



### Методические рекомендации по теме

«Символьные (строковые) переменные. Встроенные функции для обработки строк»

#### Цель:

- применение работы со строками и решение прикладных задач со строками на языке Python.

#### Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с основами работы со строками на языке Python;
- решение программных прикладных задач с работой со строками на языке Python;
- ранняя профориентация школьников, профессиональная деятельность программиста;
- развитие интеллектуальных способностей, логического и критического мышления

#### Планируемые результаты

*Личностные:* обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных задач и владение первичными навыками деятельностного анализа и критической оценки получаемой информации.

*Предметные:* обучающиеся получают представления: о типе данных string; об основных операциях со строками в языке программирования «Python»; о прикладном использовании операций со строками в программных проектах; о возможностях и особенностях применения операций со строками в практике работы программиста.

*Метапредметные:* обучающиеся получают возможность владения обще предметными понятиями «цикл», «условие», «условие в цикле»; владение информационно-логическими умениями; владение умениями самостоятельно

планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

### **Материалы к занятию**

Приложение 1: Сценарный план видеоролика

Приложение 2: Домашние задание и практика

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии

### **Ход проведения урока**

#### **1. Организационный момент.**

##### **Мотивация на учебную деятельность.**

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, что такое текстовый тип данных и как работать со строками на языке Python.; убедимся, что работа с текстовыми данными в Python не сводится к только к вводу и выводу информации, что также существует множество операторов, функций и методов для работы со строками).

##### ***Проблемная дискуссия*** по вопросам:

- Какие типы данных существуют в Python?
- Как обозначается текстовый тип данных в Python?
- Какие операции с текстовыми данными мы уже умеем делать?
- Подумайте, какие операции еще можно было бы делать с текстом при помощи программирования?
- Для чего, по вашему мнению, нужны операции со строками? Приведите примеры.

**Итоги дискуссии** (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- текстовые данные обозначаются в Python как **str**;
- с текстовыми данными можно делать такие базовые операции как сложение или умножение, но это далеко не все возможности;
- базовыми операциями со строками текстовых данных могут быть, к примеру операции форматирования текста.

Преподаватель называет ученикам тему и цели урока.

## **2. Вводный блок.**

### **Тема.**

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия)

*\*см. сцены 1 – 2 (здесь и далее приводится **Таблица «Содержание видеоролика». Приложение 1**)*

## **3. Блок повторения.**

### **Блиц-опрос.**

Преподаватель предлагает ученикам ответить на **5 вопросов** по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией **в сцене 3**; учитель должен убедиться, что всем понятна процедура голосования. *Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия*

*\*см. сцены 3 – 7*

## **4. Теоретический блок.**

### **Работа с текстами.**

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Для чего нужны методы **upper, lower, title**?
- Для каких жизненных задач нам могут пригодиться эти операции со строками?

*При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.*

*\*см. сцены 8 – 13*

## **5. Блок заданий.**

**Проекты: «Изменение регистра», «Ограничение ввода».**

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах.

**«Изменение регистра» и «Ограничение ввода»:** включает **2 практических задания** для учеников с последующим разбором. Задания представляют собой 2 небольших программных проекта с использованием методов работы со строками.

**После выполнения задания** ученики получают два работающих программных продукта – программа для регистра в тексте, программа, ограничивающая ввод информации определенным числом символов.

*На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.*

*\*см. сцены 14 – 23*

## **6. Рефлексия. Сообщение домашнего задания.**

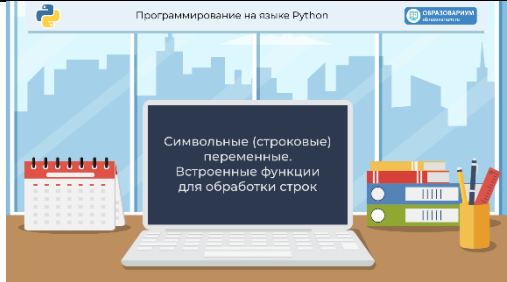
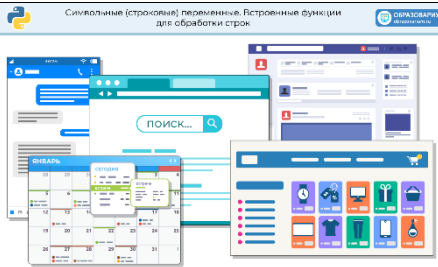
Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированному в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

