

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное                      государственное                      автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций  
«Работа со строками в языке Python»**

**Отчет по лабораторной работе № 2.3**

**по                      дисциплине  
«Программирование на Python»**

Выполнил студент группы ИВТ-б-о-22-1

Сумин Никита Сергеевич.

«        » \_\_\_\_\_ 2023г.

Подпись студента \_\_\_\_\_

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

Проверил Воронкин Р.А. \_\_\_\_\_

(подпись)

Ставрополь 2023

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

**Порядок выполнения работы:**

1. Создал общедоступный репозиторий на GitHub, в котором использована лицензия MIT и язык программирования Python.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \* Repository name \*

BrandooDi / mywork6 ✓

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

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.](#)

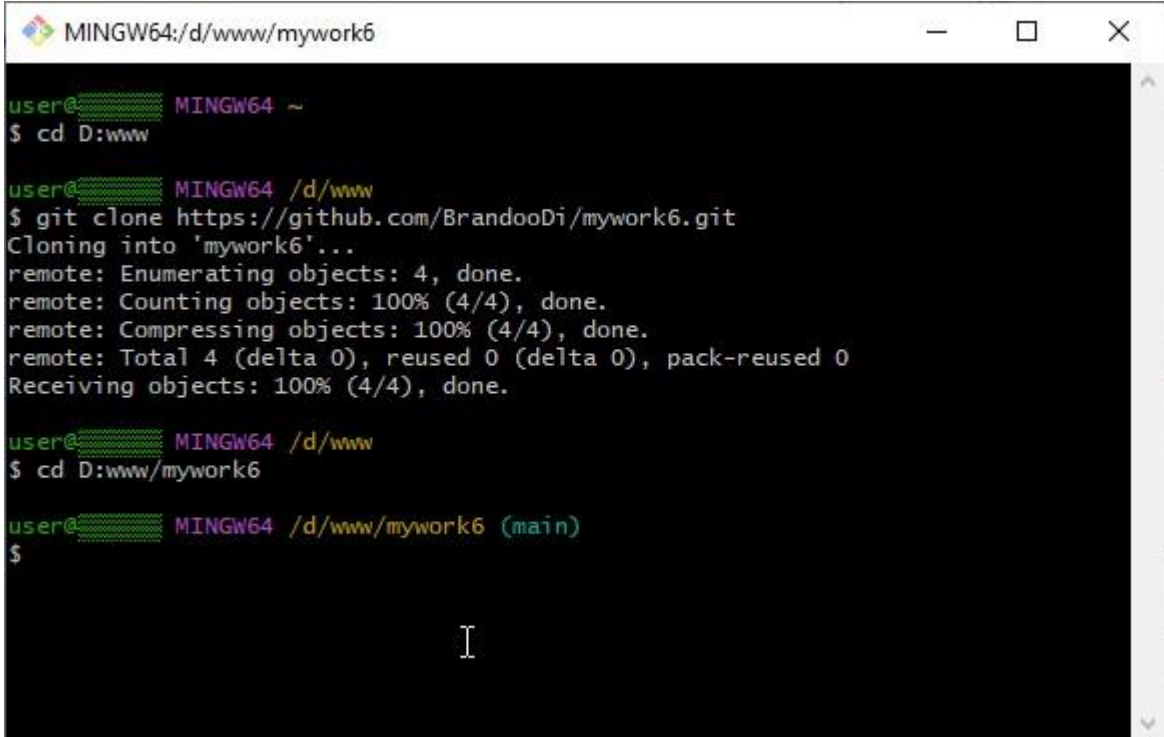
License: MIT License ▼

You are creating a public repository in your personal account.

Create repository

Рисунок 1 - Создание репозитория

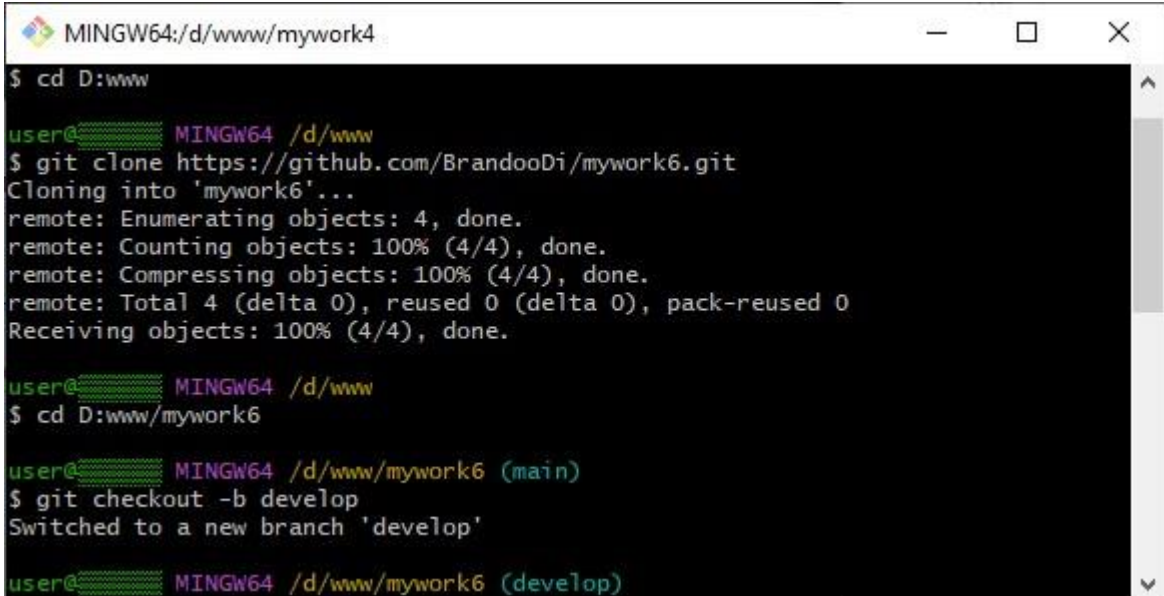
2. Выполните клонирование созданного репозитория.

