

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра  
инфокоммуникаций  
Институт цифрового  
развития**

**ОТЧЁТ**

**по лабораторной работе №2.3**

Дисциплина: «Основы программной инженерии»

Тема: «Работа со строками в языке Python»

Выполнил:  
студент 2 курса  
группы Пиж-б-о-21-1  
Коныжев Максим  
Викторович

Ставрополь 2022

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

1. Был создан общедоступного репозиторий в GitHub в котором были добавлены gitignore, правила для работы с IDE PyCharm с ЯП Python и лицензия MIT, репозиторий был клонировал на локальный сервер и организован в соответствие с моделью ветвления git-flow.

Owner \* Repository name \*

MaxDrill / lab2\_3

Great repository names are short and memorable. Need inspiration? How about [animated-parakeet?](#)

Description (optional)

☒ **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐ **Private**  
You choose who can see and commit to this repository.

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License

This will set main as the default branch. Change the default name in your [settings](#).

You are creating a public repository in your personal account.

**Create repository**

Рисунок 1.1 – Создание общедоступного репозитория

```

C:\Users\UESR\gitproj\lab2_3>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

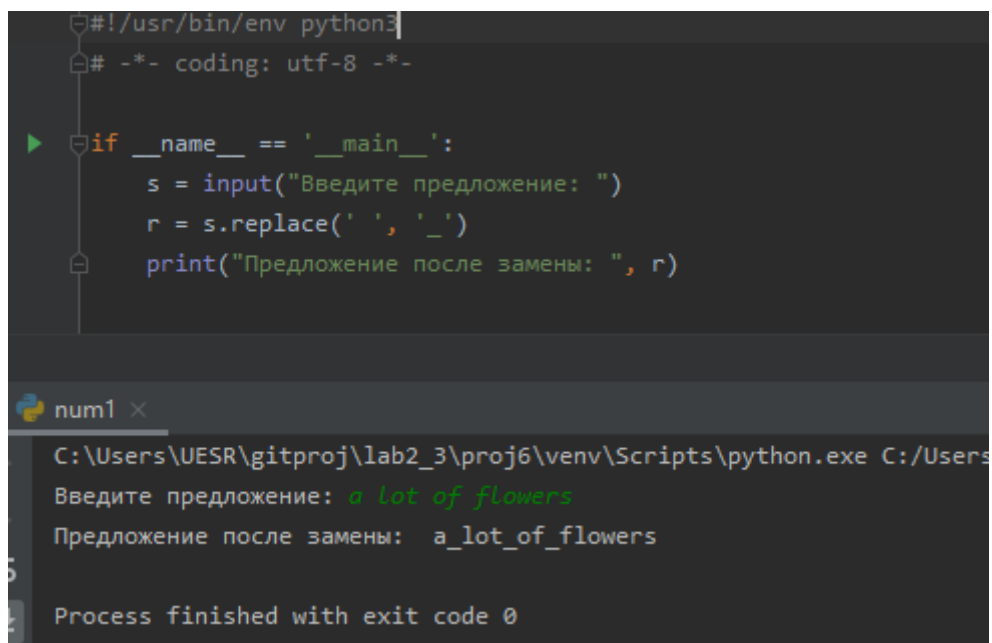
How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/UESR/gitproj/lab2_3/.git/hooks]

C:\Users\UESR\gitproj\lab2_3>git branch
* develop
  main

```

Рисунок 1.3 – Репозиторий был организован в соответствии с моделью ветвления git-flow

2. Была создана папка pycharm в которую были помещены примеры.



```

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

if __name__ == '__main__':
    s = input("Введите предложение: ")
    r = s.replace(' ', '_')
    print("Предложение после замены: ", r)

```

num1 ×

```

C:\Users\UESR\gitproj\lab2_3\proj6\venv\Scripts\python.exe C:/Users
Введите предложение: a lot of flowers
Предложение после замены: a_lot_of_flowers

Process finished with exit code 0

```

Рисунок 2.1 – Результат работы первого примера

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    word = input("Введите слово: ")
    idx = len(word) // 2
    if len(word) % 2 == 1:
        r = word[:idx] + word[idx+1:]
    else:
        r = word[:idx-1] + word[idx+1:]
    print(r)
```

if \_\_name\_\_ == '\_\_main\_\_' > else

num2 x

C:\Users\UESR\gitproj\lab2\_3\proj6\venv\Scripts\python.exe

Введите слово: *number*

nuer

Process finished with exit code 0

Рисунок 2.2 – Результат работы второго примера

```
C:\Users\UESR\gitproj\lab2_3\proj6\venv\Scripts\p
Введите предложение: a lot of flowers
Введите длину: 22
a lot of flowers

Process finished with exit code 0
```

Рисунок 2.3 – Результат работы третьего примера

3. Было выполнено три индивидуальных задания в соответствии с вариантом 7

## Задание 1

Дано предложение. Напечатать все его буквы и.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    a = input()
    print(a.count("и"))
```

```
C:\Users\UESR\gitproj\lab2_3\proj6\
мы идем за грибами
3
```

Рисунок 3.1 – Результат выполнения программы

## Задание 2

Дано предложение. Определить количество букв н, предшествующих первой запятой предложения. Рассмотреть два случая:

известно, что запятые в предложении есть;

запятых в предложении может не быть.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    a = input()
    if "," in a:
        print(a[:a.find(',')].count('н'))
    else:
        print("в предложении нет ,")
```

```
C:\Users\UESR\gitproj\lab2_3\proj6\venv\Scripts
на небе, нет
2

Process finished with exit code 0
```

Рисунок 3.2 – Результат работы программы в первом случае

```

C:\Users\UESR\gitproj\lab2_3\proj6\venv\Scripts\python.exe
на небе нет
в предложении нет ,

Process finished with exit code 0

```

Рисунок 3.3 – Результат работы программы во втором случае

### Задание 3

Дано слово. Удалить из него все повторяющиеся буквы, оставив их первые вхождения, т. е. в слове должны остаться только различные буквы.

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-
if __name__ == '__main__':
    strn = input()
    print(''.join(sorted(set(strn), key=strn.index)))

```

```

C:\Users\UESR\gitproj\lab2_3\proj6\venv\Scripts\python.exe
hhello
helo

Process finished with exit code 0

```

Рисунок 3.4 – Результат работы 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, 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`-а 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 букв?

`str.split()`

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

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