**Тематика**: [Информатика](https://tvorcheskie-proekty.ru/informatika)

**Автор**: Чеботарев Никита Сергеевич

**Руководитель**: Донец Анатолий Анатольевич

**Учреждение**: г. Старый Оскол, МБОУ СОШ №34

**Класс**: 9

**Цель**: Написать программу-парсер для поиска информации на сайте tproger.ru

**Задачи**:

1. Узнать, что такое парсинг

2. Написать программу-парсер

Актуальность: в ученическом исследовательском проекте по информатике "Использование языка Python для парсинга сайта tproger.ru" автор даёт развернутые определения понятиям «парсинг», рассматривает способ парсинга для извлечения нужной информации из сайта, для более быстрого поиска нужных статей.

**Подробнее о проекте**:

В процессе написания исследовательской работы по информатике "Использование языка Python для парсинга сайта tproger.ru" учеником 9 класса был изучен сайт tproger.ru и дано развернутое пояснение его структуре. В проекте учащийся представил подробное описание работы программы для парсинга и предложил свой вариант написания программы.

В готовом творческом проекте по информатике " Использование языка Python для парсинга сайта tproger.ru " автор также рассказал, о разновидностях сайтов для обучения программированию, которые помогут ученикам развиваться в этой сфере, используя интернет. Вся информация систематизирована и преподнесена доступным языком.

**Оглавление**

Введение  
1. Что такое парсинг?  
2. Рассмотрение библиотек Python, требующиеся для работы программы.  
3. Объяснение общей работы программы.  
Заключение

**Введение**

В интернете большое количество информации, которое требуется человеку для обучения или саморазвития, в исследовательской работе рассмотрим на примере сайта tproger.ru, сайт создал в обучающих целях, в нем мы будем искать информацию по конкретным ключевым словам.

**Что такое парсинг?**

Парсинг — автоматизированный сбор и систематизация информации из открытых источников с помощью скриптов. Другое название этого процесса — веб-скрейпинг.

Скрипты, которые собирают и систематизируют информацию, называются парсерами. Они работают так:

ищут источники по заданным параметрам — например, парсеру можно дать список сайтов, а он найдёт на них страницы с ценами;

извлекают из источников нужную информацию — несколько строк текста, ссылку или сумму;

преобразуют информацию — например, парсер может взять фрагмент из HTML-документа и преобразовать его в текст без кода;

сохраняют информацию в нужном формате — например, в виде списка или таблицы в Excel.

В исследовательской работе рассмотрим парсинг на языке Python

**Рассмотрение библиотек Python, требующиеся для работы программы.**

В данном проекте требуется две библиотеки: requests и bs4.

Библиотека Requests - это лицензированная HTTP-библиотека Apache2, которая позволяет отправлять HTTP/1.1 запросы с помощью Python. Чтобы играть с вебом, Python Requests является обязательным. Будь то использование API-интерфейсов, загрузка целых страниц Facebook и многое другое, нужно будет сделать запрос на URL-адрес.

Библиотека bs4 – это сокращение BeautifulSoup4 - это библиотека Python для извлечения данных из файлов HTML и XML. Для естественной навигации, поиска и изменения дерева HTML, модуль BeautifulSoup4, по умолчанию использует встроенный в Python парсер html.parser. BS4 так же поддерживает ряд сторонних парсеров Python, таких как lxml, html5lib и xml (для разбора XML-документов).

**Объяснение общей работы программы.**

Для работы парсера нам потребуются вышеуказанные библиотеки и их методы и функции

import requests  
import bs4  
url = 'https://tproger.ru/page/'  
HEADERS = {'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:89.0) Gecko/20100101 Firefox/89.0'}  
search = ['С++', 'Python', 'JavaScript', 'IT-команды']  
page = 1  
print("1) Использовать базовые слова для поиска('С++', 'Python', 'JavaScript', 'IT-команды' \n2) Ввести новые слова")  
com = int(input("Введите команду(номер): "))  
if com == 1:  
 print("Используются базовые слова")  
elif com == 2:  
 print('Введите 5 поисковых слов через Enter')  
 for i in range(5):  
 slov = input('Введите поисковое слово: ')  
 search.append(slov.strip(' '))  
else:  
 print("Комадна не найдена")  
ran = int(input("Введите число страниц для парсинга(не более 10): "))  
while True:  
 if ran > 10:  
 ran = int(input("Введено число больше 10, повторите ввод: "))  
 else:  
 break  
ran += 1  
for page in range(1, ran):  
 resp = requests.get(url + str(page)).text  
 soup = bs4.BeautifulSoup(resp, features='html.parser')  
 articles = soup.find\_all('article')  
 for article in articles:  
 link = article.find(class\_="article\_\_link").attrs['href']  
 temp\_art = bs4.BeautifulSoup(requests.get(link).text, features='html.parser')  
 #date = temp\_art.find(class\_='localtime meta\_\_date').text.strip()  
 heading = article.find(class\_="article\_\_link").text.strip('\n')  
 words = heading.split(' ')  
 for word in words:  
 if word in search:  
 print(f'Заголовок: {heading} \nСсылка: {link}')  
 print("Перешел к следующей странице")  
 page += 1

В первых строчках импортируем нужные библиотеки

Далее код действует по плану:

1. условие, по результату которого пользователь выбирает способ поиска. Поиск по базовым ключевым словам или поиск, по вводимым далее пользователем, словам.

2. Запрос на количество страниц для парсинга, по результату ввода программа будет искать заданные слова по N страницам сайта.

3. Запускается цикл по страницам сайта

4. Внутри цикла производится поиск заданных слов в заголовках статей

5. Выводятся данные по специальной форме: \*Название заголовка\* - \*ссылка на статью\*

Сайт tproger.ru по структуре HTML состоит из <article>, в которых мы будет искать, заданные нами слова, с помощью библиотеки bs4 будем ориентироваться по HTML разметке сайта. Найдя на сайте нужные нам блоки, изучаем HTML разметку и выделяем ключевые классы, к которым привязаны заголовок статьи и ссылка на статью. Задаем bs4 нужные нам параметры для поиска, в нашем случае это ключевые слова, если в статье есть ключевые слова, то следует вывести нужную нам информацию. Основа парсинга этой программы связана с библиотекой bs4, которая осуществляет поиск нужных заголовков и ориентацию в тегах сайта.

**Заключение**

Парсинг – это важная часть программиста, которая позволяет находить информацию быстро и качественно, поэтому она является неотъемлемой частью работы. Правильно сделанная программа для парсинга ускоряет процесс поиска информации, в исследовательской работе в пример программы-парсера привожу программу, написанную на языке Python с использованием дополнительных библиотек