



### Методические рекомендации по теме «Игра «Быки и коровы». Обработка данных»

#### Цель:

- дать представление о прикладном применении множеств в программировании при создании компьютерных игр на языке Python.

#### Задачи:

- практика применения множеств при программировании игр в Python;
- понимание множеств в программировании как основы игрового проекта;
- анализ программного кода с целью определения, что выведет программа при конкретных исходных данных;
- исправление ошибок и дописывание программного кода;
- написание программного кода.

#### Планируемые результаты

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

*Предметные:* обучающиеся получают представления об использовании возможностей множеств в Python при разработке игр.

*Метапредметные:* обучающиеся получают возможность владения общепредметными понятиями «список», «множество»; информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей;

владения умениями принятия решений и осуществления осознанного выбора; повышения уровня ИКТ – компетентности и расширение кругозора в области информатики и программирования; знакомство с профессиональной деятельностью программиста в рамках ранней профориентации; развитие интеллектуальных способностей, а также логического и критического мышления.

### **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.

#### **2. Вводный блок.**

**Тема.**

**Играем в Быки и Коровы:**

Чемпионат по «Быки и Коровы»:

Преподаватель напоминает правила игры «Быки и Коровы» и предлагает детямделиться на команды устроить чемпионат по этой игре с учетом той стратегии, которую они разработали, практикуясь в игре в качестве домашнего задания.

*\*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.

#### Игра «Быки и коровы»

Продолжим написание кода игры, поэтому сначала откроем свой проект с предыдущего занятия.

Теперь наша задача – сравнить между собой два этих списка для обнаружения совпадений. Поскольку у всех списков имеется индекс, то прибегнем к помощи циклов. Нам понадобится сравнить каждую ячейку первого списка с каждой ячейкой второго списка. Для этого организуем два цикла: внешний и внутренний с разными переменными счетчиками.

В сумме у нас получится **16 вариантов** сравнения.

Далее разработаем альтернативный вариант кода, который будет короче, и работать будет быстрее.

На финише удаляем строку-подсказку.

*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. сцена 8-9, 12, 17*

### 5. Блок заданий.

#### Проект: «Быки и коровы». Окончание.

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

**«Быки и коровы»:** включает *практическое задание 1*: организация двух циклов, вложенных один в другой, с помощью которых можно перебрать все возможные варианты совпадения ячеек в обоих списках; завершение работы циклов, в случае если угадано нужно число, вывести количество сделанных попыток.

*практическое задание 2*: написание альтернативного варианта, для перебора вариантов с большим быстродействием и более лаконичный.

**После выполнения задания** ученики получают работающий программный продукт – программа игрового проекта «Быки и коровы» для игры с компьютером.

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

*\*см. сцены 10 – 17 (кроме сцен по теории).*

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

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным 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 18*

