



Информационные ресурсы в финансовом мониторинге

НИЯУ МИФИ, КАФЕДРА ФИНАНСОВОГО МОНИТОРИНГА

КУРС ЛЕКЦИЙ

В.Ю. РАДЫГИН. ЛЕКЦИЯ 1

Часть 1

СПОСОБЫ ПРЕДСТАВЛЕНИЯ ИНФОРМАЦИИ В СЕТИ ИНТЕРНЕТ

Виды информации в сети

Сегодня в сети Интернет доступно огромное множество различной информации, использование которой возможно в задачах финансового мониторинга. По способу предоставления информации источники можно поделить на четыре основных группы.

1. Представляющие информацию в виде, удобном для машинной обработки. Могут различаться по способу доступа к информации:
 - a. Информация представляется в виде статичных файлов.
 - b. Информация представляется в виде специального API.
 - c. Информация представляется в виде FTP-репозитория.
2. Представляющие информацию в современных офисных форматах, напрямую не являющихся удобными для машинной обработки, но содержащих текстовый слой (doc, docx, rtf, txt, pdf).
3. Представляющие информацию в гипертекстовом виде (в виде страниц сайта).
4. Представляющие информацию в виде скан копий документов, не содержащих текстового слоя.

В большинстве случаев, за исключением специальных порталов с данными, предназначенными для машинной обработки, источники являются комбинированными и предоставляют информацию в разных форматах.

Пример

Информационно-правовая система «Законодательство России» (в составе Официального интернет-портала правовой информации) – <http://pravo.gov.ru/proxy/ips>.
Материалы доступны в основном в форматах txt и rtf.

The screenshot displays the 'Pravo.gov.ru' website interface. On the left, the 'Результаты поиска:' (Search Results) section shows a search for 'Текущие БПА: ФЕДЕРАЛЬНОЕ ЗАКОНОДАТЕЛЬСТВО' (Current BPA: FEDERAL LEGISLATION). It lists 440 documents, sorted by date, with page 10 of 10 selected. The first result is highlighted with a red dashed box: 'Постановление Правительства Российской Федерации от 07.12.2022 № 2239' (Decree of the Government of the Russian Federation of 07.12.2022 No. 2239). The main content area on the right shows the full text of this decree. The title is 'ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПОСТАНОВЛЕНИЕ от 7 декабря 2022 г. № 2239 МОСКВА'. The subject is 'Об утверждении Правил предоставления в 2023 году субсидий из федерального бюджета российским авиакомпаниям в целях возмещения операционных расходов на осуществление перевозок по внутренним воздушным линиям в условиях внешнего санкционного воздействия' (On approval of the Rules for providing in 2023 subsidies from the federal budget to Russian airlines for reimbursement of operational expenses for flights on domestic air lines in the conditions of external sanctions impact). The text states that the Government approves the Rules for providing subsidies in 2023 to Russian airlines for reimbursement of operational expenses for flights on domestic air lines in the conditions of external sanctions impact. The signature is 'Председатель Правительства Российской Федерации М.Михулин' (Chairman of the Government of the Russian Federation M. Mikhulin).

Результаты поиска:

Текущие БПА:
ФЕДЕРАЛЬНОЕ ЗАКОНОДАТЕЛЬСТВО

Атрибуты поиска:
Номер начинается на '223'

440 документов

сортировать: По убыванию даты

Страницы: 1 2 3 4 5 6 7 8 9 10

Аннотации: с 1 по 20 выводить по: 20

1. Постановление Правительства Российской Федерации от 07.12.2022 № 2239
Об утверждении Правил предоставления в 2023 году субсидий из федерального бюджета российским авиакомпаниям в целях возмещения операционных расходов на осуществление перевозок по внутренним воздушным линиям в условиях внешнего санкционного воздействия
• Официальный интернет-портал правовой информации (www.pravo.gov.ru) от 8.12.2022 г., ст. 0001202212080028
• Собрание законодательства Российской Федерации от 2022 г., N 50, ст. 8940 (Часть IV)

2. Постановление Правительства Российской Федерации от 07.12.2022 № 2239
О внесении изменений в постановление Правительства Российской Федерации от 19 марта 2022 г. № 411
• Официальный интернет-портал правовой информации (www.pravo.gov.ru) от 8.12.2022 г., ст. 0001202212080049
• Собрание законодательства Российской Федерации от 2022 г., N 50, ст. 8939 (Часть IV)

3. Постановление Правительства Российской Федерации от 06.12.2022 № 2239
О внесении изменений в постановление Правительства Российской Федерации от 16 апреля 2020 г. № 520
• Официальный интернет-портал правовой информации (www.pravo.gov.ru) от 8.12.2022 г., ст. 0001202212080044
• Собрание законодательства Российской Федерации от 2022 г., N 50, ст. 8938 (Часть IV)

4. Постановление Правительства Российской Федерации от 06.12.2022 № 2239
О внесении изменений в постановление Правительства Российской Федерации от 22 февраля 2021 г. № 245
• Официальный интернет-портал правовой информации (www.pravo.gov.ru) от 8.12.2022 г., ст. 0001202212080019
• Собрание законодательства Российской Федерации от 2022 г., N 50, ст. 8937 (Часть IV)

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

ПОСТАНОВЛЕНИЕ

от 7 декабря 2022 г. № 2239

МОСКВА

Об утверждении Правил предоставления в 2023 году субсидий из федерального бюджета российским авиакомпаниям в целях возмещения операционных расходов на осуществление перевозок по внутренним воздушным линиям в условиях внешнего санкционного воздействия

Правительство Российской Федерации постановляет:

Утвердить прилагаемые Правила предоставления в 2023 году субсидий из федерального бюджета российским авиакомпаниям в целях возмещения операционных расходов на осуществление перевозок по внутренним воздушным линиям в условиях внешнего санкционного воздействия.

Председатель Правительства
Российской Федерации

М.Михулин

следующий документ →

Пример

Реестр открытых данных Минфина России – <https://minfin.gov.ru/ru/opendata/>.
Материалы доступны в основном в форматах csv и xml в виде статичных файлов.

Реестр открытых данных Минфина России

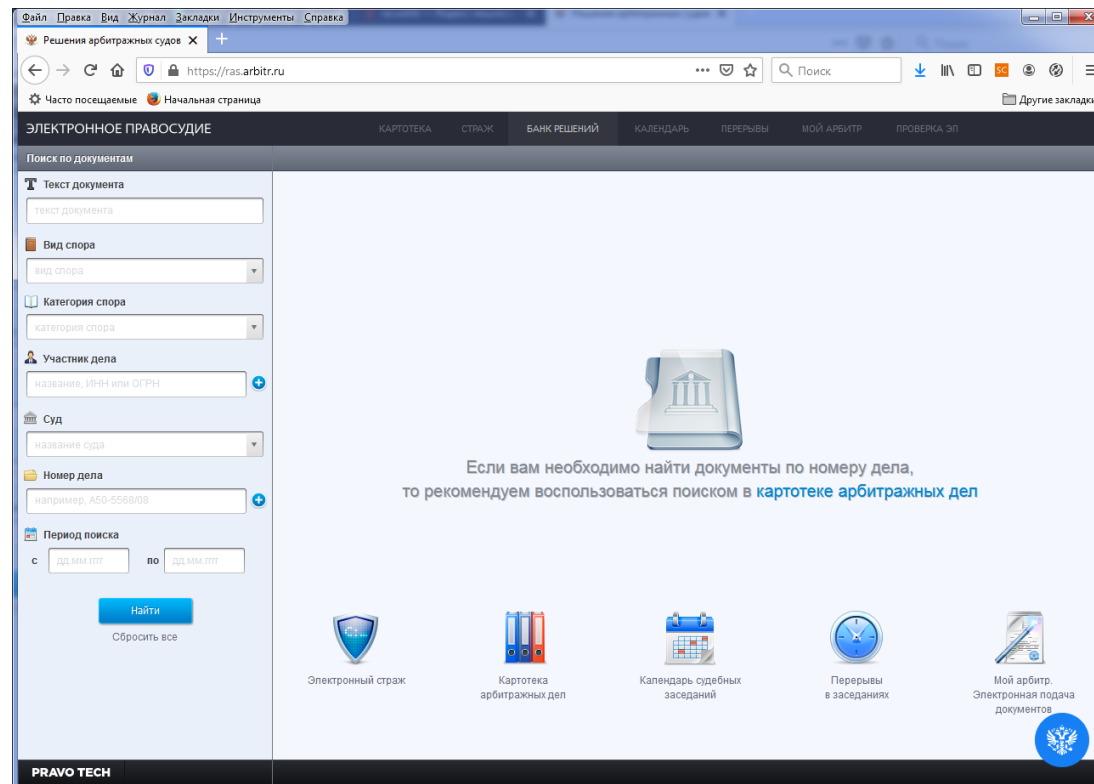
Вы можете скачать все наборы открытых данных одним документом или ознакомиться с типовыми условиями использования общедоступной информации в форме открытых данных

Поиск по реестру

№	Наименование	Статус	Скачивания	Рейтинг	Обновлено
1	Организации, находящиеся в ведении Минфина России Скачать паспорт csv 3.67 кб Скачать csv 13.89 кб	АКТУАЛЬНЫЙ	96	☆ 2,54	31.07.2020
2	План контрольной деятельности, осуществляемой Административным департаментом Министерства финансов Российской Федерации Скачать паспорт csv 2.56 кб Скачать xml 52.78 кб	АРХИВНЫЙ	14	☆ 2,57	16.12.2014
3	Отчет о контрольной деятельности Административного департамента Скачать паспорт csv 2.68 кб Скачать xml 72.26 кб	АРХИВНЫЙ	19	☆ 2,65	25.01.2016
4	Расходы консолидированного бюджета субъекта Российской Федерации на финансирование жилищно-коммунального хозяйства в части компенсации разницы между экономически обоснованными тарифами и тарифами, установленными для населения, и покрытия убытков, возникших в связи с применением регулирующих цен на жилищно-коммунальные услуги Скачать паспорт csv 2.17 кб Скачать csv 13.67 кб	АРХИВНЫЙ	76	☆ 2,67	06.06.2013

Пример

Электронное правосудие – <https://ras.arbitr.ru/>. Материалы арбитражных судов доступны в основном в форматах html и pdf (в виде статичных файлов).



