

## Домашнее задание к лекции «Регулярные выражения и основы синтаксического разбора»



### Задание

[Вопросы по заданию](#)

Преподаватель: Олег Булыгин

#### Домашнее задание к лекции "Регулярные выражения"

Файл к домашней работе можно найти здесь [github.com...egex.ipynb](https://github.com...egex.ipynb)

Внимание! При реализации всех заданий необходимо использовать именно функционал регулярных выражений для обработки текста.

#### Задание 1

Напишите функцию, которая принимает на вход строку и проверяет является ли она валидным транспортным номером (1 буква, 3 цифры, 2 буквы, 2-3 цифры). Обратите внимание, что не все буквы кириллического алфавита используются в транспортных номерах.

Примеры работы программы:

```
car_id = 'A222BC96'
```

Результат: Номер A222BC валиден. Регион: 96

```
car_id = 'АБ22ВВ193'
```

Результат: Номер не валиден

#### Задание 2

Напишите функцию, которая будет удалять все последовательные повторы слов из заданной строки при помощи регулярных выражений.

Пример работы программы:

```
some_string = 'Напишите функцию функцию, которая будет будет будет удалять все все все все последовательные повторы слов из из из из заданной строки строки при помощи регулярных выражений'
```

Результат: Напишите функцию, которая будет удалять все последовательные повторы слов из заданной строки при помощи регулярных выражений.

#### Задание 3

Напишите функцию, которая будет возвращать акроним по переданной в нее строке со словами.

Примеры работы программы:

```
some_words = 'Информационные технологии'
```

Результат: ИТ

```
some_words = 'Near Field Communication'
```

Результат: NFC

#### Задание 4

Напишите функцию, которая будет принимать на вход список email-адресов и выводить их распределение по доменным зонам.

Пример работы программы:

```
emails = ['test@gmail.com', xyz@test.in, test@ya.ru, xyz@mail.ru, xyz@ya.ru, xyz@gmail.com]
```

Результат:

gmail.com : 2

test.in : 1

**Задание 5 (необязательное)**

Напишите функцию, которая будет подсчитывать сколько слов начинается на гласные, а сколько на согласные буквы в тексте (текст может быть написан как с использованием букв кириллицы, так и латиницы).

Пример работы программы:

some\_text = 'Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные.'

Результат:

Слов на гласные буквы: 9

Слов на согласные буквы: 21

**Задание 6 (необязательное)**

Напишите функцию, которая будет проверять номер сотового телефона на валидность, если он валиден, то переводить его в формат:

+7-xxx-xxx-xx-xx

Постарайтесь предусмотреть как можно больше адекватных форматов изначального ввода номера. Примеры работы программы:

phone = '+7 955 555-55-55'

Результат: +7-950-555-55-55

phone = '8(955)555-55-55'

Результат: +7-950-555-55-55

phone = '+7 955 555 55 55'

Результат: +7-950-555-55-55

phone = '7(955) 555-55-55'

Результат: +7-950-555-55-55

phone = '423-555-55-5555'

Результат: Номер не валиден

phone = '123-456-789'

Результат: Номер не валиден

**ПРИМЕЧАНИЕ**

**Домашнее задание сдается ссылкой на репозиторий GitHub. Не сможем проверить или помочь, если вы пришлете:**

файлы;  
архивы;  
скриншоты кода.

Все обсуждения и консультации по выполнению домашнего задания ведутся только на соответствующем канале в slack.

Как правильно задавать вопросы аспирантам, преподавателям и коллегам  
Прежде чем задать вопрос, попробуйте найти ответ в интернете. Навык самостоятельного поиска информации — один из важнейших. Каждый практикующий специалист любого уровня делает это ежедневно.

Сформулируйте вопрос по алгоритму: 1) Что я делаю?  
2) Какого результата я ожидаю?  
3) Как фактический результат отличается от ожидаемого?  
4) Что я уже попробовал сделать, чтобы исправить проблему?

По возможности прикрепите к вопросу скриншоты либо ссылки на код. Не выкладывайте все решение, оставляйте только проблемный и воспроизводимый участок кода.

[https://github.com/PeterM-lab/PYDA1/blob/main/07\\_regex/regex.ipynb](https://github.com/PeterM-lab/PYDA1/blob/main/07_regex/regex.ipynb) 28 мар. 2021

LINK

Вы загрузили решение 28 мар. в 12:56

Олег Булыгин поставил(а) зачет 31 мар. в 09:32

**Олег Булыгин**

ПРЕПОДАВАТЕЛЬ

31 марта 2021 09:32

Петр, приветствую!

Спасибо за работу, вижу, что неплохо разобрались с регулярками)

Решения очень хорошие, без лишних усложнений.

В первой задаче лучше было использовать не отдельные регулярки для выделения номера и региона, а скобочные группы, например, так:

```
1 car_id = input('Введите номер: ')
2
3 def check(car_id):
4     pattern = r'([АВЕКМНОРСТУХ]\d{3}[АВЕКМНОРСТУХ]{2})(\d{2,3})'
5     result = re.match(pattern, car_id)
6     if result:
7         print(f'Номер {result.group(1)} валиден. Регион: {result.group(2)}')
8     else:
9         print('Номер не валиден')
10
11 check(car_id)
```

А в последней задаче я немного бы усложнил шаблон, чтобы он был более универсальным, например так:

```
1 def get_correct_phone_number(number):
2     pattern = r'[+]?[87]*\s*(?(\d{3})\s*)?[+]?(\d{3})\s*[+]?(\d{2})\s*[+]?(\d{2})'
3     example = r'+7(1)-2-3-4'
4     return re.sub(pattern, example, number)
```

Зачет!



Вам понравилось?

## Вопросы по заданию

Задайте вопрос — вам помогут одноклассники и эксперты

ПМ

На какой вопрос вы хотите получить ответ?

Добавьте более подробное описание вашего вопроса, если необходимо

Спросить

**Все вопросы** Помочь с ответом

