



# Web scraping



Александр  
Бардин



# Александр Бардин

Python-разработчик в Open Solutions



[Александр Бардин](#)



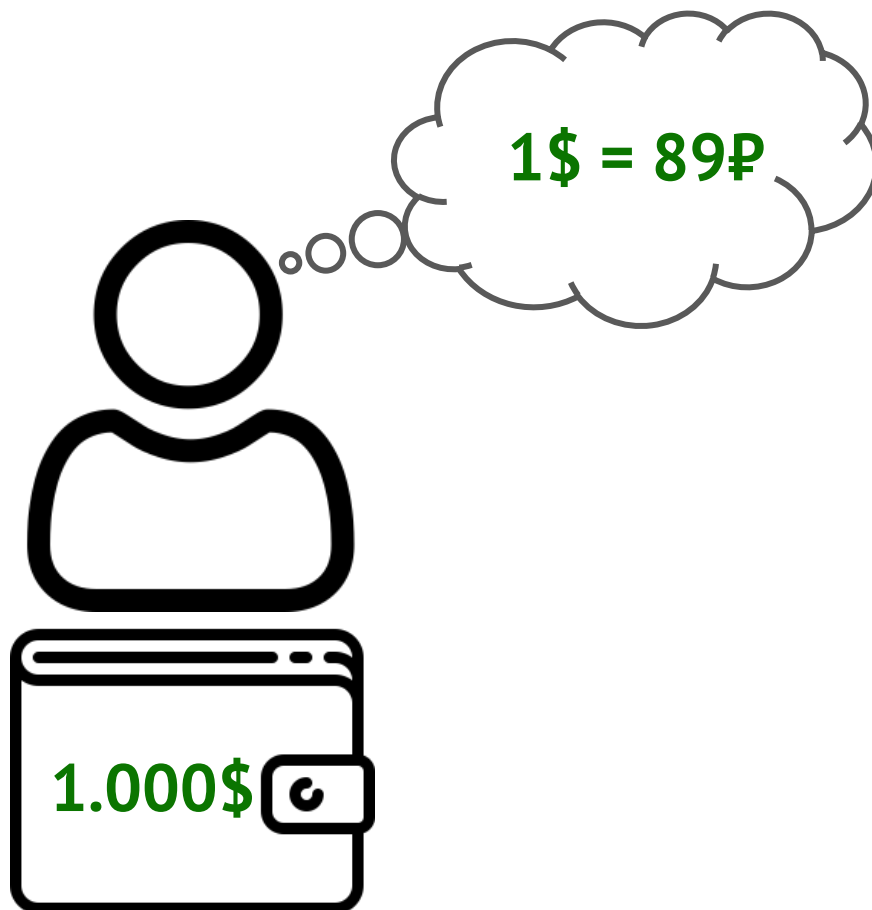
## План занятия

1. [Что такое web scraping](#)
2. [Проблемы при скрапинге](#)
3. [Инструменты](#)
4. [Извлечение информации](#)



# Что такое web scraping

# Как отследить повышение курса доллара

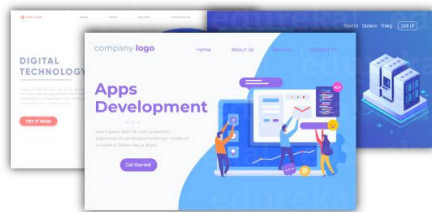


	Покупка	Продажа
\$	66₽	69₽

# Web scraping

— это процесс извлечения информации из интернета.

Когда мы говорим «**web scraping**», подразумевается именно автоматизация этого процесса.



Webpages



Web Scraping



Structured Data

Источник: [www.edureka.co](http://www.edureka.co)

---

# Правовой аспект



[Юридические аспекты скрапинга в современном вэбе.](#)



## Когда нужен скрапинг

1. Время на написание скрипта меньше, чем время, потраченное на эту же работу без скрипта.
2. Важна скорость получения результата.
3. Необходимо дождаться наступления определенного события.
4. Ваши предложения?





# Проблемы при скрапинге

# Проблемы при скрапинге



blognetology 17 июня 2019 в 11:27

## День открытых дверей факультета программирования в Нетологии

Блог компании Нетология, Программирование, Карьера в IT-индустрии

```
<dl class="post__tags">
  <dt class="post__tags-label">Хабы:</dt>
  <dd class="post__tags-list">
    <ul class="inline-list inline-list_fav-tags js-post-hubs">
      <li class="inline-list__item inline-list__item_tag">
        <a href="https://habr.com/ru/company/netologyru/" rel="tag"
class="inline-list__item-link post__tag">Блог компании Нетология</a></li>
      <li class="inline-list__item inline-list__item_tag">
        <a href="https://habr.com/ru/hub/programming/" rel="tag"
class="inline-list__item-link post__tag">Программирование</a></li>
      <li class="inline-list__item inline-list__item_tag">
        <a href="https://habr.com/ru/hub/career/" rel="tag"
class="inline-list__item-link post__tag">Карьера в IT-индустрии</a></li>
    </ul>
  </dd>
</dl>
```



# Проблемы при скрапинге

- Огромное многообразие сайтов.
- Сайты постоянно меняются.
- Сайты с динамически формируемым контентом.

