Разработка на языке Python

Регулярные выражения

Ванурин Алексей
vanurin@phystech.edu
2018 год

Задача

Автомобильные номера имеют вполне определенную структуру

Пользователь вводит в систему ГИБДД номер своей машины. Требуется понять, ввел ли пользователь номер в правильном формате или ошибся и написал ерунду

Шаблон

Автомобильные номера имеют определенную структуру, шаблон (pattern)

В символьном виде номер можно описать как **C065MK78**

Шаблон

Другими словами, будем считать номером набор символов, подходящий под описание

1 буква - 3 цифры - 2 буквы - 2 или 3 цифры

Формальная постановка задачи

С информацией, представленной строчным типом с известной и четкой структурой, которую можно задать шаблоном(pattern), требуется **быстро**, **эффективно** и **просто** выполнять ряд действий

1. Находить такую информацию в тексте
2. Заменять такие данные в тексте
3. Проверять соответствие данных структуре
4. Разбивать такие данные по подструктурам

Примеры структур данных

1. **IP-адрес** (192.168.0.3)
2. **Электронная почта** (name@yandex.ru)
3. **Дата** (7/18/2018)
4. **Время** (1:46 AM)
5. **Автомобильный номер** (C065MK78)
6. **Хайку** (Вечер за окном. / Еще один день прожит. / Жизнь скоротечна...)

Регулярные выражения

Регулярные выражения (regular expressions) — формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов.

Для этого требуется шаблон, задающий правило поиска подстроки в строке

Это просто, быстро и эффективно

Примеры спецсимволов

- \d — любая цифра
- \D — любая НЕ цифра
- \w — любая буква цифра или _
- \W — любая не буква, не цифра, не _
- {n} — вхождение подшаблона слева ровно n раз
- {m,n} — вхождение от m до n раз
- [] — набор символов
- . — любой символ
- * — 0 или более вхождений подшаблона слева
- + — 1 или более вхождений подшаблона слева

Шаблон для почтового адреса

<Слово>@<Слово>.<Слово>

С помощью специальных символов, которые мы уже упоминали, можно записать следующий шаблон

“\w+@\w+\.\w+”

Немного подумав, можно апгрейднуть шаблон и сделать его более правильным

“\w+@\w+\.(ru|com)”

Функция fullmatch

```
Import re

s = “C065MK77”

pattern = r’\w+@\w+\.(ru|com)’

match = re.fullmatch(pattern, s)

if match:

    print(“Its matched!”)

else: print (“Not matched”)
```

Методы re

re.match()	Поиск по строке шаблона от начала строки	Есть ли шаблон
re.fullmatch()	Есть ли полное соответствие паттерну	Подходит ли под шаблон
re.search()	Поиск по строке шаблона	Найти нужную подстроку
re.findall()	Поиск всех совпадений шаблона	Найти нужные подстроки
re.split()	Разбить строку по шаблону	
re.sub()	Заменить все подстроки по шаблону	