

Создание поисковых запросов через консоль в Python.

Теоретическая часть.

Модуль webbrowser.

```
>>> webbrowser.open(url, new=0, autoraise=True)
```

Открывает url, используя браузер по умолчанию. Если new равен 0, url открывается в том же окне браузера, если это возможно. Если new равен 1, по возможности открывается новое окно браузера. Если new равен 2, по возможности открывается новая страница браузера («вкладка»). Если autoraise равен True, окно вызывается, если это возможно (обратите внимание, что во многих оконных менеджерах это происходит независимо от установки этой переменной). Возможны и другие варианты открытия страницы.

Методы модуля:

- webbrowser.open_new(url)

Открывает URL в новом окне браузера по умолчанию, если это возможно, иначе открывает URL в единственном окне браузера.

- webbrowser.open_new_tab(url)

Открывает URL на новой странице («вкладке») браузера по умолчанию, если это возможно, иначе эквивалентно open_new().

- webbrowser.get(using=None)

Возвращает объект контроллера для типа браузера using. Если using — это None, возвращает контроллер для браузера по умолчанию, соответствующий среде вызывающего объекта.

- webbrowser.register(name, constructor, instance=None, *, preferred=False)

Эта функция регистрирует тип браузера name. После регистрации типа браузера, функция get() может возвращать контроллер для данного типа браузера. Constructor будет вызываться без параметров для создания экземпляра при необходимости, если instance не указан или None. Если указан instance, constructor никогда не будет вызываться и может быть None. Если установить preferred на True, то данный браузер станет предпочтительным результатом для вызова get() без аргументов. В противном случае эта функция полезна только в том случае, если вы планируете установить переменную BROWSER или вызвать get() с непустым аргументом, соответствующим имени, объявленного вами обработчика.

Пример открытия страницы.

```
>>> import webbrowser  
>>> webbrowser.open('https://www.google.com/')
```

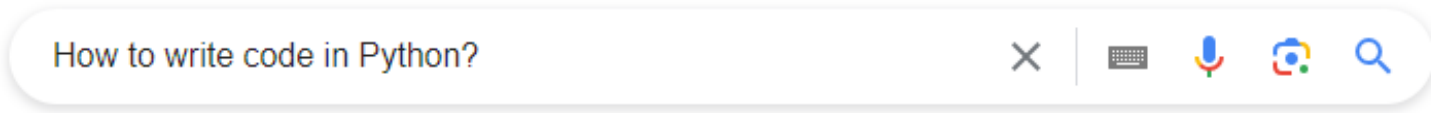
При вставке ссылки всё отобразится корректно и откроется стартовая страница Google. Но что если мы хотим сделать запрос и открыть его в определённом браузере?

```
>>> import webbrowser  
>>> webbrowser.open('How to write code in Python?')
```

Здесь будет небольшая загвоздка. Запрос выполнится, но в Internet Explorer или Microsoft Edge.

Для того, чтобы открыть запрос в нужном браузере, необходимо сделать следующее:

1. Вписать любой запрос.



2. Перейти в адресную строку и скопировать её часть. До того, как в ней будет наш запрос.



3. Вставить скопированную часть и использовать **+**, **&** или **.format**, для нашего запроса.

```
1 import webbrowser  
2 webbrowser.open('https://www.google.com/search?q=' + 'How to write code in Python?')
```

4. Готово.

Модуль urllib.

Библиотека, собирающая несколько модулей для работы с URL-адресами:

- urllib.request для открытия и чтения URL-адресов
- urllib.error содержащие исключения, установленные
- urllib.request urllib.parse для анализа URL-адресов
- urllib.robotparser для разбора robots.txt файлов

Также, с помощью неё можно проверить подключение к интернету. Просто постараться открыть и прочесть URL.

```
>>> import urllib.request  
>>> urllib.request.urlopen("http://google.com")
```

Чтобы не было ошибки, если соединение отсутствует, необходимо использовать try except с ошибкой IOError. Когда мы пытаемся открыть файл, а он не существует, возникает ошибка IOError.

