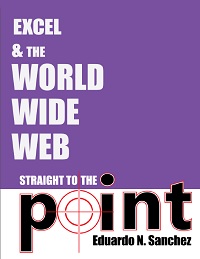
## Эдуардо Санчес. Скрапинг веб-сайтов с помощью Excel

Цель книги – предоставить инструменты для автоматизации повторяющихся задач извлечения данных из Интернета.

Eduardo Sanchez. Excel and The World Wide Web. Straight to the Point. – Holy Macro! Books, 2021. – 58 p.



Примеры кода: <https://www.mrexcel.com/download-center/excel-and-the-world-wide-web-8051/>

### Глава 1. Приступая к работе

#### Что такое HTML?

[HTML](https://ru.wikipedia.org/wiki/HTML) (Hyper Text Markup Language) – язык гипертекстовой разметки. Он используется для создания веб-сайтов. Гипертекст – это контент, который ведет себя нелинейным образом. Представьте себе веб-сайт, на каждой странице которого есть несколько ссылок на другие страницы, как того же самого сайта, так и других сайтов. Пользователь перемещается, переходя с одной страницы на другую; это гипертекстовое поведение. Обычная печатная книга – это контрпример, ее предполагается читать последовательно.

Информация в HTML помечена тегами; ниже мы поговорим об этом подробнее. Существуют и другие языки, используемые для создания веб-сайтов, такие как CSS и JavaScript, но мы не будем подробно рассматривать их здесь.

[CSS](https://ru.wikipedia.org/wiki/CSS) (Cascading Style Sheet) – каскадная таблица стилей, язык описания внешнего вида документа. Он работает вместе с HTML, который отвечать за содержимое страницы. Каскадирование означает, что можно использовать несколько CSS-файлов для создания окончательного визуального стиля. Этот язык управляет такими элементами, как размер шрифта, фоновые изображения и цветовая палитра.

[JavaScript](https://ru.wikipedia.org/wiki/JavaScript) – язык программирования для реализации динамического поведения на веб-сайтах. С его помощью разработчики могут манипулировать содержимым страницы, создавать диаграммы и взаимодействовать с API (Application Programming Interface, интерфейс прикладного программирования). Обратите внимание, что JavaScript и [Java](https://ru.wikipedia.org/wiki/Java) – это два разных языка. Говорят, что в будущем JavaScript может заменить VBA в качестве языка программирования Office.

Одна из замечательных особенностей современных браузеров заключается в том, что они предоставляют исходный код страниц. Если вы используете Google Chrome, просто щелкните правой кнопкой мыши любой элемент страницы и выберите пункт *Просмотреть код*; в правой части окна появится панель, аналогичная показанной ниже:

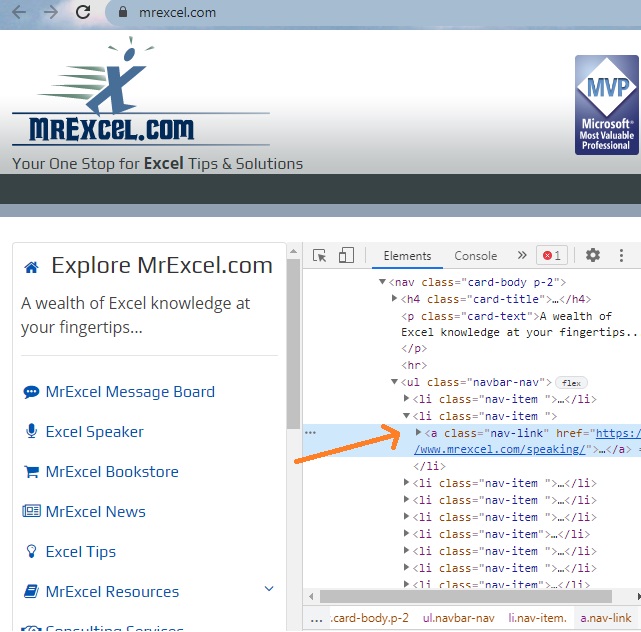


Рис. 1. Фрагмент кода HTML [веб-страницы](https://www.mrexcel.com/)

Обратите внимание:

* Ключевые слова *header, div, aside, nav, h4, li* и другие являются тегами.
* Когда маленькие черные треугольники указывают вправо, это означает, что их можно щелкнуть, чтобы развернуть и отобразить дополнительную информацию.
* Ключевые слова *id, class* и *href* являются атрибутами.
* <li – это открывающий тег, а </li> – закрывающий.
* Элемент может принадлежать к нескольким различным классам.

Если вы чувствуете себя подавленным всем этим, не паникуйте; не обязательно быть программистом HTML, чтобы работать с такого рода автоматизацией. Позже мы увидим, как читать код HTML и извлекать информацию с помощью VBA.

Хотя веб-дизайнеры используют профессиональные редакторы, можно создавать HTML-файлы в стандартных приложениях Windows, например, в *Блокноте*. Это особенно быть полезно, когда вы хотите протестировать код VBA, который будет взаимодействовать с веб-страницей, но по какой-то причине реальный сайт недоступен.

Чтобы создать локальный HTML-файл, выполните следующие действия:

* Откройте *Блокнот* и введите исходный код, как ниже.
* Сохраните его с расширением *htm*.
* Откройте этот файл с помощью браузера. Для этого, например, можно в *Проводнике* просто кликнуть на файле (если вы всё сделали верно, файл будет иметь иконку вашего браузера по умолчанию).
* Чтобы позже отредактировать исходный код, переименуйте расширение в *txt*.

<!DOCTYPE html>

<html>

<body style="background-color:powderblue;">

<h1>My Heading</h1>

<p><i>This is italic.</i></p>

<p style="color:red;">Red paragraph.</p>

<button>Click me</button>

<bdo dir="rtl"><br><br>12345<br><br></bdo>

</body>

</html>

Например, в Chrome файл выглядит так (я кликнул *Просмотреть код*):

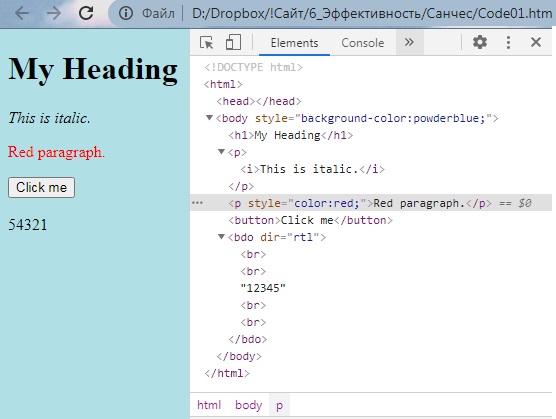


Рис. 2. Файл, созданный в Блокноте и открытый в браузере

#### Выполнение запросов

Веб-запрос позволяет извлечь информацию, хранящуюся в HTML-файле. Обычно он используется для подключения к какому-либо веб-сайту, но также можно запросить локальный файл.

Если вы планируете извлекать данные из одного и того же источника снова и снова, сохраните запрос и просто обновляйте его. Веб-запросы Excel идентифицируют таблицы на выбранной странице в Интернете и позволяют импортировать таблицы в книгу.

#### Статический запрос

Стандартный запрос Excel является статическим, то есть он не принимает параметры. Чтобы выполнить статический веб-запрос, выполните следующие действия:

1. Пройдите по меню *Данные* –> *Получить и преобразовать данные* –> *из Интернета.* (Хотя книга датируется 2021-м г., автор описывает импорт из Интернета в версии Excel 2013. На одном из ПК у меня установлена эта «древняя» версия, но, наверное, движок импорта настолько отстал от развития www, что мне не удалось с его помощью импортировать ни одной из страниц. Примеры в этом разделе выполнены в Excel 365.)
2. Откроется диалоговое окно, введите адрес страницы, которую хотите импортировать. Я ввел [ссылку](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%B3%D0%BE%D1%81%D1%83%D0%B4%D0%B0%D1%80%D1%81%D1%82%D0%B2_%D0%B8_%D0%B7%D0%B0%D0%B2%D0%B8%D1%81%D0%B8%D0%BC%D1%8B%D1%85_%D1%82%D0%B5%D1%80%D1%80%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9_%D0%BF%D0%BE_%D0%BD%D0%B0%D1%81%D0%B5%D0%BB%D0%B5%D0%BD%D0%B8%D1%8E) на страницу Википедии *Список государств и зависимых территорий по населению*.
3. Нажмите Ok.

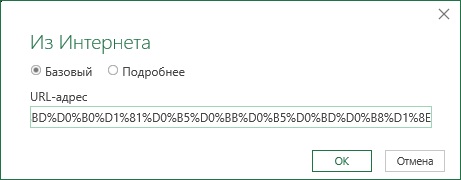


Рис. 3. Окно стандартного запроса Excel

1. При первом обращении к сайту появится окно запроса уровня доступа. Оставьте настройку по умолчанию – *Анонимно*. Нажмите *Подключение*.
2. Откроется окно навигатора Power Query. Кликайте по очереди на таблицы в левой части. Остановите свой выбор на подходящей (Table 0). Нажмите *Преобразовать данные*.

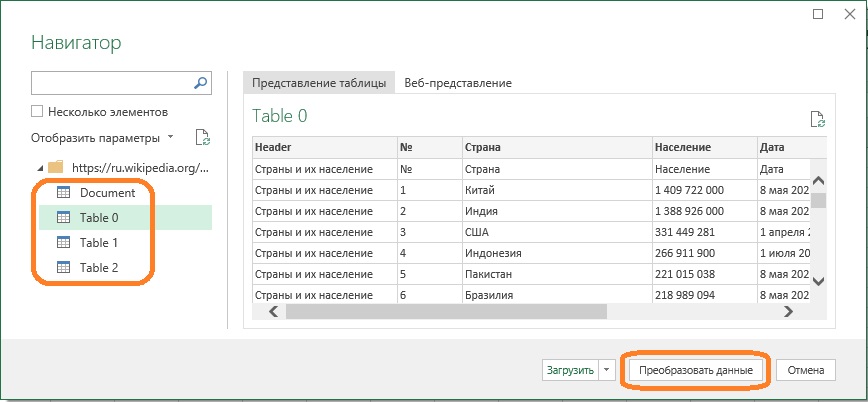


Рис. 4. Выбор таблицы для подключения

Удалите все столбцы, кроме *Страна* и *Население*. Преобразуйте тип данных в столбце *Население* в *Целое число*. Загрузите данные в умную таблицу на лист Excel. У вас получится что-то вроде:



Рис. 5. Начало умной таблицы с населением стран, импортированной из Интернета

Подробнее об импорте из Интернет с использование Power Query см. [здесь](http://baguzin.ru/wp/?p=21124).

#### Динамический веб-запрос

Допустим, вам нужно получить информацию с нескольких страниц одного веб-сайта, и в адресах страниц можно выявить шаблон. Можно передать переменную в веб-запрос с помощью функции Power Query. (Автор продолжает рассказ на основе версии Excel 2013. В ней запросы хранятся в отдельных текстовых файлах с расширением IQY. Поэтому метод основан на редактировании таких файлов.)

На первом шаге я создал обычный статический запрос к странице рейтинга АТР:

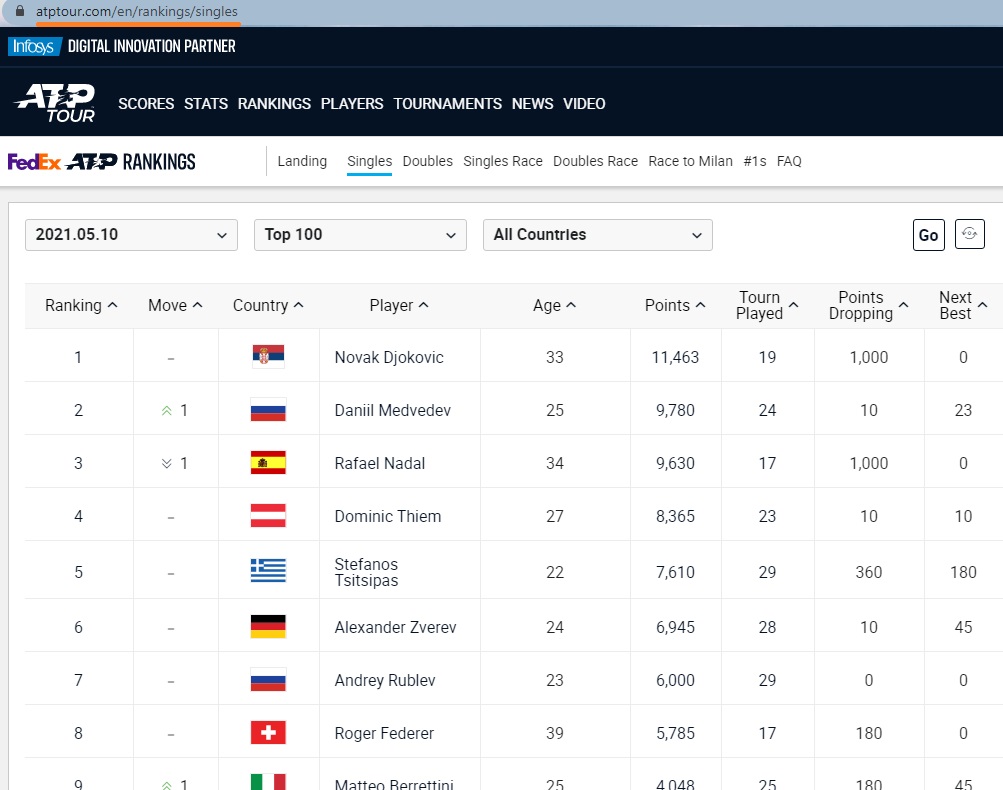


Рис. 6. Сайт АТР

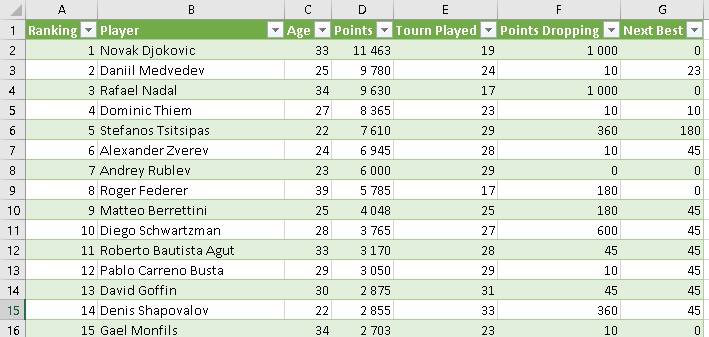


Рис. 7. Умная таблица, полученная статическим запросом Power Query

Обратите внимание: вверху на рис. 6 адрес страницы <https://www.atptour.com/en/rankings/singles>. Страница рейтинга парных игроков отличается лишь последним фрагментом: <https://www.atptour.com/en/rankings/doubles>. В нашем учебном примере мы создадим функцию в Power Query, которая будет считывать с листа Excel, какой рейтинг запросить из Интернета.