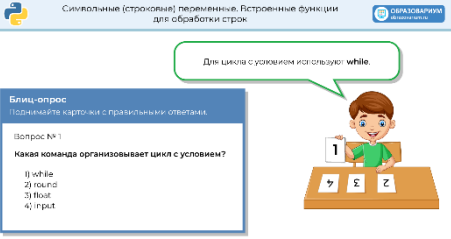
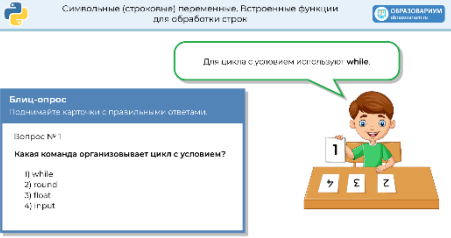
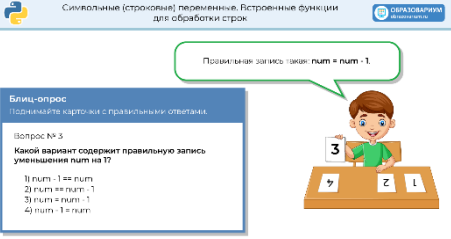
Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*). *\*см. сцена 24*

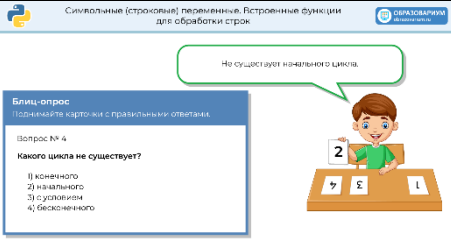
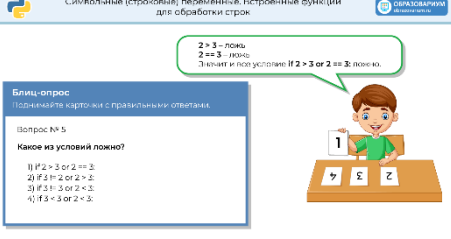
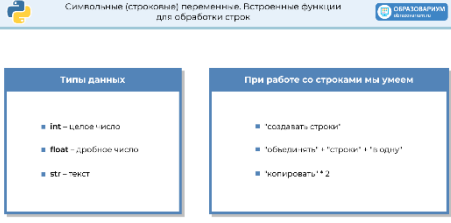
## Сценарный план видеоролика

В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

Таблица. Содержание видеоролика

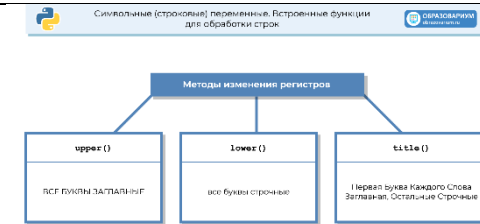
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Символьные (строковые) переменные. Встроенные функции для обработки строк.</p>	 <p>Сцена 1</p>	1 2
	<p>Редко можно встретить приложение, которое не использует строковый тип данных.</p> <p>Python обладает большим количеством операторов, функций и методов для работы со строками.</p> <p>Сегодня мы начнем с ними знакомство.</p>	 <p>Сцена 2</p>	

<p>Блок повторения.</p> <p><b>Блиц-опрос</b></p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p><b>Первый вопрос.</b> Какая команда организывает цикл с условием?</p> <ol style="list-style-type: none"> <li>1) while</li> <li>2) round</li> <li>3) float</li> <li>4) input</li> </ol> <p><i>Ответ 1. Для цикла с условием используется <b>while</b>.</i></p>	 <p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p> <p>Для цикла с условием используем: <b>while</b>.</p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос №1: Какая команда организует цикл с условием?</p> <ol style="list-style-type: none"> <li>while</li> <li>round</li> <li>float</li> <li>input</li> </ol> <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p><b>Второй вопрос.</b> При каком значении не будет работать цикл <b>while</b> num != 5: ?</p> <ol style="list-style-type: none"> <li>1) num = 4</li> <li>2) num = 5</li> <li>3) num = 6</li> <li>4) При любом</li> </ol> <p><i>Ответ 2. При num = 5 цикл не сработает.</i></p>	 <p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p> <p>Для цикла с условием используем: <b>while</b>.</p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос №1: Какая команда организует цикл с условием?</p> <ol style="list-style-type: none"> <li>while</li> <li>round</li> <li>float</li> <li>input</li> </ol> <p>Сцена 4</p>	
	<p><b>Третий вопрос.</b> Какой вариант содержит правильную запись уменьшения num на 1?</p> <ol style="list-style-type: none"> <li>1) num - 1 == num</li> <li>2) num == num - 1</li> <li>3) num = num - 1</li> <li>4) num - 1 = num</li> </ol> <p><i>Ответ 3. Правильная запись такая: num = num - 1.</i></p>	 <p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p> <p>Правильная запись такая: num = num - 1.</p> <p>Блиц-опрос: Поднимайте карточки с правильными ответами.</p> <p>Вопрос №3: Какой вариант содержит правильную запись уменьшения num на 1?</p> <ol style="list-style-type: none"> <li>num - 1 == num</li> <li>num == num - 1</li> <li>num = num - 1</li> <li>num - 1 = num</li> </ol> <p>Сцена 5</p>	