Машино обрабатываемые форматы

Сегодня в сети Интернет есть чёткий набор общепринятых форматов представления данных для машинной обработки, в том числе:

1. CSV (Comma-Separated Values) – текстовый формат для представления данных электронных таблиц.
2. XML (eXtensible Markup Language) – специальный язык разметки данных, построенный по принципам, схожим с форматом HTML.
3. JSON (JavaScript Object Notation) – язык разметки данных, применяемые в языке программирования JavaScript.
4. SQLITE – файлы репозитория базы данных, предназначенные для обработки в простой СУБД SQLite.
5. XLS, XLSX – файлы Excel.
6. Специальным образом отформатированный TXT – текстовые файлы, в которых информация отформатирована особым образом. Например, определённым числом пробелов и т.д.
7. Другие форматы.

CSV-формат

CSV-файл – это по сути простейший язык разметки данных. В нём есть следующие основные составляющие:

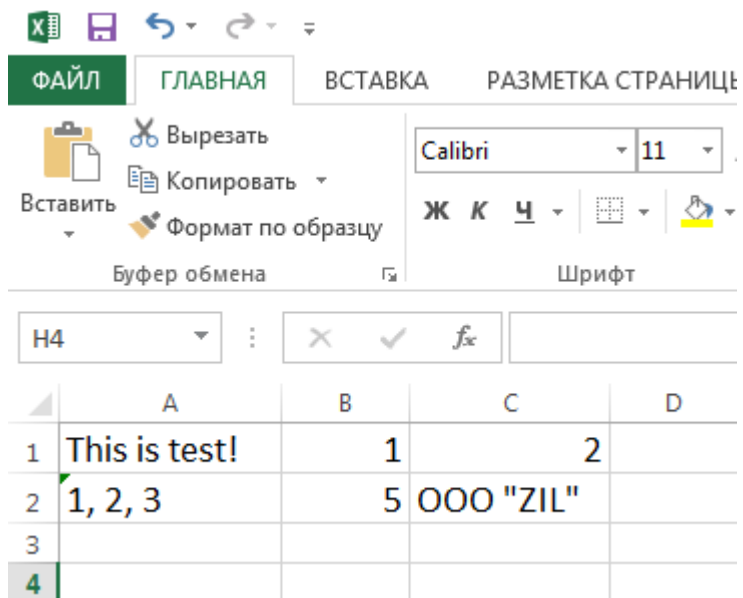
- разделитель строк;
- разделитель ячеек;
- ограничитель строк;
- данные.

Разделители и ограничители в CSV

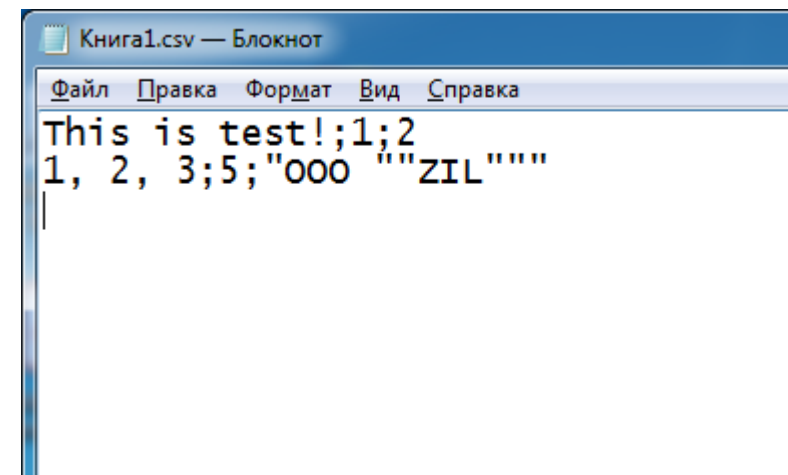
Канонический формат CSV подразумевает, что каждая строка отделяется от другой символом конца строки и (но не обязательно) возврата каретки (либо `\n`, либо `\n\r`), каждая ячейка отделяется от другой символом запятой (`,`), а для сложных строковых значений (например, содержащих запятую) данные берутся внутрь двойных кавычек ("`...`"). При этом, если строка содержит ещё и сами кавычки, то оно экранируются заменой на две кавычки сразу: `"ООО ""ЗИЛ"""`.

На сегодняшний день большинство систем вместо канонического формат CSV работают с форматом DSV (Delimiter-separated values). Данный формат позволяет использовать другие разделители ячеек и ограничители строк. Такой переход связан с различными факторами. Одним из них является необходимость учитывать локализацию. К примеру, в России целую часть числа от десятичной дроби принято отделять не точкой, а запятой, которая совпадает с разделителем по умолчанию. Поэтому Microsoft Excel с русской локализацией при сохранении данных в формате «CSV (разделитель – запятые)» выполняет сохранение с разделителем точка с запятой (`;`).