## **Приложение 1**

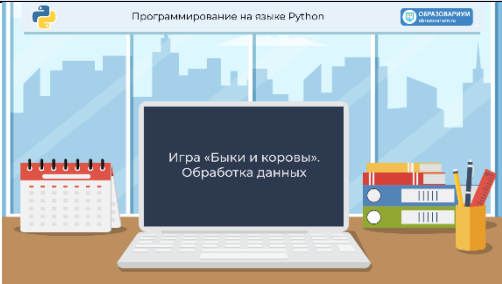
### **Сценарный план видеоролика**

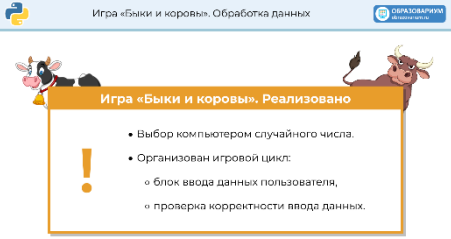
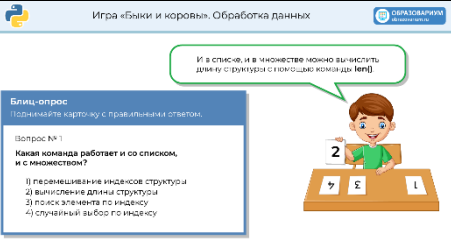
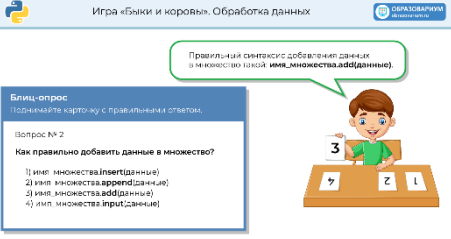
В таблице «Содержание видеоролика» представлены:

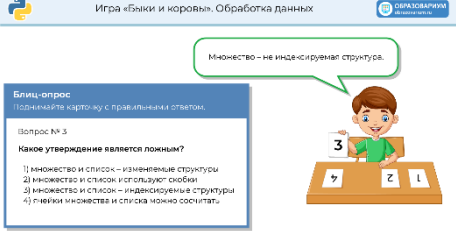
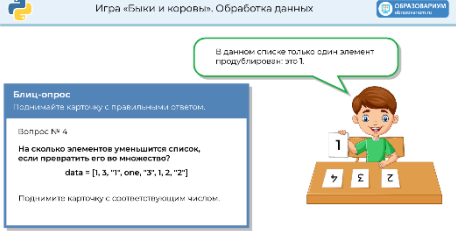
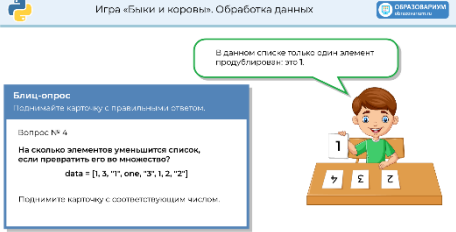
- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.

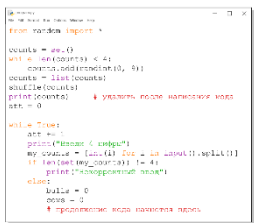
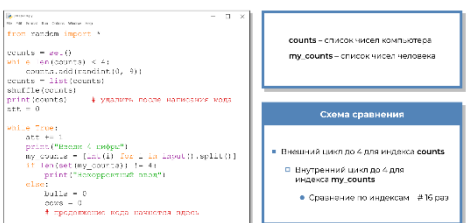
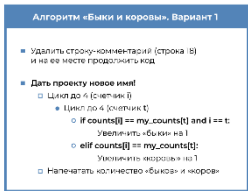

*Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.*

Таблица. Содержание видеоролика

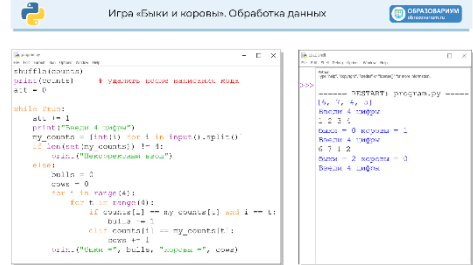
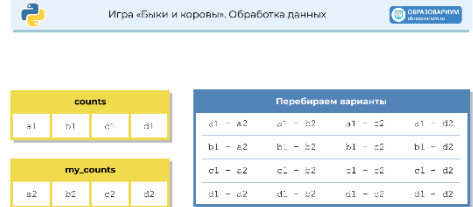
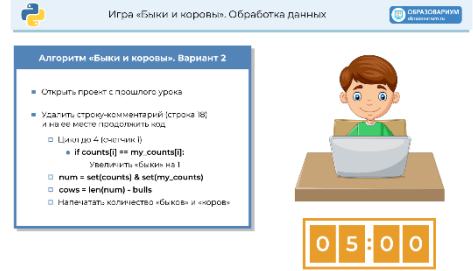
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Игра «Быки и коровы». Обработка данных	 <p>Сцена 1</p>	1 2