Шаг 2. На новом листе Excel создайте умную таблицу:

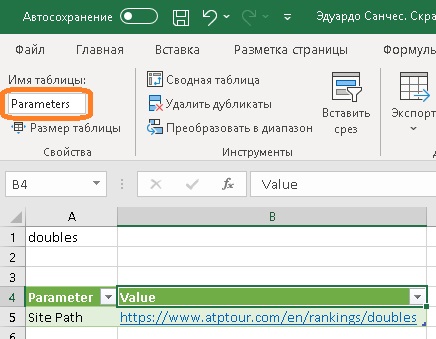


Рис. 8. Таблица параметров на листе Excel

Следующие названия должны быть такими же, как на рисунке: заголовок первого столбца *Parameter*, заголовок второго столбца *Value*, имя таблицы *Parameters*. Это позволит использовать заготовленный текст функции. В ячейке А1 я применил инструмент *Проверка данных*, чтобы создать список выбора рейтинга:

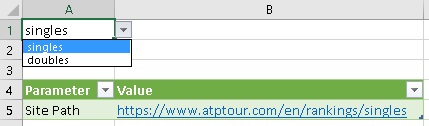


Рис. 9. Список для выбора рейтинга

Формула в В5 считывает значение в А1 и возвращает полный путь к странице сайта: ="https://www.atptour.com/en/rankings/"&A1

Шаг. 3. Создадим пользовательскую функцию *fnGetParameter* в Power Query:

= (ParameterName as text) =>

let

ParamSource = Excel.CurrentWorkbook(){[Name="Parameters"]}[Content],

ParamRow = Table.SelectRows(ParamSource, each ([Parameter]=ParameterName)),

Value=

if Table.IsEmpty(ParamRow)=true

then null

else Record.Field(ParamRow{0},"Value")

in

Value

Функция подключается к таблице *Parameters* в книге Excel, и извлекает путь к сайту. Скопируйте текст функции в буфер. В Excel пройдите по меню *Данные* –> *Получить данные* –> *Из других источников* –> *Пустой запрос*. В редакторе Power Query перейдите на вкладку *Главная* –> *Расширенный редактор*. Выделите все строки кода и нажмите Ctrl+V, чтобы вставить содержимое из буфера обмена. Нажмите *Готово*. Измените имя функции на *fnGetParameter*. Пройдите по меню *Главная* –> *Закрыть и загрузить*. Для функций используется единственный тип загрузки – *Только создать подключение*.

### Вызов функции *fnGetParameter*

Теперь в ранее созданном статическом запросе *АТР* заменим фиксированный URL на динамический, возвращаемый функцией *fnGetParameter*. В файле Excel пройдите по меню *Данные* –> *Запросы и подключения*. В области *Запросы и подключения* кликните правой кнопкой мыши на запросе *АТР* –> *Изменить*. В редакторе Power Query кликните *Расширенный редактор*:

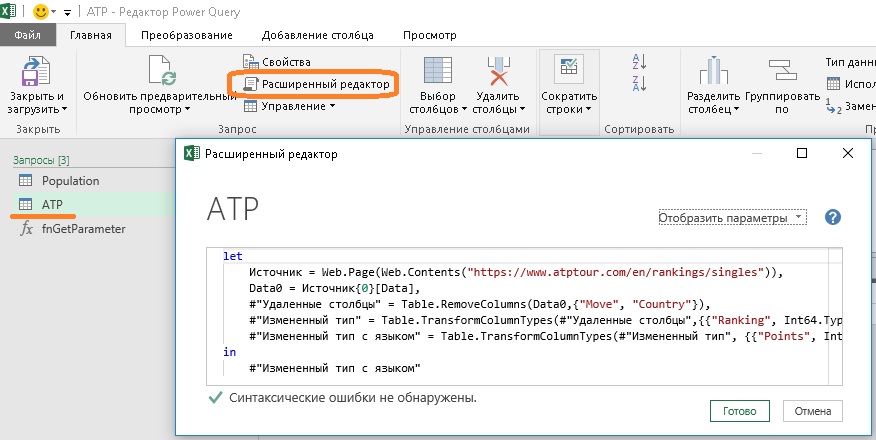


Рис. 10. Исходный код статического запроса с именем *ATP*

Вставьте строку кода сразу после *let* (не забудьте про запятую в конце строки):

sitepath = fnGetParameter("Site Path"),

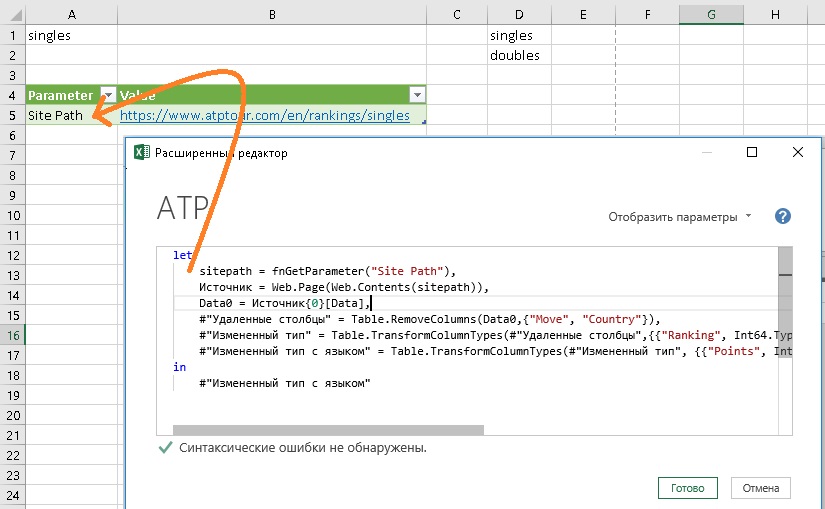


Рис. 11. Функция *fnGetParameter* возвращает значение из таблицы листа Excel

Функция вернет путь к сайту из ячейки В5 таблицы *Parameters*. Вы создали новую переменную *sitepath* для хранения значения из строки *File Path* таблицы Excel. Нажмите *Готово*. Во второй строке кода в *Расширенном редакторе* выделите путь к сайту включая кавычки, и замените его на имя переменной – *sitepath* (как показано на рис. 11). Если вы всё делали верно, как только вы наберете букву *s*, редактор выдаст контекстную подсказку, в которой в том числе будет и имя переменной *sitepath*. Откроется окно:

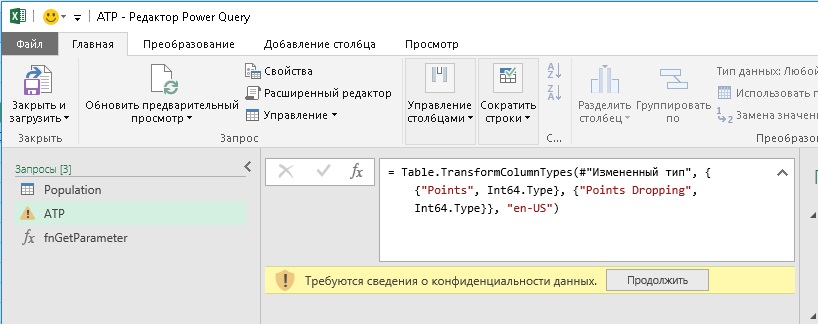


Рис. 12. Подтверждение конфиденциальности данных

Нажмите *Продолжить*. Откроется еще одно окно:

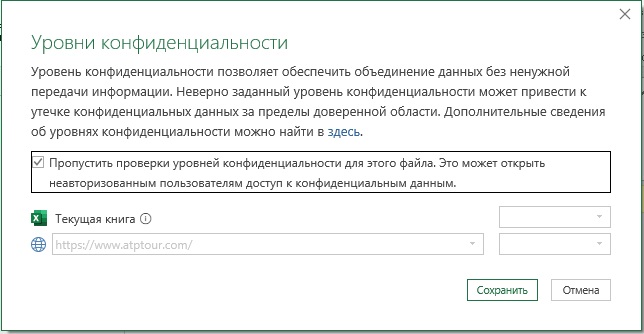


Рис. 13. Не проверять уровень конфиденциальности для этого файла

Поставьте галочку, как показано выше.

Вы можете проверить, как работает переменная *sitepath*. В редакторе Power Query перейдите в область *ПРИМЕНЕННЫЕ ШАГИ*, и кликните на первый шаг. Отразится имя страницы сайта:

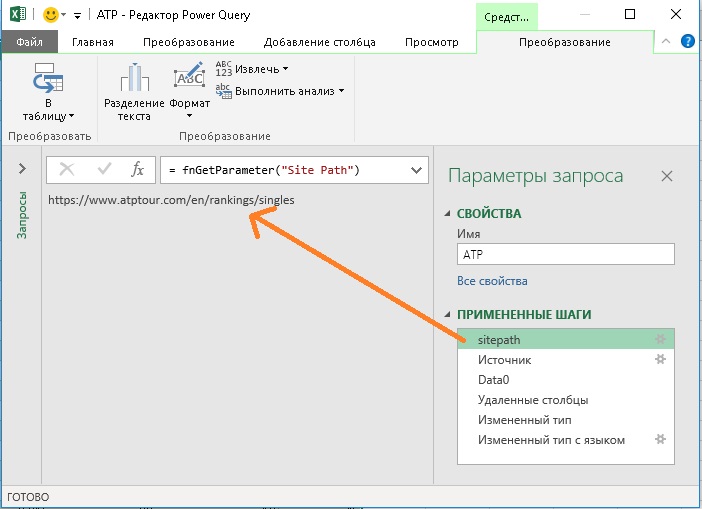


Рис. 14. Переменная *sitepath* правильно определяет название страницы сайта

В редакторе Power Query пройдите по меню *Главная* –> *Закрыть и загрузить*. Осталось проверить, как работает динамический запрос. В Excel перейдите на лист с таблицей *Parameters*, и в ячейке А1 выберите *doubles*. Перейдите на лист умной таблицы с рейтингом АТР, правой кнопкой щелкните на любой ее ячейке, и выберите Обновить. Таблица отразит данные рейтинга парных игроков:

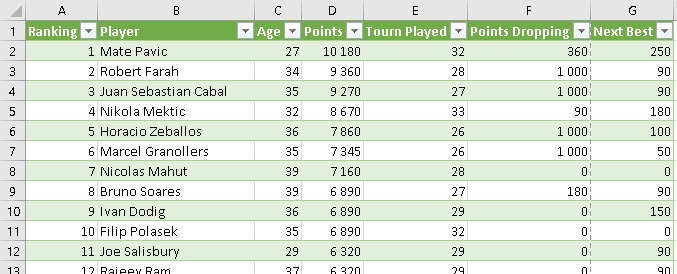


Рис. 15. Рейтинг парных игроков, извлеченный с сайта АТР динамическим запросом Power Query

#### Запрос с помощью VBA

Можно написать код VBA, который автоматизирует этот процесс. Обратите внимание, что строковые переменные могут быть жестко закодированы или извлечены из рабочего листа. Результирующая таблица создается, начиная с ячейки A5 активного листа.

Sub Query\_one()

Dim part1$, part2$

'part1 = "www.atptour.com/en/rankings/"

'part2 = "singles"

part1 = [b2] ' from cell B2

part2 = [b3] ' from cell B3

Application.CutCopyMode = False

With ActiveSheet.QueryTables.Add(Connection:= \_

"URL;https://" & part1 & part2, Destination:=Range("$A$5"))

.Name = "wash"

.FieldNames = True

.RowNumbers = False

.FillAdjacentFormulas = False

.PreserveFormatting = True

.RefreshOnFileOpen = False

.BackgroundQuery = True

.SavePassword = False

.SaveData = True

.AdjustColumnWidth = True

.RefreshPeriod = 0

.WebSelectionType = xlAllTables

.WebFormatting = xlWebFormattingNone

.WebPreFormattedTextToColumns = True

.WebSingleBlockTextImport = False

.WebDisableDateRecognition = False

.WebDisableRedirections = False

.Refresh BackgroundQuery:=False

End With

End Sub

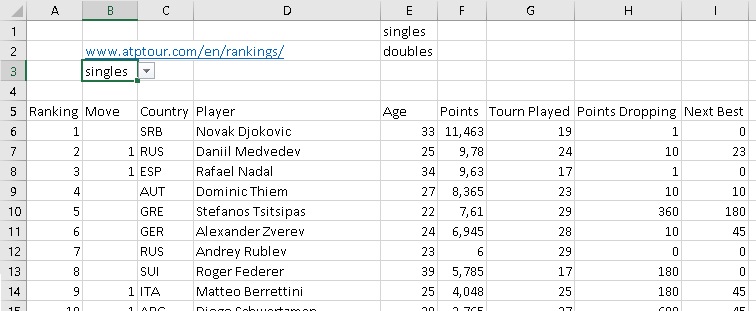


Рис. 16. Данные, полученные запросом на основе кода VBA

### Глава 2. Взаимодействие с сайтами без использования браузера

Когда мы говорим о получении данных из Интернета, некоторые люди думают об автоматизации браузера, например, Internet Explorer. На самом деле, в зависимости от конкретных потребностей пользователя, это довольно неэффективно. В этой главе будут представлены методы получения или публикации данных через Интернет. Говоря о данных, давайте сделаем шаг назад и обсудим JSON.

#### Что такое JSON?

JSON (JavaScript Object Notation) – формат для хранения и обмена информацией, доступной для чтения человеком, как и XML. Поскольку он очень популярен в наши дни, мы будем использовать примеры JSON, чтобы продемонстрировать некоторые из методов в этой главе. JSON в основном работает с двумя сущностями, а именно объектами и массивами:

{

"name": "This is my site",

"url": "https://bananas.com"

}

Имя и URL-адрес называются ключами с соответствующими значениями. Следующий синтаксис определяет массив:

["Nicole","Elaine","Samantha","Tom"]

Эти два основных элемента могут быть объединены, как показано ниже; значение ключа Files представляет собой массив.

{

"id": 18,

"personal": {

"name": "Jean Ferguson",

"Files": ["ftpA", "ftpB"]

}

}

Это почти все, что вам нужно знать, чтобы начать работать с файлами JSON.

#### Получение данных из Интернета с помощью кода VBA

Объекты запроса VBA XMLHTTP/ServerXMLHTTP предоставляют необходимые ресурсы для выполнения задач связи с веб-страницами и службами. Чтобы использовать их в своем коде, выполните следующие действия:

* Перейдите в редактор VBA
* Нажмите кнопку *Tools* в строке меню и выберите *References*.
* Отметьте строку *Microsoft XML v6.0*:

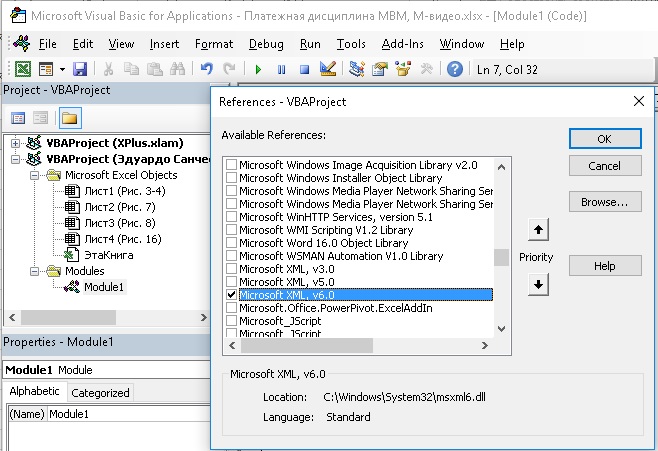


Рис. 17. Диалоговое окно *VBA* *References*

Когда Интернет начал расти в размерах, возникла необходимость в выделенных компьютерах для предоставления различных услуг (серверы), в то время как пользователи получают доступ к услугам (клиенты). Общие серверы – это веб-службы, электронная почта и FTP. Когда браузер открывает веб-страницу, он действует как клиент, а сайт, предлагающий информацию, размещен на сервере. Это называется архитектурой клиент-сервер.