Пример



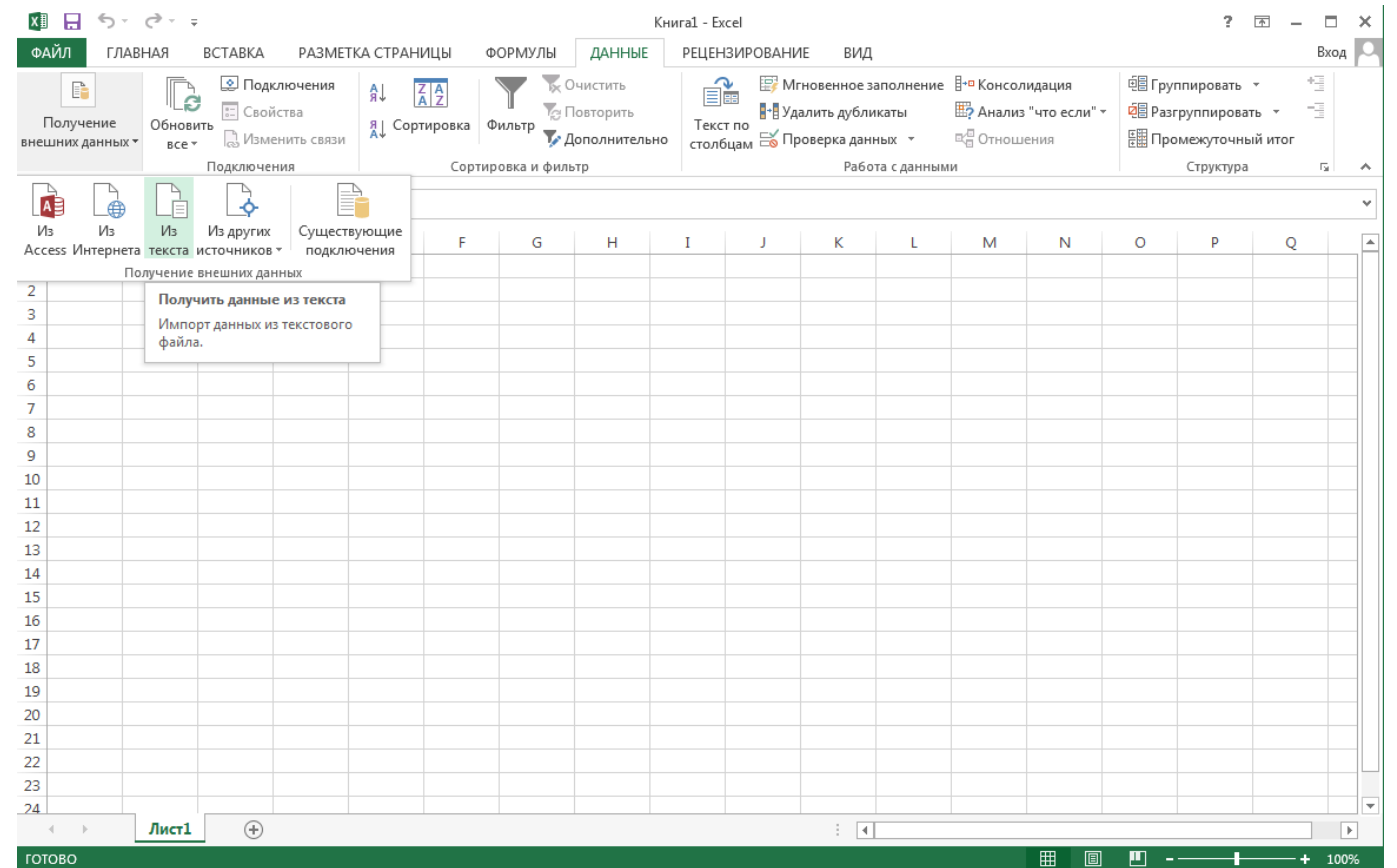
Excel



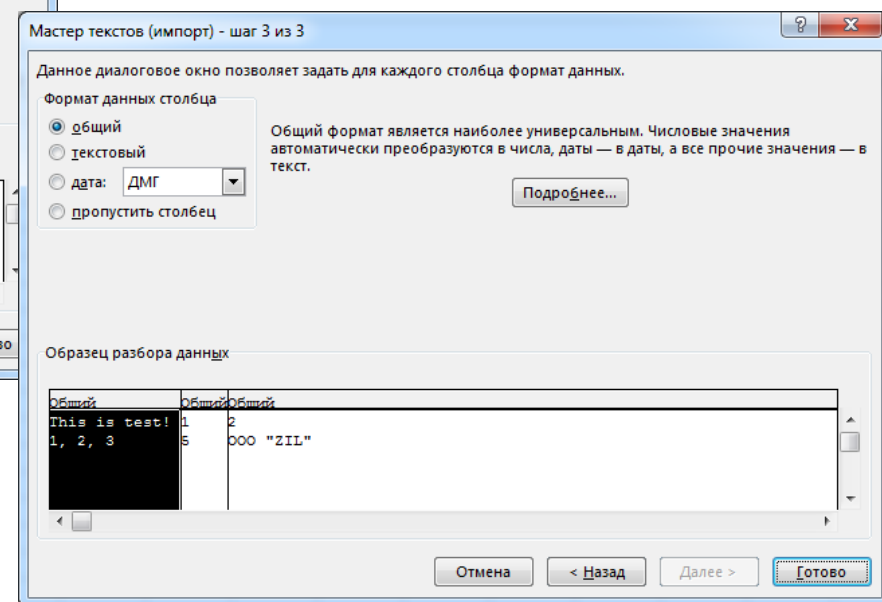
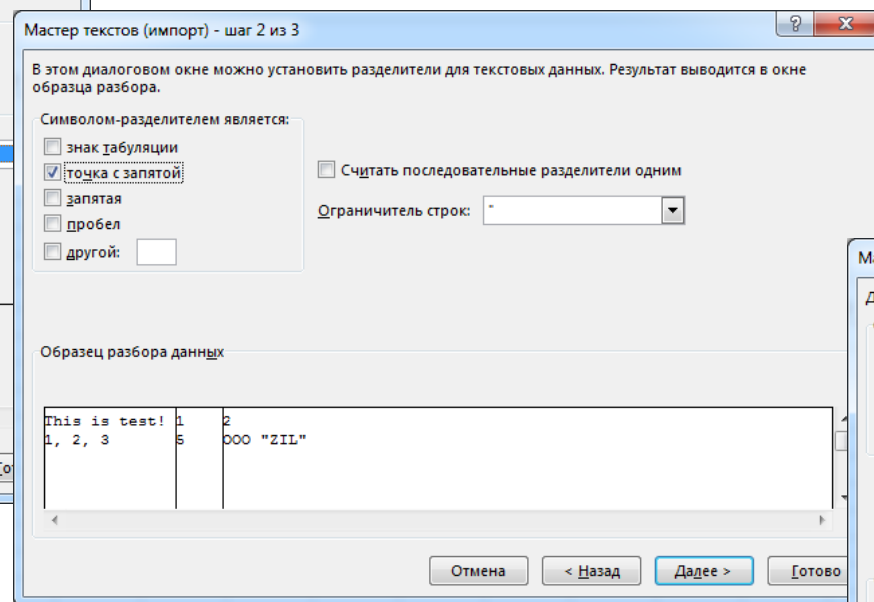
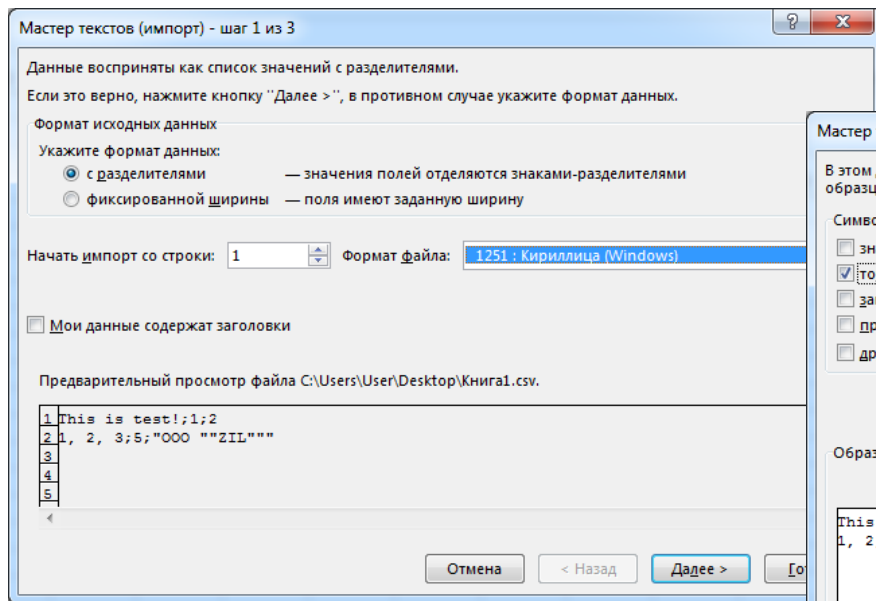
Реальное содержимое CSV

«Правильная» работа с CSV в Excel

Учитывая, что формат CSV сейчас превратился в формат DSV, то не всегда корректно открывать CSV-файл обычным образом. Более правильно будет создать новый пустой документ и импортировать данные посредством пункта «Из текста» действия «Получение внешних данных» раздела «Данные» верхнего меню.



Импорт CSV



CSV в языках программирования

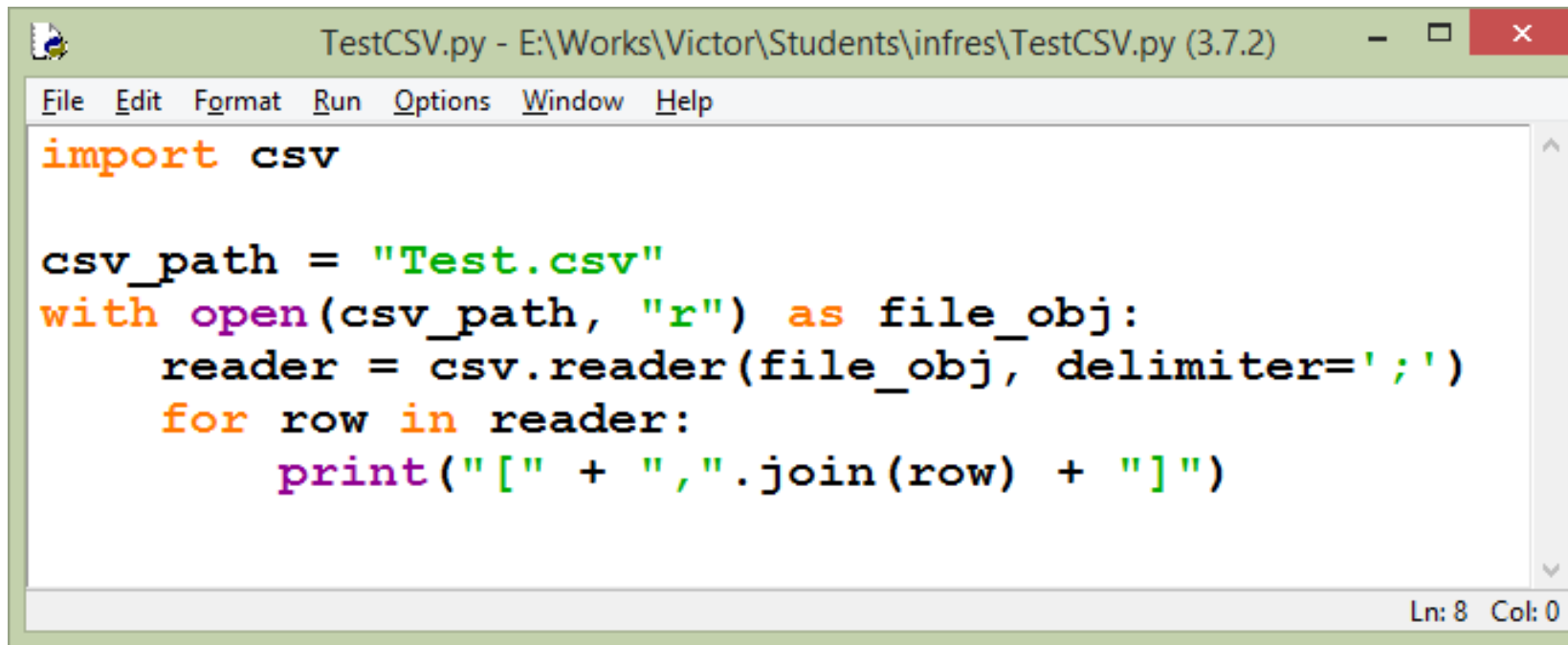
Сегодня для работы с CSV-документами во всех популярных языках программирования есть специальные библиотеки, обеспечивающие как чтение и разбор подобных файлов, так и их запись. Мы разберём два примера:

- библиотеку csv для языка Python;
- средства для работы с CSV и Excel-форматами данных библиотеки Pandas.

csv для Python

Библиотека csv встроена в стандартную поставку языка Python и позволяет выполнять как чтение и разбор файлов в данном формате, так и их запись.

