

**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 кроссплатформенного программирования»

Тема: «Работа со строками в языке Python»

Выполнил: студент 1 курса

группы ИВТ-б-о-21-1

Кочкаров Умар Ахматович

Ставрополь 2022

## Ход работы


1. Создал репозиторий в GitHub, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

Owner \*

 umarkochkarov ▾


Repository name \*

/ lb2.3 ✓

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

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 репозитория

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/Универ/Основы крос/Лаб2.3
$ git clone https://github.com/umarkochkarov/lb2.3.git
Cloning into 'lb2.3'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.
```

Рисунок 1.2 Клонирование репозитория

```
erken@LAPTOP-ESTC60GF MINGW64 ~/Desktop/Универ/Основы крос/Лаб2.3/lb2.3 (main)
$ git flow init

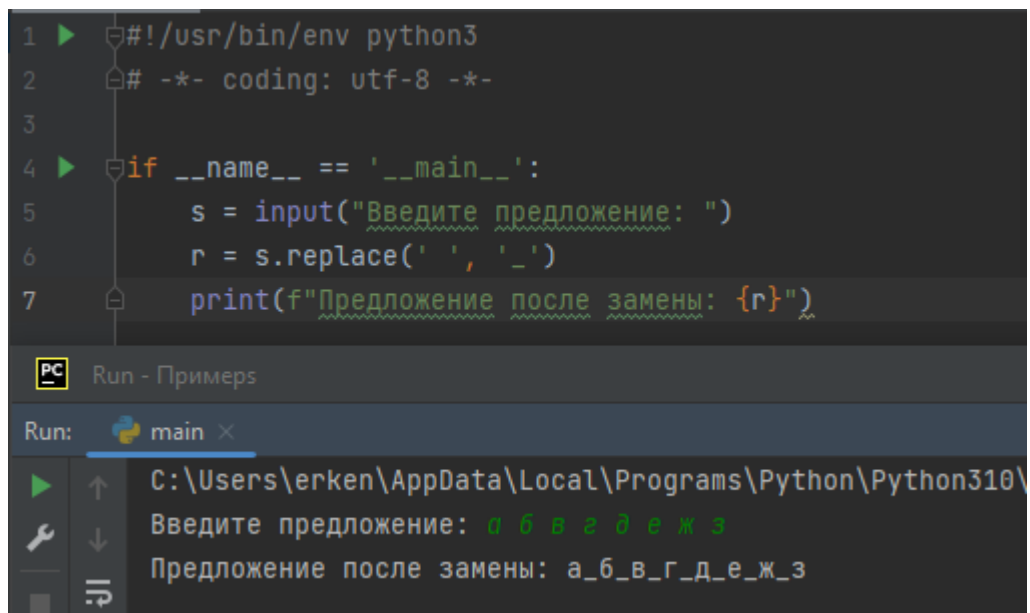
Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

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/erken/Desktop/Универ/Основы крос/Лаб2.3/lb2.3/.git/hooks]
```

Рисунок 1.3 Организация репозитория согласно модели ветвления git-flow

## 2. Проработал примеры из лабораторной работы

1) Дано предложение. Все пробелы в нем заменить символом «\_».



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     s = input("Введите предложение: ")
6     r = s.replace(' ', '_')
7     print(f"Предложение после замены: {r}")
```

Run - Примеры

Run: main x

C:\Users\erken\AppData\Local\Programs\Python\Python310\

Введите предложение: a b v g d e j z

Предложение после замены: a\_b\_v\_g\_d\_e\_j\_z

Рисунок 2.1 Код и результат выполнения программы

- 2) Дано слово. Если его длина нечетная, то удалить среднюю букву, в противном случае – две средние буквы.



```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 if __name__ == '__main__':
5     word = input("Введите слово: ")
6
7     idx = len(word) // 2
8     if len(word) % 2 == 1:
9         # Длина слова нечетная.
10        r = word[:idx] + word[idx+1:]
11    else:
12        # Длина слова четная.
13        r = word[:idx-1] + word[idx+1:]
14    print(r)
```

