

# Разбор домашнего задания

**Роман Булгаков**

Спикер курса

Skillbox

# Регулярные выражения: модуль re и его методы

**Роман Булгаков**

Спикер курса

Skillbox

# Регулярные выражения: шаблоны

**Роман Булгаков**

Спикер курса

Skillbox

# Задача «Слова на гласную»

## ✓ Входные данные:

```
text = 'AV is largest Analytics community of India'
```

## ✓ Выходные данные:

- слова, которые начинаются на гласную букву

## Правила для паттерна:

- 1 Начало слова на одну из гласных букв: **aeiou**
- 2 Любой регистр: **aeiou** и **AEIOU**
- 3 Нет ограничений на количество букв в слове

# Зачем нужны регулярные выражения?

- Проверка формата данных (телефонный номер, email)
- Разбиение строки на подстроки; поиск, извлечение и замена символов
- Высокая скорость выполнения сложных операций по поиску и замене подстрок



Источник: [habr.com](http://habr.com)

# Задача «В поисках Немо»

✓ Входные данные:

```
""oCeAn Marlin OcEaN oceAN ocEAN oCEAN OCEAN OCEAn OCean Ocean ocean oCeAn OcEaN oceAN ocEAN
OCEAN OCEAn OCEan OCean Ocean ocean oCeAn OcEaN oceAN ocEAN oCEAN OCEAN OCEAn OCean OCean Ocean
ocean oCeAn OcEaN nemaa ocEAN oCEAN OCEAN OCEAn OCean OCean nemo0 ocean oCeAn OcEaN oceAN ocEAN
oCEAN OCEAN OCEAn OCEan OCean ocean oNeMa OcEaN oceAN ocEAN oCEAN OCEAN OCEAn OCEan OCean
Ocean ocean oCeAn OcEaN oceAN nenemo oCEAN OCEAN OCEAn OCean OCean Ocean Nemo ocean oCeAn OcEaN
oceAN ocEAN oCEAN OCEAN OCEAn OCean OCean Ocean ocean oCeAn OcEaN oceAN ocEAN oCEAN OCEAN OCEAn
OCEan OCean Ocean ocean ""
```

Возможна ошибка в окончании имени

✓ Выходные данные:

- найти все упоминания Немо

Правила для паттерна:

- 1 Нужны слова, которые начинаются на Nem или nem
- 2 Окончание слова может быть разным (от 0 до 2 букв)

# Основы парсинга: модуль request и формат json

**Роман Булгаков**

Спикер курса

Skillbox

# Формат JSON

**JSON (JavaScript Object Notation)** — текстовый формат обмена данными. Является одним из стандартизированных форматов.

JSON представляет собой:

- набор пар ключей: значение (словарь, хеш-таблица и т. д.)
- упорядоченный набор значений (например, список)

```
{
  "firstName": "Иван",
  "lastName": "Иванов",
  "address": {
    "streetAddress": "Московское ш., 101, кв.101",
    "city": "Ленинград",
    "postalCode": 101101
  },
  "phoneNumbers": [
    "812 123-1234",
    "916 123-4567"
  ]
}
```

Содержание типичного JSON-файла



# Итоги урока

- `import requests`  
`req = requests.get('https://swapi.dev/api/people/3/')`
- Формат JSON
- `import json`
- # десериализация JSON  
`data = json.load(file)`  
`data = json.loads(req.text)`
- # сериализация JSON  
`json.dump(data, file)`  
`j_data = json.loads(req.text)`



# Модуль itertools

**Роман Булгаков**

Спикер курса

Skillbox

# Итоги модуля

- `import re` # работа с regular expressions
- `my_pattern = r'\b[aeiouAEIOU]\w+'`  
`result = re.findall(my_pattern, text)`
- `import requests` # запросы
- `import json` # работа с json форматом
- `collections, itertools, datetime, decimal, matplotlib, numpy, pandas ...`

