



### Методические рекомендации по теме

«Работа со списками, перемешивание элементов. Игра «Угадай столицу»»

#### Цель:

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

#### Задачи:

- практика применения сложных списков при программировании игр в Python;
- понимание сложного списка 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 сложных списков в Python при разработке игр.

*Метапредметные:* обучающиеся получают возможность владения общепредметными понятиями «список», «индекс», «сложные список», «сортировка»; 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.

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

- Какую механику могла бы иметь игра «Угадай столицу»?
- Как в рамках этой механики была бы организована работа со сложными списками?

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

- Определена возможная общая механика игры «Угадай столицу»
- Определены возможности использования списков для игр типа «Викторина»

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.**

### **Игра «Угадай столицу».**

Новый материал излагается в сопровождении видеоролика, рекомендуется разместить на доске или флип-чарте изображения объектов, сопровождающих материалы по теме.

Обсуждением вопросов по просмотренным материалам:

- В чем смысл проекта?
- Какова механика работы проекта?
- Как задействованы списки в игре?

*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*

## **5. Блок заданий.**

### **Проект: «Угадай столицу».**

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

**«Угадай столицу»:** включает *практическое задание 1* которое выполняется в несколько этапов: создание списка «страна-столица» и применить команду перемешивания списка; организовать цикл с условием «пока» на длину списка и организовать вывод информации названия страны и прием ответа информации названия столицы; организовать проверку полученного ответа и удаление пары страна-столица после этого и увеличением счетчика правильных ответов.

**После выполнения задания** ученики получают работающий программный продукт – программа вида викторина, позволяющая проверить знание столиц разных стран с выводом количества правильных ответов.

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

*\*см. сцены 10 – 20.*

## **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 21*

## Приложение 1

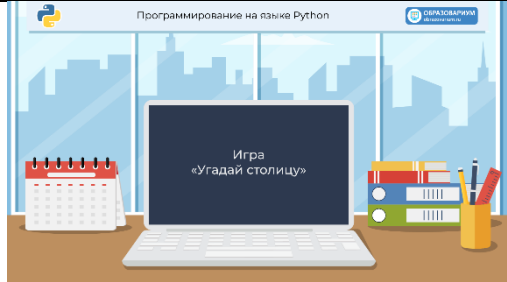
### Сценарный план видеоролика

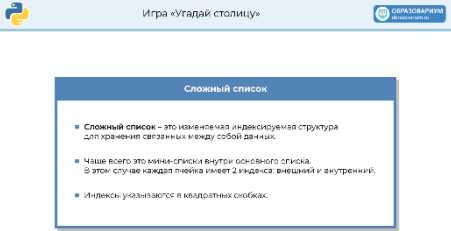
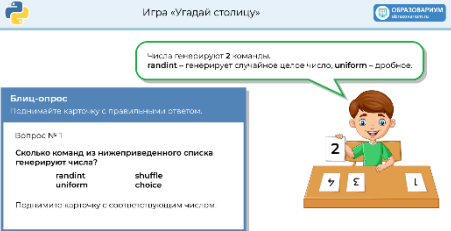
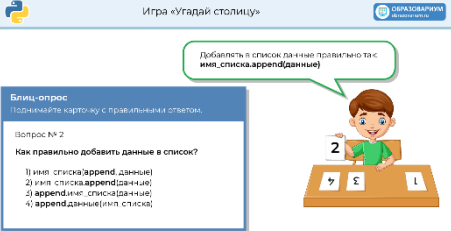
В таблице «Содержание видеоролика» представлены:

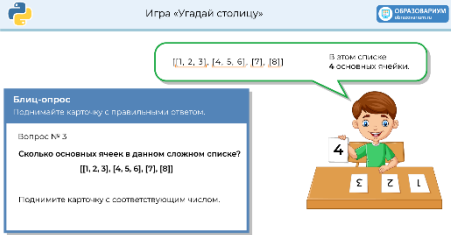
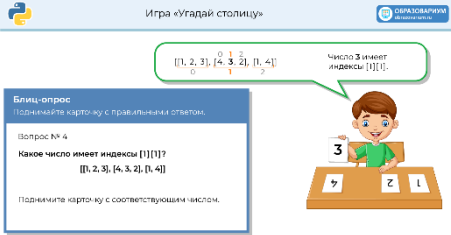
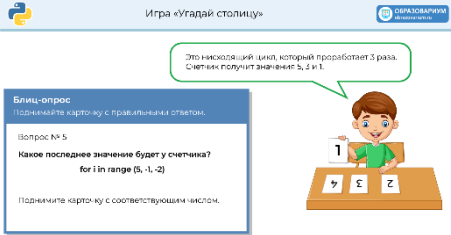
- название блоков видеоролика (тайминг);
- краткое описание содержания в каждом блоке;
- фрагменты из видеоролика, относящиеся к соответствующему блоку;
- номера сцен в каждом блоке.


