

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

Кафедра инфокоммуникаций

Основы кроссплатформенного программирования

Отчет по лабораторной работе №2.3

Работа со строками в языке Python

Выполнил студент группы

ИВТ-б-о-21-1 (2)

Стригалов Д.М. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

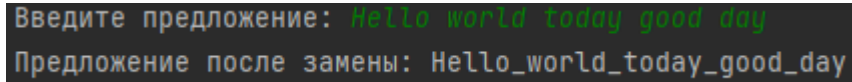
(подпись)

Ставрополь 2022

Цель работы: приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

Ход работы:

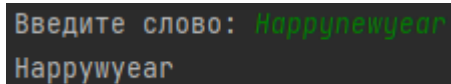
Пример 1.



```
Введите предложение: Hello world today good day
Предложение после замены: Hello_world_today_good_day
```

Рисунок 1 – результат выполнения программы

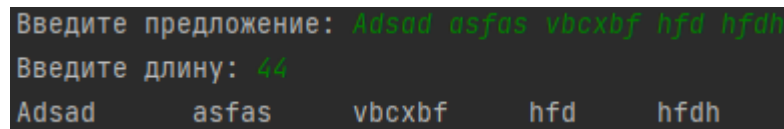
Пример 2.



```
Введите слово: Happynewyear
Happywyear
```

Рисунок 2 – результат выполнения программы

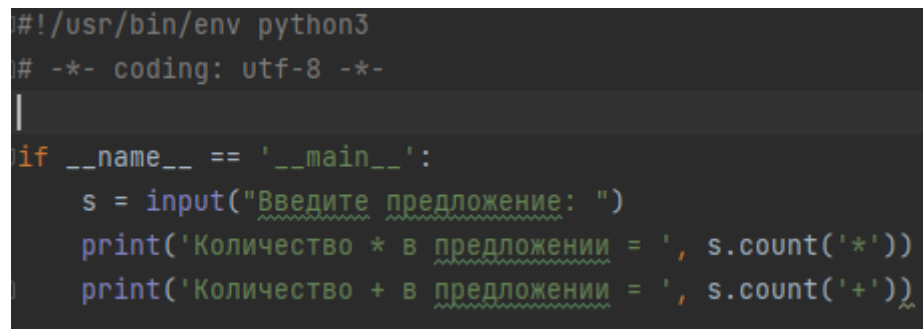
Пример 3.



```
Введите предложение: Adsad asfas vbcxbf hfd hfdh
Введите длину: 44
Adsad asfas vbcxbf hfd hfdh
```

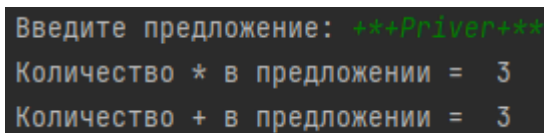
Рисунок 3 – результат выполнения программы

Индивидуальное задание 1. Дан текст. Сколько раз в нем встречается символ «+» и сколько раз символ «*»? ?



```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
|
if __name__ == '__main__':
    s = input("Введите предложение: ")
    print('Количество * в предложении = ', s.count('*'))
    print('Количество + в предложении = ', s.count('+'))
```

Рисунок 4 – Листинг программы



```
Введите предложение: +++Priver+++
Количество * в предложении = 3
Количество + в предложении = 3
```

Рисунок 5 – результат выполнения программы

Индивидуальное задание 2. Дано предложение. Заменить в нем все вхождения буквосочетания про на нет.

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

if __name__ == '__main__':
    s = input("Введите предложение: ")
    z = s.replace('про', 'нет')
    print(z)
```

Рисунок 6 – Листинг программы

```
Введите предложение: про про нет нет
нет нет нет нет
```

Рисунок 7 – результат выполнения программы

Индивидуальное задание 3. Дано ошибочно написанное слово роцессорп. Путем перемещения его букв получить слово процессор.

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

if __name__ == '__main__':
    p = 'роцессорп'
    tmp = list(p)

    s = tmp[-1]
    for i in range(len(p)-1, -1, -1):
        tmp[i] = tmp[i-1]
    tmp[0] = s

    p = ''.join(tmp)
    print(p)
```

Рисунок 8 – Листинг программы

```
процессор

Process finished with exit code 0
```

Рисунок 9 – результат выполнения программы

Задание повышенной сложности. Дано слово. Определить, сколько различных букв в нем.

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

if __name__ == '__main__':
    s = input("Введите слово: ")
    print("Разных букв в слове: ", len(set(s)))
```

Рисунок 10 – Листинг программы

```
Введите слово: крем
Разных букв в слове: 5
```

Рисунок 11 – результат выполнения программы

Контрольные вопросы:

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, 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?

Asd*3 = AsdAsdAsd

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