

Практическая работа №11

Обработка строковых данных с помощью регулярных выражений в Python

1 Цель работы

- 1.1 Научиться составлять регулярные выражения в программах на Python;
- 1.2 Научиться применять регулярные выражения для поиска и замены подстрок в программах на Python.

2 Литература

2.1 Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С. Р. Гуриков. – Москва : ФОРУМ : ИНФРА-М, 2022. URL:<https://znanium.com/read?id=390096>. – Режим доступа: для зарегистрир. пользователей. – Текст: электронный. – гл.7.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование 4.1 Персональный компьютер.

5 Задание

5.1 Написать регулярное выражение, разделяющее строку на предложения и вывести каждое предложение с новой строки. Признак конца предложения: точка, вопросительный знак, восклицательный знак, многоточие.

Для разделения строки на список использовать:

```
re.split(r'паттерн', строка)
```

5.2 Написать регулярное выражение, заменяющее слова «редиска» и фразу «нехороший человек» (и то и другое в любых склонениях и числе) на текст «*давайте жить дружно*».

Для замены подстроки на новую подстроку использовать:

```
re.sub(r'паттерн', 'новая подстрока', строка)
```

5.3 Написать регулярное выражение, выделяющее из строки все даты в формате дд.мм.гггг и выводящее их на экран. Вместо дд и мм может быть от 1 до

2 чисел, вместо гггг 2 или 4 числа. Месяц может быть от 1 до 12, день может быть от 1 до 31.

Для возврата списка всех совпадений использовать:

```
re.findall(r'паттерн', строка)
```

5.4 Написать регулярное выражение (или набор регулярных выражений), проверяющее, является ли введенная строка надежным паролем. Требования к паролю: длина от 6 символов, обязательно наличие цифры, строчной и прописной буквы латинского алфавита. Если требования не выполнены, запрашивать повторный ввод до тех пор, пока пароль не будет соответствовать требованиям.

Для определения, есть ли в строке определённый текст, использовать:

```
re.search(r'паттерн', строка)
```

6 Порядок выполнения работы

6.1 Запустить Python IDLE и выполнить все задания из п.5.

6.2 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «регулярное выражение»?

8.2 Для чего используются регулярные выражения?

8.3 Какие методы применяются в Python для работы с регулярными выражениями?

8.4 Как задать количество повторений символов в регулярных выражениях?

8.5 Как задать набор символов в регулярных выражениях?

8.6 Как привязать поиск к началу и концу строки в регулярном выражении?

8.7 Как задать буквенные, числовые и пробельные символы в регулярном выражении?