Run - Примеры

Run: primer\_2 x

C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe

Введите слово: Чечня

Ченя

Рисунок 2.2 Код и результат выполнения программы

- 3) Дана строка текста, в котором нет начальных и конечных пробелов. Необходимо изменить ее так, чтобы длина строки стала равна заданной длине (предполагается, что требуемая длина не меньше исходной). Это следует сделать путем вставки между словами дополнительных пробелов. Количество пробелов между отдельными словами должно отличаться не более чем на 1.

```
23 print(
24     "Предложение должно содержать несколько слов",
25     file=sys.stderr
26 )
27 exit(1)
28
29 # Количество пробелов для добавления.
30 delta = n
31 for word in words:
32     delta -= len(word)
33
34 # Количество пробелов на каждое слово.
35 w, r = delta // (len(words) - 1), delta % (len(words) - 1)
36
37 # Сформировать список для хранения слов и пробелов.
38 lst = []
39
40 # Пронумеровать все слова в списке и перебрать их.
41 for i, word in enumerate(words):
42     lst.append(word)
43
44     # Если слово не является последним, добавить пробелы.
45     if i < len(words) - 1:
46         # Определить количество пробелов.
47         width = w
48         if r > 0:
49             width += 1
50             r -= 1
51
52         # Добавить заданное количество пробелов в список.
53         if width > 0:
54             lst.append(' ' * width)
55
56 # Вывести новое предложение, объединив все элементы списка lst.
57 print(''.join(lst))
```

Run - Примеры

primer\_3 x

C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe C:/Users/erken/Desktop/

Введите предложение: *Пример работы программы*

Введите длину: *50*

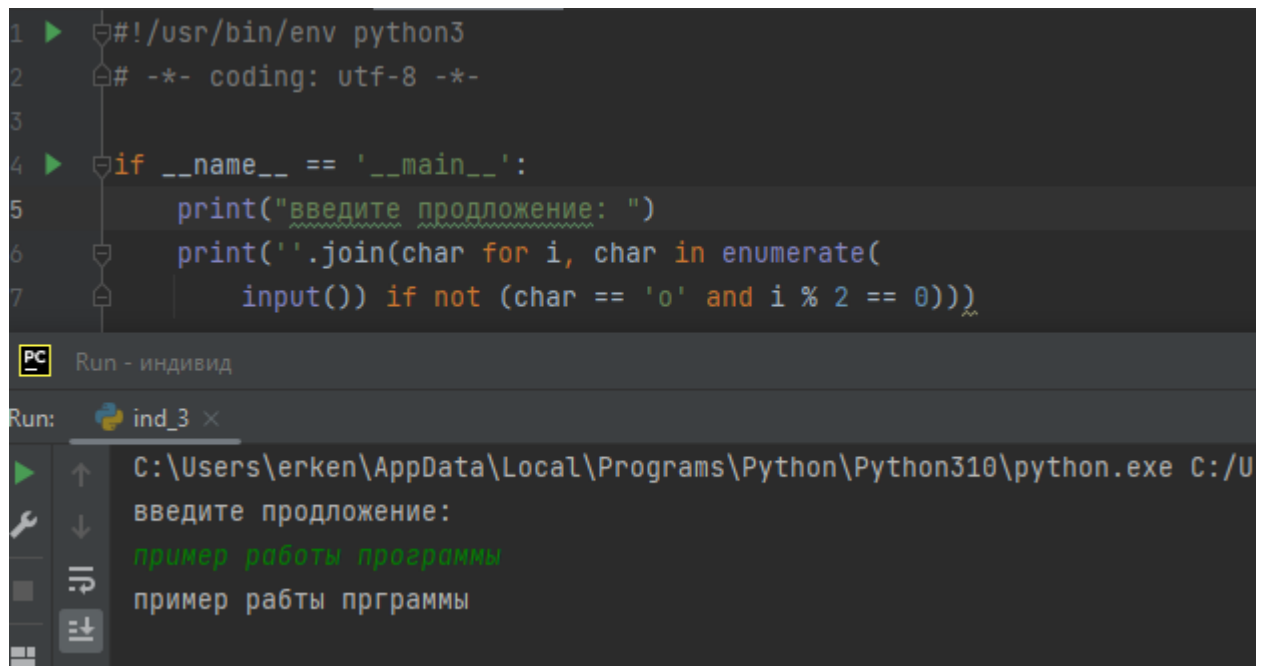
Рисунок 2.3 Код и результат выполнения программы

## Индивидуальные задания

- 1) Дано предложение. Составить программу, которая печатает «столбиком» все вхождения в предложение некоторого символа.



- 3) Дано предложение. Удалить из него все буквы о, стоящие на нечетных местах.



```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 ▶ if __name__ == '__main__':
5     print("введите предложение: ")
6     print(''.join(char for i, char in enumerate(
7         input()) if not (char == 'о' and i % 2 == 0)))
```

Run - индивид

Run: ind\_3 ×

C:\Users\erken\AppData\Local\Programs\Python\Python310\python.exe C:/U  
введите предложение:  
пример работы программы  
пример рабты программы

Рисунок 2.6 Код программы

## Ответы на 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(<sub>)>` возвращает индекс последнего вхождения подстроки

`<sub>` в `s`, который соответствует началу `<sub>`