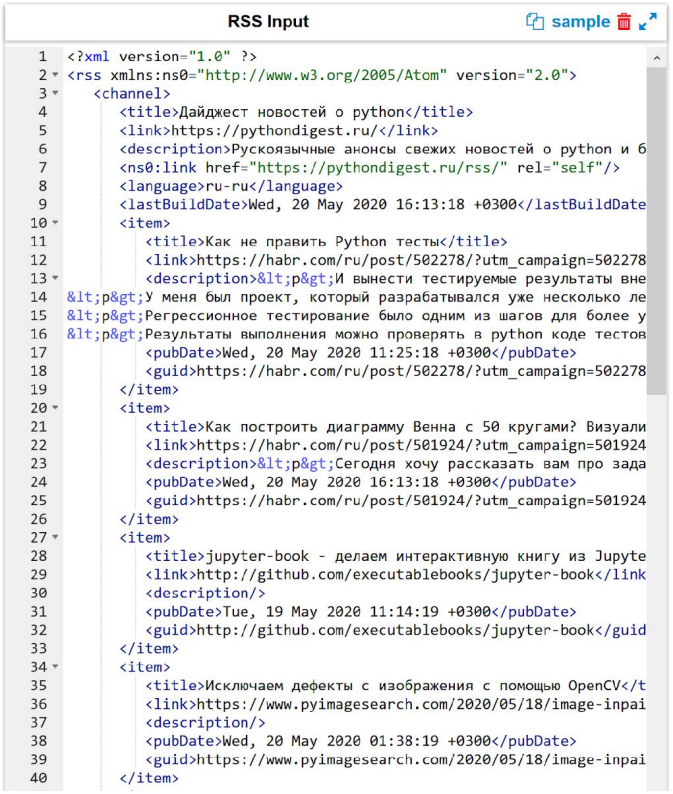
Сериализация

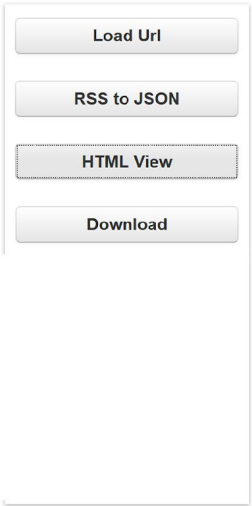
Сериализация — процесс преобразования объекта в поток байтов для сохранения или передачи в память, базу данных или файл.

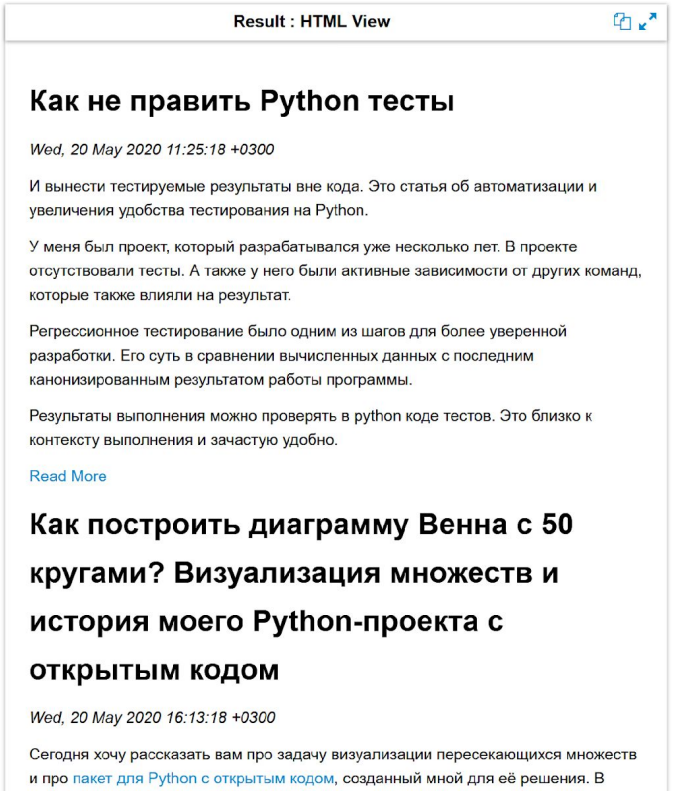
Эта операция предназначена для того, чтобы сохранить состояния объекта для последующего воссоздания при необходимости.