	<p><b>Четвертый вопрос.</b> Какого цикла не существует?</p> <ol style="list-style-type: none"> <li>1) конечный</li> <li>2) начальный</li> <li>3) с условием</li> <li>4) бесконечный</li> </ol> <p><i>Ответ 2. Не существует начального цикла.</i></p>	 <p>Сценарий 6</p>	
	<p><b>Пятый вопрос.</b> Какое из условий ложно?</p> <ol style="list-style-type: none"> <li>1) if <math>2 &gt; 3</math> or <math>2 == 3</math>:</li> <li>2) if <math>3 != 2</math> or <math>2 &gt; 3</math>:</li> <li>3) if <math>3 != 3</math> or <math>2 &lt; 3</math>:</li> <li>4) if <math>3 &lt; 3</math> or <math>2 &lt; 3</math>:</li> </ol> <p><i>Ответ 1. <math>2 &gt; 3</math> – ложь, <math>2 == 3</math> – ложь. Значит и все условие if <math>2 &gt; 3</math> or <math>2 == 3</math>: ложно.</i></p>	 <p>Сценарий 7</p>	
<p>Теоретический блок.</p> <p><b>Работа с текстами.</b></p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Нам уже известно, что в Python существует несколько типов данных. Для математических вычислений мы используем типы <b>int</b> и <b>float</b>, а для работы с текстами – <b>str</b>.</p> <p>Можно создавать строки самостоятельно или через функцию <b>input</b>, объединять строку с другими строками используя сложение, копировать текстовую информацию (умножая на целое число)</p>	 <p>Сценарий 8</p>	<p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p>

Для начала рассмотрим три метода, которые позволяют изменять регистр букв. Это методы: **upper()**, **lower()** и **title()**.

*Рассмотрите работу функции на примерах, анализируя отображения редактора и терминала.*



## Сцена 9

Если текстовая переменная уже создана, ей можно поменять регистр букв. Возьмем для примера переменную `текст`, в которой буквы разных регистров.

Если перезаписать переменную и добавить нужный код, то метод **lower** сделает их маленькими...

The screenshot shows a Jupyter Notebook interface. The top bar has the Google Colab logo and the text 'Симуляторы (страницы) переменные Встроенные функции для обработки строк'. On the right, there is a 'СЕРВИСЫ ГИТ' button with a GitHub logo. The notebook contains two code cells. The first cell defines a function:

```
def f(x, y):
    return x + y
```

The second cell calls the function and prints the result:

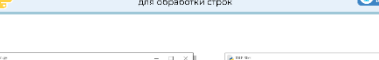
```
print(f(10, 20))
```

The output of the second cell is 30.

## Сцена 10

Метод **upper** сделает все буквы большими.

Обратите особое внимание на синтаксис: перед методом всегда ставится точка, а после него – пустые круглые скобки.



