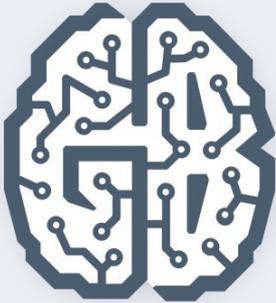


Методы сбора и обработки данных при помощи Python

Урок 2



Веб запросы, работа с API, парсинг

План урока

- 1) Немножко про ДЗ
- 2) Библиотека requests
- 3) Запросы к API Aviasales, парсинг JSON в python
- 4) Регулярные выражения — краткий экскурс
- 5) Парсинг статей Википедии, подсчет слов



Библиотека requests

<https://2.python-requests.org/en/master/> - документация

Behold, the power of Requests:

```
>>> r = requests.get('https://api.github.com/user', auth=('user', 'pass'))
>>> r.status_code
200
>>> r.headers['content-type']
'application/json; charset=utf8'
>>> r.encoding
'utf-8'
>>> r.text
u'{"type": "User"...}'
>>> r.json()
{u'private_gists': 419, u'total_private_repos': 77, ...}
```



Запросы

```
>>> r = requests.get('https://api.github.com/events')
>>> r = requests.post('https://httpbin.org/post', data =
{'key': 'value'})
>>> r = requests.put('https://httpbin.org/put', data =
{'key': 'value'})
>>> r = requests.delete('https://httpbin.org/delete')
>>> r = requests.head('https://httpbin.org/get')
>>> r = requests.options('https://httpbin.org/get')
```



GET

```
>>> payload = {'key1': 'value1', 'key2': 'value2'}
>>> r = requests.get('https://httpbin.org/get', params=payload)
URL = https://httpbin.org/get?key1=value1&key2=value2

>>> payload = {'key1': 'value1', 'key2': ['value2', 'value3']}
>>> r = requests.get('https://httpbin.org/get', params=payload)
URL = https://httpbin.org/getkey1=value1&key2=value2&key2=value3
```



POST

```
>>> payload = {'key1': 'value1', 'key2': 'value2'}
>>> r = requests.post("https://httpbin.org/post", data=payload)
>>> payload_tuples = [('key1', 'value1'), ('key1', 'value2')]
>>> r1 = requests.post('https://httpbin.org/post',
data=payload_tuples)
>>> payload_dict = {'key1': ['value1', 'value2']}
>>> r2 = requests.post('https://httpbin.org/post',
data=payload_dict)
```



Обработка ответа

```
>>> r = requests.get('https://api.github.com/events')
>>> r.text
u' [{"repository":{"open_issues":0,"url":"https://github.com/...

>>> r.content
b' [{"repository":{"open_issues":0,"url":"https://github.com/...

>>> from PIL import Image
>>> from io import BytesIO
>>> i = Image.open(BytesIO(r.content))
```



Обработка ответа

```
>>> r =
requests.get('https://api.github.com/events')
>>> r.json()
[{'repository': {'open_issues': 0, 'url':
'https://github.com/...
```



Во время изучения чего-то нового, я самозабвенно выдумываю невероятные ситуации, в которых это умение поможет мне спасти мир

О нет! Убийца должно быть последовал за ней в отпуск!

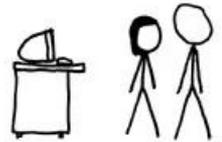


Но чтобы узнать где он, нам нужно прочесть 200 Мб писем в поисках чего-то схожего по формату с адресом!