Пример чтения и разбора CSV-файла

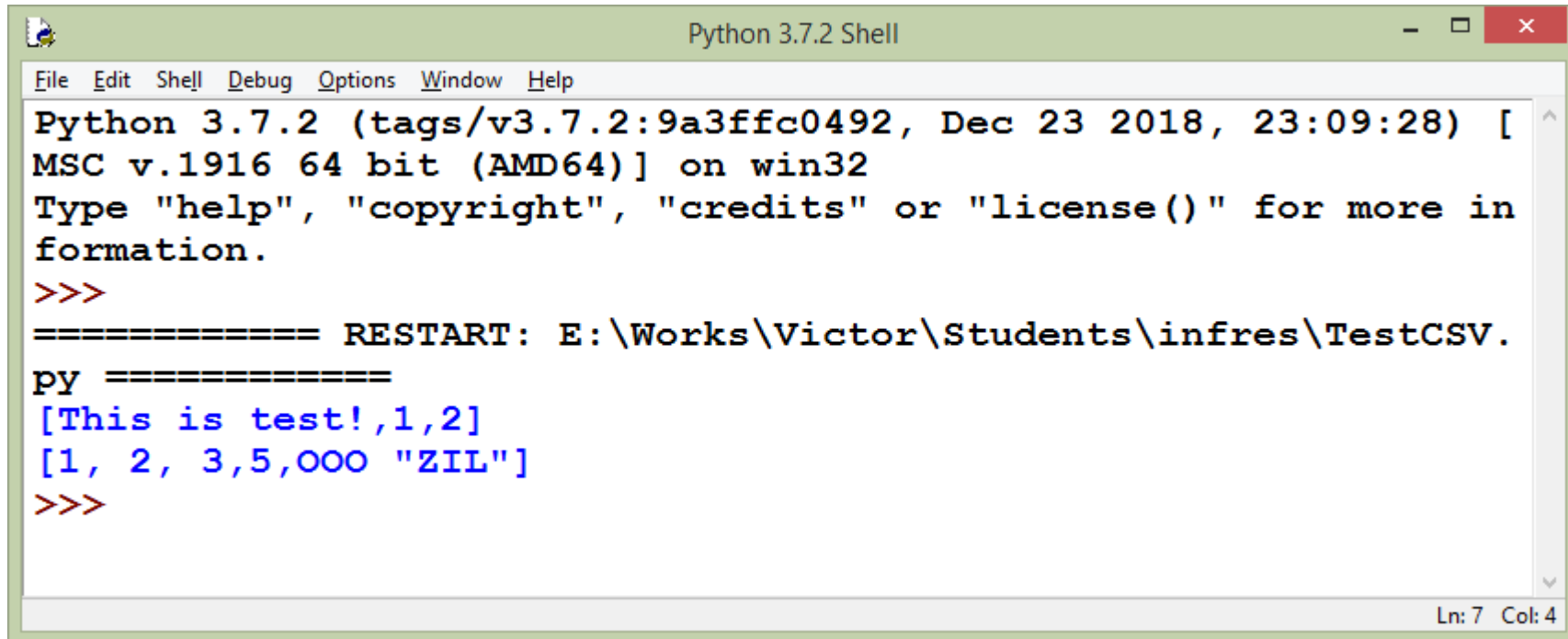
A screenshot of a Python IDE window titled "TestCSV.py - E:\Works\Victor\Students\infres\TestCSV.py (3.7.2)". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code editor contains the following Python code:

```
import csv

csv_path = "Test.csv"
with open(csv_path, "r") as file_obj:
    reader = csv.reader(file_obj, delimiter=';')
    for row in reader:
        print "[" + ",".join(row) + "]"
```

The status bar at the bottom right indicates "Ln: 8 Col: 0".

Результат



```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [
MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
===== RESTART: E:\Works\Victor\Students\infres\TestCSV.
py =====
[This is test!,1,2]
[1, 2, 3,5,000 "ZIL"]
>>>
```

Ln: 7 Col: 4

Лучше использовать Pandas!

Pandas (название происходит не от наименования животного, а является сокращением слов «panel data» – табличные данные) создавалась прежде всего для обеспечения быстрой и удобной работы с более сложными структурами данных, чем просто многомерные массивы.

За 10 лет развития библиотеки Pandas в неё были добавлены многочисленные иные возможности и в настоящее время она рассматривается, прежде всего, как инструмент для работы с «большими данными» (BigData) и как основа для «машинного обучения» (Machine Learning).

Pandas доступна по адресу [1].

Для установки pandas можно использовать команду*:

`python.exe -m pip install pandas`

* Здесь и далее приводятся команды для ОС Windows. Для ОС Linux команды будут без префикса `python.exe -m`.

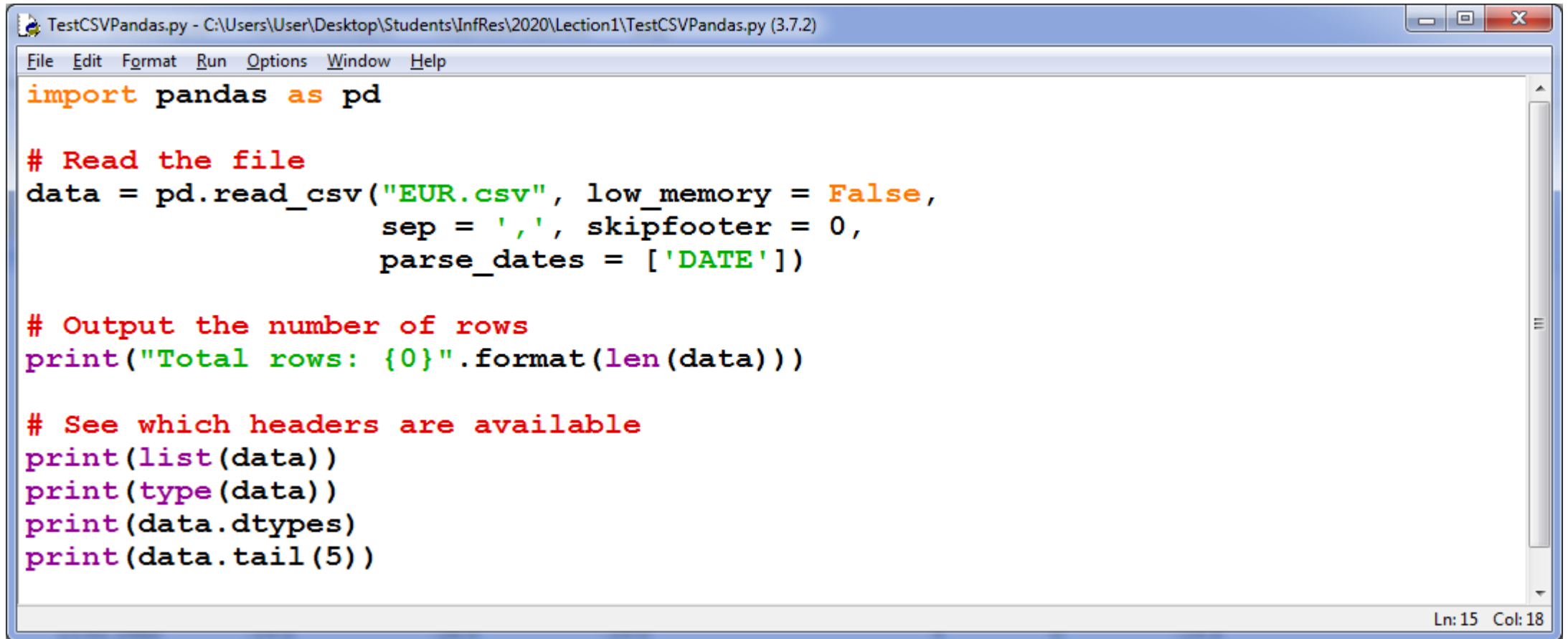
Загрузка из CSV с помощью Pandas

Библиотека Pandas обеспечивает загрузку данных из CSV-формата с помощью метода `read_csv`.

В качестве примера загрузим файл с курсом Евро с сайта <http://export.rbc.ru/expdocs/free.cb.0.shtml>.