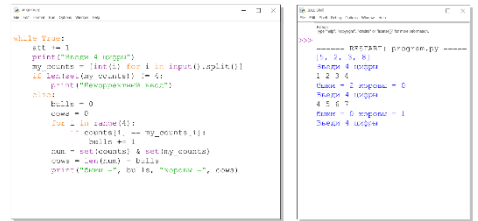
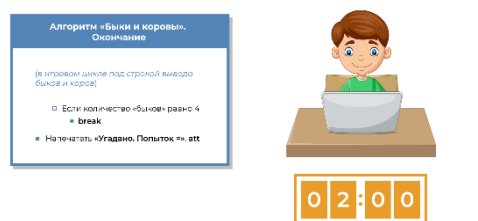
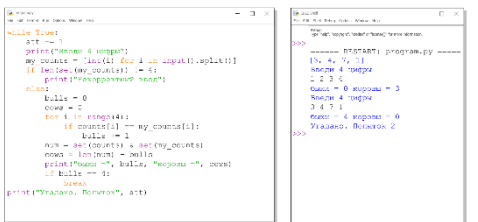
	<p>На прошлом занятии мы начали работу над игровым проектом.</p> <p>Написали блок выбора компьютером случайного псевдо-числа (в виде четырех отдельных чисел), а также организовали игровой цикл, а внутри создали блок ввода данных пользователя с проверкой на корректность.</p> <p>Теперь нам предстоит закончить этот проект.</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 работает и со списком, и с множеством?</p> <ol style="list-style-type: none"> <li>1) перемешивание индексов структуры</li> <li>2) вычисление длины структуры</li> <li>3) поиск элемента по индексу</li> <li>4) случайный выбор по индексу</li> </ol> <p><i>Ответ 2. И в списке, и в множестве можно вычислить длину структуры с помощью команды <b>len()</b>.</i></p>	 <p>Сцена 3</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p><b>Второй вопрос.</b> Как правильно добавить данные в множество?</p> <ol style="list-style-type: none"> <li>1) имя_множества.insert()</li> <li>2) имя_множества.append()</li> <li>3) имя_множества.add()</li> <li>4) имя_множества.input()</li> </ol> <p><i>Ответ 3. Правильный синтаксис добавления данных в множество такой: имя_множества.add()</i></p>	 <p>Сцена 4</p>	

