

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**  
**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**«Основы ветвления Git»**

**Отчет по лабораторной работе № 2.3**

**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Образцова М. Д.. «05» ноября 2022г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_

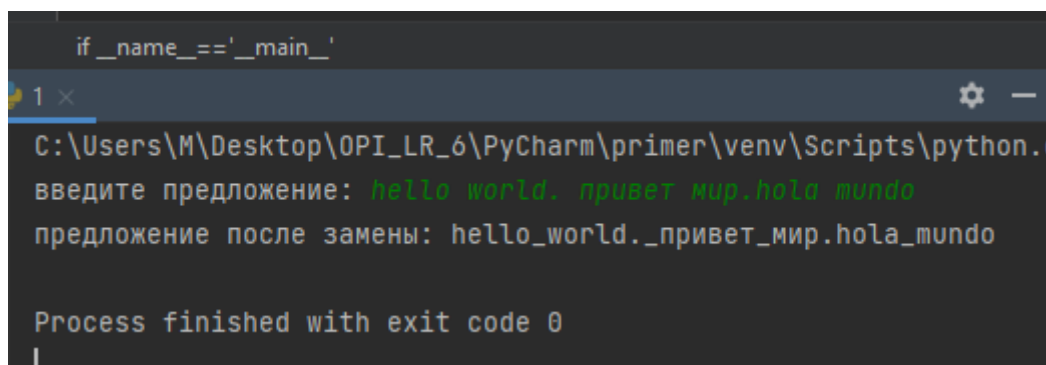
(подпись)

Методика и порядок выполнения работы

Создайте проект PyCharm в папке репозитория.

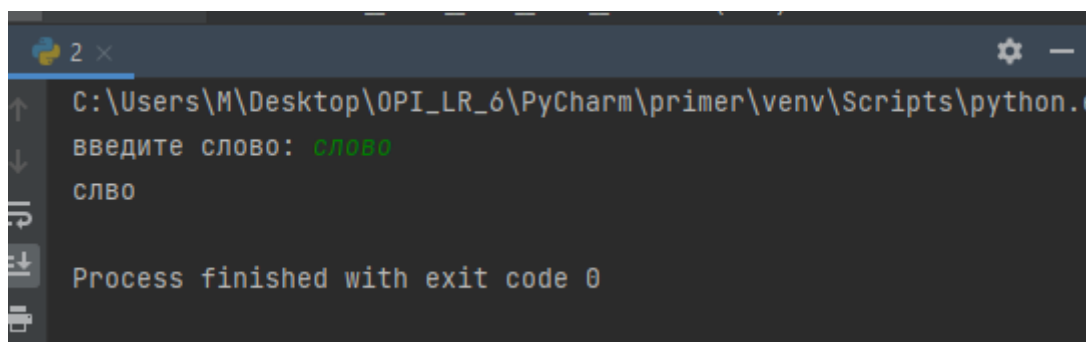
Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.

Приведите в отчете скриншоты результатов выполнения каждой из программ примеров при различных исходных данных вводимых с клавиатуры.



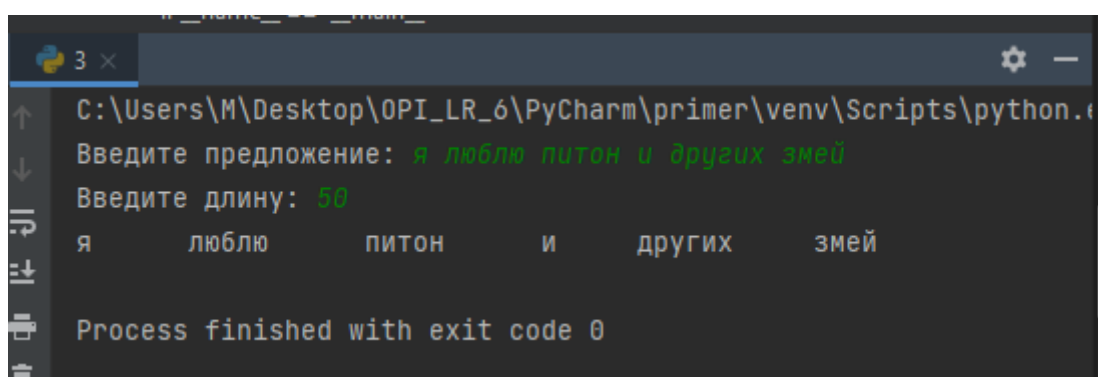
```
if __name__ == '__main__':  
1 x  
C:\Users\M\Desktop\OPI_LR_6\PyCharm\primer\venv\Scripts\python.exe  
введите предложение: hello world. привет мир.hola mundo  
предложение после замены: hello_world._привет_мир.hola_mundo  
  
Process finished with exit code 0
```