The screenshot shows a Jupyter Notebook with two cells. The first cell contains a function definition:

```
def say_hello(name):
    print('Hello, ' + name + '!')
```

The second cell contains a function call:

```
say_hello('John')
```



The output of the second cell is:


```
Hello, John!
```

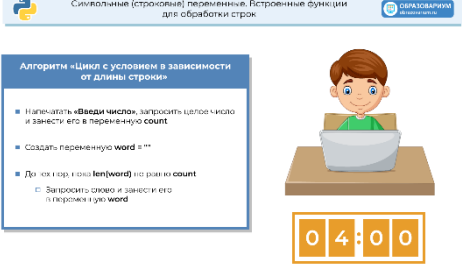
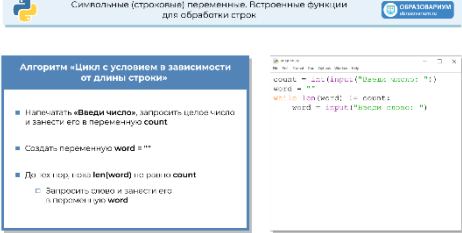
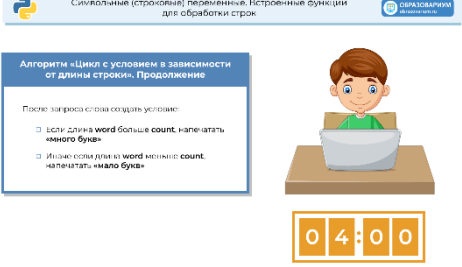
On the right side of the notebook, there is a sidebar with a search icon and a link to 'Создайте свой проект' (Create your project).


## Сцена 11

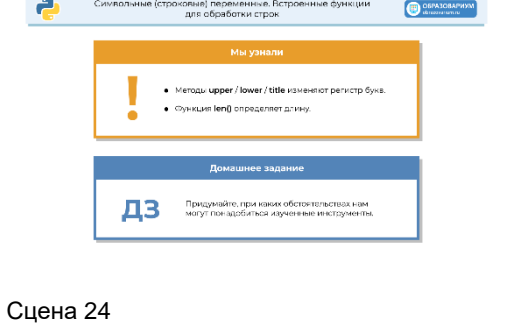


	<p>А метод <b>title</b> первую букву каждого слова сделает заглавной, а остальные — строчными.</p>	<div><div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div><pre>text = "one two three four five" text = text.title() print(text)</pre></div><div><pre>&gt;&gt;&gt; one Two three Four five &gt;&gt;&gt;</pre></div></div><div><div>Сцена 12</div><div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div><pre>name = input("Введите имя: ") print(name)</pre></div><div><pre>&gt;&gt;&gt; Илья &gt;&gt;&gt;</pre></div></div></div></div>	
	<p>Если переменная еще не создана, тогда эти методы можно применить вместе с функцией <b>input</b>.</p> <p><b>Пример.</b> Нам надо создать переменную <b>name</b> и занести туда имя. По правилам орфографии имя надо писать с большой буквы. Но пользователь может это правило проигнорировать. Сделаем это за него. В конце строки поставим точку и припишем метод <b>title</b> с пустыми круглыми скобками. Выведем переменную, чтобы проверить программу. Теперь введенное имя станет с большой буквы.</p>	<div><div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div><pre>name = input("Введите имя: ").title() print(name)</pre></div><div><pre>&gt;&gt;&gt; илья &gt;&gt;&gt; Илья &gt;&gt;&gt;</pre></div></div><div><div>Сцена 13</div><div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div><div>Алгоритм «upper – lower – title»</div><ul style="list-style-type: none"><li>■ Запросить название города и занести ответ в переменную <b>city</b></li><li>■ Запросить название страны и занести ответ в переменную <b>country</b></li><li>■ Запросить название дня и занести ответ в переменную <b>day</b></li><li>■ Добавить в каждую строку ввода свой метод, чтобы получилось так: Город, СТРАНА, день</li><li>■ Вывести переменные на экран</li></ul></div><div><div>04:00</div></div></div></div></div>	
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 1</p> <p>Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p><b>Задание 1. Проект «upper – lower – title»</b></p> <ul style="list-style-type: none"><li>– Запросить название города и занести ответ в переменную <b>city</b></li><li>– Запросить название страны и занести ответ в переменную <b>country</b></li><li>– Запросить название дня и занести ответ в переменную <b>day</b></li><li>– Добавить в каждую строку ввода свой метод, чтобы получилось так:</li><li>– Город, СТРАНА, день</li><li>– Вывести переменные на экран</li></ul>	<div><div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div><div>Алгоритм «upper – lower – title»</div><ul style="list-style-type: none"><li>■ Запросить название города и занести ответ в переменную <b>city</b></li><li>■ Запросить название страны и занести ответ в переменную <b>country</b></li><li>■ Запросить название дня и занести ответ в переменную <b>day</b></li><li>■ Добавить в каждую строку ввода свой метод, чтобы получилось так: Город, СТРАНА, день</li><li>■ Вывести переменные на экран</li></ul></div><div><div>04:00</div></div></div><div><div>Сцена 14</div></div></div>	<p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p> <p>19</p> <p>20</p> <p>21</p> <p>22</p> <p>23</p>