# Можно без разметки?

Можно, нужен лишь API.

```
{
  "count": 1,
  "next": null,
  "previous": null,
  "results": [
    {
      "url": "http://tbacco.ru/api/baskets/1/lines/12/?format=json",
      "product": "http://tbacco.ru/api/products/29/?format=json",
      "quantity": 4,
      "attributes": [],
      "price_currency": "RUB",
      "price_excl_tax": "760.00",
      "price_incl_tax": "760.00",
      "price_incl_tax_excl_discounts": "760.00",
      "price_excl_tax_excl_discounts": "760.00",
      "is_tax_known": true,
      "warning": null,
      "basket": "http://tbacco.ru/api/baskets/1/?format=json",
      "stockrecord": "http://tbacco.ru/api/products/29/stockrecords/22/?format=json",
      "date_created": "2020-05-30T19:48:16.172000+03:00"
    }
  ]
}
```



# Инструменты




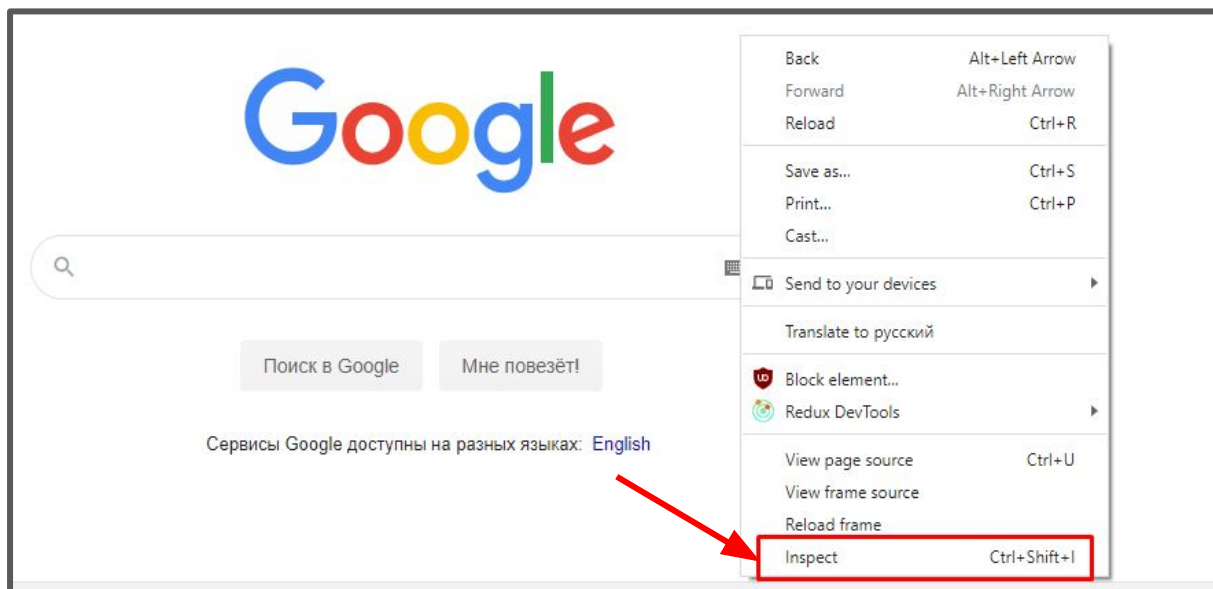
# Инструменты для скрапинга

1. Developer Tools в браузере.
2. Requests.
3. BeautifulSoup.
4. Requests-html.
5. Selenium.

# Как открыть Developer Tools

Developer Tools можно открыть несколькими способами:

- F12;
-  -> Inspect/Просмотреть код;
- Сочетания клавиш, например: **Ctrl + Shift + I**.



# Инструменты на python

Инструмент	Как установить	Описание
requests	<code>pip install requests</code>	извлечение сырой разметки
Beautiful Soup	<code>pip install beautifulsoup4</code>	удобная библиотека для легкого извлечения данных из сырой разметки
requests-html	<code>pip install requests-html</code> (только python 3.6+)	извлечение сырой разметки на сайтах с динамически формируемым контентом + удобное извлечение данных из полученной разметки
selenium	<code>pip install selenium</code> + нужен драйвер	эмулятор браузера или даже полноценный браузер (в зависимости от драйвера)





# Извлечение информации

# Извлечение информации

1. **Всегда начинаем с изучения особенностей целевого сайта!** Возможно, там есть удобный API или можно вытянуть информацию через тот же механизм, что и сам сайт (это тоже API, только не публичный).
2. Если никакого API нет, то переходим к плану Б.

Например, узнаем свой внешний ip:

```
import requests

ret = requests.get('https://2ip.ru/')
print(ret.text)

id_pos = ret.text.find('id="d_clip_button"')
ip_start_pos = ret.text.find('span>', id_pos) + 5
ip_end_pos = ret.text.find('</', ip_start_pos)

ip = ret.text[ip_start_pos:ip_end_pos]
print(ip)
```

# Извлечение информации. BeautifulSoup

А теперь тоже самое, но с библиотекой BeautifulSoup:

```
import requests
from bs4 import BeautifulSoup

ret = requests.get('https://2ip.ru/')
print(ret.text)

soup = BeautifulSoup(ret.text, 'html.parser')
el = soup.find(id='d_clip_button')

ip = el.text
print(ip)
```

[Документация.](#)



# Практика

Попробуем извлечь все посты с `habr.ru`, в которых есть интересующие нас хабы.

На экран надо напечатать название подходящей статьи и ссылку на неё.

[Решение.](#)



# Итоги

Сегодня на занятии мы:

1. Разобрались, что такое web scraping.
2. Посмотрели, с какими проблемами можно столкнуться.
3. Рассмотрели инструменты для скрапинга.
4. Изучили основные приёмы для извлечения информации.



# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задаём в чате Slack!
- Задачи можно сдавать по частям.
- Зачёт по домашней работе проставляется после того, как приняты **все задачи**.

**Задавайте вопросы и  
пишите отзыв о лекции!**

 [Александр Бардин](#)

**Александр Бардин**