Рисунок 1–результат выполнения первого примера



```
2 x  
C:\Users\M\Desktop\OPI_LR_6\PyCharm\primer\venv\Scripts\python.exe  
введите слово: слово  
слово  
  
Process finished with exit code 0
```

Рисунок 2–результат выполнения второго примера



```
3 x  
C:\Users\M\Desktop\OPI_LR_6\PyCharm\primer\venv\Scripts\python.exe  
Введите предложение: я люблю питон и других змей  
Введите длину: 50  
я_люблю_питон_и_других_змей  
  
Process finished with exit code 0
```

Рисунок 3–результат выполнения третьего примера

Выполните индивидуальные задания, согласно своего варианта. Для заданий повышенной сложности номер варианта должен быть получен у преподавателя.

#### Индивидуальные задания

##### Задание 1

Дано предложение. Определить:

- число вхождений в него буквосочетания ро;
- число вхождений в него некоторого буквосочетания из двух букв;
- число вхождений в него некоторого буквосочетания.

```
1.py x
1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  ▶ if __name__ == '__main__':
7
8      # число вхождений в него буквосочетания po
9      s = input('введите предложение: ')
10     r = s.count('po')
11     print(f'количесвто буквосочетания po= {r} ')
12
13     # число вхождений в него некоторого буквосочетания из двух букв
14     sub = input('введите некоторое буквосочетание из 2 букв: ')
15     if len(sub) == 2 and sub.isalpha() == 1:
16         e = s.count(sub)
17         print(f'количество некоторого буквосочетания из 2 букв {e}')
18
19     else:
20         print(
21             "буквосочетание должно состоять из 2 букв",
22             file=sys.stderr
23         )
24         exit(1)
25
26     # число вхождений в него некоторого буквосочетания
27     sub2 = input('введите некоторое буквосочетание : ')
28     print(f'количество некоторого буквосочетания {s.count(sub2)}')
```