<p><b>Разбор задания 1.</b></p> <pre>city = input("Введи город: ").title() country = input("Введи страну: ").upper() day = input("Введи день: ").lower()  print(city, country, day)</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<div data-bbox="1480 228 1962 478"><p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p><div data-bbox="1503 284 1733 472"><p>Алгоритм «upper – lower – title»</p><ul style="list-style-type: none"><li>■ запросить название города и занести ответ в переменную <code>city</code></li><li>■ Запросить название страны и занести ответ в переменную <code>country</code></li><li>■ Запросить название дня и занести ответ в переменную <code>day</code></li><li>■ Добавить в каждую строку ввода свой метод, чтобы получилось так: Город, СТРАНА, день</li><li>■ Вывести переменные на экран</li></ul></div><div data-bbox="1742 284 1962 472"><pre>city = input("Введи город: ").title() country = input("Введи страну: ").upper() day = input("Введи день: ").lower()  print(city, country, day)</pre></div></div> <p>Сцена 15</p>
<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 1. Проект «upper – lower – title» (продолжение)</b></p> <p>Теперь рассмотрим функцию <code>len()</code>, которая помогает узнать длину строки или текстовой переменной. Доработаем проект:</p> <ul style="list-style-type: none"><li>– Добавить в проект строку: <code>print(len(city), len(country), len(day))</code></li></ul>	<div data-bbox="1480 608 1962 879"><p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p><div data-bbox="1503 663 1733 852"><p>Функция <code>len()</code> считает длину!</p><p>Алгоритм «upper – lower – title». Дополнение</p><ul style="list-style-type: none"><li>■ Добавить в проект строку: <code>print(len(city), len(country), len(day))</code></li></ul></div><div data-bbox="1783 663 1935 879"></div></div> <p>Сцена 16</p>
<p><b>Разбор задания 1 (продолжение).</b></p> <p>Запустим проект и посмотрим, что получилось.</p> <p>В последней строке вывелось три числа, каждое из которых показывает, сколько букв содержится в наших переменных.</p> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<div data-bbox="1480 943 1962 1190"><p>Символьные (строковые) переменные. Встроенные функции для обработки строк</p><div data-bbox="1503 999 1720 1187"><pre>city = input("Введи город: ").title() country = input("Введи страну: ").upper() day = input("Введи день: ").lower()  print(city, country, day)</pre></div><div data-bbox="1742 999 1962 1187"><pre>city = input("Введи город: ").title() country = input("Введи страну: ").upper() day = input("Введи день: ").lower()  print(len(city), len(country), len(day))</pre></div></div> <p>Сцена 17</p>

	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p><b>Задание 2. Алгоритм «Цикл с условием в зависимости от длины строки»</b></p> <p>Напечатать «Введи число», запросить целое число и занести его в переменную count</p> <p>Создать переменную word = ""</p> <p>До тех пор, пока len(word) не равно count</p> <p>Запросить слово и занести его в переменную word.</p>	 <p>Сцена 18</p>	
	<p><b>Разбор задания 2.</b></p> <p>Код программы может выглядеть так.</p> <pre>count = int(input("Введи число: ")) word = "" while len(word) != count:     word = input("Введи слово: ")</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	 <p>Сцена 19</p>	
	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p><b>Задание 2. Продолжение</b></p> <p>После запроса слова создать условие:</p> <p><b>Если</b> длина word больше count, напечатать «много букв»</p> <p><b>Иначе</b> если длина word меньше count, напечатать «мало букв»</p>	 <p>Сцена 20</p>	