В следующем примере мы получим данные JSON с [веб-сайта](http://jsonplaceholder.typicode.com/posts/1/comments), специально созданного для целей тестирования. Вот как выглядит этот сайт:

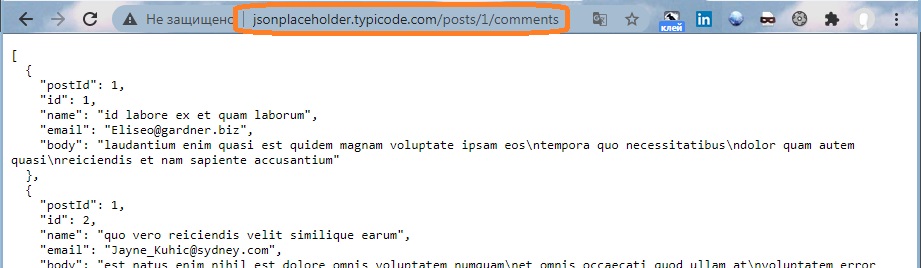


Рис. 18. Тестовый сайт для извлечения данных JSON

К сожалению, в Excel нет встроенной поддержки JSON. Следующий код просто поместит текст JSON в строковую переменную:

Public Sub Direct()

Dim http As Object, resp$

Set http = CreateObject("MSXML2.XMLHTTP")

http.Open "GET", "http://jsonplaceholder.typicode.com/" \_

& "posts/1/comments", False

http.send

' get the response

resp = http.responseText

MsgBox Left(resp, 350), 64, "The first 350 characters"

End Sub



Рис. 19. Данные JSON просто записаны в строковую переменную

Содержимое удалось извлечь, но работа с информацией, хранящейся в переменной, потребует неуклюжих манипуляций со строками.

Хорошей новостью является то, что в Интернете есть несколько бесплатных конвертеров JSON, которые обеспечат структурированный способ обработки данных. Один из них называется VBA-JSON. Чтобы воспользоваться им:

1. Перейдите на главную страницу <https://github.com/VBA-tools/VBA-JSON> и нажмите кнопку *Code*, а затем *Download ZIP*.
2. Извлеките папку *VBA-JSON-master* на свой ПК.
3. Откройте книгу Excel, в которой вы хотите использовать код, и перейдите в редактор VBA.
4. Щелкните правой кнопкой мыши любой модуль на левой панели и выберите опцию Import File...
5. В появившемся диалоговом окне найдите и выберите файл JsonConverter.bas из разархивированного пакета.
6. Нажмите кнопку *Открыть*, и в ваш проект VBA будет добавлен новый модуль. Исходный код не защищен, поэтому вы можете изучить его.
7. Добавьте ссылку на библиотеку *Microsoft Scripting Runtime*. Нажмите Ok. Вы можете получать ошибки, если эта ссылка будет иметь низкий приоритет в списке. Чтобы избежать этого, вернитесь в окно *Tools* –> *References* и нажмите стрелку вверх, чтобы увеличить его приоритет, как показано ниже:

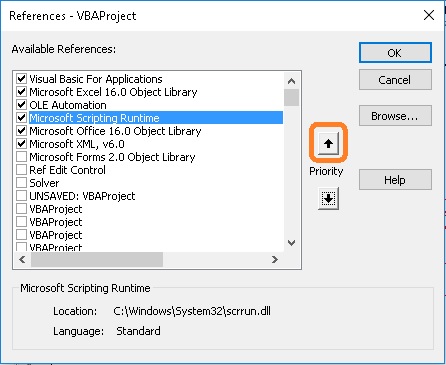


Рис. 20. Повышение приоритета ссылки на библиотеку *Microsoft Scripting Runtime*

Теперь можно запустить следующий код, чтобы перенести данные JSON на рабочий лист:

Public Sub Use\_Library()

Dim http As Object, json As Object, i%, item, \_

head, dest As Range

Set http = CreateObject("MSXML2.XMLHTTP")

' create array with keys

head = Array("id", "name", "postId", "email", "body")

Set dest = [a1].Resize(1, UBound(head) + 1)

dest.Value = head ' first row

http.Open "GET", "http://jsonplaceholder.typicode.com/" \_

& "posts/1/comments", False

http.send

Set json = ParseJson(http.responseText)

i = 2

For Each item In json

Sheets(2).Cells(i, 1) = item(head(0))

Sheets(2).Cells(i, 2) = item(head(1))

Sheets(2).Cells(i, 3) = item(head(2))

Sheets(2).Cells(i, 4) = item(head(3))

Sheets(2).Cells(i, 5) = item(head(4))

i = i + 1 ' next row

Next

End Sub

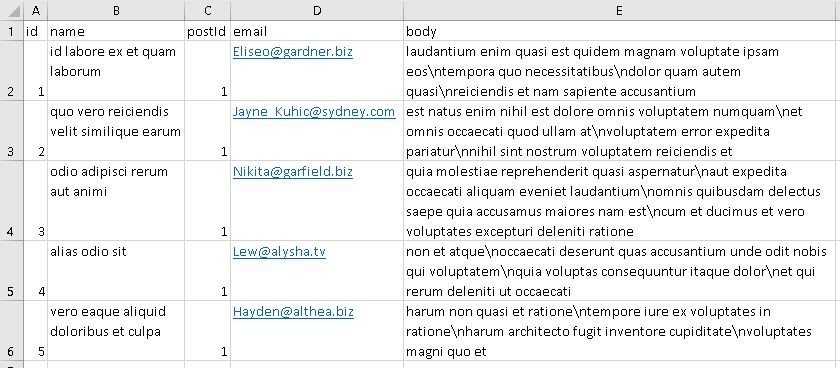


Рис. 21. Эта таблица Excel была создана на основе данных в формате JSON

#### Получение данных из элементов веб-страницы

Ранее я показал HTML-код с веб-страницы (см. рис. 1), которая содержала класс с именем "nav-link". Следующий код найдет все элементы, принадлежащие этому классу, и извлечет строку, представляющую атрибут *href*. Этот атрибут указывает адрес назначения для ссылки. Затем все адреса поместятся на рабочий лист Excel. Но начните с подключения библиотеки Microsoft HTML Object Library.

Обратите внимание, что, набрав *Н* в строке объявления переменных, редактор VBA предложит доступные варианты для продолжения набора. Эта функция известна как *Intellisense.* Она позволяет выбирать свойства и методы для переменных, а также служит для автоматического завершения набора кода (аналогичная функция есть и в Excel, облегчая набор формул).

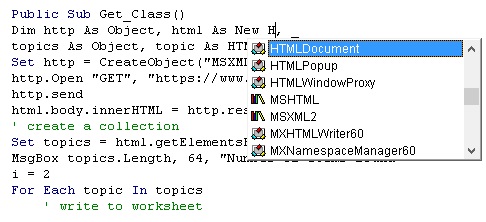


Рис. 22. Intellisense в действии

Public Sub Get\_Class()

Dim http As Object, html As New H, \_

topics As Object, topic As HTMLHtmlElement, i%

Set http = CreateObject("MSXML2.XMLHTTP")

http.Open "GET", "https://www.mrexcel.com/", False

http.send

html.body.innerHTML = http.responseText

' create a collection

Set topics = html.getElementsByClassName("nav-link")

MsgBox topics.Length, 64, "Number of items found"

i = 2

For Each topic In topics

' write to worksheet

Cells(i, 1) = topic.getAttribute("href")

i = i + 1

Next

End Sub

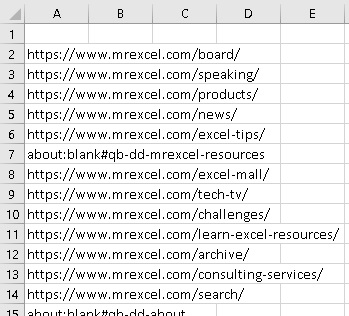


Рис. 23. Просмотр части очищенных данных

#### Размещение данных в Интернете

Как правило, мы используем код для получения данных с сайта, но бывают случаи, когда нам нужно отправить данные, например, чтобы сообщить приложению, сколько акций вы хотите купить. Приведенный ниже код отправляет данные JSON из ячейки D1 в точку тестирования.

Sub Post\_example()

Dim HTTP As Object, json$, URL$

' get the JSON string from worksheet

json = Range("d1")

' use server object

Set HTTP = CreateObject("MSXML2.ServerXMLHTTP")

URL = "https://jsonplaceholder.typicode.com/posts"

HTTP.Open "POST", URL, False

HTTP.setRequestHeader "Content-type", "application/json"

HTTP.send (json)

MsgBox HTTP.responseText, 64, "This is the result:"

Set HTTP = Nothing

End Sub

На рисунке показан текст ответа:

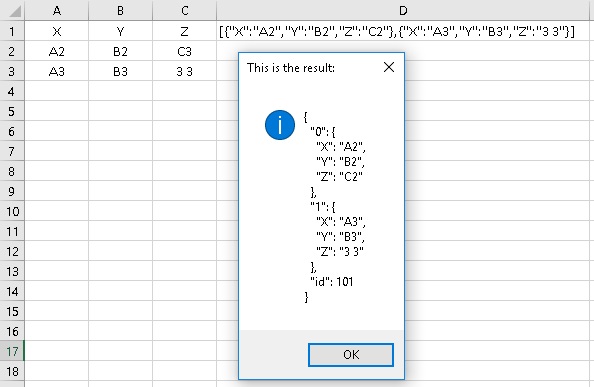


Рис. 24. В окне сообщения отображаются опубликованные данные

Некоторые важные моменты:

* Строка в ячейке D1 – JSON-представление диапазона A1:C3. Он начинается и заканчивается квадратными скобками; следовательно, это массив.
* В тексте ответа были созданы индексы "0", "1" и ключ идентификатора со значением 101.
* Серверный XML-объект, используемый в этом примере, может запрашивать или отправлять данные.

#### Аутентифицированные запросы

Предыдущие разделы показали, что вполне возможно получить веб-данные с общедоступных сайтов. Однако некоторые приложения потребуют, чтобы вы вошли в службу. Например, финансовые приложения. Очевидно, что без учетных данных пользователя приложение не будет знать, к какой учетной записи принадлежит запрос.

В приведенном ниже примере кода используется учетная запись на веб-сайте Alpaca. Этот вид учетной записи предоставляет возможность выполнять файловые операции в целях тестирования, но у них также есть живая часть, где вы можете рисковать реальными деньгами... Веб-API дает пользователю взаимодействовать с интернет-сервером, извлекая информацию из его базы данных или отправляя определенные данные с помощью почтовых запросов. Адреса назначения часто называются конечными точками, и ответы обычно приходят в форматах JSON или XML.

Sub authenticating()

Dim req As MSXML2.ServerXMLHTTP60, key\_id, \_

secretk$, secreth$, URL$, json As Object, item, s$

' create object

Set req = New MSXML2.ServerXMLHTTP60

key\_id = CStr([b18]) ' id from cell B18

secretk = CStr([b19]) ' secret key from cell B19

' get Microsoft information

URL = "https://paper-api.alpaca.markets/v1/assets/MSFT"

req.Open "GET", URL, False

req.setRequestHeader "APCA-API-KEY-ID", key\_id

req.setRequestHeader "APCA-API-SECRET-KEY", secretk

req.send

' start building string

s = "Raw data:" & vbLf & req.responseText

s = s & vbLf & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*" & vbLf & \_

"Available data:" & vbLf

' use the library

Set json = ParseJson(req.responseText)

For Each item In json ' loop the collection

s = s & item & vbLf

Next

MsgBox s, 64

End Sub

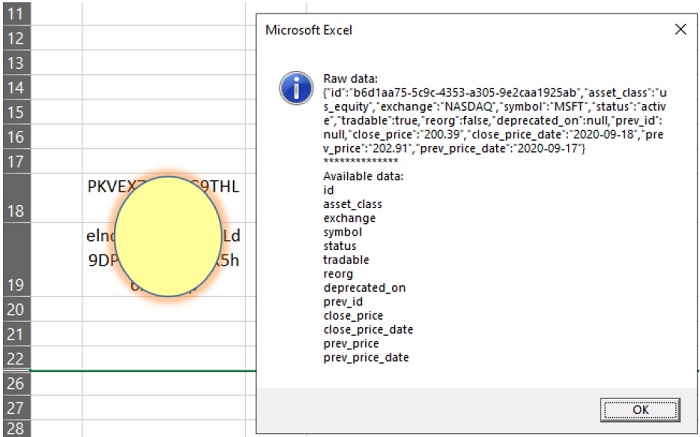


Рис. 25. Два ключа на листе и полученные данные

Чтобы получить нужные данные, вам необходимо сообщить идентификатор, а также секретный ключ, который может храниться в ячейках рабочего листа, как показано на рисунке выше. Данные извлекаются в формате JSON; окно сообщения показывает их. (Поскольку автор скрыл ключ и пароль на листе Excel, протестировать код не удалось.)

#### Глава 3. Internet Explorer и VBA

В настоящее время браузером по умолчанию для Microsoft является Edge, и мы обсудим его позже. Однако Internet Explorer – единственный браузер, у которого есть встроенная поддержка VBA. Поскольку он все еще существует, стоит потратить несколько страниц на его обсуждение.

Первое, что нужно сделать, это добавить ссылку в VBA на библиотеку *Microsoft Internet Controls*. Если вы используете Windows 10, для запуска IE попробуйте один из вариантов:

* Нажмите *Пуск* –> *Все приложения* и посмотрите в разделе *Стандартные – Windows*
* На панели задач кликните кнопку *Поиск в Windows* и введите *Internet Explorer*
* Нажмите клавиши Windows + R и введите iexplore.exe

Некоторые сайты, такие как YouTube, больше не будут открываться в Internet Explorer, поэтому рекомендуется вручную открыть сайт, который вы планируете автоматизировать с помощью IE.

#### Перенос веб-таблицы на рабочий лист

Рассмотрим [веб-сайт](http://rates.currencywholesale.com/rates/rates.html), показанный ниже; задача состоит в том, чтобы перенести таблицу на лист Excel вместе с флагами стран.