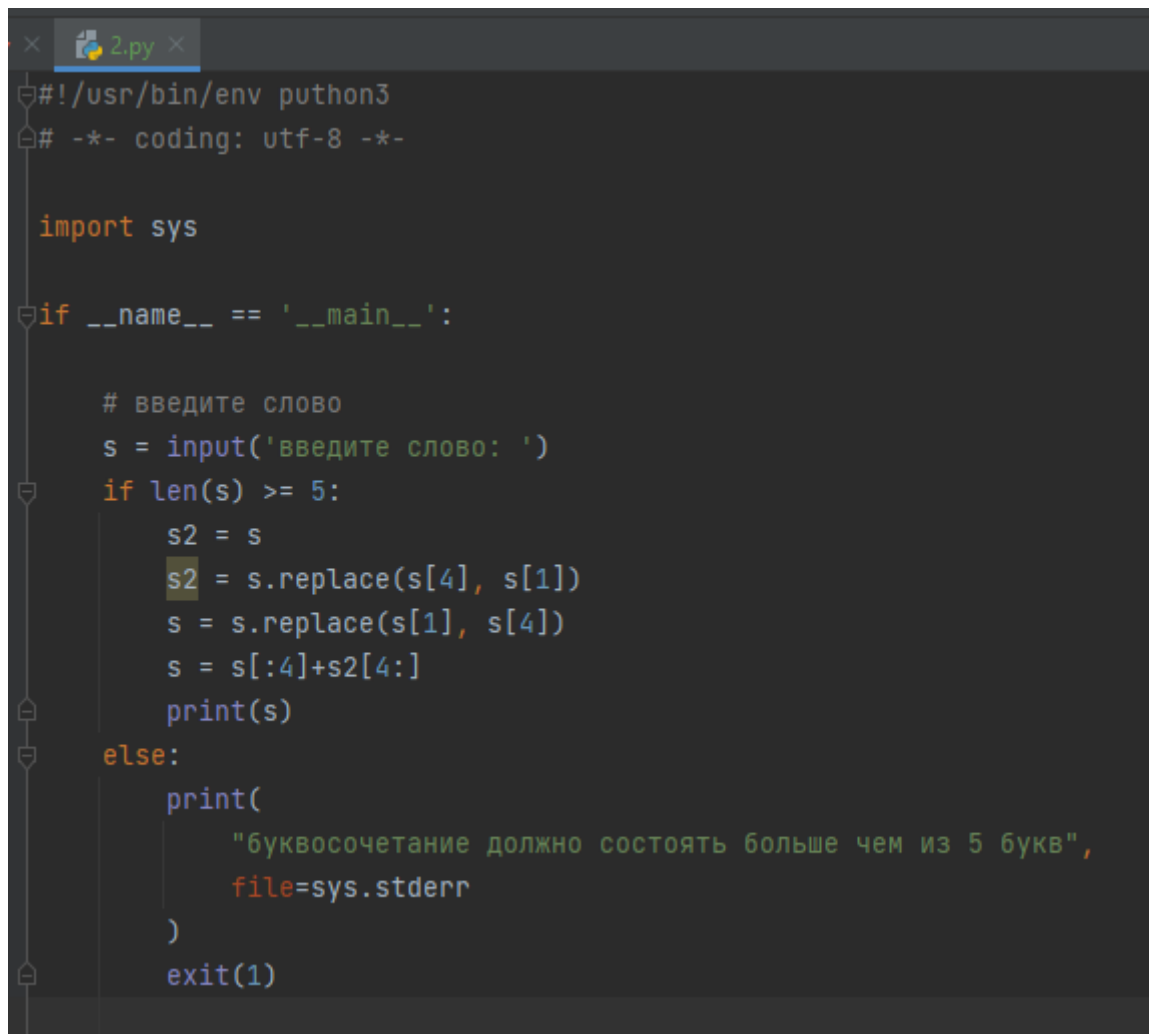
Рисунок 4—листинг программы

```
1 x
C:\Users\M\Desktop\OPI_LR_6\PyCharm\ind\venv\Scripts\python.exe
введите предложение: i love python and other snakes
количесвто буквосочетания po= 0
введите некоторое буквосочетание из 2 букв: lo
количество некоторого буквосочетания из 2 букв 1
введите некоторое буквосочетание : df
количество некоторого буквосочетания 0
```

Рисунок 5—результат выполнения

## Задание 2

Дано слово. Поменять местами его вторую и пятую буквы.

A screenshot of a code editor window titled '2.py'. The code is a Python script that takes a word as input and swaps the second and fifth characters if the word is at least 5 characters long. It uses string slicing and the replace method. The script includes a shebang, a UTF-8 encoding declaration, and a main block.