	<p><b>Разбор задания 2. Продолжение</b></p> <p>Код программы может выглядеть так.</p> <pre>If len(word) &gt; count:     print("много букв") elif len(word) &lt; count:     print("мало букв")</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div>Алгоритм «Цикл с условием в зависимости от длины строки». Продолжение</div><div>После запроса слова создать условие:<ul style="list-style-type: none"><li>Если длина word больше count, напечатать «много букв»</li><li>Иначе если длина word меньше count, напечатать «мало букв»</li></ul></div><div><pre>word = input("Введите слово: ") count = 10 if len(word) &gt; count:     print("много букв") elif len(word) &lt; count:     print("мало букв")</pre></div><div>Почему нам нельзя было после условия использовать оператор else?</div><div>Он бы сработал также и когда количество букв равно count</div></div> <p>Сцена 21</p>	
	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 2. Окончание</b></p> <p>После цикла: <b>напечатать</b> введенное слово сначала маленькими, а потом большими буквами</p>	<div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div>Алгоритм «Цикл с условием в зависимости от длины строки». Продолжение</div><div>После цикла:<ul style="list-style-type: none"><li>напечатать введенное слово сначала маленькими, а потом большими буквами</li></ul></div><div></div></div> <p>Сцена 22</p>	
	<p><b>Разбор задания 2. Окончание</b></p> <p>Код программы может выглядеть так.</p> <pre>print(word.lower()) print(word.upper())</pre> <p><i>После окончания времени провести разбор решения, анализируя код.</i></p>	<div><div>Символьные (строковые) переменные. Встроенные функции для обработки строк</div><div>Алгоритм «Цикл с условием в зависимости от длины строки». Продолжение</div><div>После цикла:<ul style="list-style-type: none"><li>напечатать введенное слово сначала маленькими, а потом большими буквами</li></ul></div><div><pre>word = input("Введите слово: ") count = 10 if len(word) &gt; count:     print("много букв") elif len(word) &lt; count:     print("мало букв")  print(word.lower()) print(word.upper())</pre></div></div> <p>Сцена 23</p>	

Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги:</b></p> <ul style="list-style-type: none"> <li>- узнали, что методы <b>upper / lower / title</b> изменяют регистр букв;</li> <li>- функция <b>len()</b> определяет длину.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 24</p>	24
---	--	---	----

## Приложение 2

### Домашнее задание

Придумать еще несколько ситуаций в которых нам могут понадобиться инструменты, изученные на занятии.

### Практика

Проект «Изменение текста»

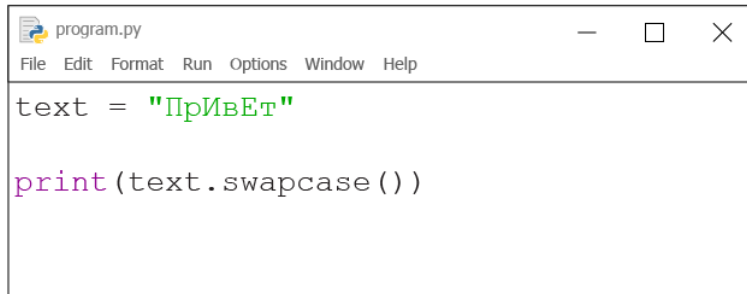
Запросите у пользователя ввод текстового фрагмента.

Выведите предложение по режиму работы:

- 1 – перевести текст в верхний регистр,
- 2 – перевести текст в нижний регистр,
- 3 – поменять регистры местами,
- 0 – закончить работу.

Запросите у пользователя желаемый режим и выведите в соответствии с ним текст. Повторяйте запрос о режиме работы до тех пор, пока пользователь не введет 0.

Для того, чтобы поменять регистры местами используйте метод **swapcase()**.



```
program.py
File Edit Format Run Options Window Help
text = "ПрИвЕт"
print(text.swapcase())
```

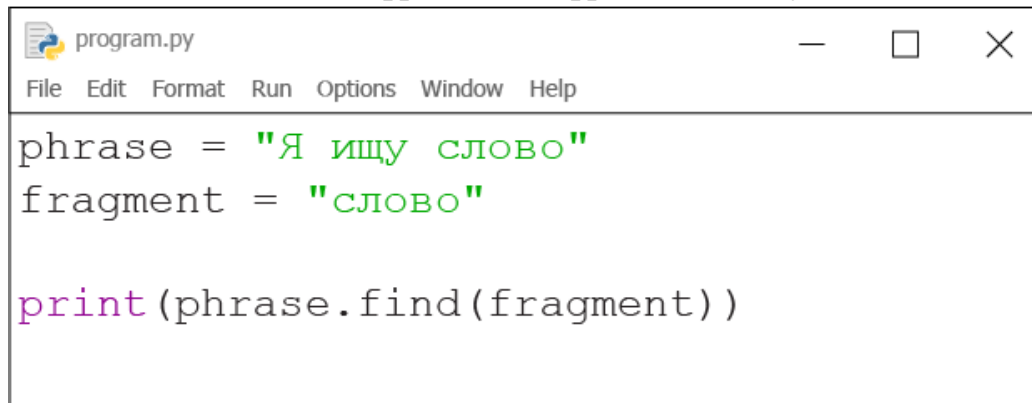
Такой код выведет **"пРиВеТ"**.