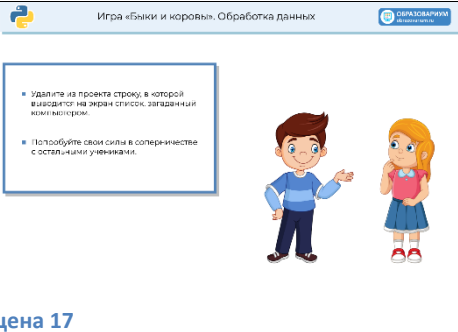
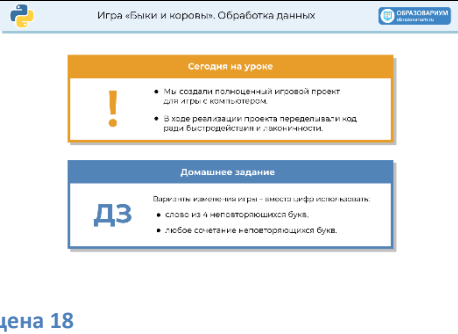
	<p><b>Третий вопрос.</b> Какое утверждение является ложным?</p> <ol style="list-style-type: none"> <li>1) множество и список – изменяемые структуры</li> <li>2) множество и список использует скобки</li> <li>3) множество и список – индексируемые структуры</li> <li>4) ячейки множества и списка можно сосчитать</li> </ol> <p><i>Ответ 3. Множество – не индексируемая структура.</i></p>	 <p>Игра «Бъки и коровы». Обработка данных</p> <p>Множество – не индексируемая структура.</p> <p>Блиц-опрос Поднимайте карту, жу с правильными ответом.</p> <p>Вопрос № 3 Какое утверждение является ложным? 1) множество и список – изменяемые структуры 2) множество и список использует скобки 3) множество и список – индексируемые структуры 4) ячейки множества и списка можно сосчитать.</p>	
	<p><b>Четвертый вопрос.</b> На сколько элементов уменьшится список, если превратить его во множество?</p> <p><code>data = [1, 3, "1", one, "3", 1, 2, "2"]</code></p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 1. В данном списке только один элемент продублирован: это 1.</i></p>	 <p>Игра «Бъки и коровы». Обработка данных</p> <p>В данном списке только один элемент продублирован: это 1.</p> <p>Блиц-опрос Поднимайте карту, жу с правильными ответом.</p> <p>Вопрос № 4 На сколько элементов уменьшится список, если превратить его во множество? <code>data = [1, 3, "1", one, "3", 1, 2, "2"]</code></p> <p>Поднимите карточку с соответствующим числом.</p>	Сцена 5
	<p><b>Пятый вопрос.</b> Какие индексы у числа 7?</p> <p><code>[[1, 2], [3], [4, 5, 6, 7], [8]]</code></p> <p>Поднимите карточки с соответствующими числами.</p> <p><i>Ответ 2 и 3. Индексами числа 7 будут [2][3].</i></p> <p><code>[[1, 2], [3], [4, 5, 6, 7], [8]]</code></p> <p>0 1 2 3</p>	 <p>Игра «Бъки и коровы». Обработка данных</p> <p>В данном списке только один элемент продублирован: это 1.</p> <p>Блиц-опрос Поднимайте карту, жу с правильными ответом.</p> <p>Вопрос № 4 На сколько элементов уменьшится список, если превратить его во множество? <code>data = [1, 3, "1", one, "3", 1, 2, "2"]</code></p> <p>Поднимите карточку с соответствующим числом.</p>	Сцена 6
		Сцена 7	

<p>Теоретический блок.</p> <p><b>Игра «Быки и коровы».</b> Обработка данных</p>	<p>Продолжим написание кода игры, поэтому сначала откроем свой проект с предыдущего занятия. Он должен иметь следующий вид.</p> <p>В строках 1-8 подключается модуль <b>random</b>, создается множество из четырех чисел. Потом множество преобразуется в перемешанный список. И создается переменная, которая будет подсчитывать попытки (<i>строка 7 – подсказка</i>)</p> <p>Начиная со строки 9 - игровой цикл, внутри которого уже имеется блок ввода данных пользователем – строка 11 и 12, и блок их проверки на правильность, который представлен условной конструкцией <b>If - else</b></p>	 <p>Сцена 8</p>	<p>8 9</p>
	<p>В итоге мы имеем два списка: <b>counts</b> – числа, загаданные компьютером и <b>my_counts</b> – числа, которые назвал пользователь.</p> <p>Теперь наша задача – сравнить между собой два этих списка для обнаружения совпадений. Поскольку у всех списков имеется индекс, то прибегнем к помощи циклов. Нам понадобится сравнить каждую ячейку первого списка с каждой ячейкой второго списка. Для этого организуем два цикла: внешний и внутренний с разными переменными счетчиками.</p> <p>В сумме у нас получится 16 вариантов сравнения</p>	 <p>Сцена 9</p>	
<p>Блок заданий.</p> <p><b>Практические задания:</b> Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 6 мин.</i></p> <p><b>Задание 1. Алгоритм «Быки и коровы». Вариант 1</b></p> <ul style="list-style-type: none"> <li>Удалить строку-комментарий (строка 18) и на ее месте продолжить код.</li> <li>Дать проекту новое имя!</li> <li>Цикл до 4 (счетчик i)</li> <li>Цикл до 4 (счетчик t)</li> <li><b>if counts[i] == my_counts[t] and i == t:</b></li> <li>Увеличить «быки» на 1</li> <li><b>elif counts[i] == my_counts[t]:</b></li> <li>Увеличить «коровы» на 1</li> <li>Напечатать количество «быков» и «коров»</li> </ul>	  <p>Сцена 10</p>	<p>10 11</p>