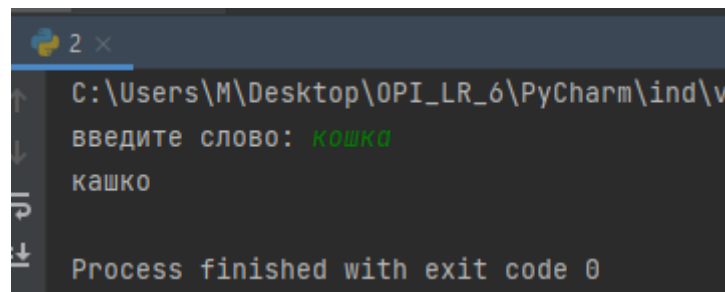
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':

    # введите слово
    s = input('введите слово: ')
    if len(s) >= 5:
        s2 = s
        s2 = s.replace(s[4], s[1])
        s = s.replace(s[1], s[4])
        s = s[:4]+s2[4:]
        print(s)
    else:
        print(
            "буквосочетание должно состоять больше чем из 5 букв",
            file=sys.stderr
        )
    exit(1)
```

Рисунок 6—листинг программы

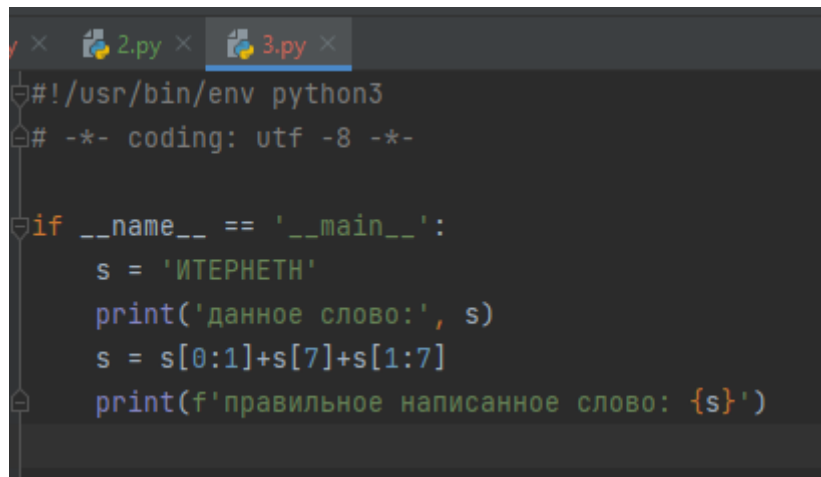
A screenshot of a terminal window titled '2'. It shows the execution of the program. The prompt 'введите слово:' is followed by the input 'кошка'. The output is 'кашко'. The terminal also shows the path 'C:\Users\M\Desktop\OPI\_LR\_6\PyCharm\ind\venv' and the message 'Process finished with exit code 0'.

```
C:\Users\M\Desktop\OPI_LR_6\PyCharm\ind\venv
введите слово: кошка
кашко
Process finished with exit code 0
```

Рисунок 7—результат выполнения

## Задание 3

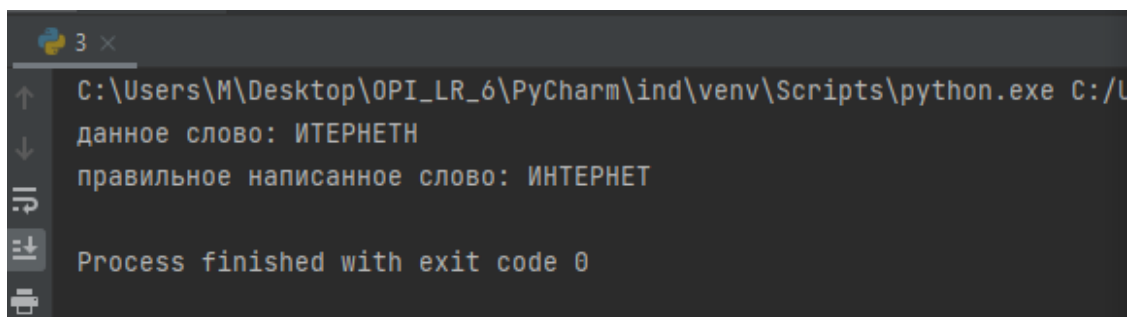
Дано ошибочно написанное слово ИТЕРНЕТН. Путем перемещения его букв получить слово ИНТЕРНЕТ.



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = 'ИТЕРНЕТН'
    print('данное слово:', s)
    s = s[0:1]+s[7]+s[1:7]
    print(f'правильное написанное слово: {s}')
```