### Проект «Поиск в строке»

Запросите у пользователя ввод фразы и фрагмента, который нужно найти в этой фразе.

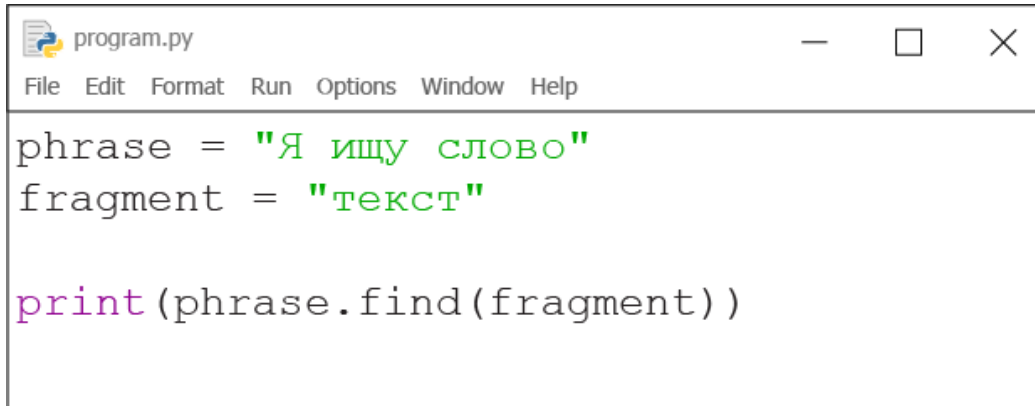
Выведите номер символа, с которого начинается искомый фрагмент в исходной фразе. Если фрагмент не найден, выведите соответствующее сообщение.

Для того, чтобы найти фрагмент в фразе используйте метод **find()**.



```
program.py
File Edit Format Run Options Window Help
phrase = "Я ищу слово"
fragment = "слово"
print(phrase.find(fragment))
```

Такой код выведет **6** – индекс символа, с которого начинается фрагмент (**fragment**) в фразе (**phrase**).



```
program.py
File Edit Format Run Options Window Help

phrase = "Я ищу слово"
fragment = "текст"

print(phrase.find(fragment))
```

Такой код выведет **-1**. Это значит, что фрагмент (**fragment**) не найден в фразе (**phrase**).

### Приложение 3

#### Краткие организационно-методические рекомендации по организации работы на занятии

«Символьные (строковые) переменные. Встроенные функции для обработки строк».

**В начале занятия** необходимо вспомнить: что такое цикл, каким образом осуществляется условие в циклах и для чего это необходимо. А также проверить домашнее задание – собственный циклический алгоритм с условием.

Также нужно вспомнить чем условие в циклах отличается от условных операторов (**if, elif, else**).

**Перед просмотром блока повторения** из ролика необходимо раздать дидактический материал для выполнения заданий из блока повторение (по 4 пронумерованных карточки)

Во время голосований карточками можно останавливать ролик и вести учет правильных ответов. По окончании блока – отметить тех, у кого наилучший результат.

Далее карточки необходимо собрать.

*После теоретического блока* можно остановить ролик и вспомнить какие типы данных ребята помнят и привести примеры операций с текстовыми переменными (сложение с другим текстом, умножение на число). А также – чего нельзя делать с текстовыми переменными.

Особое внимание уделите тому, что команда **while** не требует дополнительных условных операторов. А также тому, что при формулировке условия желательно проговаривать его как «**до тех пор, пока...**»

Перед выполнением *задания 1* можно написать на доске название методов (**upper, lower, title**) и что они делают. Аналогичным образом можно поступить с функцией **len** и заодно рассказать, что ее название происходит от английского слова **length** – длина.

*Задание 2* выполняется в три этапа. После написания первой части программы - дайте ребятам протестировать код, прежде чем переходить к следующему шагу. Обратите внимание, что вторая часть проекта располагается внутри созданного нами цикла, в отличие от третьей части, которая срабатывает только по завершению цикла при срабатывании условия.