	A	B	C	D	E	F	G	H	I
1	TICKER	DATE	OPEN	HIGH	LOW	CLOSE	VOL	WAPRICE	NOMINAL
2	EUR	18.09.2018				79.3595			1
3	EUR	19.09.2018				79.1749			1
4	EUR	20.09.2018				78.3613			1
5	EUR	21.09.2018				77.7529			1
6	EUR	22.09.2018				78.0753			1
7	EUR	25.09.2018				77.6844			1

Пример



The screenshot shows a Python IDE window titled "TestCSVpandas.py - C:\Users\User\Desktop\Students\InfRes\2020\Lecture1\TestCSVpandas.py (3.7.2)". The menu bar includes File, Edit, Format, Run, Options, Window, and Help. The code in the editor is as follows:

```
import pandas as pd

# Read the file
data = pd.read_csv("EUR.csv", low_memory = False,
                   sep = ',', skipfooter = 0,
                   parse_dates = ['DATE'])

# Output the number of rows
print("Total rows: {0}".format(len(data)))

# See which headers are available
print(list(data))
print(type(data))
print(data.dtypes)
print(data.tail(5))
```

The status bar at the bottom right indicates "Ln: 15 Col: 18".

Результат

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:\Users\User\Desktop\Students\InfRes\2020\Lecton1\TestCSVPandas.py
Total rows: 99
['TICKER', 'DATE', 'OPEN', 'HIGH', 'LOW', 'CLOSE', 'VOL', 'WAPRICE', 'NOMINAL']
<class 'pandas.core.frame.DataFrame'>
TICKER      object
DATE        datetime64[ns]
OPEN         float64
HIGH         float64
LOW          float64
CLOSE        float64
VOL          float64
WAPRICE      float64
NOMINAL      int64
dtype: object
   TICKER  DATE      OPEN  HIGH  LOW  CLOSE  VOL  WAPRICE  NOMINAL
94  EUR 2020-02-05   NaN   NaN  NaN  70.1265  NaN     NaN      1
95  EUR 2020-02-06   NaN   NaN  NaN  69.7443  NaN     NaN      1
96  EUR 2020-02-07   NaN   NaN  NaN  69.0837  NaN     NaN      1
97  EUR 2020-02-08   NaN   NaN  NaN  69.6288  NaN     NaN      1
98  EUR 2020-02-11   NaN   NaN  NaN  69.8226  NaN     NaN      1
>>>
```

Ln: 24 Col: 4

XML

XML – это способ разметки информации, позволяющий заключить полезные данные внутрь системных обёрток – тегов. Сегодня язык XML является средством промежуточного представления данных. Иногда набор большого числа данных, описанных посредством языка XML называют XML базой данных. Хотя в полной мере такую модель нельзя назвать БД, но в каком-то смысле это верно.

XML расшифровывается как Extensible Markup Language – расширяемый язык разметки.

Язык XML является ограниченным подмножеством языка SGML, который был разработан для интернет публикаций.

Теги и элементы

Как и в языке HTML слова, заключенные в символы < > называются XML-тегами. Теги всегда используются парами – открывающийся тег + закрывающийся тег:

```
<name>Маша</name>
```

Между открывающимся и закрывающимся тегами может располагаться какая-либо информация. Все вместе (информация и пара тегов) образуют XML-элемент. Поэтому информацию между тегами обычно называют содержимым элемента.

Правила определения элементов

1. Каждому открывающемуся тегу всегда соответствует закрывающийся;
2. Теги не могут перекрываться;
3. Есть только один корневой элемент;
4. Регистр символов (верхний/нижний) для XML существенен;
5. Должны использоваться правильные XML-имена.

Стандарт языка XML хорошо описан на сайте консорциума W3C [2].

Атрибуты

Теги могут содержать атрибуты:

```
<person age = '23'>Маша</person>
```

```
<person age = "28">Даша</person>
```

Атрибуты всегда содержат значения! Кавычки могут быть как одинарные, так и двойные.

Пустые элементы

Пустые элементы могут быть записаны сокращенно:

```
<name/>
```

```
<!-- Это пустой тег -->.
```

Комментарии пишутся как в языке HTML

Пример

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type='text/xsl' href='population3.xsl'?>
<population>
  <year number = "2011">142865433</year>
  <year number = "2012">143056383</year>
  <year number = "2013">143347059</year>
  <year number = "2014">143666931</year>
  <year number = "2015">146267288</year>
  <year number = "2016">146544710</year>
  <year number = "2017">146804372</year>
  <year number = "2018">146880432</year>
  <year number = "2019">146780720</year>
  <year number = "2020">146748590</year>
  <year number = "2021">147182123</year>
  <year number = "2022">146980061</year>
  <year number = "2023">146424729</year>
  <year number = "2024">146150789</year>
</population>
```

XSLT

Язык XML сам по себе не предназначен для каких-либо операций. Он лишь позволяет структурировать данные. По своей сути XML – это язык разметки документов.

Для получения из XML-документа какого-либо нормального выходного результата обычно применяются дополнительные специальные средства. Наиболее популярное из них – это XSLT.

The Extensible Stylesheet Language Family (XSL) - это набор рекомендаций для преобразования и отображения XML-документов.

XSLT – это язык XSL-преобразований для XML-документа.

XSLT

Язык XSLT состоит из набора тегов, как и язык XML. Теги языка XSLT – это правила разметки некоторого XML-документа. Без применения к XML-документу XSLT бесполезен.

Описание языка XSLT приведено на сайте консорциума W3C [3, 4].

Каждый тег XSLT либо сам по себе что-то добавляет в разметку XML-документа, либо выполняет какие-то действия на основе конкретного XML-тега. Выбор XML-тега, обрабатываемого XSLT-тегов осуществляется в соответствии с правилами XPATH [5].

Пример преобразования в HTML 1 ч.

```
<?xml version="1.0" encoding="utf-8"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:template match = "/">  
    <html>  
      <head>  
        <title>Население России</title>  
      </head>
```

Пример преобразования в HTML 2 ч.

```
<body>
<h1>
  Население России
</h1>
<dl>
  <xsl:for-each select = "population/year">
    <dt>
      <b><xsl:value-of select = "@number"/></b>
    </dt>
```

```
<dd>
  <xsl:value-of select = "text()"/>
</dd>
</xsl:for-each>
</dl>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Как применить XSLT к XML?

Пусть наше xsl-преобразование лежит в файле population.xls. Тогда для его применения добавим в XML следующий код:

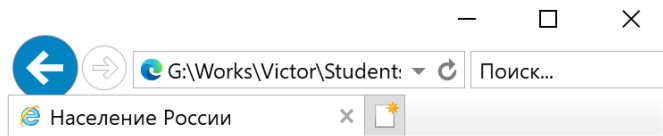
```
<?xml-stylesheet type='text/xsl' href='population.xsl'?>
```

После этого можно открыть XML-файл в старом браузере:

- ☐ Internet Explorer.

Преобразование сработает автоматически.

Или использовать более сложные конвертеры. Самый простой из них xsltproc из libxslt.



Население России

2011

142865433

2012

143056383

2013

143347059

2014

143666931

2015

146267288

2016

146544710

2017

146804372

2018

146880432

2019

146780720

2020

146748590

2021

147182123

2022

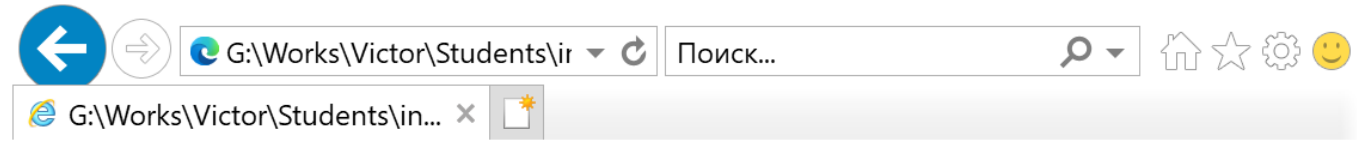
146980061

2023

146424729

2024

146150789



142865433 143056383 143347059 143666931 146267288 146544710 146804372
146880432 146780720 146748590 147182123 146980061 146424729 146150789

Отображение XML-файла в IE
до применения XLS

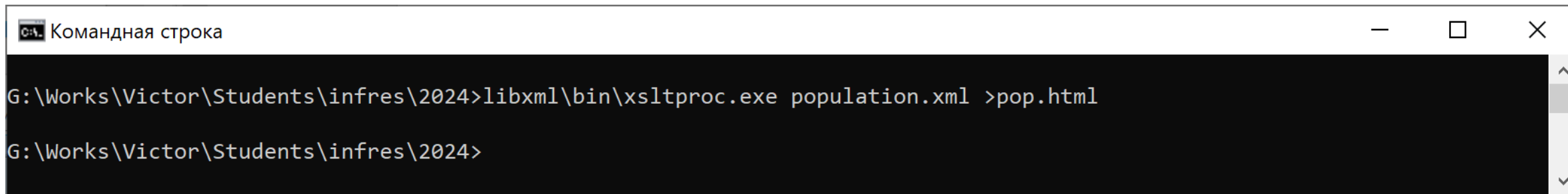
Отображение XML-файла в IE после
применения XLS

Как применить XSLTPROC?

Для начала скачаем по ссылке [6] или для windows [7]. Предположим, что он теперь лежит в папке libxml\bin. Тогда, чтобы из xml-файла получить результат xslt преобразования (html-файл) нужно запустить cmd и выполнить команду:

```
libxml\bin\xsltproc.exe population.xml >pop.html
```

Полученный файл pop.html можно открыть любым браузером.

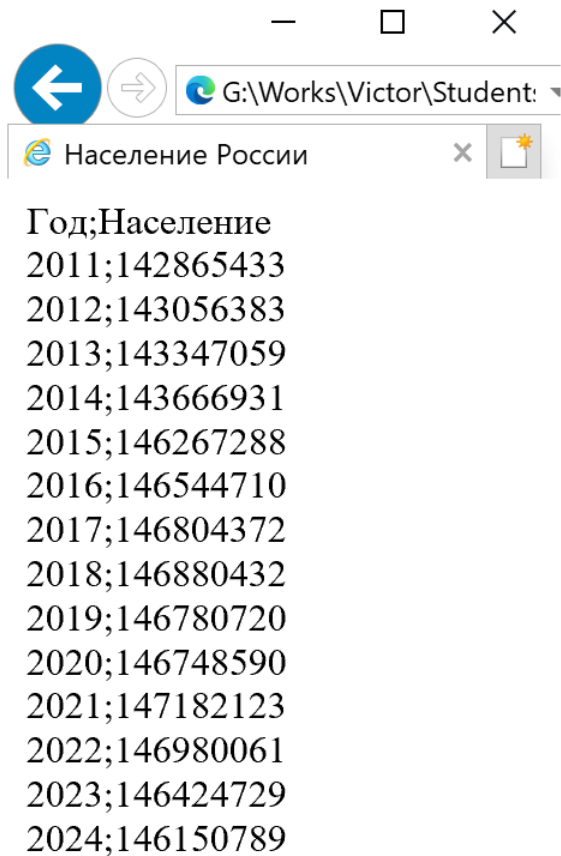


```
Командная строка
G:\Works\Victor\Students\infres\2024>libxml\bin\xsltproc.exe population.xml >pop.html
G:\Works\Victor\Students\infres\2024>
```

Почти CSV

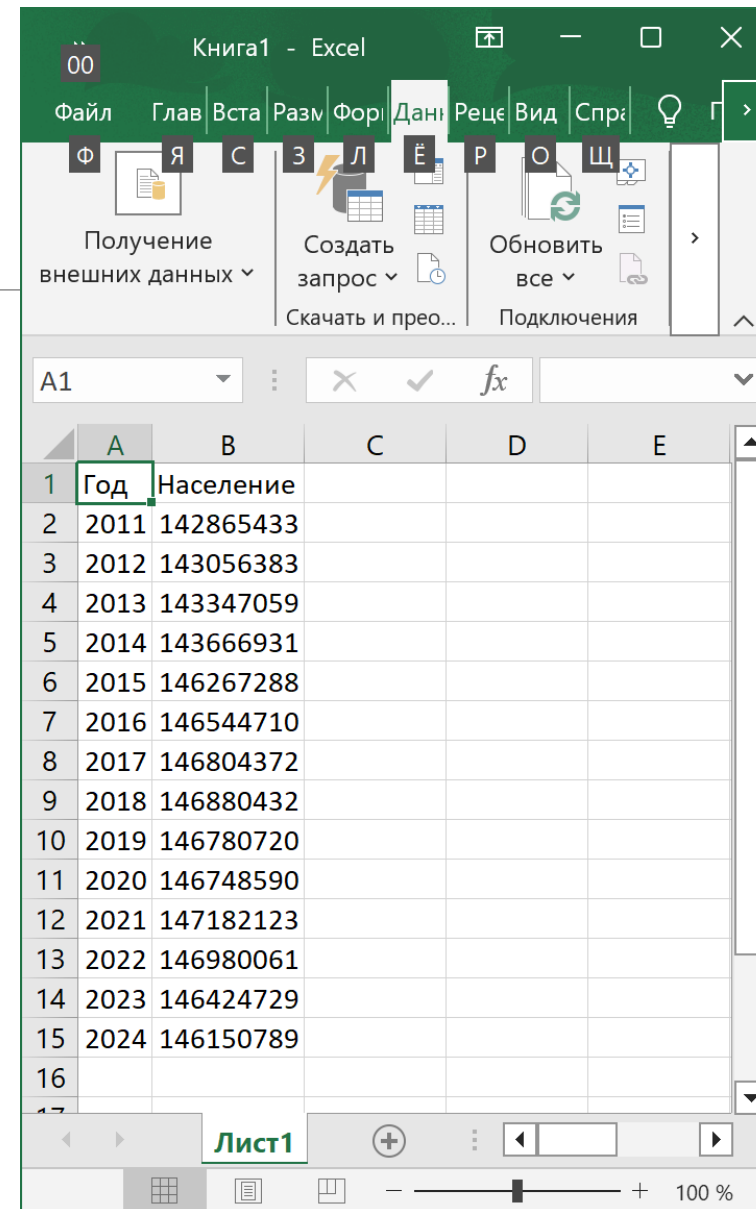
```
<body>
  Год;Население
  <br/>
  <xsl:for-each select = "population/year">
    <xsl:value-of select = "@number"/>;<xsl:value-of select = "text()"/><br/>
  </xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Результат



A screenshot of a web browser window. The address bar shows the path G:\Works\Victor\Student:. The page title is "Население России". The table contains two columns: "Год" (Year) and "Население" (Population). The data spans from 2011 to 2024.

Год	Население
2011	142865433
2012	143056383
2013	143347059
2014	143666931
2015	146267288
2016	146544710
2017	146804372
2018	146880432
2019	146780720
2020	146748590
2021	147182123
2022	146980061
2023	146424729
2024	146150789



A screenshot of the Microsoft Excel application window titled "Книга1 - Excel". The "Данные" (Data) tab is active in the ribbon. The data is entered into a table with columns A and B. Column A is labeled "Год" and column B is labeled "Население". The data spans from row 2 to row 15.

Год	Население
2011	142865433
2012	143056383
2013	143347059
2014	143666931
2015	146267288
2016	146544710
2017	146804372
2018	146880432
2019	146780720
2020	146748590
2021	147182123
2022	146980061
2023	146424729
2024	146150789

Основные элементы XLS. Шаблон

Основная единица языка XLS – это шаблон (template). Шаблон – это описание правил отображения для конкретного XML-тега. Создаётся шаблон при помощи конструкции `<xsl:template>`.

Поэтому наше решение с использованием конструкции `<xsl:for-each>` не совсем концептуально правильное. Более правильно применять для каждого объекта `year` шаблон его отображения.

Такой подход позволяет быстро перейти от одного способа визуализации к другому, просто подменив нужный шаблон.

Перепишем первый пример с использованием шаблонов.

Использование шаблонов

```
<body>
  <h1>
    Население России
  </h1>
  <dl>
    <xsl:apply-templates select = "population/year"/>
  </dl>
</body>
</html>
</xsl:template>
```

```
<xsl:template match = "year">
  <dt>
    <b>
      <xsl:value-of select = "@number"/>
    </b>
  </dt>
  <dd>
    <xsl:value-of select = "text()"/>
  </dd>
</xsl:template>
</xsl:stylesheet>
```

Использование разных форматов

В наших примерах численность населения России отображалась визуально плохо. Девять цифр подряд воспринимаются глазом человека не очень хорошо. Обычно каждые три цифры выделяют пробелом.

Например, вместо **146880432** пишут **146 880 432**. Как это сделать?

Прежде всего, стоит отметить, что в России и на Западе приняты разные форматы. Например, в Америке в такой ситуации вставляют не пробелы, а запятые: **146,880,432**. Поэтому нам придётся настроить собственный формат чисел. Это делается при помощи тега `<xsl:decimal-format/>`.

Для применения формата к числу используем функцию `format-number` [8].

Пример использования формата 1 ч.

```
<?xml version="1.0" encoding="utf-8"?>  
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
  <xsl:decimal-format name = "tst" grouping-separator = " " />  
  <xsl:template match = "/">  
    <html>  
      <head>  
        <title>Население России</title>  
      </head>
```

Пример использования формата 2 ч.

```
<body>
  <h1>
    Население России
  </h1>
  <dl>
    <xsl:apply-templates select = "population/year"/>
  </dl>
</body>
</html>
</xsl:template>
<xsl:template match = "year">
```

```
<dt>
  <b>
    <xsl:value-of select = "@number"/>
  </b>
</dt>
<dd>
  <xsl:value-of
    select = "format-number(text(), '### ## #', 'tst')"/>
</dd>
</xsl:template>
</xsl:stylesheet>
```


Результат



The screenshot shows a web browser window with the address bar displaying 'G:\Works\Victor\Student...' and a search bar with the text 'Поиск...'. The browser tab is titled 'Население России'. The main content area displays the title 'Население России' followed by a table of population data for Russia from 2011 to 2024. The table has two columns: the first column contains the years from 2011 to 2024, and the second column contains the corresponding population figures. The browser window includes standard navigation buttons (back, forward, home, search) and window control buttons (minimize, maximize, close).

Население России	
2011	142 865 433
2012	143 056 383
2013	143 347 059
2014	143 666 931
2015	146 267 288
2016	146 544 710
2017	146 804 372
2018	146 880 432
2019	146 780 720
2020	146 748 590
2021	147 182 123
2022	146 980 061
2023	146 424 729
2024	146 150 789

Использование агрегатных функций

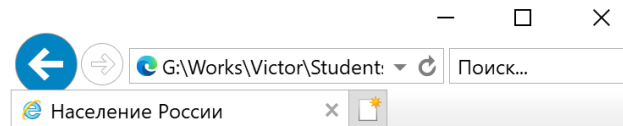
В XSLT, как и в SQL есть многострочные (агрегатные) функции, которые позволяют собрать аккумулярованную информацию сразу из нескольких строк. Для знакомства с ними рассмотрим простой пример. Добавим в предыдущие задачи строку со средним населением России за эти 13 лет.