Практическая часть.

Написать программу по следующему алгоритму:

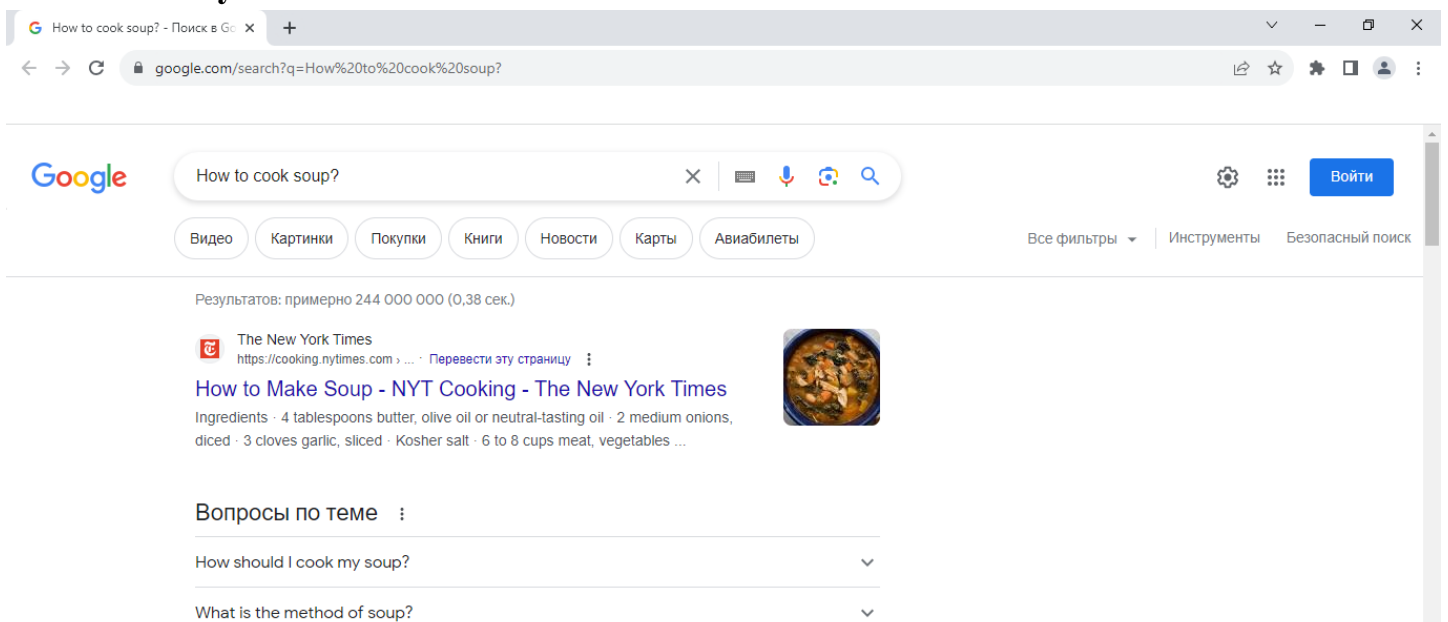
1. Программа проверяет подключён ли пользователь к интернету. В зависимости от результата, вывести соответствующее сообщение.
2. Если соединение есть, предлагаем пользователю ввести свой запрос и выбрать 2 поисковые системы на выбор (Яндекс и Google, например). Далее, открыть его запрос в определённом поисковике.
3. Если пользователь выбрал вариант не из предложенных, вывести сообщение «Выберете из предложенных вариантов» и повторить выбор.

Пример входных/выходных данных 1.

Ввод данных в консоль:

```
Проверка подключения к интернету...  
Соединение стабильно  
Введите запрос: How to cook soup?  
Выбор браузера для поиска  
[1] - Google, [2] - Яндекс.  
> 1  
  
Process finished with exit code 0
```

Результат:



Пример входных/выходных данных 2. Интернет отключён.