Обратный процесс называется десериализацией.

Тренировочные данные: новости в RSS







Источник: http://pythondigest.ru/rss/
https://codebeautify.org/rssviewer онлайн просмотр ленты RSS

ФОРМАТ CSV

Формат CSV

```
1  title,link,description,pubDate
2  Как не править Python тесты,https://habr.com/ru/post/502278/?utm_campaign=502278&
   utm_source=habrahabr&utm_medium=rss,"<p>И вынести тестируемые результаты вне кода. Это
   статья об автоматизации и увеличения удобства тестирования на Python.</p><p>У меня был
   проект, который разрабатывался уже несколько лет. В проекте отсутствовали тесты. А также
   у него были активные зависимости от других команд, которые также влияли на
   результат.</p><p>Регрессионное тестирование было одним из шагов для более уверенной
   разработки. Его суть в сравнении вычисленных данных с последним канонизированным
   результатом работы программы.</p><p>Результаты выполнения можно проверять в python коде
   тестов. Это близко к контексту выполнения и зачастую удобно.</p>","Wed, 20 May 2020
   11:25:18 +0300"
3  Как построить диаграмму Венна с 50 кругами? Визуализация множеств и история моего
   Python-проекта с открытым кодом,https://habr.com/ru/post/501924/?utm_campaign=501924&
   utm_source=habrahabr&utm_medium=rss,"<p>Сегодня хочу рассказать вам про задачу
```

	title	link	description	pubDate
2	Как не править Python тесты	https://habr.com/ru/post/502278	<p>И вынести тестируемые резуль	20.05.2020 11:25
3	Как построить диаграмму Венна с 50 кругами? Визуализация мно	https://habr.com/ru/post/501924	<p>Сегодня хочу рассказать вам п	20.05.2020 16:13
4	jupyter-book - делаем интерактивную книгу из Jupyter Notebooks	http://github.com/executablebook		19.05.2020 11:14
5	Исключаем дефекты с изображения с помощью OpenCV	https://www.pyimagesearch.com/		20.05.2020 1:38

Формат CSV

Основное применение: **выгрузки данных**

- Для хранения больших объемов (до нескольких Гб) единообразных данных
- Самый компактный формат из всех
- Самый популярный формат для обмена данными у аналитиков
- Поддерживается MS Excel (эквивалентен плоской таблице)
- Не подходит для иерархических данных

https://docs.python.org/3/library/csv.html

Формат CSV

Десериализация в список
reader = csv.reader(file)
data = list(reader)

Десериализация в словарь
reader = csv.DictReader(file)

Сериализация
writer = csv.writer(file)
writer.writerows(data)
writer.writerow(data)

Настройки форматирования
csv.register_dialect()

Формат CSV

Настройки форматирования через csv.register_dialect()

delimiter=“,”
quoting=csv.QUOTE_MINIMAL (QUOTE_ALL, QUOTE_NONNUMERIC, QUOTE_NONE)
quotechar='“’
escapechar='\\'

Сравните результат с csv.QUOTE_MINIMAL и csv.QUOTE_ALL:

```
5    Исключаем дефекты с изображения с помощью OpenCV,
     https://www.pyimagesearch.com/2020/05/18/image-inpainting-with-opencv-and-python/,,"Wed, 20 May
     2020 01:38:19 +0300"
```

```
5    "Исключаем дефекты с изображения с помощью OpenCV",
     "https://www.pyimagesearch.com/2020/05/18/image-inpainting-with-opencv-and-python/","",
     "Wed, 20 May 2020 01:38:19 +0300"
```

https://docs.python.org/3/library/csv.html#csv-fmt-params форматирование

Формат CSV: Десериализация и сериализация

```
73  import csv
74  csv.register_dialect('customcsv', delimiter=',', quoting=csv.QUOTE_MINIMAL,
    quotechar='"', escapechar='\\')
75  with open("files/sample.csv", "w", encoding="utf-8") as f:
76      writer = csv.writer(f, "customcsv")
77      writer.writerows(data2)
78
79  with open("files/sample.csv", newline="") as f:
80      reader = csv.reader(f)
81      print(list(reader))
82
83  with open("files/sample.csv", newline="") as f:
84      reader = csv.DictReader(f)
85      for row in reader:
86          print(row["title"])
```