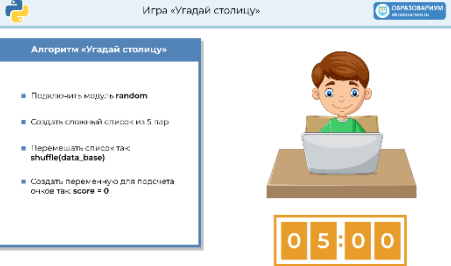
*Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного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

Таблица. Содержание видеоролика

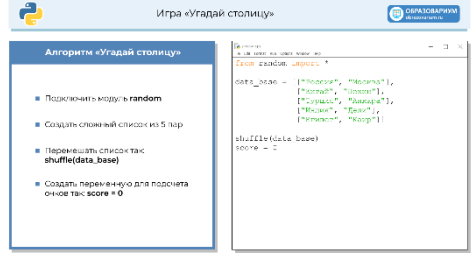
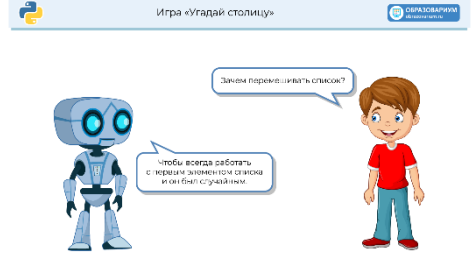
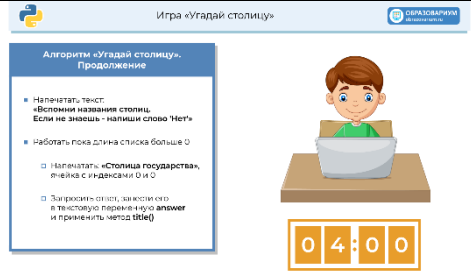
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Работа со списками, перемешивание элементов. Игра «Угадай столицу»	 Сцена 1	1 2

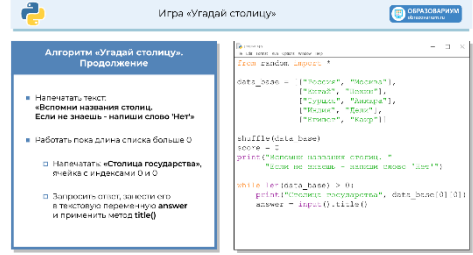
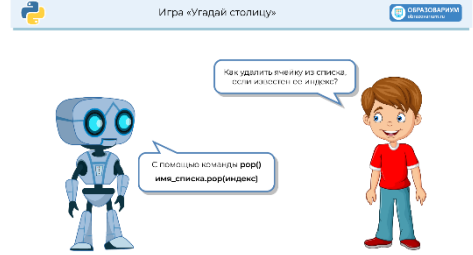
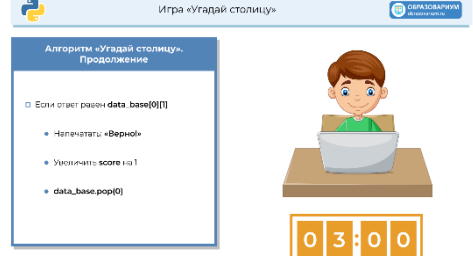
	<p><b>Сложный список</b>, это изменяемая индексируемая структура для хранения связанных между собой данных.</p> <p>Чаще всего это основной список, внутри которого находятся отдельные мини-списки.</p> <p>В отличие от простого списка – здесь каждая ячейка имеет целых два индекса: внешний и внутренний. Индексы указываются в квадратных скобках. Именно на основе сложного списка и будет наш сегодняшний проект.</p>	 <p>Игра «Угадай столицу»</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> Сколько команд из нижеприведенного списка генерируют числа?</p> <p>randint uniform shuffle choice</p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 2. Числа генерируют 2 команды. <b>randint</b> – генерирует случайное целое число, <b>uniform</b> – дробное.</i></p>	 <p>Игра «Угадай столицу»</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 список?</p> <p>1) имя_списка(append, данные) 2) имя_списка.append(данные) 3) append.имя_списка(данные) 4) append.данные(имя_списка)</p> <p><i>Ответ 2. Добавлять в список данные правильно так: <b>имя_списка.append(данные)</b></i></p>	 <p>Игра «Угадай столицу»</p> <p>Сцена 4</p>	