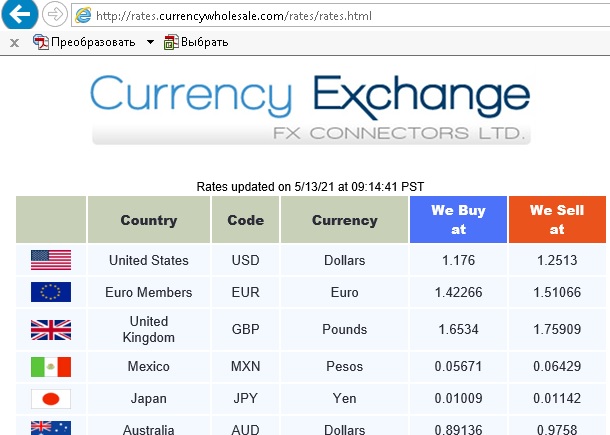


Рис. 26. Эта таблица будет скопирована в Excel

Глядя на HTML-код ниже, вы заметите теги tr, td и th. Тег <tr> определяет строку в таблице HTML. Элемент <tr> содержит один или несколько элементов <th> или <td>. Таблица HTML имеет два типа ячеек: ячейки заголовка, созданные с помощью элемента <th>, и ячейки данных, созданные с помощью элемента <td>. Также обратите внимание, что существуют элементы, определенные тегом img и содержащие атрибут src; они используются для хранения флагов стран.

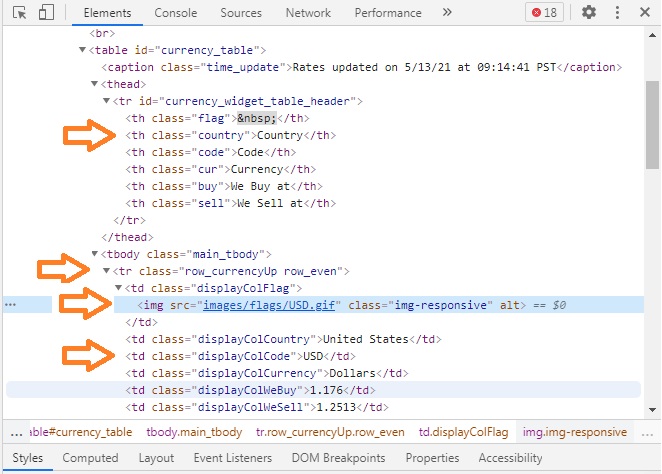


Рис. 27. HTML-код веб-страницы с рис. 26

Вот ключевые шаги приведенного ниже кода:

1. Назовите активный лист Excel *sheet1*, или измените его имя в коде.
2. Перейдите на веб-страницу.
3. Создайте коллекцию всех объектов с тегом img.
4. Создайте коллекцию всех таблиц страниц; нас будет интересовать вторая таблица этой группы. Поскольку коллекция начинается с индекса #0, мы ссылаемся на индекс #1.
5. Выполните цикл по строкам таблицы, перенося информацию о каждой стране в отдельные строки рабочего листа. Существует внутренний цикл, который повторяет столбцы таблицы.

При взаимодействии с веб-страницей мы, как правило, ищем определенные элементов в структуре HTML. Вот три самых популярных метода:

* GetElementbyID – находит один элемент, который имеет уникальный идентификатор. К сожалению, не все элементы страницы будут иметь такой идентификатор.
* GetElementsbyClassName – создает коллекцию элементов, принадлежащих определенному классу.
* GetElementsbyTagName – создает коллекцию элементов с определенным тегом.

Sub Web\_Table()

Dim tob As Object, ws As Worksheet, im As shape, rowc%, \_

colv%, objie As InternetExplorer, bod As HTMLBody, ic, s$

s = "https://rates.currencywholesale.com/rates/"

Set ws = ThisWorkbook.ActiveSheet

Set objie = New InternetExplorer

objie.Visible = True

objie.navigate s & "rates.html"

' wait for page loading

Application.Wait (Now + TimeValue("0:00:05"))

Set bod = objie.Document.body

Set ic = bod.getElementsByTagName("img")

' get all tables

Set tob = bod.getElementsByTagName("table")

On Error Resume Next ' if a flag is missing

For rowc = 0 To tob(1).Rows.Length - 1

Set im = ws.Shapes.AddPicture(s & \_

ic(rowc + 1).getAttribute("src"), msoFalse, msoTrue, \_

50, 50, 50, 50)

For colv = 0 To tob(1).Rows(rowc).Cells.Length - 1

ws.Cells(rowc + 1, colv + 1) = \_

tob(1).Rows(rowc).Cells(colv).innerText

With im

.LockAspectRatio = 0

.Top = ws.Cells(rowc + 2, 1).Top

.Left = ws.Cells(rowc + 2, 1).Left

.Width = ws.Cells(rowc + 2, 1).Width

.Height = ws.Cells(rowc + 2, 1).Height

End With

Next

Next

On Error GoTo 0

End Sub

У меня макрос работал около 15 минут.

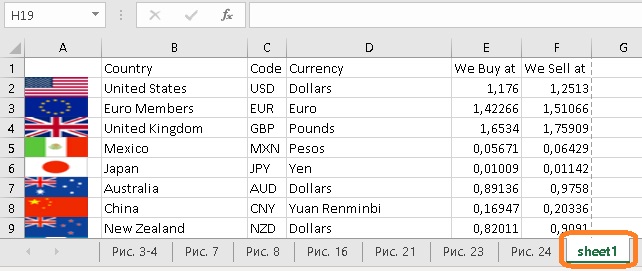


Рис. 28. Таблица в Excel, полученная макросом

#### Работа с событиями

Некоторые веб-сайты запрограммированы так, чтобы реагировать на события, генерируемые пользователем; это позволяет создавать более интерактивную и динамичную страницу. В следующем примере используется локальный HTML-файл, чтобы показать основы такого поведения.

В *Блокноте* создайте файл Code2.txt со следующим кодом:

<!DOCTYPE html>

<html>

<body style="background-color:powderblue;">

<h1>My Heading</h1>

<p><i>This is italic.</i></p>

<p style="color:red;">Red paragraph.</p>

<button>Click me</button>

<input name="q" onchange="alert('Alert message');"/input>

<bdo dir="rtl"><br><br>12345<br><br></bdo>

</body>

</html>

В *Проводнике* переименуйте расширение – \*.hmt. Чтобы убедиться в работоспособности, дважды кликните на файле, и он откроется в браузере по умолчанию. Обратите внимание, что определено событие *on-change*, то есть оно будет срабатывать при изменении значения поля ввода. В Excel выполните следующий код. Но перед этим отредактируйте путь к файлу Code2.hmt.

Sub IE\_event()

Dim iea As InternetExplorer, doc, search, \_

event\_onChange As Object

Set iea = CreateObject("InternetExplorer.Application")

iea.Visible = True

iea.navigate "C:\Users\public\Code2.htm"

Do Until Not iea.Busy

DoEvents

Loop

Application.Wait Now + TimeValue("0:00:03")

Set doc = iea.document

Set search = doc.getElementsByName("q")(0)

search.Value = "Pogacar"

'Exit Sub 'exiting here does not display the message

Set event\_onChange = iea.document.createEvent("HTMLEvents")

event\_onChange.initEvent "change", True, False

search.dispatchEvent event\_onChange

'iea.Quit close IE window

Set iea = Nothing

End Sub

Макрос открывает локальный файл с помощью Internet Explorer, определяет переменную поиска в качестве поля ввода и записывает в него. После этого код отправляет событие on-change, вызывая появление предупреждающего сообщения:

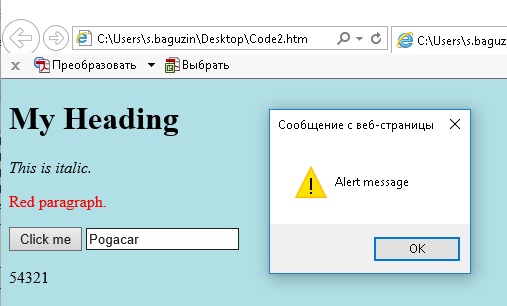


Рис. 29. Это сообщение страницы было вызвано кодом VBA

При открытии IE с помощью VBA может возникнуть ошибка, связанная с параметрами безопасности. Если вы не используете IE в качестве браузера, измените два параметра. Перейдите в меню *Свойства браузера* –> *Безопасность* и снимите флажок *Включить защищенный режим*. Далее на вкладке *Дополнительно* установите флажок *Разрешить запуск активного содержимого файлов на моем компьютере*. Если вы все еще используете IE для просмотра веб-страниц или загрузки файлов для локального использования, безопаснее восстановить эти параметры до значений по умолчанию после завершения задач VBA.

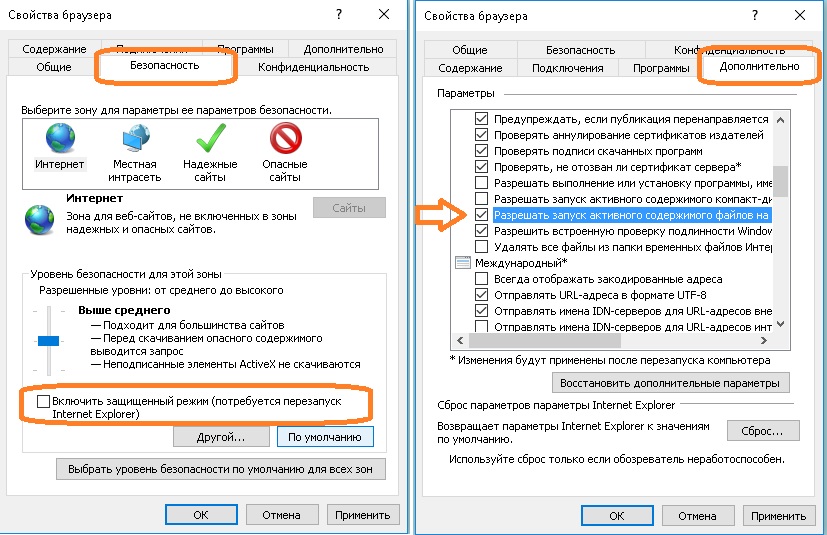


Рис. 30. Изменение параметров безопасности Internet Explorer

### Глава 4. Введение в Selenium

Selenium – это инструмент с открытым исходным кодом, используемый для автоматизации тестирования веб-приложений. Он может быстро выполнять десятки функциональных тестов и тестов на совместимость, взаимодействуя с бета-скриптами браузеров. При разработке сложного коммерческого программного обеспечения этап тестирования необходим для минимизации ошибок при развертывании продукта на нескольких платформах.

Флорент Бреер (Florent Breheret) разработал Selenium Basic, который представляет собой фреймворк, позволяющий напрямую использовать Selenium в Excel; так что это две разные вещи:

* Selenium – платформа автоматизации браузера и экосистема, первоначально созданная Джейсоном Хаггинсом.
* Selenium Basic (SB) – платформа автоматизации браузера на основе Selenium для VB.Net, VBA и VBScript.

Автор далее называет SB просто Selenium.

Selenium предоставляет интерфейс, который позволяет вашей электронной таблице автоматизировать различные браузеры (кроме Internet Explorer). Например, более стабильный Chrome.

#### Установка Selenium

Шаг. 1. Загрузите и установите исполняемый файл со страницы [GitHub](https://github.com/florentbr/SeleniumBasic/releases/tag/v2.0.9.0)

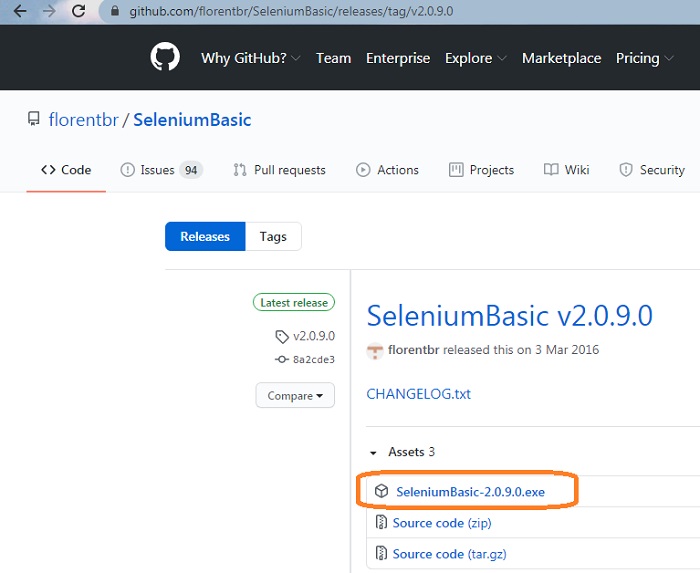


Рис. 31. Загрузите и установите исполняемый файл Selenium

Шаг. 2. В редакторе VBА добавьте ссылку на библиотеку Selenium Type Library:

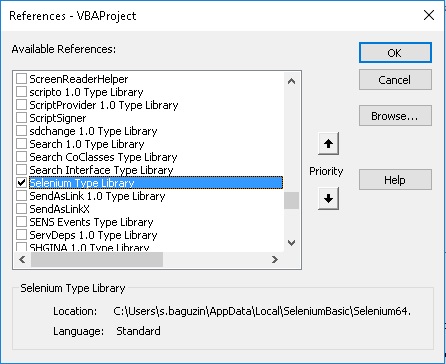


Рис. 32. Добавление ссылки на Selenium Type Library

Шаг. 3. Обновите драйвер Chrome, установленный на первом шаге. Дело в том, что пакет установки Selenium включает не самую свежую версию этого драйвера. Для этого перейдите на сайт <https://chromedriver.chromium.org/downloads> и скачайте драйвер, совместимый с вашей текущей версией Chrome.