ФОРМАТ JSON

Формат JSON

```
{
  "channel": {
    "title": "Дайджест новостей о python",
    "link": "https://pythondigest.ru/",
    "description": "Рускоязычные анонсы свежих новостей о python и близлежащих технологиях.",
    "lastBuildDate": "Wed, 20 May 2020 16:13:18 +0300",
    "items": [
      {
        "title": "Как не править Python тесты",
        "link": "https://habr.com/ru/post/502278/?utm_campaign=502278&utm_source=habrahabr&
        utm_medium=rss",
        "description": "<p>И вынести тестируемые результаты вне кода. Это статья об автоматизации и
        увеличения удобства тестирования на Python.</p>",
        "pubDate": "Wed, 20 May 2020 11:25:18 +0300"
      }
    ]
  }
}
```

Формат JSON

Основное применение: **базы данных, выгрузки данных**

- Для импорта/экспорта данных в базы данных (в т.ч. bson)
- Для сохранения вложенных структур данных
- Также применяется при передаче данных клиент<->сервер для сериализации иерархических объектов
- Самый популярный и простой в использовании формат для Python и Java программистов
- Является подмножеством формата YAML

https://docs.python.org/3/library/json.html документация JSON
https://jsoneditoronline.org/ онлайн редактор JSON

Формат JSON

Десериализация
Из файла: json.load(file)
Из строки: json.loads(str)

Сериализация
В файл: json.dump()
В строку: json.dumps()

Печать не-ascii символов, отступы
ensure_ascii=False, indent=2

Формат JSON: Десериализация и сериализация

```
38  import json
39  from pprint import pprint
40
41  data = {"channel": {"title": "Дайджест новостей о python",
42                      "link": "https://pythondigest.ru/"}}
43  with open("files/sample.json", "w") as f:
44      json.dump(data, f, ensure_ascii=False, indent=2)
45
46  with open("files/sample.json", encoding = "utf-8") as f:
47      data = json.load(f)
48      pprint(data)
```

ФОРМАТ YAML

Формат YAML

```yaml
channel:
  description: Рускоязычные анонсы свежих новостей о python и близлежащих технологиях.
  items:
  - description: '<p>И вынести тестируемые результаты вне кода. Это статья об автоматизации
      и увеличения удобства тестирования на Python.</p>'
    link: https://habr.com/ru/post/502278/?utm_campaign=502278&utm_source=habrahabr&
    utm_medium=rss
    pubDate: Wed, 20 May 2020 11:25:18 +0300
    title: Как не править Python тесты
  lastBuildDate: Wed, 20 May 2020 16:13:18 +0300
  link: https://pythondigest.ru/
  title: Дайджест новостей о python
```

Формат YAML

Основное применение: **файлы конфигурации**

- Самый компактный язык разметки
- Для создания файлов настроек
- Используется для описания классов, ресурсов и манифестов в API

https://yaml.org/
https://github.com/yaml/pyyaml

Формат YAML

Десериализация
Из файла: yaml.load(file)
Из строки: yaml.loads(str)

Сериализация
В файл: yaml.dump
В строку: yaml.dumps

Печать не-ascii символов, отступы
allow_unicode=True, default_flow_style=False

Формат YAML: Десериализация и сериализация

```
50  import yaml
51  from pprint import pprint
52
53  data = {"channel": {"title": "Дайджест новостей о python",
54                      "link": "https://pythondigest.ru/"}}
55  with open("files/sample.yml", "w") as f:
56      yaml.dump(data, f, allow_unicode=True, default_flow_style=False)
57
58  with open("files/sample.yml", encoding = "utf-8") as f:
59      data = yaml.load(f, Loader=yaml.FullLoader)
60      pprint(data)
```