Рисунок 8—листинг программы

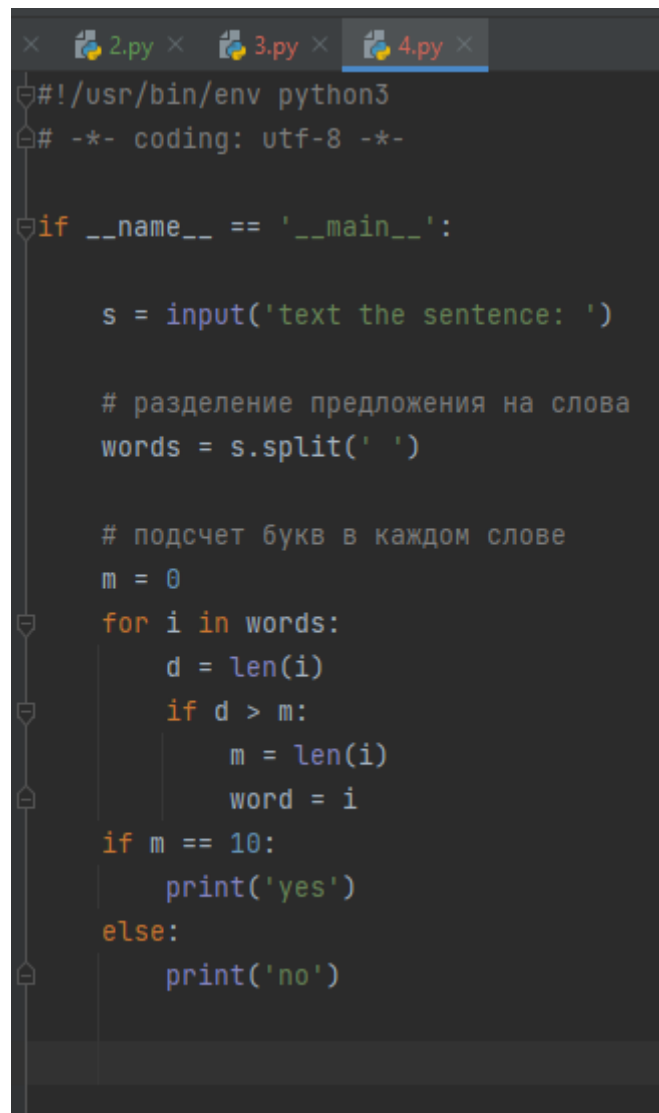


```
C:\Users\M\Desktop\OPI_LR_6\PyCharm\ind\venv\Scripts\python.exe C:/U
данное слово: ИТЕРНЕТН
правильное написанное слово: ИНТЕРНЕТ
Process finished with exit code 0
```

Рисунок 9—результат выполнения

Задание повышенной сложности

Дано предложение. Верно ли, что его самое длинное слово имеет больше 10 символов?



```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':

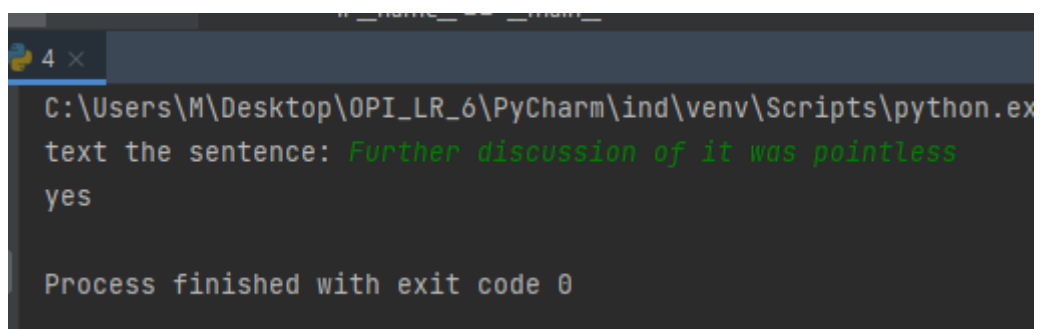
    s = input('text the sentence: ')

    # разделение предложения на слова
    words = s.split(' ')

    # подсчет букв в каждом слове
    m = 0
    for i in words:
        d = len(i)
        if d > m:
            m = len(i)
            word = i
    if m == 10:
        print('yes')
    else:
        print('no')