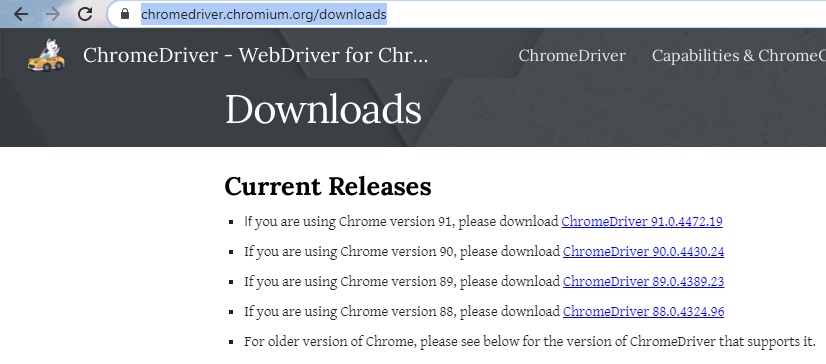


Рис. 33. Загрузите последнюю версию драйвера Chrome

Чтобы узнать, какая у вас версия, в Chrome пройдите по меню:

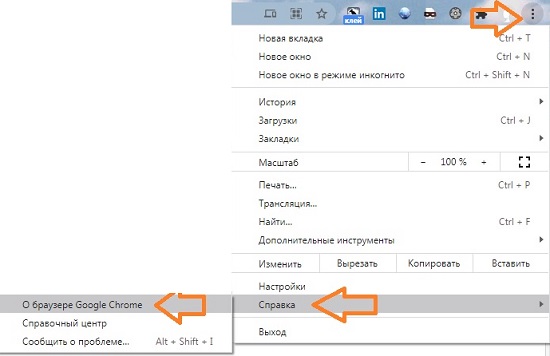


Рис. 34. О браузере Chrome

В открывшемся окне, вы увидите номер версии:

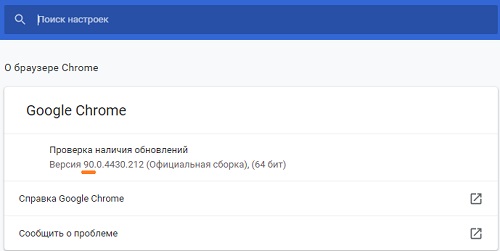


Рис. 35. Версия Chrome 90

Шаг. 4. Замените старую версию драйвера Chrome новой. Сделайте это в папке:

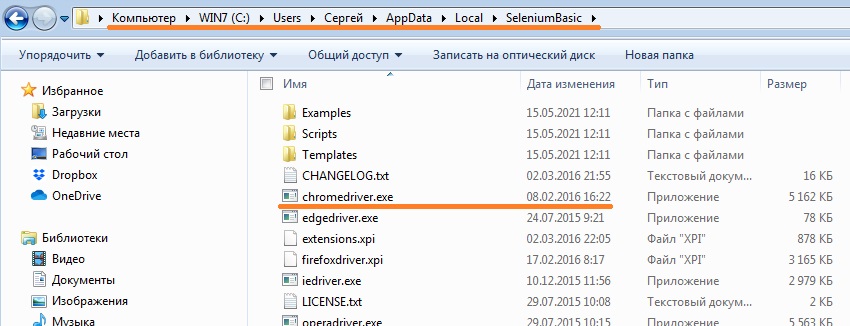


Рис. 36. Папка с компонентами Selenium

Chrome имеет быстрый цикл разработки, и через некоторое время используемый вами драйвер станет несовместимым с новой версией браузера. Когда это произойдет, вы получите ошибки, подобные тем, которые показаны ниже. Чтобы всё снова заработало, опять обновите драйвер.

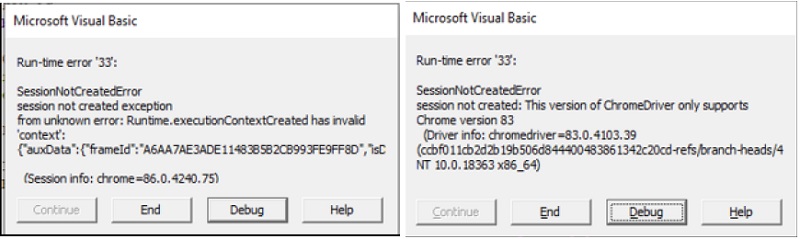


Рис. 37. Сообщения об ошибках, связанных с версией драйвера

#### Internet Explorer и Selenium

В большинстве случаев мы будем использовать Selenium для автоматизации современных браузеров, таких как Chrome или Edge. Тем не менее, вы можете взаимодействовать со старым браузером IE; пример ниже показывает, как это сделать.

Public bot As New IEDriver

Sub Test()

With bot

.get "https://www.mrexcel.com/"

.Window.Maximize

'.Quit

End With

End Sub

Просто обратитесь к установленному драйверу Internet Explorer, и все будет работать нормально.

### Глава 5. Google Chrome и VBA

Установив Selenium займемся автоматизацией браузера Chrome из кода VBA. Selenium имеет свое собственное дерево свойств и методов, но он будет работать аналогично Internet Explorer.

#### Запуск JavaScript на странице

Для первого примера создадим еще один локальный HTML-файл с исходным кодом, который очень похож на реальную веб-страницу. Он содержит несколько элементов, а также встроенный код JavaScript, завернутый в теги <script>. В Блокноте введите текст:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

</head>

<body>

<input id="B1" type="button" value="click to change textbox value"

onclick="TAlert()" />

<form class="OF\_form">

<ul class="OF\_t2">

<li class="OF\_3T OF\_active">FIRST</li>

<li class="OF\_3T">SECOND</li>

<li class="OF\_3T">STOP</li>

</ul>

<ul class="OF\_toggle">

<li class="OF\_ttab\_ OF\_buy OF\_active">THIRD</li>

<li class="OF\_ttab OF\_sell">FOURTH</li>

</ul>

<div class="MOr">

<div class="OF\_section">

<div class="OF\_section-h">Amount</div>

<div class="OF\_input-box\_Xk">

<input type="number" step="0.1" min="0" name="amount"

placeholder="0.00" value=""

autocomplete="off" oninput="OIHandler()">

<span>USD</span>

</div>

</div>

</div>

<div class="OF\_OT">

<div>

<b>Total</b>

<span>(corrected)</span>

<b>≈</b>

</div>

<div class="OF\_total" >0.00000</div>

</div>

</form>

<script language="javascript">

function OIHandler() {

print\_call\_stack(); alert('input detected...');}

function print\_call\_stack() { console.trace(); }

function TAlert() { setIB(document); }

function setIB() {

try {

var inputBox = document.querySelector('div.OF\_input-box\_Xk input');

inputBox.value = 50;

if (document.createEvent) { var ev2 = document.createEvent("HTMLEvents");

ev2.initEvent("input", true, true); ev2.eventName = "input";

inputBox.dispatchEvent(ev2); }

return ({ success: true }); }

catch (ex) {

return ({ exception: ex, myMsg: '#error in setIB!' });

}

}

</script>

</body>

</html>

Измените расширение файла на Code3.htm. В Проводнике дважды кликните на файле. Он откроется в Chrome:

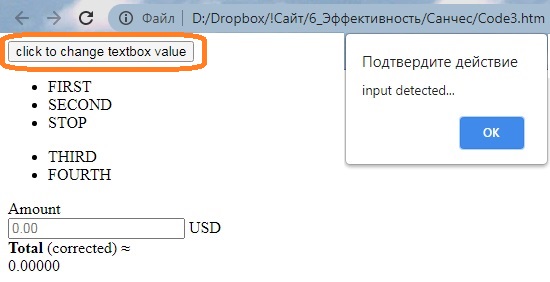


Рис. 38. Локальная страница с окном сообщения, вызванным событием ввода

Кликните на кнопку в левом верхнем углу. Появится окно для подтверждения действия. Нажмите Ok. Значение в поле Amount изменится. Щелчок по стрелкам вверх и вниз, появляющимся при наведении курсора мыши на поле Amount, изменяет его значение. Любые действия ввода приводят к появлению окна сообщения *Подтвердите действие*.

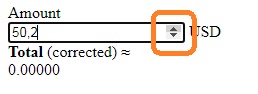


Рис. 39. Изменение значения ввода

Следующий VBA выполняет функцию настройки, содержащуюся в файле Code3.htm. Это приводит к запуску нового окна Chrome и открытию файла Code3.htm, а затем к его закрытию. Укажите вашу папку, где вы разместили файл Code3.htm. Если вы хотите, чтобы загруженная веб-страница оставалась открытой после завершения выполнения кода, объявите переменную веб-драйвера вне подпрограммы. Т.е., разместите первую строку кода в отдельном модуле. Если переменная объявлена внутри подпрограммы, память освобождается, когда код заканчивается и страница автоматически закрывается.

Public d As WebDriver

Public Sub Exec()

Dim url$

url = " D:\Dropbox\!Сайт\6\_Эффективность\Санчес\Code3.htm"

Set d = New ChromeDriver

With d

.Start "Chrome"

.get url

.Wait 1000

.ExecuteScript ("setIB()")

End With

End Sub

#### Iframes и перенос таблиц без цикла

Iframes – это встроенные элементы HTML, которые можно использовать для включения содержимого с других страниц в текущую страницу. Допустим, вам нужно программно ввести фрейм, чтобы получить доступ к его элементам. Например, цель состоит в том, чтобы перенести приведенную ниже веб-таблицу на рабочий лист Excel. Понаблюдайте за HTML-кодом, чтобы увидеть, что таблица находится внутри фрейма, который, в свою очередь, вложен в родительский фрейм. Следовательно, код должен ввести два фрейма, чтобы получить нужную таблицу.

Помните, как в главе 3 наш макрос мучительно перебирали строки и столбцы, чтобы получить таблицу с флагами? У Selenium есть классный метод, который скопирует таблицу за один проход!

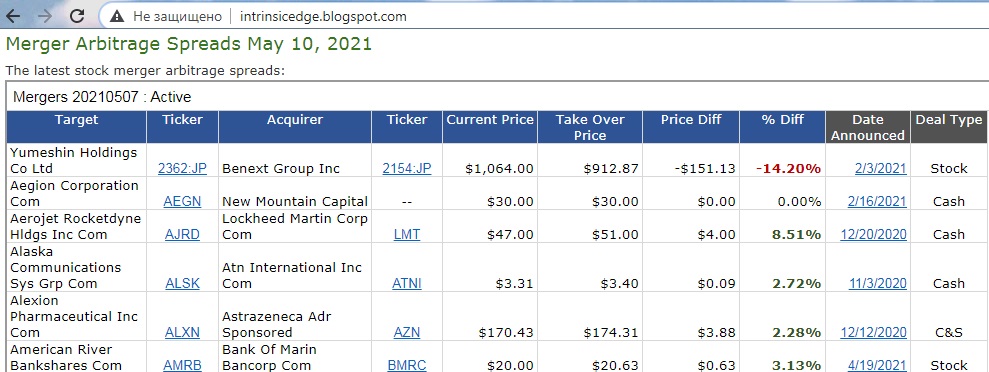


Рис. 40. [Сайт](http://intrinsicedge.blogspot.com/) с нужной таблицей

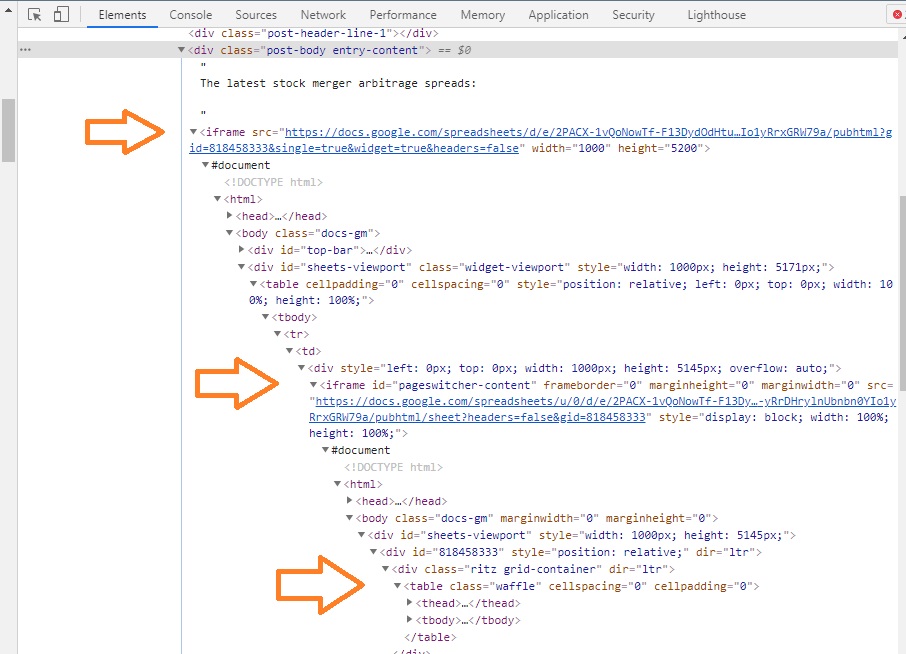


Рис. 41. Исходный html-код

Public d As WebDriver

Public Sub Table\_no\_loop()

Dim col As Object, i%, col2 As Object, s$, col3 As Object

Const url As String = "http://intrinsicedge.blogspot.com/"

Set d = New ChromeDriver

With d

.Start "Chrome"

.get url

.Wait 800

Set col = d.FindElementsByTag("iframe")

For i = 1 To col.Count

s = col.Item(i).Attribute("src")

If s Like "\*docs.google\*" Then Exit For

Next

.Wait 1000

.SwitchToFrame col.Item(3)

Set col2 = .FindElementsByTag("iframe")

.SwitchToFrame col2.Item(1)

Set col3 = .FindElementsByTag("table")

.Wait 500

End With

col3.Item(1).AsTable.ToExcel \_

ThisWorkbook.ActiveSheet.[a1]

End Sub

Код открывает Chrome, загружает страницу и создает коллекцию всех присутствующих Iframes. Затем он находит нужный, используя атрибут scr. После этого входит в первый фрейм, находит дочерний фрейм и переключается на него. Наконец, таблица идентифицируется и копируется с помощью одного оператора:

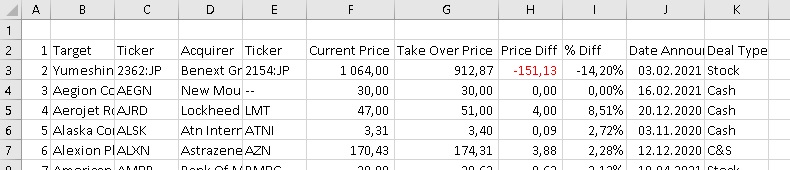


Рис. 42. Представление данных в Excel

Как и Internet Explorer, Selenium предлагает несколько методов поиска объектов страницы: FindElementbyID, FindElementsbyClass и многие другие. Используйте функцию Intellisense, чтобы просмотреть полный список:

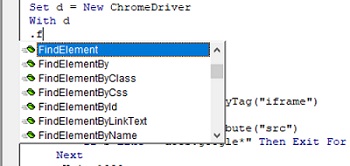


Рис. 43. Использование Intellisense для визуализации доступных опций

#### Использование XPath для поиска элементов

[XPath](https://ru.wikipedia.org/wiki/XPath#:~:text=XPath%20(XML%20Path%20Language)%20%E2%80%94,%D0%BE%D1%82%D0%BB%D0%B8%D1%87%D0%BD%D1%8B%D0%B9%20%D0%BE%D1%82%20%D0%BF%D1%80%D0%B8%D0%BD%D1%8F%D1%82%D0%BE%D0%B3%D0%BE%20%D0%B2%20XML.), XML Path Language – язык запросов к элементам XML-документа. Он реализован в Selenium и может использоваться для навигации по HTML-структуре веб-страницы, поиска любого элемента с помощью определенного синтаксиса. Существует две основные категории XPath:

* Абсолютный: это полный путь для элемента, начиная с корневого элемента. Недостатком этой категории является то, что она может быть довольно длинной и более подвержена сбою при изменении структуры HTML DOM (Document Object Model). Вы можете распознать абсолютный путь по начальной единственной прямой косой черте.
* Относительный: эта ссылка начинается где-то в середине структуры DOM. Относительный XPath начинается с двойной прямой косой черты и обычно является предпочтительным вариантом.