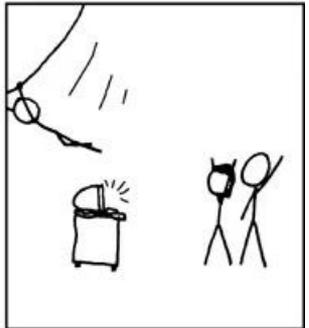
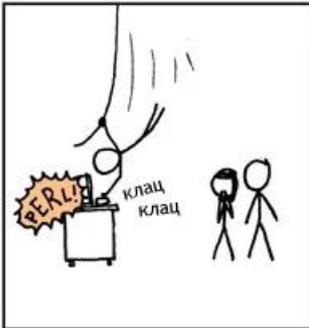
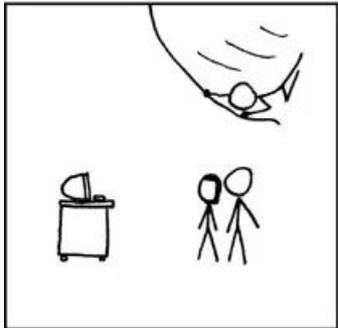


Это безнадежно!

Всем расступиться



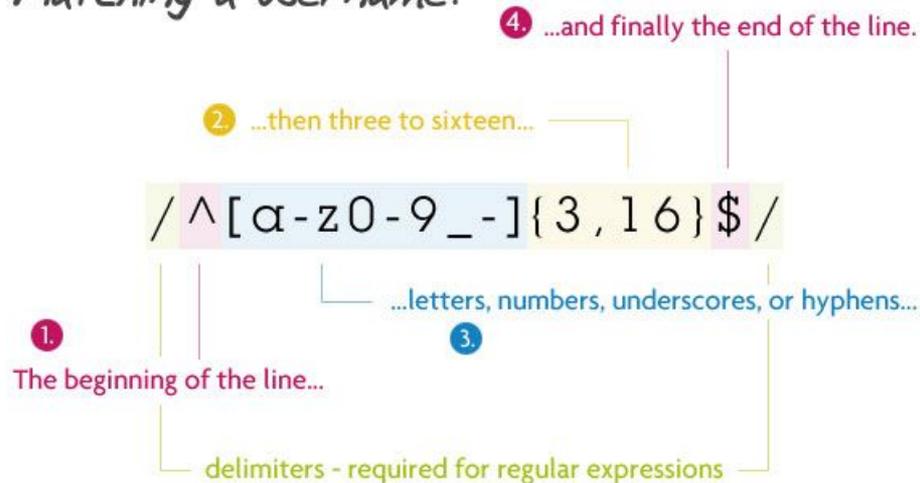
Я знаю регулярные выражения



Регулярные выражения

<https://habr.com/ru/post/66931/> - простые примеры

Matching a username:



Регулярные выражения

| Спец. символ | Зачем нужен |
|--------------|---|
| . | Задаёт один произвольный символ |
| [] | Заменяет символ из квадратных скобок |
| - | Задаёт один символ, которого не должно быть в скобках |
| [^] | Задаёт один символ из не содержащихся в квадратных скобках |
| ^ | Обозначает начало последовательности |
| \$ | Обозначает окончание строки |
| * | Обозначает произвольное число повторений одного символа |
| ? | Обозначает строго одно повторение символа |
| + | Обозначает один символ, который повторяется несколько раз |
| | Логическое ИЛИ . Либо выражение до, либо выражение после символа |
| \ | Экранирование. Для использования метасимволов в качестве обычных |
| () | Группирует символы внутри |
| { } | Указывается число повторений предыдущего символа |



Домашнее задание

- 1) Доработать приложение по поиску авиабилетов, чтобы оно возвращало билеты по названию города, а не по IATA коду. (У aviasales есть для этого дополнительное API) Пункт отправления и пункт назначения должны передаваться в качестве параметров. Сделать форматированный вывод, который содержит в себе пункт отправления, пункт назначения, дату вылета, цену билета (можно добавить еще другие параметры по желанию)
- 2) В приложении парсинга википедии получить первую ссылку на другую страницу и вывести все значимые слова из неё. Результат записать в файл в форматированном виде
- 2.* Научить приложение определять количество ссылок в статье. Спарсить каждую ссылку и результаты записать в отдельные файлы.



Ваши вопросы?

