

**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-6-о-21-1  
Урусов Максим  
Андреевич**

у

Ставрополь 2022

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

Выполнение работы:

1. Создал репозиторий в GitHub «rep 2.2» в который добавил .gitignore, который дополнил правила для работы с IDE PyCharm с ЯП Python, выбрал лицензию MIT, клонировал его на лок. сервер и организовал в соответствии с моделью ветвления git-flow.

Рисунок1.1 Создание репозитория

```
C:\Users\den-n>git clone https://github.com/MaksimUrusov/Laba2.3.git
Cloning into 'Laba2.3'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.
```

Рисунок1.2 Клонирование репозитория

```

C:\Users\den-n\L\Laba-2.3>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/den-n/L/Laba-2.3/.git/hooks]

```

Рисунок1.3 Организация репозитория в соответствии с моделью ветвления git-flow

## 2.Создание проекта PyCharm в папке проекта, проработка примеров

```

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

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

```

Пример 1

```

C:\Users\den-n\L\Laba-2.3\примеры\prim_1\Scripts\python.exe C:/Users/den-n/L/Laba-2.3/примеры/main.py
Введите предложение: привет как дела
Предложение после замены: привет_как_дела

Process finished with exit code 0

```

Работа примера 1

```

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

if __name__ == '__main__':
    word = input("Введите слово: ")
    idx = len(word) // 2
    if len(word) % 2 == 1:
        # Длина слова нечетная.
        r = word[:idx] + word[idx+1:]
    else:
        # Длина слова четная.
        r = word[:idx-1] + word[idx+1:]
    print(r)

```

## Пример2

```
C:\Users\den-n\L\Laba-2.3\примеры\prim\Scripts\python.exe C:/Users/den-n/L/Laba-2.3/примеры/main.py
Введите слово: hello
helo

Process finished with exit code 0
|
```

## Работа примера 2

```
import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)

    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)
```

## Пример 3

```
C:\ProgramData\Anaconda3\envs\Laba_2_3\python.exe D:/git/Laba_2_3/prim_3.py
Введите предложение: sdagag a h s hf dh s dfh sd hfa h
Введите длину: 49
sdagag a h s hf dh s dfh sd hfa h
```

## Работа примера 3

3. (16 вариант). Выполнил 3 индивидуальных задания и задание повышенной сложности.

Задание1. Дано предложение. Определить число пробелов в нем.

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

"""
16. Дано предложение. Определить число пробелов в нем.
"""

if __name__ == '__main__':
    s = str(input("Введите предложение - "))
    prob = 0
    for i in range(0, len(s)):
        if s[i] == ' ':
            prob += 1
    print("Кол-во пробелов равно - ", prob)
```

Рисунок3.1 Листинг программы

```
C:\Users\den-n\L\Laba-2.3\индивидуальные\ind_1\Scripts\python.exe C:/Users/den-n/L/Laba-2.3/индивидуальные/main.py
Введите предложение - Hello world
Кол-во пробелов равно - 1

Process finished with exit code 0
|
```

Рисунок3.2 Работа программы

Задание2. Дано предложение. Все его символы, стоящие на третьем, шестом, девятом и т. д. местах, заменить буквой а.

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

"""
Дано предложение. Все его символы, стоящие на третьем, шестом, девятом и т. д. местах, заменить буквой а.
"""

if __name__ == '__main__':
    s = str(input("Введите предложение - "))
    schet = 0
    for i in range(0, len(s)):
        if i == schet:
            s[i] = s.replace(s[i], 'a')
            schet += 1
    print(s)
```

Рисунок3.3 Листинг программы

```
Введите предложение - привет как дела
привеа как аела
```

### Рисунок3.4 Работа программы

Задание 3. Дано слово. Переставить его первую букву на место -й. При этом вторую, третью, ..., -ю буквы сдвинуть влево на одну позицию.

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

"""
Дано слово. Переставить его первую букву на место -й. При этом вторую, третью, ..., -ю
буквы сдвинуть влево на одну позицию.
"""

if __name__ == '__main__':
    w = str(input('Введите слово: '))
    k = int(input('Введите k: '))
    tmp = list(w)

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

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

### Рисунок3.5 Листинг программы

```
Введите слово: привет
Введите k: 2
ппвет
```

### Рисунок 3.6 Работа программы

Усложнённое задание: Дано предложение. Определить количество слов:  
начинающихся с буквы н;  
оканчивающихся буквой р.

```

"""
Дано слово. Переставить его первую букву на место -й. При этом вторую, третью, ..., -ю
буквы сдвинуть влево на одну позицию.
"""

if __name__ == '__main__':
    w = str(input('Введите слово: '))
    k = int(input('Введите k: '))
    tmp = list(w)

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

    w = ''.join(tmp)
    print(w)

```

Рисунок4.1 Листинг программы

```

Введите слово: pp nn pp
На n - 1
Оканчивающихся на p - 2

```

Рисунок 4.2 Выполнение программы

5. Сделал коммит, выполнил слияние с веткой main, и запустил изменения в уд. репозиторий.

```

D:\git\Laba_2_3>git add .

D:\git\Laba_2_3>git commit -m
error: switch `m' requires a value

D:\git\Laba_2_3>
D:\git\Laba_2_3>git commit -m "Выполненные задания"
[develop e9e953d] Выполненные задания

```

Рисунок5.1 Фиксация и коммит файлов

```

C:\Users\den-n\L\Laba-2.3>git merge readme
merge: readme - not something we can merge

C:\Users\den-n\L\Laba-2.3>git merge develop
Updating be5e18b..4a1a909
Fast-forward
 ind/ind_1.py      | 14 ++++++++
 ind/ind_2.py      | 15 ++++++++
 ind/ind_3.py      | 20 ++++++++
 prim/prim-1.py    | 7 +++++
 prim/prim-2.py    | 13 +++++
 prim/prim-3.py    | 54 +++++
 6 files changed, 123 insertions(+)
 create mode 100644 ind/ind_1.py
 create mode 100644 ind/ind_2.py
 create mode 100644 ind/ind_3.py
 create mode 100644 prim/prim-1.py
 create mode 100644 prim/prim-2.py
 create mode 100644 prim/prim-3.py
C:\Users\den-n\L\Laba-2.3>

```

Рисунок5.2 слияние веток develop и main

```

D:\git\Laba_2_3>git push
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 657 bytes | 657.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/LenkaGolovach/Laba_2_3.git
 e71c96f..0b7bab8  main -> main

```

Рисунок5.3 Пуш коммитов

## Контр. вопросы и ответы на них:

### 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?**

`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>`.