DOM – это API для документов HTML и XML. Он обеспечивает структурное представление документа, позволяя изменять его содержимое и визуальное представление с помощью языка сценариев, такого как JavaScript. Для наших целей думайте о DOM как о виртуальном представлении структуры веб-страницы в виде дерева со всеми ее объектами, стилями, содержимым и событиями. Вот шаги, чтобы получить XPath для элемента страницы:

* Щелкните правой кнопкой мыши интересующий объект страницы (1) и выберите *Просмотреть код* (2)
* Когда исходный код появится в правом окне, строка, относящаяся к интересующему объекту будет подсвечена. Щелкните на ней правой кнопкой мыши (3)
* Пройдите по меню Copy (4) –> Copy XPath (5)
* Информация о XPath скопирована в буфер обмена. Для нашего примера она выглядит так: //\*[@id="Blog1"]/div/div/div/div/div[1]/h3/a

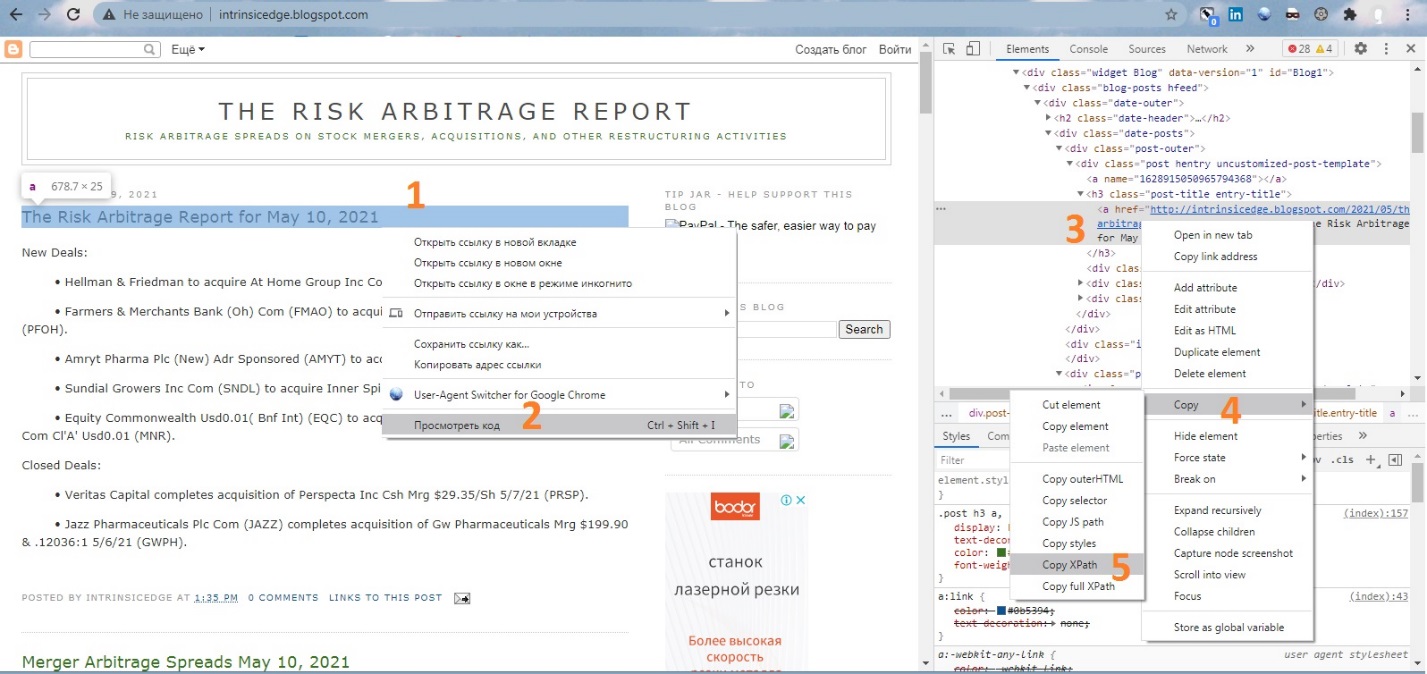


Рис. 44. Получение XPath для элемента страницы

Предложенный ниже код делает следующее:

* Открывает финансовый сайт и заполняет учетные данные для входа на него, беря их из ячеек рабочего листа В85 и В86.
* Получает дескриптор для кнопки всплывающего окна и нажимает на нее, чтобы закрыть всплывающее окно.
* Переходит на личную страницу автора на этом сайте; результат представлен на рис. 45.

Public driver As New ChromeDriver

Sub Filling\_Fields()

Dim pt As WebElement, ac As WebElement

driver.get "https://www.investing.com/portfolio/"

Application.Wait Now + TimeValue("0:00:04")

Set pt = driver.FindElementByXPath \_

("//\*[@id=""loginFormUser\_email""]").SendKeys \_

(CStr(Sheets("test").[b85]))

Set pt = driver.FindElementByXPath \_

("//\*[@id=""loginForm\_password""]").SendKeys \_

(CStr(Sheets("test").[b86]))

Set pt = driver.FindElementByXPath("//\*[@id=""signup""]/a")

DoEvents

' get popup button

Set ac = driver.FindElementById("onetrust-accept-btn-handler")

ac.Click

DoEvents

pt.Click

Application.Wait Now + TimeValue("0:00:03")

Set pt = driver.FindElementByXPath \_

("//\*[@id=""navMenu""]/ul/li[9]/a")

pt.Click: DoEvents

Application.Wait Now + TimeValue("0:00:02")

End Sub

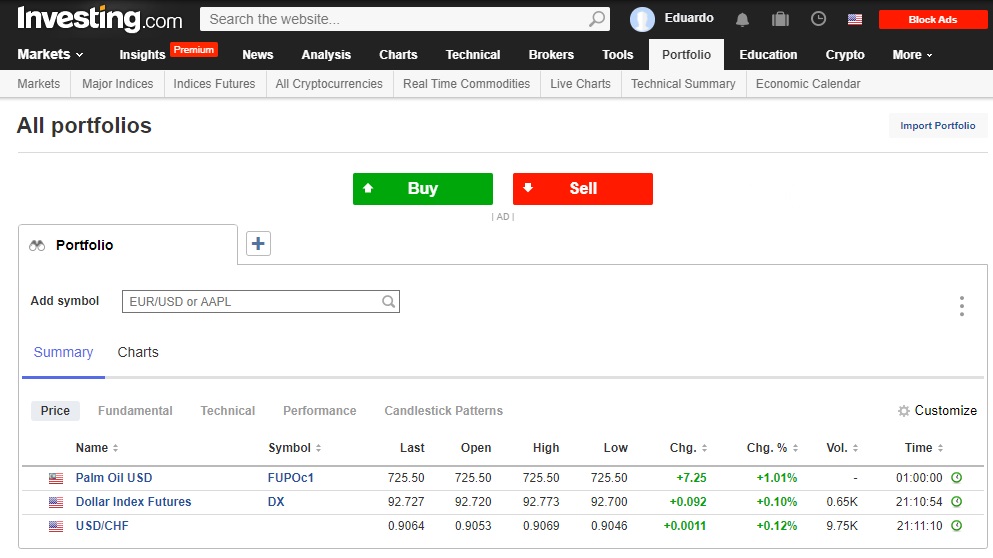


Рис. 45. Вход в систему с помощью кода

### Глава 6. Microsoft Edge и VBA

Microsoft разработала одиннадцать версий Internet Explorer с 1995 по 2013 год. Затем они объявили, что Internet Explorer 11 станет последним в этой линейке, и начали фокусироваться на Edge. Хорошей новостью для пользователей IE является то, что он является частью Windows 10; поэтому, если у вас есть какое-то приложение, которое использует определенные функции IE 11, нет причин для немедленного беспокойства. Следующая большая новость появилась в 2019 году, когда стало известно, что эта начальная версия Edge, часто называемая Edge HTML, была заменена новой версией Edge на основе Chromium. Т.е. они используют ту же реализацию с открытым исходным кодом, которая является основой для Google Chrome, Opera и других менее известных браузеров. Вы можете визуально распознать семейство Microsoft, как показано ниже:

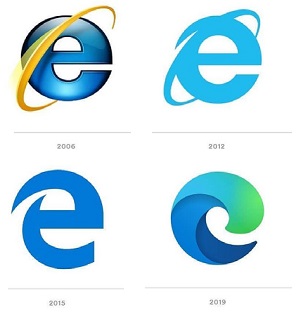


Рис. 46. Браузеры Microsoft на протяжении последних лет

Поскольку я не использую Edge, то для начала скачал версию 90 со страницы [Microsoft](https://www.microsoft.com/ru-ru/edge). Далее необходимо обновить драйвер Selenium. Проверьте версию Edge и тип операционной системы, и загрузите драйвер с этой [страницы](https://developer.microsoft.com/en-us/microsoft-edge/tools/webdriver/). Найдите папку с элементами Selenium – C:\Users\Сергей\AppData\Local\SeleniumBasic. Удалите файл edgedriver.exe. Поместите в эту папку загруженный драйвер и переименуйте его в edgedriver.exe. Мы готовы к первому примеру этой главы.

#### Поиск элементов с помощью селекторов CSS

Еще один способ найти элемент веб-страницы – это селектор CSS. CSS – это язык, используемый для применения стилей к странице.

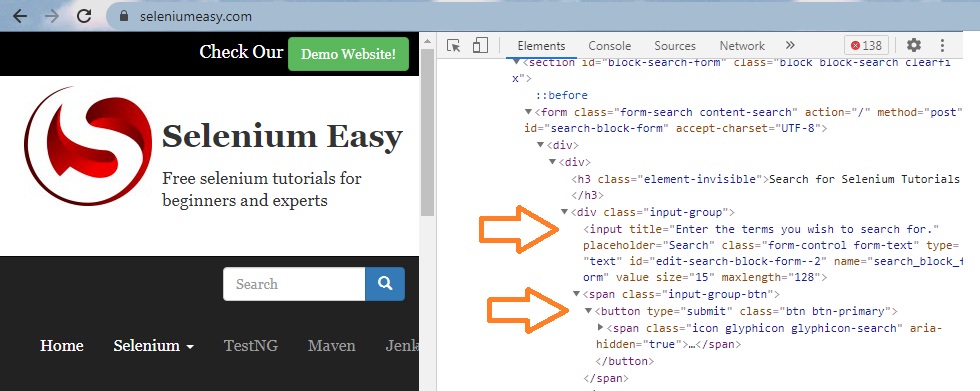


Рис. 47. Ссылка на два конкретных элемента веб-страницы

Допустим, вы хотите заполнить поле поиска на этом веб-сайте, и нажать кнопку поиска, чтобы получить результаты. Обратите внимание, что поле имеет входной тег, в то время как кнопка имеет тег кнопки. Поскольку поле ввода имеет уникальный идентификатор, на него можно ссылаться напрямую. Кнопка, с другой стороны, не имеет идентификатора, поэтому она идентифицируется по пути DOM, начиная с элемента формы. Когда код завершает выполнение, на вкладке Edge отображаются результаты поиска по выбранному объекту.

Чтобы узнать, что такое CSS-селектор для конкретного элемента, выполните те же действия, описанные в главе Chrome, чтобы получить XPath, но вместо этого выберите опцию Copy selector. Для кнопки это: #search-block-form > div > div > div.input-group > span > button.

Public driver As New EdgeDriver

Private Sub Use\_Edge()

Dim e As WebElement

driver.get "https://www.seleniumeasy.com/"

driver.Wait 3000

Set e = driver.FindElementByCss("#edit-search-block-form--2")

If e.IsDisplayed Then

e.SendKeys ("css") ' write search string

Set e = driver.FindElementByCss \_

("#search-block-form > div > div >" & \_

" div.input-group > span > button")

driver.Wait 1000

e.Click

End If

'driver.Quit

End Sub

У меня этот код не заработал((

#### Загрузка файла

Обычная задача при работе в Интернете – загрузить файл. Для этого существуют эффективные методы, такие как клиенты FTP (протокол передачи файлов) или P2P (одноранговые) системы. Однако вы можете столкнуться с веб-сайтом, где единственным способом загрузки является щелчок по значку. В следующем примере показано, как автоматизировать этот процесс.

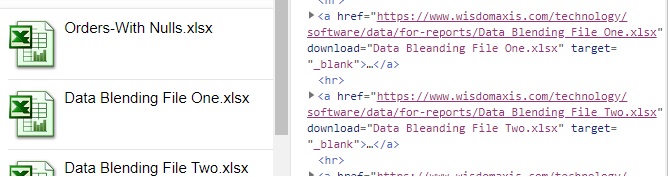


Рис. 48. Определение адресов файлов

На рисунке показана часть исходного кода HTML страницы. Каждый файл Excel имеет свой адрес, содержащийся в атрибуте href, поэтому все, что нам нужно сделать, это использовать эту информацию для загрузки.

Public ed As New EdgeDriver

Public Sub Dload()

Dim col, i%, s$, p$, v

ed.get "https://www.wisdomaxis.com/technology/" & \_

"software/data/for-reports/"

ed.Wait 2000

Set col = ed.FindElementsByTag("a")

For i = 1 To col.Count

s = col.item(i).Attribute("href")

If s Like "\*Refresh\*" Then Exit For ' desired file

Next

ed.SetProfile "Download", persistant:=True

ed.SetPreference "browser.download.folderList", 2

ed.SetPreference "browser.helperApps.neverAsk.saveToDisk", \_

"application/vnd.ms-excel, application/msword," & \_

" application/msexcel," & \_

" application/xls, application/csv, text/csv," & \_

" application/pdf, text/html, text/plain"

ed.SetPreference "browser.download.manager.showWhenStarting", 0

ed.SetPreference "browser.download.manager.focusWhenStarting", 0

ed.SetPreference "browser.download.useDownloadDir", True

ed.SetPreference "browser.helperApps.alwaysAsk.force", False

ed.SetPreference "browser.download.manager.alertOnEXEOpen", False

ed.SetPreference "browser.download.manager.closeWhenDone", True

ed.SetPreference "browser.download.manager.showAlertOnComplete", 1

ed.SetPreference "browser.download.manager.useWindow", False

ed.SetPreference "pdfjs.disabled", True

ed.get s ' download file

p = Split(Environ$(41), "=")(1) & "\downloads\" ' build path

v = Split(s, "/")

Workbooks.Open p & Replace(v(UBound(v)), "%20", " ")

'ed.Window.Maximize

Workbooks(Workbooks.Count).Activate

End Sub

Этот код:

* Создает коллекцию элементов с тегом <a>
* Организует цикл, чтобы найти нужный файл
* Устанавливает несколько настроек загрузки
* Загружает и открывает книгу

Если вы хотите открыть Microsoft Edge без установки Selenium, следующий небольшой код позволит вам это сделать. Обратите внимание, что нет необходимости знать путь установки Edge.

Sub ShellEdge()

VBA.Shell Environ$(“comspec”) & " /c start microsoft-edge:" & \_

"http://www.mrexcel.com"

End Sub

#### Создание PDF-файла с нуля

Допустим, вы хотите каталогизировать несколько веб-страниц в формате PDF для последующего анализа. В следующем примере показано, как создать файл.