A terminal window titled 'MINGW64:/d/www/mywork6' with standard Windows window controls. The terminal shows the following commands and output:

```
user@MINGW64 ~  
$ cd D:www  
  
user@MINGW64 /d/www  
$ git clone https://github.com/BrandooDi/mywork6.git  
Cloning into 'mywork6'...  
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.  
  
user@MINGW64 /d/www  
$ cd D:www/mywork6  
  
user@MINGW64 /d/www/mywork6 (main)  
$
```

Рисунок 2 - Клонирование репозитория

3. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.

A terminal window titled 'MINGW64:/d/www/mywork4' with standard Windows window controls. The terminal shows the following commands and output:

```
$ cd D:www  
  
user@MINGW64 /d/www  
$ git clone https://github.com/BrandooDi/mywork6.git  
Cloning into 'mywork6'...  
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.  
  
user@MINGW64 /d/www  
$ cd D:www/mywork6  
  
user@MINGW64 /d/www/mywork6 (main)  
$ git checkout -b develop  
Switched to a new branch 'develop'  
  
user@MINGW64 /d/www/mywork6 (develop)
```

Рисунок 3 - Ветвление по модели git-flow 4.

Создайте проект PyCharm в папке репозитория.

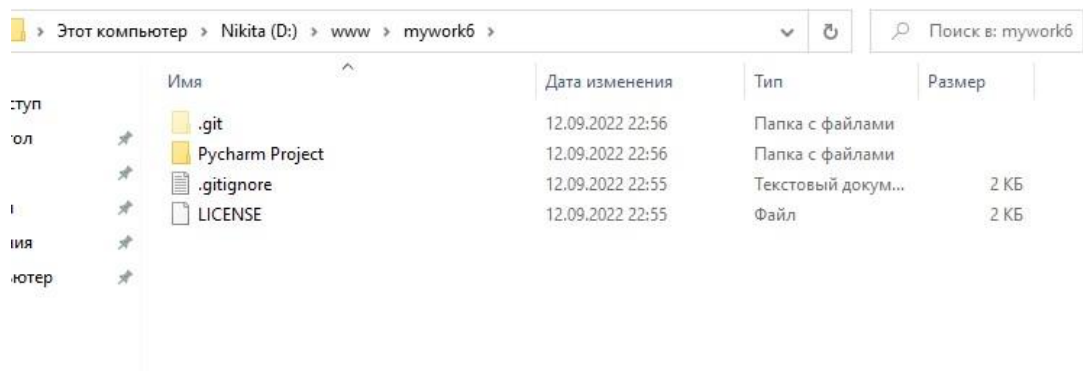


Рисунок 4 - Проект PyCharm

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

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

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:/www/mywork6/Pycharm Project/pr1.py"
Введите предложение: Вперед и с песней
Предложение после замены: {'Вперед_и_с_песней'}

Process finished with exit code 0
```

Рисунок 5 – Результат выполнения программы

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

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:/www/mywork6/Pycharm Project/pr2.py"
Введите слово: слон
Слн

Process finished with exit code 0
```

Рисунок 6 - Результат выполнения программы

Пример 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.

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:/www/mywork6/Pycharm Project/pr3.py"
Введите предложение: Угадай с двух раз
Введите длину: 24
Угадай с двух раз
Process finished with exit code 0
```

Рисунок 7 - Результат выполнения программы

#### 6. Выполните индивидуальные задания.

Задание 1: . Дано предложение. Определить, какая из букв – о или а – встречается в нем чаще (принять, что указанные буквы в строке есть).

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:/www/mywork6/Pycharm Project/ind1.py"
Введите предложение: Дай посмотреть
Запятых нет
Process finished with exit code 1
```

Рисунок 8 - Результат выполнения программы

Задание 2: Дано слово из 15 букв. Переставить в обратном порядке буквы, расположенные между k-й и s-й буквами (т. е. с (k+1)-й по (s-1)-ю).

Значения k и s вводятся с клавиатуры,  $k < s$ .

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:/www/mywork6/Pycharm Project/ind2.py"
Введите слово: Второй
['В', 'Т', 'О', 'Р', 'О', 'Й']
Результат работы программы: тВройо
Process finished with exit code 0
|
```

## Рисунок 9 - Результат выполнения программы

Задание 3: Дан текст. Напечатать все имеющиеся в нем цифры.

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:\www\mywork6\Pycharm Project\ind3.py"
Введите фразу: Сейчас слишком много времени
Чисел нет
I
Process finished with exit code 2
```

## Рисунок 10 - Результат выполнения программы

Задание повышенной сложности: Дана последовательность слов. Напечатать те слова последовательности, которые отличны от первого слова и удовлетворяют следующему свойству: ☐ В слове нет повторяющихся букв; ☐ слово симметрично.

```
"D:\www\mywork6\Pycharm Project\venv\Scripts\python.exe" "D:\www\mywork6\Pycharm Project\usl.py"
Введите первое предложение: I need more power
Введите второе предложение: Motivation
Слова удовлетворяющие условию: I need more power Motivation
Process finished with exit code 0
```

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

Вывод: приобрел навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.