	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.  <i>...дописывается к существующему коду</i></p> <pre> for i in range(4):     for t in range(4):         if counts[i] == my_counts[t] and i == t:             bulls += 1         elif counts[i] == my_counts[t]:             cows += 1 print("быки =", bulls, "коровы =", cows) </pre>	 <p>Сцена 11</p>	
<p>Теоретический блок.</p>	<p>Подведем предварительный итог. При помощи двух циклов, вложенных один в другой, нам удалось перебрать все возможные варианты совпадения ячеек в обоих списках. В целом у нас получилось 16 проверок.</p> <p>Данный тип перебора вариантов, по компьютерным меркам, очень долгий. Давайте сделаем альтернативный вариант кода, который будет и короче, и работать будет быстрее.</p>	 <p>Сцена 12</p>	12
<p>Блок заданий.  <b>Практические задания:</b>  Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 2. Алгоритм «Быки и коровы». Вариант 2</b></p> <ul style="list-style-type: none"> <li>Удалить строку-комментарий (строка 18) и на ее месте продолжить код. <ul style="list-style-type: none"> <li>Цикл до 4 (счетчик i) <ul style="list-style-type: none"> <li><b>if counts[i] == my_counts[i]:</b>  Увеличить «быки» на 1</li> </ul> </li> <li><b>num = set(counts) &amp; set(my_counts)</b></li> <li><b>cow = len(num) - bul</b></li> <li>Напечатать количество «быков» и «коров»</li> </ul> </li> </ul>	 <p>Сцена 13</p>	13 14

	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.  <i>... дописывается к существующему коду</i></p> <pre> for i in range(4):     if counts[i] == my_counts[i]:         bulls += 1 num = set(counts) &amp; set(my_counts) cows = len(num) - bulls print("быки =", bulls, "коровы =", cows) </pre>	 <p>Сцена 14</p>	
<p>Блок заданий.  <b>Практические задания:</b>  Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 1. Алгоритм «Быки и коровы». Окончание</b>  <i>* Перейдем в игровой цикл, под строку вывода быков и коров</i></p> <ul style="list-style-type: none"> <li>○ Если быки = 4 <ul style="list-style-type: none"> <li>• break</li> </ul> </li> <li>■ Напечатать «Угадано. Попыток =», att</li> </ul>	 <p>Сцена 15</p>	<p>15 16</p>
	<p><b>Разбор задание 1.</b> Код программы будет выглядеть так:  <i>... дописывается к существующему коду</i></p> <pre> if bul == 4:     break print("Угадано. Попыток", att) </pre>	 <p>Сцена 16</p>	

<p>Теоретический блок.</p>	<p>Осталось последнее – удалить строку-подсказку из нашего кода. Она имеет предположительно номер семь.</p> <p>Теперь наш код полностью готов, и мы можем провести соревнование: кому потребуется наименьшее количество попыток.</p>		<p>17</p>
<p>Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b></p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги.</b></p> <p>Мы создали:</p> <ul style="list-style-type: none"> <li>■ полноценный игровой проект для игры с компьютером;</li> <li>■ в ходе реализации проекта редактировали код ради быстродействия и лаконичности.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>		<p>18</p>

## Приложение 2

### Домашнее задание

Поиграйте в игру «Быки и Коровы» с друзьями, родителями, чтобы лучше понять механику и стратегию игры.