ФОРМАТ XML

XML vs JSON

```xml
<?xml version="1.0" ?>
<rss xmlns:ns0="http://www.w3.org/2005/Atom" version="2.0">
    <channel>
        <title>Дайджест новостей о python</title>
        <link>https://pythondigest.ru/</link>
        <description>Рускоязычные анонсы свежих новостей о python и близлежащих технологиях.</description>
        <ns0:link href="https://pythondigest.ru/rss/" rel="self"/>
        <language>ru-ru</language>
        <lastBuildDate>Wed, 20 May 2020 16:13:18 +0300</lastBuildDate>
        <item>
            <title>Как не править Python тесты</title>
            <link>https://habr.com/ru/post/502278/?utm_campaign=502278&amp;utm_source=habrahabr&amp;utm_medium=rss</link>
            <description>&lt;p&gt;И вынести тестируемые результаты вне кода. Это статья об автоматизации и увеличения удобства тестирования на Python.&lt;/p&gt;</description>
            <pubDate>Wed, 20 May 2020 11:25:18 +0300</pubDate>
            <guid>https://habr.com/ru/post/502278/?utm_campaign=502278&amp;utm_source=habrahabr&amp;utm_medium=rss</guid>
        </item>
        <item>
            <title>Как построить диаграмму Венна с 50 кругами? Визуализация множеств и история моего Python-проекта с открытым кодом</title>
            <link>https://habr.com/ru/post/501924/?utm_campaign=501924&amp;utm_source=habrahabr&amp;utm_medium=rss</link>
            <description>&lt;p&gt;Сегодня хочу рассказать вам про задачу визуализации пересекающихся множеств. Поехали!&lt;/p&gt;</description>
            <pubDate>Wed, 20 May 2020 16:13:18 +0300</pubDate>
            <guid>https://habr.com/ru/post/501924/?utm_campaign=501924&amp;utm_source=habrahabr&amp;utm_medium=rss</guid>
        </item>
```

```json
{
  "channel": {
    "title": "Дайджест новостей о python",
    "link": "https://pythondigest.ru/",
    "description": "Рускоязычные анонсы свежих новостей о ру
    технологиях.",
    "lastBuildDate": "Wed, 20 May 2020 16:13:18 +0300",
    "items": [
      {
        "title": "Как не править Python тесты",
        "link": "https://habr.com/ru/post/502278/?utm_campai
        utm_source=habrahabr&utm_medium=rss",
        "description": "<p>И вынести тестируемые результаты
        статья об автоматизации и увеличения удобства тестир
        Python.</p>",
        "pubDate": "Wed, 20 May 2020 11:25:18 +0300"
      },
      {
        "title": "Как построить диаграмму Венна с 50 кругами
        множеств и история моего Python-проекта с открытым к
        "link": "https://habr.com/ru/post/501924/?utm_campai
        utm_source=habrahabr&utm_medium=rss",
        "description": "<p>Сегодня хочу рассказать вам про з
        пересекающихся множеств. Поехали!</p>",
        "pubDate": "Wed, 20 May 2020 16:13:18 +0300"
      },
```

Формат XML

Основное применение: **сериализация объектов любой сложности**

- Применяется при передаче данных клиент<->сервер для сериализации объектов
- Стандарт обмена данными и сообщениями большинства информационных систем
- Для создания файлов конфигурации

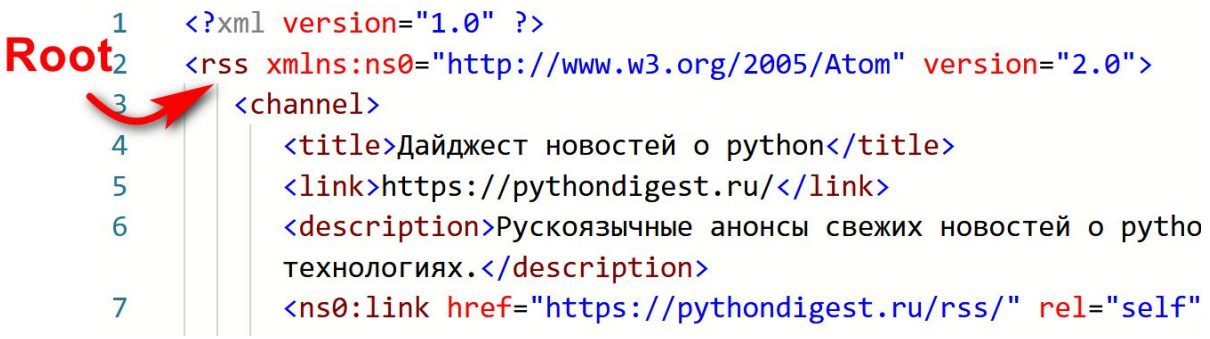
https://docs.python.org/3/library/xml.etree.elementtree.html