Пример среднее значение

```
<body>
  <h1>
    Население России
  </h1>
  <dl>
    <xsl:apply-templates select = "population/year"/>
  </dl>
  <h2>
    Среднее население России в данные года:
    <xsl:value-of
      select = "sum(population/year/text()) div
                count(population/year)" />
  </h2>
</body>
</html>
</xsl:template>
```

```
<xsl:template match = "year">
  <dt>
    <b>
      <xsl:value-of select = "@number"/>
    </b>
  </dt>
  <dd>
    <xsl:value-of
      select = "format-number(text(), '### ### ##', 'tst')"/>
  </dd>
</xsl:template>
</xsl:stylesheet>
```

Результат



Население России

2011	142 865 433
2012	143 056 383
2013	143 347 059
2014	143 666 931
2015	146 267 288
2016	146 544 710
2017	146 804 372
2018	146 880 432
2019	146 780 720
2020	146 748 590
2021	147 182 123
2022	146 980 061
2023	146 424 729
2024	146 150 789

**Среднее население России в
данные года: 145692830**

XML в языках программирования

Сегодня для работы с XML-документами во всех популярных языках программирования есть специальные библиотеки, обеспечивающие как чтение и разбор подобных файлов, так и их запись. Мы разберём

библиотеку `minidom` (`xml.dom.minidom`) для языка Python [9-11].

xml.dom.minidom для Python

Библиотека `xml.dom.minidom` встроена в стандартную поставку языка Python и позволяет выполнять как чтение и разбор файлов в данном формате, так и их запись.

Пример чтения и разбора XML-файла

```
import xml.dom.minidom

xml_path = "population.xml"
with open(xml_path, "r") as file_obj:
    doc = xml.dom.minidom.parse(file_obj)
    node = doc.documentElement
    print('Население России')
    years = node.getElementsByTagName("year")
    for year in years:
        print('%s: %s' % (year.attributes['number'].value, year.childNodes[0].nodeValue))
```

Результат



```
*IDLE Shell 3.12.4*
File Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: G:\Works\Victor\Students\infres\2024\TestXML.py
Население России
2011: 142865433
2012: 143056383
2013: 143347059
2014: 143666931
2015: 146267288
2016: 146544710
2017: 146804372
2018: 146880432
2019: 146780720
2020: 146748590
2021: 147182123
2022: 146980061
2023: 146424729
2024: 146150789
Ln: 5 Col: 0
```


Часть 2

ВЫБОРКА ДАННЫХ ИЗ HTML-СТРАНИЦ

Информация в виде HTML-страниц

К сожалению, часть информации в сети Интернет не всегда представлена в удобном для машинной обработки виде. Наиболее распространённым видом представления данных сегодня являются HTML-страниц. Их организация может быть удобна для использования в современных офисных продуктах, а может быть специально затруднена для копирования и использования в собственных целях.

Для получения данных из таких источников существуют специальные средства.

Загрузка HTML-страниц в Excel

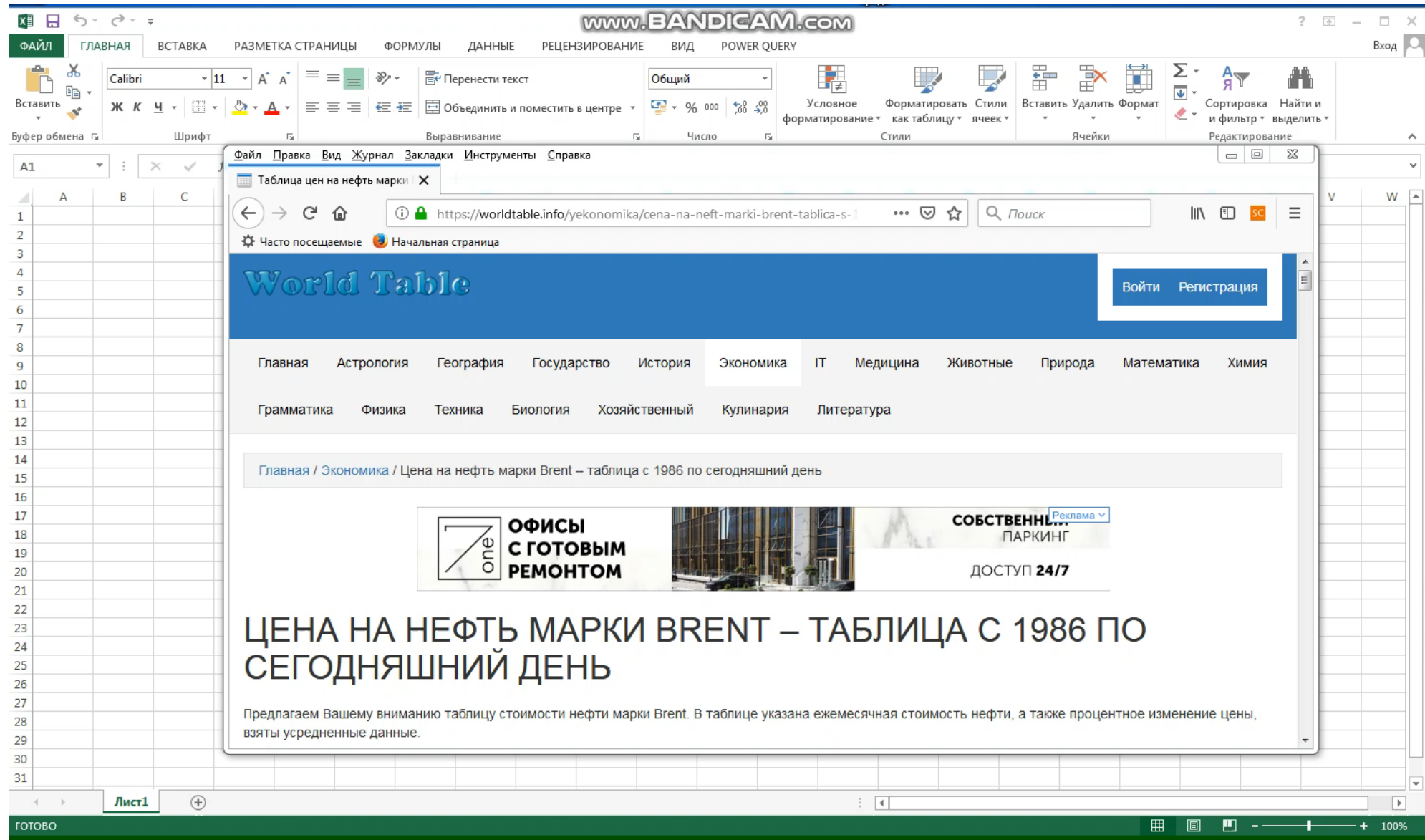
Самый простой способ работы с информацией – это её импорт в Excel.

В обычной ситуации можно выделить данные мышкой и вставить в Excel-лист. Но, если данные сложные и их много, то сделать это не всегда удобно и не всегда получается. Тогда можно использовать функцию импорта «Из Интернета» раздела «Данные» верхнего меню.

Для лучшего понимания проблема рассмотрим видео-пример.

Пример 1

Данный слайд – это видеоролик. В PDF-версии лекции он не доступен. Ссылка на ролик размещена отдельно в учебных материалах.



The screenshot displays a Microsoft Excel spreadsheet in the background, with a web browser window overlaid on top. The browser window shows the 'World Table' website, which is displaying a table of Brent oil prices. The website's header includes a search bar and navigation links. The main content area features a large title 'ЦЕНА НА НЕФТЬ МАРКИ BRENT – ТАБЛИЦА С 1986 ПО СЕГОДНЯШНИЙ ДЕНЬ' (Brent Oil Price – Table from 1986 to Today). Below the title, there is a brief description of the table's content. The Excel spreadsheet in the background shows a grid with columns A, B, and C, and rows 1 through 31. The status bar at the bottom of the Excel window indicates 'Готово' (Ready) and 'Лист1' (Sheet1).

Загрузка HTML-страниц с Power Query

Ещё более хороший результат можно получить с помощью расширения Power Query. Рассмотрим видео-пример для тех же самых данных.