*Задание можно выполнить на компьютере и представить результат и код в виде файла или снимка экрана, или распечатки.*

## Практика

Проект «Словарик программиста. Перемешивание»

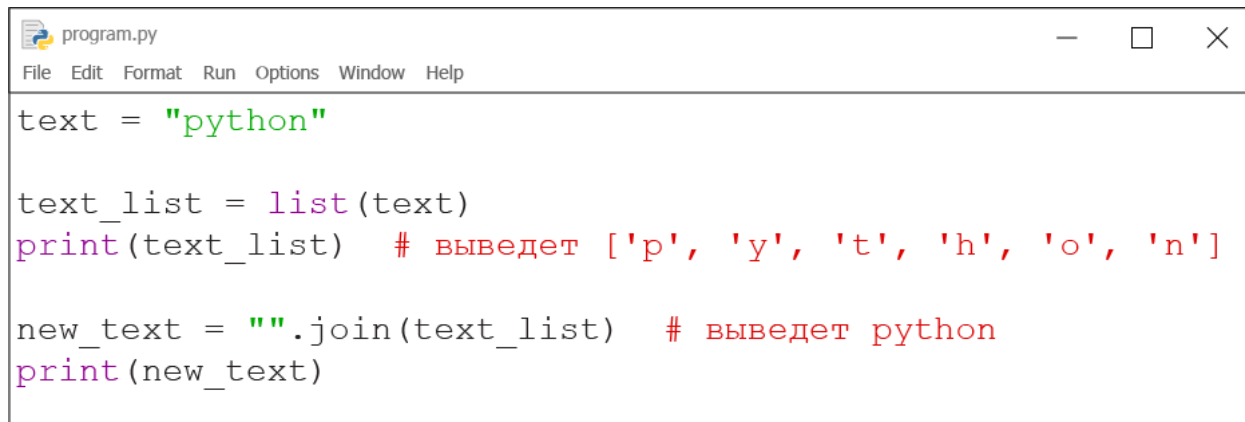
Цель проекта: сделать игру, в которой нужно будет отгадать зашифрованное слово.

Создайте список терминов, которые вы освоили на этом курсе. Случайным образом выберите одно слово из этого списка, перемешайте буквы в слове и выведите полученный результат на экран.

Попросите пользователя отгадать зашифрованное слово. Продолжайте до тех пор, пока слово не будет отгадано или пользователь не сдастся. Подсчитайте и выведите на экран, какое количество попыток предпринял пользователь, чтобы отгадать слово.

Вы можете усовершенствовать проект, сделав его многократным: чтобы пользователю поочередно предлагалось угадать несколько слов из списка.

Для того, чтобы превратить слово в список используйте команду **list**. А для того, чтобы список из букв объединить в одно слово используйте команду **join**.



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

text = "python"

text_list = list(text)
print(text_list)  # выведет ['p', 'y', 't', 'h', 'o', 'n']

new_text = "".join(text_list)  # выведет python
print(new_text)
```

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

«Игра «Быки и коровы». Обработка данных».

**В начале урока** необходимо повторить материал по теме «Множества». Лучше это делать в сравнении со списками. Для наглядности можно использовать доску, на которой в виде таблицы отмечать общие и различные методы свойства. Также напомните ребятам о том, что в коде встречается генератор списка – «волшебная строка», которая заменяет несколько строк кода.

**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.

**Перед началом работы**, убедитесь, что у всех загружены проекты, которые были начаты на прошлом уроке. На уроке будет сделано два варианта кода, поэтому сохраняйте код под новым именем, чтобы иметь возможность еще раз загрузить исходный проект. Сравнение списков предлагается в двух вариантах: через два цикла, вложенных друг в друга путем перебора ячеек по индексам. Это классический, но долгий способ. Второй вариант – используя свойства множеств (конъюнкция), в которые можно преобразовать списки. По желанию, можно сделать ограничение попыток на игру (например- 20), после чего цикл прерывается и выводится загаданное число.