Формат XML

```
1  <rss xmlns:ns0="http://www.w3.org/2005/Atom" version="2.0">
2  <channel>
3    <title>Дайджест новостей о python</title>
4    <link>https://pythondigest.ru/</link>
5    <description>Рускоязычные анонсы свежих новостей о python и близлежащих
     технологиях.</description>
6  </channel></rss>
```

Элемент: <title>Дайджест новостей о python</title>
Тег: title
Текст: Дайджест новостей о python
Атрибут: version="2.0"

Формат XML



Чтение дерева: xml.etree.ElementTree.parse()
Корень дерева: tree.getroot()
<rss xmlns:ns0="http://www.w3.org/2005/Atom" version="2.0">

Имя тега: root.tag
Текст внутри тега: root.text
Атрибуты тега: root.attrib

Сохранение: tree.write()

Формат XML

XPath

Поиск одного элемента: root.find(query)

Поиск нескольких элементов: root.findall(query)

query = XPath

https://www.w3schools.com/xml/xpath_syntax.asp синтаксис XPath

XML: Загрузка из файла и словаря

```
1   import xml.etree.ElementTree as ET
2
3   # чтение из файла
4   # создать объект парсера и указать верную кодировку
5   parser = ET.XMLParser(encoding="utf-8")
6   # прочитать DOM-дерево документа
7   tree = ET.parse("files/sample.xml", parser)
8   # получить корневой элемент дерева
9   root = tree.getroot()
10
11  # xml из словаря
12  import dicttoxml
13  data = {"channel": {"title": "Дайджест новостей о python",
14                      "link": "https://pythondigest.ru/"}}
15  xml = dicttoxml.dicttoxml(data)
```

XML: Работа с элементами

```
18  # получить корневой элемент дерева
19  root = tree.getroot()
20  # название тега (на примере корневого элемента)
21  print(root.tag)
22  # получение атрибутов тега
23  print(root.attrib)
24  # текст внутри тега
25  print(root.text)
26  # поиск элемента с помощью xpath
27  xml_title = root.find("channel/title")
28  # текст внутри тега
29  print(xml_title.text)
30  # поиск всех элементов с помощью xpath
31  xml_items = root.findall("channel/item")
32  print(len(xml_items))
33  for xmli in xml_items:
34      print(xmli.find("title").text)
```

XML: Сохранение

```
63   # простое сохранение
64   tree.write("files/result.xml", encoding="utf-8")
65
66   # сохранение с отступами
67   from xml.dom import minidom
68   xmlstr = minidom.parseString(ET.tostring(root)).toprettyxml(indent="   ")
69   with open("files/result_indent.xml", "w", encoding= "utf-8") as f:
70       f.write(xmlstr)
```

Проблема кодировок

Кодировки: проблемы с кириллицей

Один и тот же текст в кодировке Windows 1251 и utf-8

Windows 1251 (один байт на букву)

```
00000000: C8 F1 F2 EE F0 E8 FF 20|F3 20 EC E5 ED FF 20 F1  | История у меня с
00000010: EB E5 E4 F3 FE F9 E0 FF|3A 20 EF EE E7 ED E0 EA  | ледующая: познак
00000020: EE EC E8 EB F1 FF 20 F1|20 E4 E5 E2 F3 F8 EA EE  | омился с девушко
00000030: E9 20 E8 E7 20 F1 E2 EE|E5 E3 EE 20 E8 ED F1 F2  | й из своего инст
00000040: E8 F2 F3 F2 E0 2E 0D 0A|D1 ED E0 F7 E0 EB E0 20  | итута...Сначала
```