Ввод данных в консоль:

```
Проверка подключения к интернету...
Соединения не установлено
Проверьте подключение к интернету.

Process finished with exit code 0
```

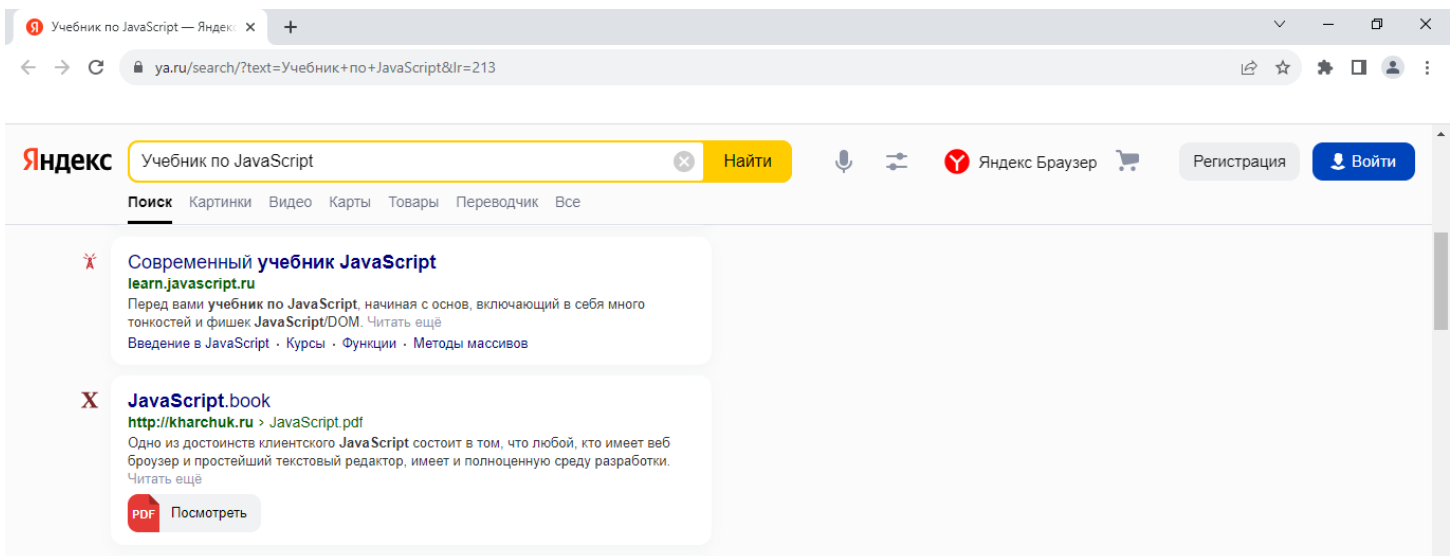
Пример входных/выходных данных 3. Ошибка при выборе.

Ввод данных в консоль:

```
Проверка подключения к интернету...
Соединение стабильно
Введите запрос: Учебник по JavaScript
Выбор браузера для поиска
[1] - Google, [2] - Яндекс.
> 3
Выберете из предложенных вариантов.
Выбор браузера для поиска
[1] - Google, [2] - Яндекс.
> 2

Process finished with exit code 0
```

Результат:



Возможный алгоритм выполнения.

1. Добавление библиотек webbrowser и urllib.request.
2. Проверка подключения к интернету, с помощью urllib.request. Всё должно происходить в блоке исключения try.
3. В качестве ошибки выставить IOError. «Соединения нет»
4. Продолжить можно в else (если блок try выполнится).
5. Переменная для ввода запроса пользователя.

6. Бесконечный цикл.
7. Предложить выбор поисковых систем.
8. Переменная для выбора пользователем.
9. Ветвление. Если выбор 1, или выбор 2. То webbrowser открыть в соответствующем браузере.
10. Иначе, повторите ввод, такого варианта нет.