* Код использует массив для хранения поисковых запросов, которые будут переданы в Google. В качестве альтернативы можно получать информацию из диапазона ячеек Excel.
* Код использует атрибут заголовка поля ввода; эта строка будет отличаться в зависимости от вашего языка. На рисунке ниже показано, как это выглядит для португальской версии автора.
* Макрос использует цикл создавая новую вкладку Google для каждого элемента. После выполнения поиска будет сделан снимок экрана и добавлен в PDF-файл. В конце файл сохраняется на диск.

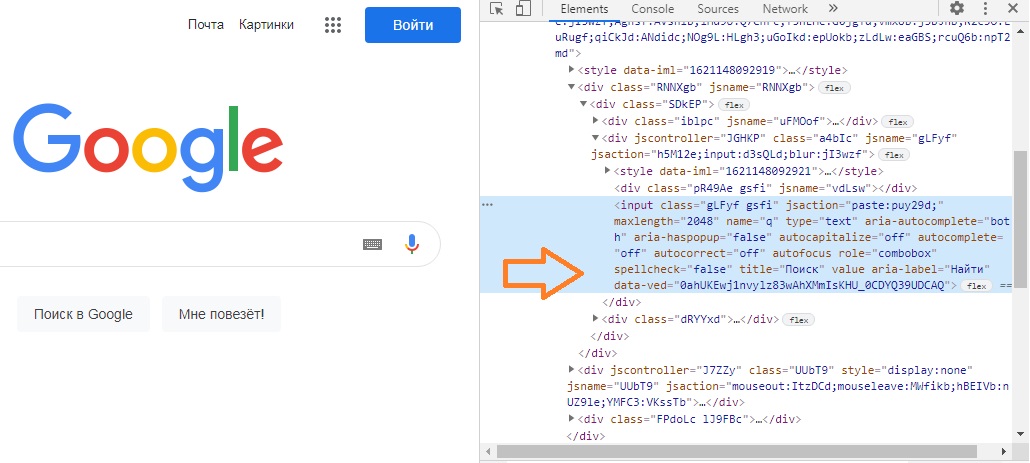


Рис. 49. Код HTML для поля поиска Google, русская версия

Public Sub CreatePDF()

Dim ed As EdgeDriver, arr(), i%, pdf As Object

Set ed = New EdgeDriver

Set pdf = CreateObject("Selenium.PdfFile")

pdf.SetPageSize 210, 300, "mm"

pdf.SetMargins 5, 5, 5, 15, "mm"

arr = Array("burj khalifa", "petronas towers")

With ed

.Start "edge"

.get "https://google.com/"

For i = LBound(arr) To UBound(arr)

If Not IsEmpty(arr(i)) Then

If i > 0 Then

.ExecuteScript "window.open(arguments[0])", \_

"https://google.com/"

.SwitchToNextWindow

End If

.FindElementByCss("[title=Pesquisar]").SendKeys \_

arr(i) & " images"

.SendKeys .keys.Enter

pdf.AddImage .TakeScreenshot, True

End If

Next

End With

pdf.SaveAs "c:\temp\selenium\arch.pdf"

'Stop

End Sub

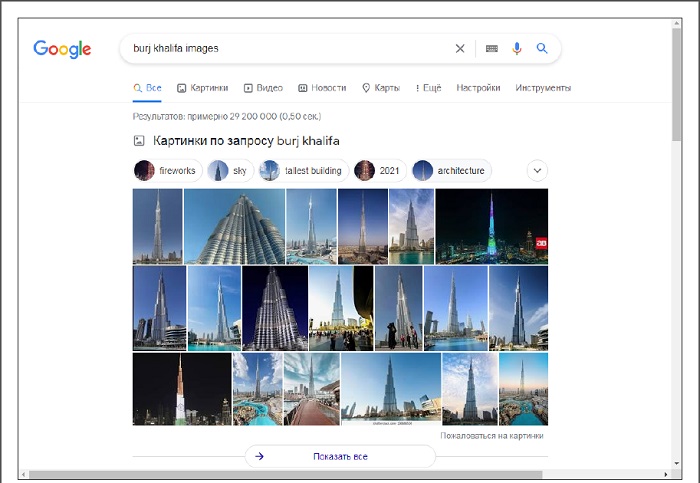


Рис. 50. Этот PDF-файл был создан макросом из Excel

#### Оболочка Selenium

Как вы, возможно, уже заметили, не имеет значения, какой браузер мы используем с Selenium, поскольку его дерево свойств и методов одно и то же. Последний фрагмент кода VBA иллюстрирует это; все три браузера открываются с одинаковыми операторами. Выбранный сайт – это еще один сайт, на котором Internet Explorer больше не может работать хорошо...

Public ed As New EdgeDriver

Public cr As New ChromeDriver

Public ie As New IEDriver

Sub Choice()

Dim url$

url = "http://www.startrek.com"

Select Case InputBox("1 - Edge / 2 - Chrome / 3 - IE", \_

"Select a browser:")

Case 1: ed.get url

Case 2: cr.get url

Case 3: ie.get url ' does not load completely

End Select

End Sub

### Глава 7. Power Query и Интернет

Мы живем в эпоху бизнес-аналитики, и Power Query является частью пакета инструментов BI от Microsoft. Он позволит извлекать данные из различных источников, включая Интернет, изменять их и переносить готовый продукт на рабочий лист Excel. Для Excel 2010 и 2013 Power Query был надстройкой; начиная с Excel 2016 и далее, это встроенный ресурс. Примеры в этой главе будут основаны на Excel 365.

#### Простое подключение к веб-таблице

В первом примере мы увидим простую процедуру создания соединения с данными, хранящимися на веб-странице.

Шаг 1. В Excel пройдите по меню *Данные* –> *Из Интернета:*

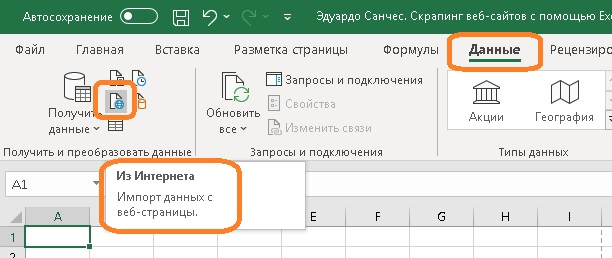


Рис. 51. Запуск Power Query

Шаг 2. Вставьте адрес веб-страницы в диалоговое окно и нажмите Ok. Автор использовал страницу [Вики](https://en.wikipedia.org/wiki/Athletics_at_the_2013_Mediterranean_Games_%E2%80%93_Results) с результатами соревнований

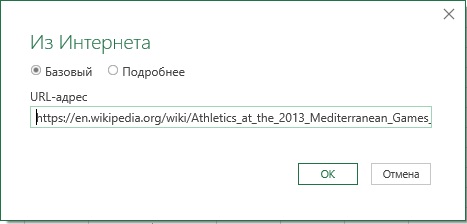


Рис. 52. Запрос в странице <https://en.wikipedia.org/wiki/Athletics_at_the_2013_Mediterranean_Games_%E2%80%93_Results>

Если вы первый раз обращаетесь к Википедии, появится запрос об уровне доступа:

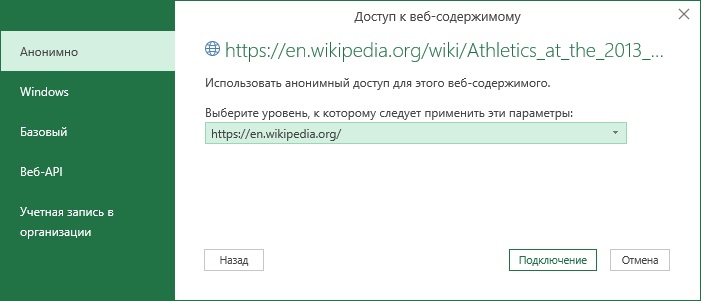


Рис. 53. Уровень доступа

Оставьте установку по умолчанию – *Анонимно*. Нажмите *Подключение*. При последующих обращениях к Википедии это окно не будет появляться.

Шаг 3. В окне *Навигатора* отобразятся все доступные таблицы. Я выбрал таблицу *5000 metres*. В правой части окна отобразилось содержимое таблицы. Видно, что это результаты женского финала в беге на 5000 м. Нажмите *Преобразовать данные*:

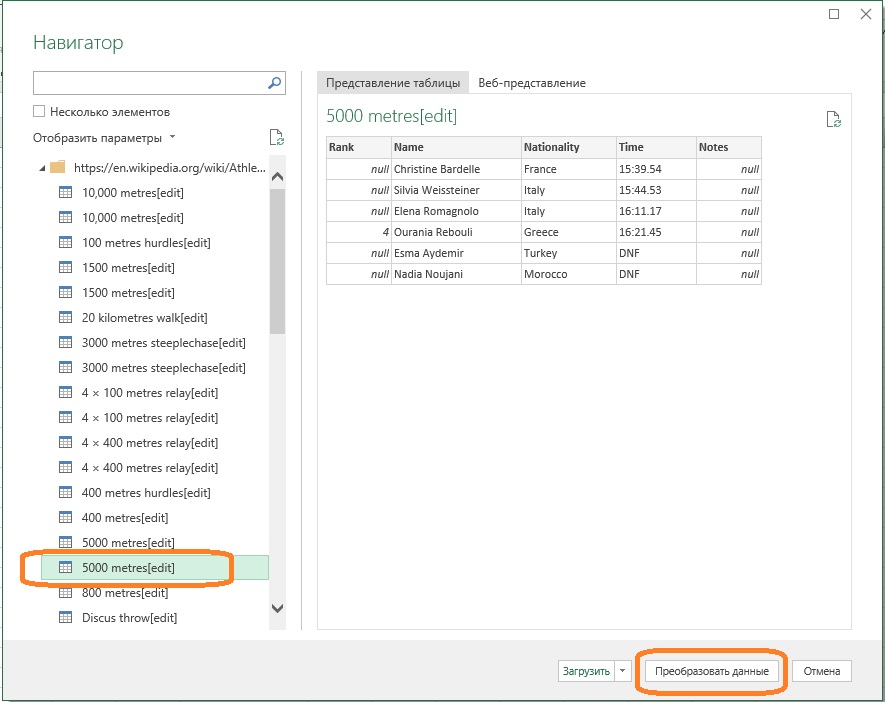


Рис. 54. Окно *Навигатора*

Откроется окно Редактор Power Query:



Рис. 55. Окно редактора Power Query

Шаг 4. Важно понимать, что, хотя у вас есть лента, строка формул, строки и столбцы с данными и язык программирования (называемый M), Power Query – это не Excel. Это два разных приложения, но они очень тесно взаимосвязаны друг с другом.

Как вы видите на рис. 55, данные уже есть, но они требуют обработки. Удалите столбцы *Ранг* и *Примечания*. Для этого выделите столбец, который вы хотите удалить, и на вкладке *Главная* нажмите кнопку *Удалить столбцы*. Или кликните на столбец правой кнопкой мыши, и выберите *Удалить*. Можно также удалить столбцы, которые не выбраны. Для выделения нескольких столбцов выделяйте их при нажатой клавише Ctrl. Чтобы выделить диапазон столбцов, кликните на первом, нажмите Shift, и кликните на последнем.

Шаг 5. Добавьте столбец, чтобы отразить место спортсмена. Для этого на вкладке *Добавить столбец* выберите *Столбец индекса* > *От 1*. После вставки перетащите его по строке заголовка влево. Таблица готова для экспорта в Excel.

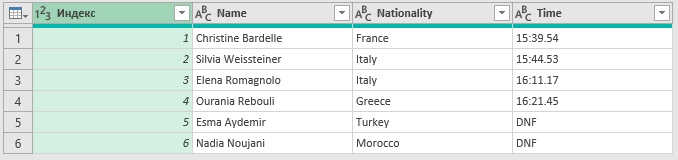


Рис. 56. Таблица обработана в Power Query

Шаг 6. Выберите *Файл* > *Закрыть и загрузить в…*, чтобы определить, куда следует поместить данные:

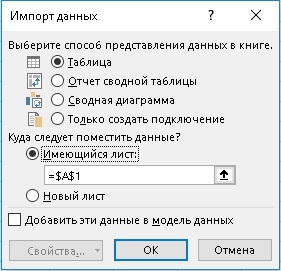


Рис. 57. Опции загрузки запроса в книгу Excel

Нажмите *Ok*. Запрос будет помещен в умную таблицу на текущий лист, начиная с ячейки А1:

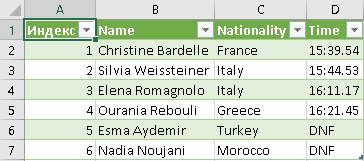


Рис. 58. Лист Excel с данными из запроса

Запрос построен и может быть обновлен в любое время; описанные выше шаги повторять не нужно.

#### Обходной путь подключения к веб-таблице

Power Query – развивающийся продукт, но пока он не может автоматически обнаруживать все таблицы на веб-странице. Иногда мы вообще не получаем таблиц при открытии диалогового окна *Навигатор*. Однако есть обходной путь, предложенный Microsoft MVP Gil Raviv. Мы можем вручную находить и импортировать данные.

В следующем примере мы пройдем по HTML-структуре страницы, пока не найдем нужную таблицу. На самом деле, Power Query находит эту таблицу в автоматическом режиме, но мы сделаем вид, что не находит, чтобы потренироваться.

Шаг 1. Начините, как в предыдущем примере: *Данные* –> *Из Интернета*. Укажите страницу подключения: <https://climatedata.eu/climate.php?loc=szxx0038&lang=en>

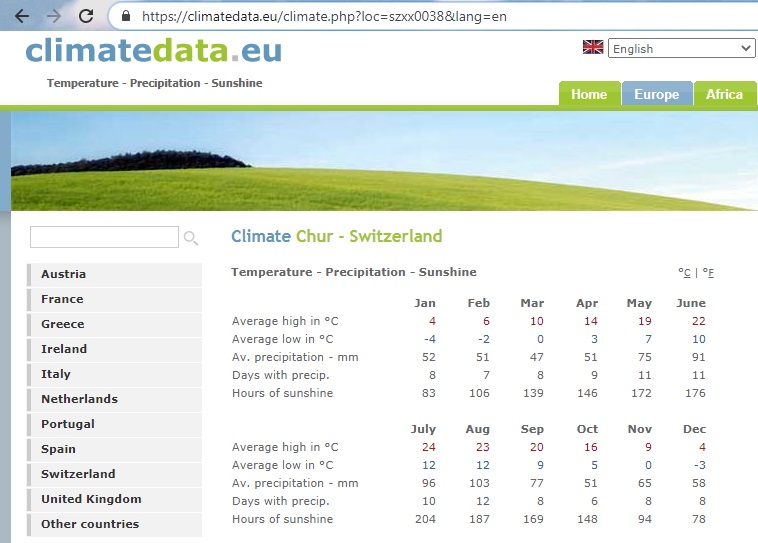


Рис. 59. Сайт для запроса таблицы

Оставьте уровень доступа *Анонимно*, выберите для подключения *Document*. Нажмите *Преобразовать данные*, чтобы загрузить Редактор Power Query. Идея состоит в том, чтобы кликать правильные узлы в столбце *Children* для перемещения по дереву HTML к нужным данным. Вот как выглядит HTML-код. Здесь также указаны некоторые из шагов, упомянутых ниже. Целевая таблица подсвечивается слева.