```

Рисунок 10—листинг программы



```

C:\Users\M\Desktop\OPI_LR_6\PyCharm\ind\venv\Scripts\python.exe
text the sentence: Further discussion of it was pointless
yes

Process finished with exit code 0

```

Рисунок 11—результат выполнения

## Вопросы для защиты работы

### 1. Что такое строки в языке Python?

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

### 2. Какие существуют способы задания строковых литералов в языке Python?

Строки в апострофах и в кавычках, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках

### 3. Какие операции и функции существуют для строк?

Сложение, дублирование, длина строки, длина строки, извлечение среза и т. д.

### 4. Как осуществляется индексирование строк?

Доступ к символам в строках основан на операции индексирования – после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

### 5. Как осуществляется работа со срезами для строк?

Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, `S[i]` — это срез, состоящий из одного символа, который имеет номер `i`, при этом считая, что нумерация начинается с числа 0. То есть если `S = 'Hello'`, то `S[0]=='H'`, `S[1]=='e'`, `S[2]=='l'`, `S[3]=='l'`, `S[4]=='o'`.

Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.

Срез с двумя параметрами: `S[a:b]` возвращает подстроку из `b-a` символов, начиная с символа с индексом `a`, то есть до символа с индексом `b`, не включая его.



6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

7. Как проверить то, что каждое слово в строке начинается с заглавной буквы?

```
string.istitle()
```

8. Как проверить строку на вхождение в неё другой строки?

```
string.find()
```

9. Как найти индекс первого вхождения подстроки в строку?

```
s.partition(<sep>)
```

10. Как подсчитать количество символов в строке?

```
len(s)
```

11. Как подсчитать то, сколько раз определённый символ встречается в строке?

```
s.count(<sub>)
```

12. Что такое f-строки и как ими пользоваться?

Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: `print(f"Меня зовут {name} Мне {age} лет.")`

13. Как найти подстроку в заданной части строки?

```
s.find(значение, начало, конец)
```

14. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`?

```
print('{}'.format(s))
```

15. Как узнать о том, что в строке содержатся только цифры?

```
s.isdigit()
```

16. Как разделить строку по заданному символу?

`str.split()`

17. Как проверить строку на то, что она составлена только из строчных букв?

`s.isalpha()`

18. Как проверить то, что строка начинается со строчной буквы?

`s.istitle()`

19. Можно ли в Python прибавить целое число к строке?

Нет

20. Как «перевернуть» строку?

`s.reverse()`

21. Как объединить список строк в одну строку, элементы которой разделены дефисами?

`str.split('-')`

22. Как привести всю строку к верхнему или нижнему регистру?

`s.upper()`  
`s.lower`

23. Как преобразовать первый символ строки к верхнему регистру?

`s.capitalize()`

24. Как проверить строку на то, что она составлена только из прописных букв?

`s.isupper()`

25. В какой ситуации вы воспользовались бы методом `splitlines()`?

`s.splitlines()` делит `s` на строки и возвращает их в списке. Любой из следующих символов или последовательностей символов считается границей строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

`s.replace(old, new)`

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

`str.startswith()` и `str.endswith()`

28. Как узнать о том, что строка включает в себя только пробелы?

`s.isspace()`

29. Что случится, если умножить некую строку на 3?

`AQ*3 = AQAQAQ`

30. Как привести к верхнему регистру первый символ каждого слова в строке?

`s.title()`

31. Как пользоваться методом `partition()`?

Метод `partition()` разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом `rfind()`?

`s.rfind(<sub>)` возвращает индекс последнего вхождения подстроки `<sub>` в `s`, который соответствует началу `<sub>`.