Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

МФКТиУ, кафедра ПИиКТ

Лабораторная работа №4 по "Информатике"

Регулярные выражения.

Выполнил: Романов Артём

Группа: Р3110

Вариант: 307360

Санкт-Петербург

2020 г.

**Задание 1:**

Хайку – жанр традиционной японской лирической поэзии века, известный с XIV века.

Оригинальное японское хайку состоит из 17 слогов, составляющих один столбец иероглифов. Особыми разделительными словами – кирэдзи – текст хайку делится на части из 5, 7 и снова 5 слогов. При переводе хайку на западные языки традиционно вместо разделительного слова использую разрыв строки и, таким образом, хайку записываются как трёхстишия.

Вам надо проверить, что текст является Хайку. В качестве разделителя строк используются символы «/». Если разделители делят текст на строки, в которых 5/7/5 слогов, то выведите «Хайку!». Если число строк не равно 3, то выведите строку «Не хайку. Должно быть 3 строки.». Иначе выведите строку вида «Не хайку.»

Для простоты будем считать, что слогов ровно столько же, сколько гласных, не задумываясь о тонкостях.

**Задание 2:**

Реализуйте программный продукт на языке Python, используя регулярные выражения, выделите все предложения из вашего текста, соответствующие критерию по вашему варианту.

Придумайте способ, как проверить корректность работы вашего регулярного выражения и проверьте его.

**Вариант:**

Все предложения, в которых используется две и более запятых

**Задание 3:**

Реализуйте программный продукт на языке Python, используя регулярные выражения по варианту, представленному в таблице.

Для своей программы придумайте минимум 5 тестов.

Протестируйте свою программу на этих тестах.

**Вариант:**

С помощью регулярного выражения найти в тексте все слова, в которых две гласные стоят подряд, а после этого слова идёт слово, в котором не больше 3 согласных.

**Код задания 1:**

1. **import** re
3. # teststr = "Вечер за окном. / Еще один день прожит./ Жизнь скоротечна..."  # Хайку!
4. # teststr = "Вечер за окном. / Еще один день прожит./ Жизнь скоротечна.../" # Не Хайку. Должно быть 3 строки
5. # teststr = "Вечер за окном. / Еще один день прожит./ Жизнь скоротечн..." # Не Хайку
6. # teststr = "Вечер за окном. / Еще один день прожит./ Жизнь скоротечнаааа..." # Не Хайку
7. teststr = "Прошу отчислить. / меня по собственному./ Желаниюю."  # Хайку!
8. lines = teststr.split("/")
9. result = ''
10. i = 0
11. q = [0] \* 10000
12. pattern = '[бвгджзйклмнпрстфхцчшщБВГДЖЗЙКЛМНПРСТФХЦЧШЩ]{0,}[аеёиоуэюяыАЕЁИИОУЭЮЯЫ]'
13. **for** string **in** lines:
14. q[i] = len(re.findall(pattern, string))
15. i = i + 1
16. **if** len(lines) != 3:
17. result = 'Не Хайку. Должно быть 3 строки'
18. **else**:
19. **if** q[0] == 5 **and** q[1] == 7 **and** q[2] == 5:
20. result = "Хайку!"
21. **else**:
22. result = "Не Хайку"
23. **print**(result)

**Код задания 2:**

1. **import** re
3. file = open("Macbeth.txt")
4. test\_str = file.read()
5. file.close()
6. regex = '[!?.;]{0,}([^!?.;]\*,){2,}[^!?.;]{1,}[!?.;]'
7. matches = re.finditer(regex, test\_str)
8. **for** matchNum, match **in** enumerate(matches, start=1):
9. **print**("{matchNum} : {match}".format(matchNum=matchNum, match=match.group()))

**Код задания 3:**

1. **import** re
3. test\_str = "кривошеее существо гуляет по парку"  # гуляет
4. # test\_str = "кривошеее существо гуляет попопопо парку" # Empty output
5. # test\_str = "кривошеее существо гулят по парку" # Empty output
6. # test\_str = "отчисление по собственному желанию это выход" # отчисление; желанию
7. # test\_str = "пятый тест, на котором моя фантазия уже совсем на исходе"  # фантазия
8. regex = '[а-я]\*[бвгджзйклмнпрстфхцчшщ][аеёиоуэюяы][аеёиоуэюяы][^аеёиоуэюяы]\* ([бвгджзйклмнпрстфхцчшщ]\*[аеёиоуэюяы][бвгджзйклмнпрстфхцчшщь]\*){1,3}([ ]|$)'
9. matches = re.finditer(regex, test\_str)
10. **for** matchNum, match **in** enumerate(matches, start=1):
11. **print**("{matchNum} : {match}".format(matchNum=matchNum, match=match.group().split(" ")[0]))

**Вывод:**

Данная лабораторная работа научила меня работать с регулярными выражениями regex с помощью библиотеки re в Python и производить их отладку на сайте regex101.com