utf-8 (два байта на букву)

```
00000000: EF BB BF D0 98 D1 81 D1|82 D0 BE D1 80 D0 B8 D1  | п»ї.СЃС‚РѕСЂРёС
00000010: 8F 20 D1 83 20 D0 BC D0|B5 D0 BD D1 8F 20 D1 81  | Џ Сѓ РјРµРSСЏ СЃ
00000020: D0 BB D0 B5 D0 B4 D1 83|D1 8E D1 89 D0 B0 D1 8F  | Р»РµРґСѓСЋС‰Р°СЏ
00000030: 3A 20 D0 BF D0 BE D0 B7|D0 BD D0 B0 D0 BA D0 BE  | : РїРѕР·РSР°РєРѕ
00000040: D0 BC D0 B8 D0 BB D1 81|D1 8F 20 D1 81 20 D0 B4  | РјРёР»СЃСЏ СЃ Рґ
```

Кодировки: проблемы с кириллицей

Ожидания при сохранении в JSON:

```
2     "channel": {
3       "title": "Дайджест новостей о python",
4       "link": "https://pythondigest.ru/",
5       "description": "Рускоязычные анонсы свежих новостей о python и близлежащих
        технологиях.",
6       "lastBuildDate": "Wed, 20 May 2020 16:13:18 +0300",
```

Результат при сохранении в JSON:

```
2     "channel": {
3       "title": "\u0414\u0430\u0439\u0434\u0436\u0435\u0441\u0442
        \u043d\u043e\u0432\u043e\u0441\u0442\u0435\u0439 \u043e python",
4       "link": "https://pythondigest.ru/",
5       "description":
        "\u0420\u0443\u0441\u043a\u043e\u044f\u0437\u044b\u0447\u043d\u04
        \u0430\u043d\u043e\u043d\u0441\u044b \u0441\u0432\u0435\u0436\u04
        \u043d\u043e\u0432\u043e\u0441\u0442\u0435\u0439 \u043e python \u
        \u0431\u043b\u0438\u0437\u043b\u0435\u0436\u0430\u0449\u0438\u04
        \u0442\u0435\u0445\u043d\u043e\u043b\u043e\u0433\u0438\u044f\u04
6       "lastBuildDate": "Wed, 20 May 2020 16:13:18 +0300",
```

Кодировки: utf-8, cp1251

Для решения проблемы явно указывайте кодировку:

```
59  with open("files/sample.json", encoding = "utf-8") as f:
60      data = json.load(f)
```

Результат:

```
2   "channel": {
3     "title": "Дайджест новостей о python",
4     "link": "https://pythondigest.ru/",
5     "description": "Рускоязычные анонсы свежих новостей о python и близлежащих
      технологиях.",
6     "lastBuildDate": "Wed, 20 May 2020 16:13:18 +0300",
```

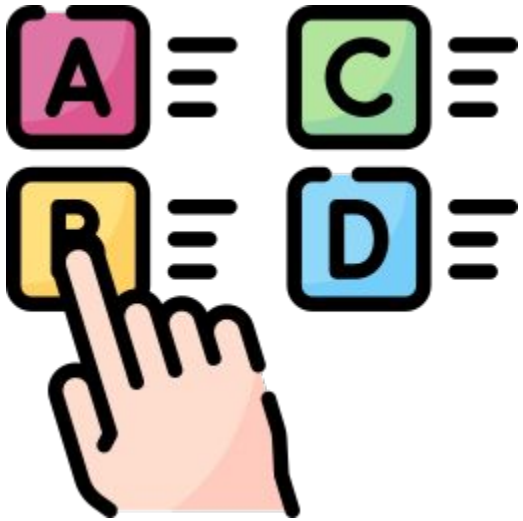
https://ru.hexlet.io/courses/python_101/lessons/python_unicode/theory_unit Unicode в Python

Домашнее задание

Закрепите тему сегодняшней лекции — пройдите **квиз**!

В квизе вас ждут:

- пояснения к каждому варианту ответа,
- неограниченное количество попыток.



нетология

Задавайте вопросы и пишите отзыв о лекции!

Елена Никитина