Пример 2

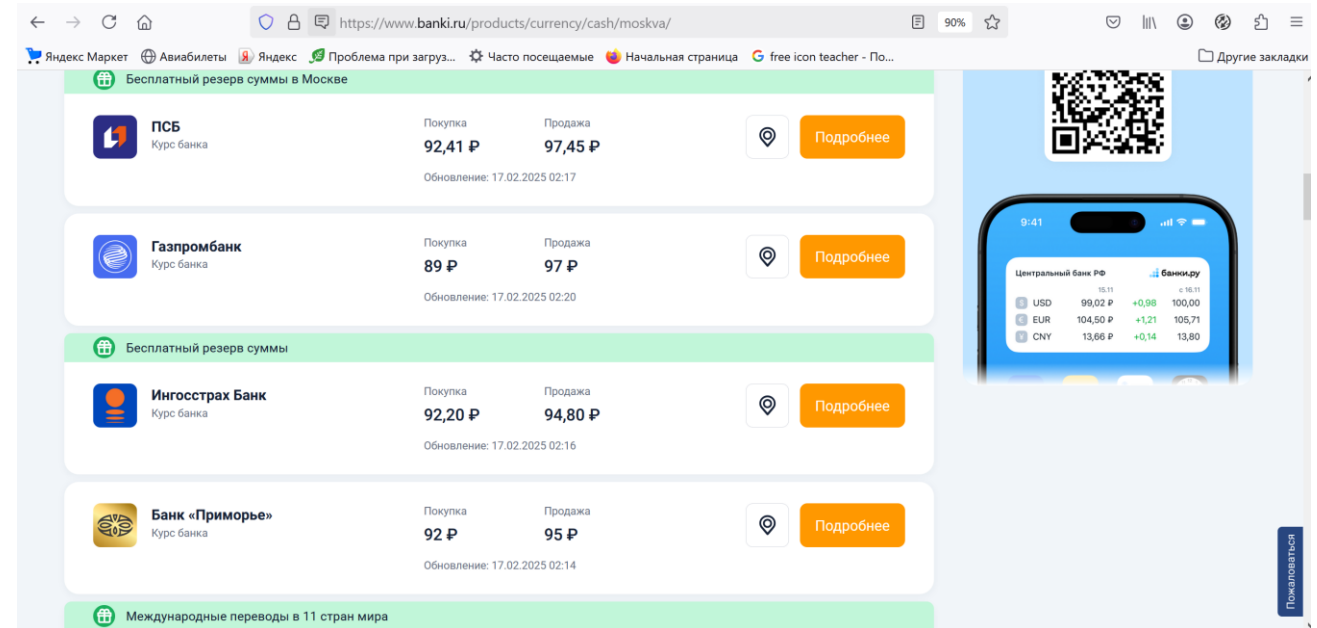
Данный слайд – это видеоролик. В PDF-версии лекции он не доступен. Ссылка на ролик размещена отдельно в учебных материалах.

The screenshot shows a Microsoft Excel spreadsheet in the background with the 'Главная' (Home) tab selected. Overlaid on the spreadsheet is a web browser window displaying the 'World Table' website. The browser's address bar shows the URL: <https://worldtable.info/yekonomika/cena-na-neft-marki-brent-tablica-s-1>. The website has a blue header with the 'World Table' logo and navigation links for 'Главная', 'Астрология', 'География', 'Государство', 'История', 'Экономика', 'IT', 'Медицина', 'Животные', 'Природа', 'Математика', and 'Химия'. Below the header, there is a section for 'Главная / Экономика / Цена на нефть марки Brent – таблица с 1986 по сегодняшний день'. An advertisement for 'ОФИСЫ С ГОТОВЫМ РЕМОНТОМ' (Offices with ready-made repair) is visible, along with a 'СОБСТВЕННЫЙ ПАРКИНГ' (Private Parking) sign. The main title of the page is 'ЦЕНА НА НЕФТЬ МАРКИ BRENT – ТАБЛИЦА С 1986 ПО СЕГОДНЯШНИЙ ДЕНЬ'. Below the title, there is a brief description: 'Предлагаем Вашему вниманию таблицу стоимости нефти марки Brent. В таблице указана ежемесячная стоимость нефти, а также процентное изменение цены, взяты усредненные данные.'

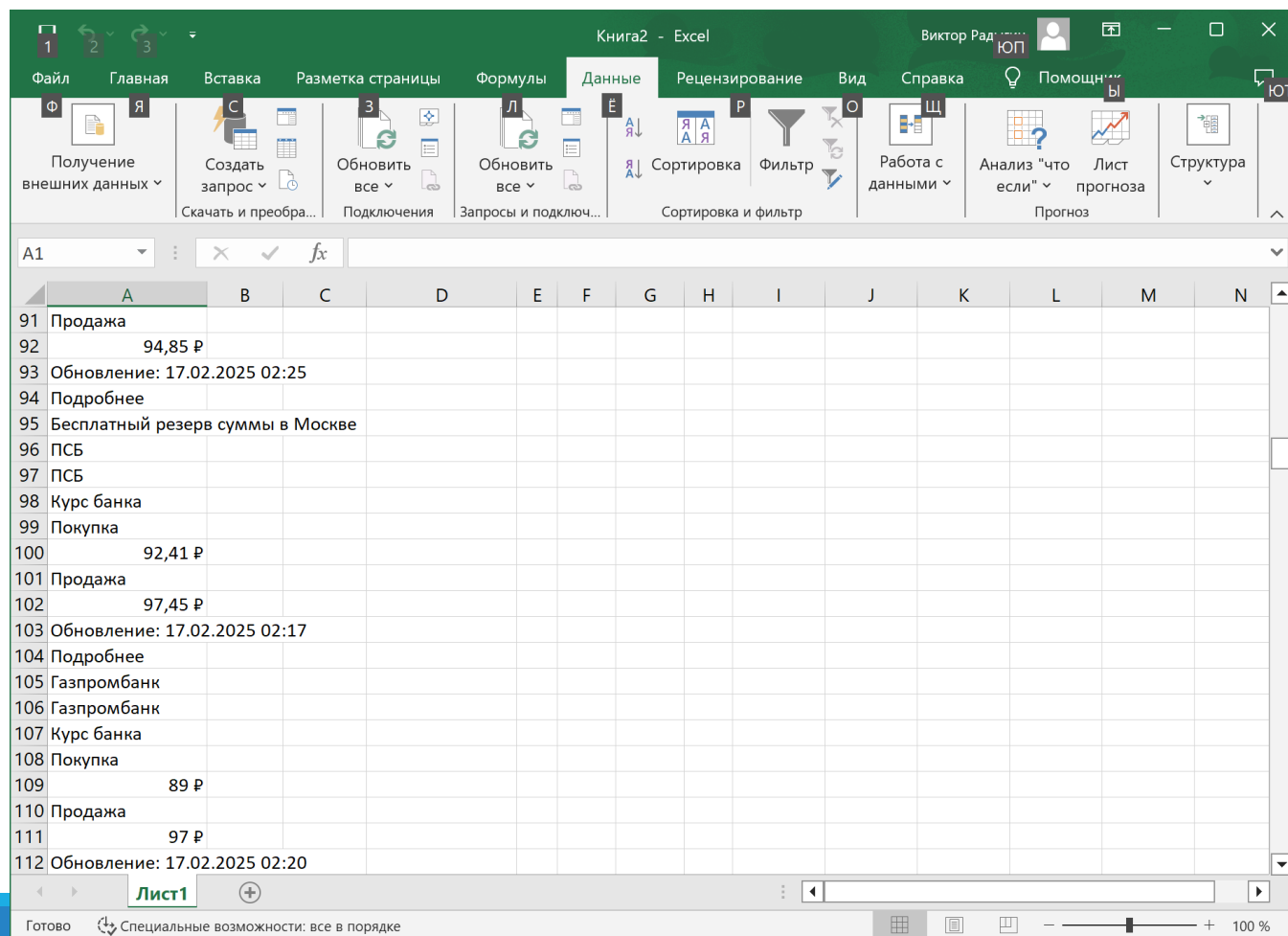
Не всё Excel под силу!

Существуют сайты, вёрстка которых устроена в современном адаптивном дизайне и не использует таблиц для представления информации.

В качестве примера можем рассмотреть страницу <https://www.banki.ru/products/currency/cash/moskva/>, на которой показана информация о курса покупки и продажи валюты в московских банках.



Не всё Excel под силу!



Полезные ссылки

1. <https://pandas.pydata.org/> – библиотека Pandas для Python.
2. <https://www.w3.org/TR/xml/> – рекомендации W3C по стандарту XML.
3. <https://www.w3.org/TR/1999/REC-xslt-19991116> – рекомендации W3C по стандарту XSLT 1.0 (поддерживаются большинством браузеров).
4. <https://www.w3.org/TR/2017/REC-xslt-30-20170608/> – рекомендации W3C по стандарту XSLT 3.0 (новейший стандарт).
5. <https://www.w3.org/TR/1999/REC-xpath-19991116/> – рекомендации W3C по стандарту XPATH 1.0 (поддерживаются большинством браузеров).
6. <https://gitlab.gnome.org/GNOME/libxslt/-/wikis/home> – страница libxslt.
7. <https://www.zlatkovic.com/pub/libxml/> – скачать libxslt для windows.
8. <https://www.w3.org/TR/1999/REC-xslt-19991116#format-number> – описание работы с функцией format-number.

Полезные ссылки

7. <https://docs.python.org/2/library/xml.etree.elementtree.html> – библиотека контейнер для представления иерархических структур в Python
8. <https://docs.python.org/2/library/xml.dom.html> – библиотека для работы с DOM-объектами в Python
9. <https://docs.python.org/2/library/xml.dom.minidom.html> – библиотека для минимальных действия с DOM-объектами в Python.
10. <https://pypi.org/project/lxml/> – библиотека lxml.
11. <https://python-scripts.com/parsing-lxml> – о библиотеке lxml.
12. https://www.w3schools.com/xml/xpath_intro.asp – основы XPath на сайте консорциума W3C.
13. <https://www.banki.ru/products/currency/cash/moskva/>
14. http://pogoda-service.ru/archive_gsod_res.php