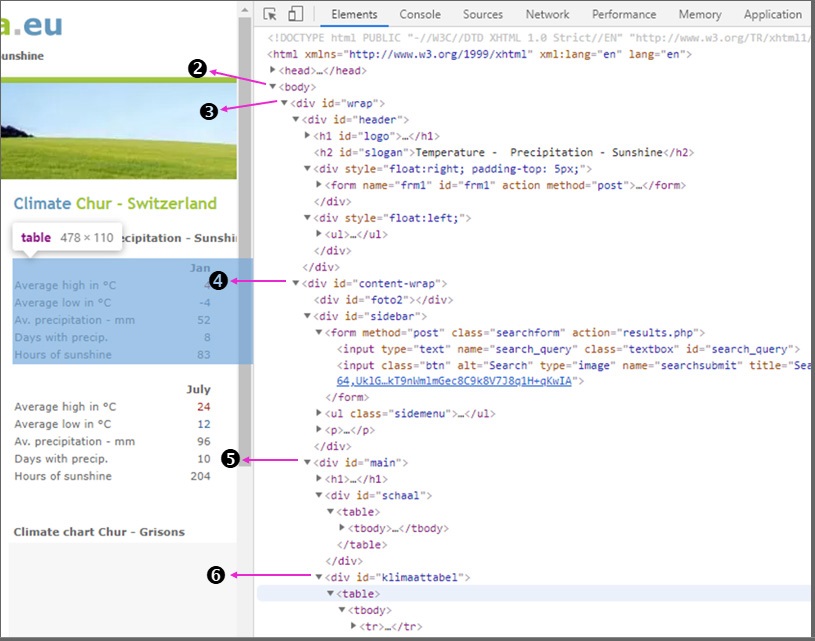


Рис. 60. HTML-код страницы

Выполняя следующие шаги, сравнивайте их с приведенным выше HTML-кодом, чтобы лучше понять процедуру:

1. Щелкните ссылку на таблицу *Table* в единственной доступной строке.
2. Щелкните *Table* во второй строке (элемент BODY).
3. Щелкните *Table* в первой строке (элемент DIV).
4. Щелкните *Table* во второй строке (элемент DIV).
5. Щелкните *Table* в третьей строке (элемент DIV).
6. Щелкните *Table* в третьей строке (элемент DIV).
7. Щелкните *Table* в первой строке (элемент TABLE), Это уже наша таблица!
8. Щелкните *Table* на единственной доступной строке, которая является элементом TBODY.
9. На данный момент у нас уже есть шесть строк таблицы; теперь разверните столбец *Children*, щелкнув две маленькие стрелки, обращенные друг к другу в заголовке таблицы. В диалоговом окне снимите флажки с других столбцов, так как нам не нужно их разворачивать (рис. 63). Снимите флажок *Использовать исходное имя столбца, как префикс*.

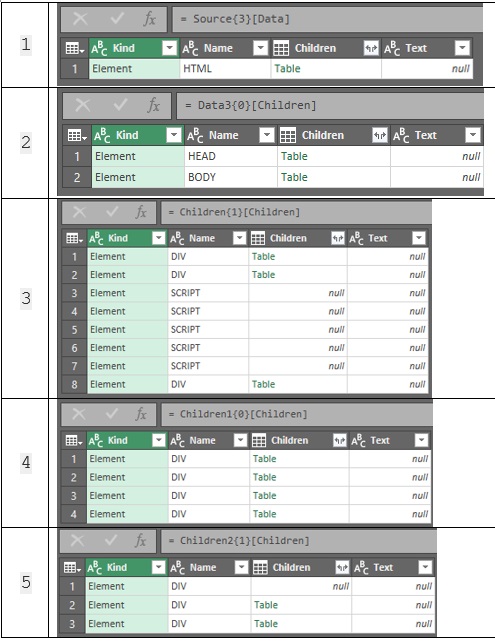


Рис. 61. Шаги с первого по пятый

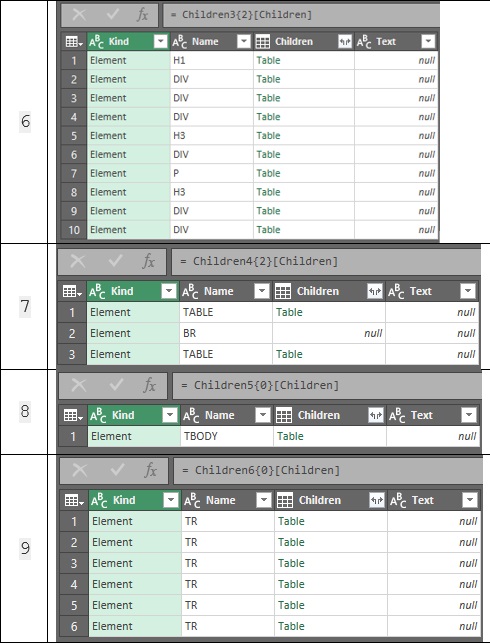


Рис. 62. Шаги с шестого по девятый

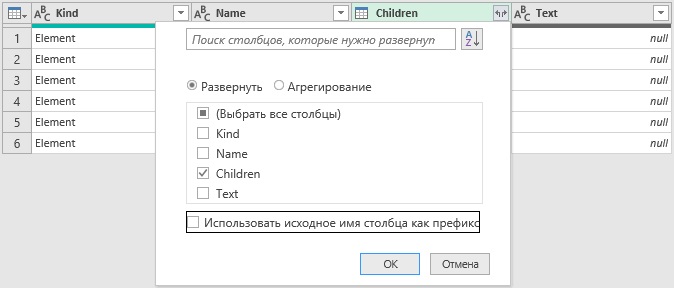


Рис. 63. Настройки окна *Развернуть*

Power Query показывает 42 одинаковые строки:

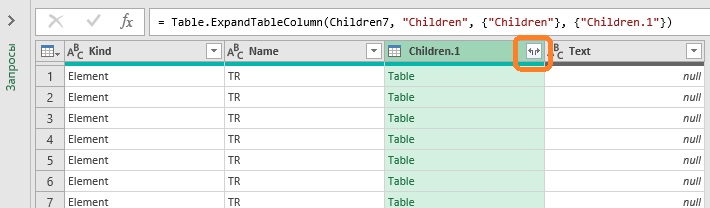


Рис. 64. Очередной шаг обработки

Разверните столбец Children1. Для раскрытия укажите только поле Text:

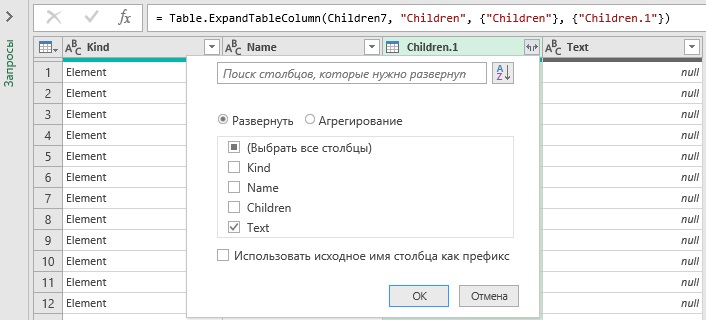


Рис. 65. Настройки второго окна *Развернуть*

Наши данные отобразятся, правда все в одном столбце:



Рис. 66. Столбец Text после команды *Развернуть*

Удалить все столбцы, кроме столбца Text.1. На вкладке *Добавить столбец* выберите *Столбец индекса* –> *От 0*:

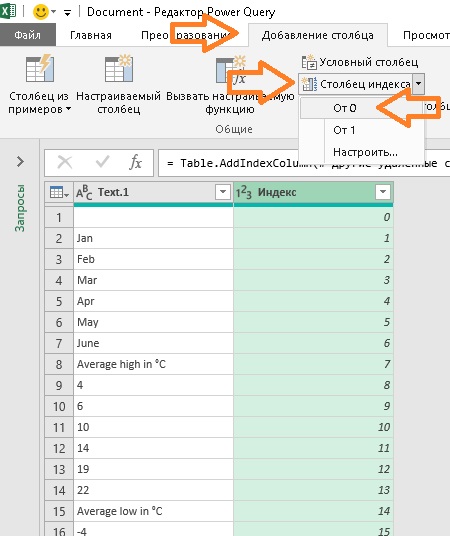


Рис. 67. Добавлен столбец индекса

На вкладке *Преобразование* выберите *Стандартный* –> *Остаток от деления*. Когда появится диалоговое окно, введите число 7, которое является числом строк в исходной таблице:

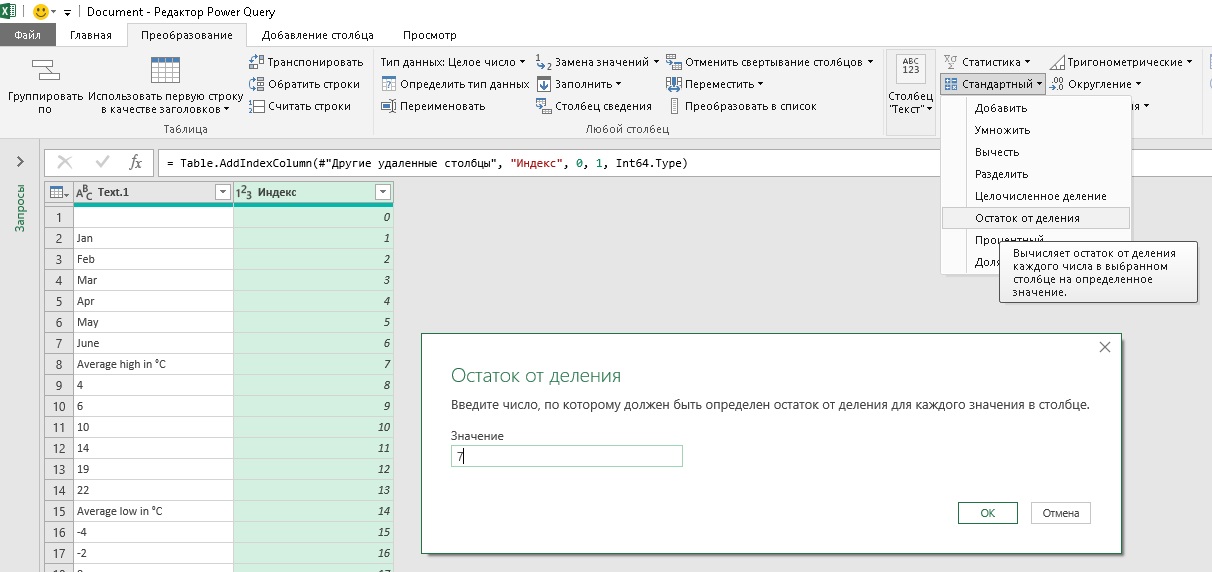


Рис. 68. Преобразование индекса в остаток от деления на 7

На вкладке *Добавить столбец* выберите *Столбец индекса* –> *От 0*. Добавится второй столбец индекса. Снова перейдите на вкладку *Преобразование*, но теперь выберите *Стандартный* –> *Целочисленное значение*. Снова введите число 7:

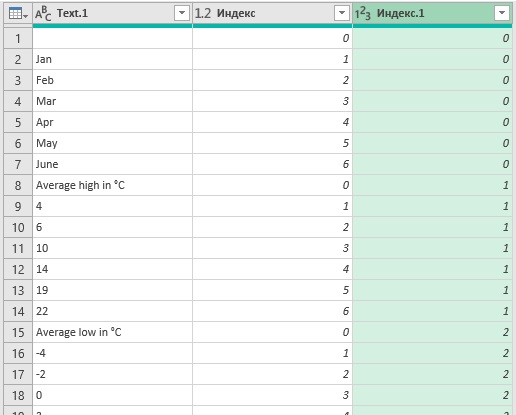


Рис. 69. Второй индекс показывает целое от деления на 7

Выберите столбец *Индекс* и на вкладке *Преобразование* щелкните *Столбец сведения*. Когда появится диалоговое окно, выберите следующие параметры:

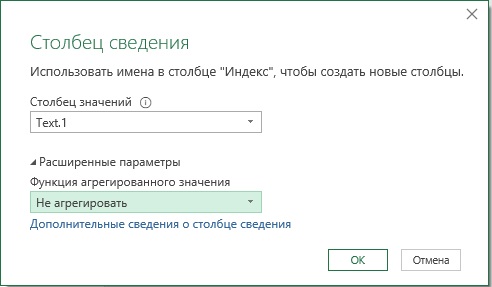


Рис. 70. Параметры сведения

Нажмите кнопку Оk. Далее удалите столбец *Индекс.1*. Пройдите по меню *Главная* –> *Использовать первую строку в качестве заголовков*. Присвойте первому столбцу имя *Параметр*. Ваш запрос принял нужный формат, и его можно загрузить!

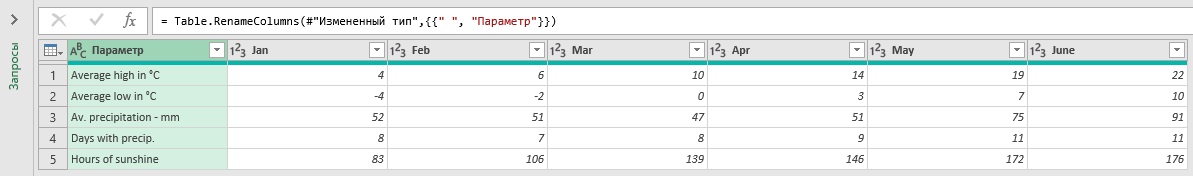


Рис. 71. Обработанный запрос

Выберите *Файл* > *Закрыть и загрузить в…*, укажите *Таблица* на *Имеющийся лист*, как на рис. 57.

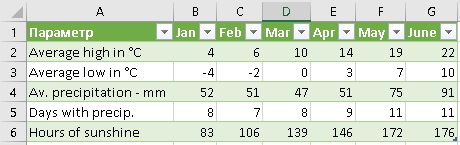


Рис. 72. Умная таблица на листе Excel, созданная запросом

#### Заключение

Excel и Интернет – две быстро развивающиеся динамические сущности, а эта небольшая книга способна показать вам интересные возможности, и придать импульс, чтобы вы начали заниматься этой увлекательной темой подробнее. Последний совет: вполне вероятно, что в какой-то момент вы застрянете при разработке своих проектов. Если это произойдет, подумайте о том, чтобы обратиться за помощью на веб-форумы, таких как [MrExcel Message Board](https://www.mrexcel.com/board/) или [Stack Overflow](https://ru.stackoverflow.com/) (есть на русском). Там вы найдете знающих людей, которые всегда готовы помочь. Удачи!