	<p><b>Третий вопрос.</b> Сколько основных ячеек в данном сложном списке?  <code>[[1, 2, 3], [4, 5, 6], [7], [8]]</code>          Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 4. В этом списке 4 основных ячейки.</i></p> <p><code>[[1, 2, 3], [4, 5, 6], [7], [8]]</code></p>	 <p>Игра «Угадай столицу»</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос № 3 Сколько основных ячеек в данном сложном списке? [[1, 2, 3], [4, 5, 6], [7], [8]]</p> <p>Поднимите карточку с соответствующим числом.</p> <p>В этом списке 4 основных ячейки.</p>	
	<p><b>Четвертый вопрос.</b> Какое число имеет индексы [1] [1]?</p> <p><code>[[1, 2, 3], [4, 3, 2], [1,4]]</code></p> <p><i>Ответ 3. Правильный ответ - число 3</i></p> <p><code>[[1, 2, 3], [4, 3, 2], [1,4]]</code></p>	 <p>Игра «Угадай столицу»</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос № 4 Какое число имеет индексы [1][1]? [[1, 2, 3], [4, 3, 2], [1, 4]]</p> <p>Поднимите карточку с соответствующим числом.</p> <p>Число 3 имеет индексы [1][1].</p>	<p>Сцена 5</p> <p>Сцена 6</p>
	<p><b>Пятый вопрос.</b> Какое последнее значение будет у счетчика?</p> <p><code>for i in range (5, -1, -2)</code></p> <p><i>Ответ 1. Это нисходящий цикл, который проработает 3 раза. Счетчик получит значения 5, 3 и 1.</i></p>	 <p>Игра «Угадай столицу»</p> <p>Блиц-опрос Поднимайте карточку с правильным ответом.</p> <p>Вопрос № 5 Какое последнее значение будет у счетчика? for i in range (5, -1, -2)</p> <p>Поднимите карточку с соответствующим числом.</p> <p>Это нисходящий цикл, который проработает 3 раза. Счетчик получит значения 5, 3 и 1.</p>	<p>Сцена 7</p>

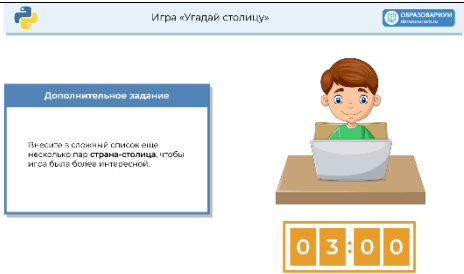

<p>Теоретический блок.</p> <p><b>Игра «Угадай столицу»</b></p>	<p>Сегодня нам предстоит создать игровой проект на основе <b>random</b> и сложного списка.</p> <p>Правила нашей игры достаточно просты. Компьютер называет нам страну, а наша задача написать ее столицу.</p> <p>За правильный ответ получаем балл.</p> <p>Если дан неправильный ответ – вопрос повторяется.</p> <p>В конце игры узнаем количество правильных ответов.</p>	 <p>Сцена 8</p>	<p>8 9</p>
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 1. Алгоритм «Угадай столицу»</b></p> <ul style="list-style-type: none"> <li>Подключить модуль <b>random</b></li> <li>Создать сложный список с из 5 пар</li> <li>Перемешать список так: <b>shuffle(data_base)</b></li> <li>Создать переменную для подсчета очков так: <b>score = 0</b></li> </ul>	 <p>Сцена 10</p>	<p>10 11</p>



	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.</p> <pre>from random import * data_base = ["Россия", "Москва"], ["Китай", "Пекин"], ["Турция", "Анкара"],             ["Индия", "Дели"], ["Египет", "Каир"]]</pre> <p>Для начала подключили модуль <b>random</b> и создадим базу данных в виде сложного списка. Для начала нам достаточно пяти стран.</p>	 <p>Сцена 11</p>	
Теоретический блок.	<p>Может возникнуть вопрос – зачем мы перемешали список? Для удобства! Мы всегда будем задавать вопрос, который идет первым в списке (по принципу колоды карт) и надо чтобы он был случайным.</p> <p>Также нам будет удобно удалять использованные вопросы и отслеживать количество оставшихся.</p>	 <p>Сцена 12</p>	12
Блок заданий. <b>Практические задания:</b> Задание 1	<p>После окончания дикторского текста запускается таймер на 4 мин. Перед началом игрового цикла пусть программа расскажет нам правила.</p> <p><b>Задание 1. Алгоритм «Угадай столицу». Продолжение</b></p> <ul style="list-style-type: none"> <li>Напечатать текст: Вспомни названия столиц. Если не знаешь - напиши слово 'Нет'</li> <li>Работать пока длина списка &gt; 0 <ul style="list-style-type: none"> <li>Напечатать: «Столица государства», ячейка с индексами 0 и 0</li> <li>Запросить ответ, занести его в текстовую переменную <b>answer</b> и применить метод <b>title()</b></li> </ul> </li> </ul>	 <p>Сцена 13</p>	13 14

	<p><b>Разбор задание 1.</b> Код программы будет выглядеть так: ... <i>дописывается к существующему программному коду.</i></p> <pre>print("Вспомни названия столиц. Если не знаешь - напиши слово 'Нет'") while len(data_base) &gt; 0:     print("Столица государства", data_base[0][0])     answer = input().title()</pre>	 <p>Сцена 14</p>	
Теоретический блок.	<p>После того, как вопрос будет задан и ответ на него будет получен (или напечатано слово НЕТ), его надо будет удалить из нашего списка, чтобы программа не задала его повторно</p> <p>Для удаления ячейки с определенным номером имеется команда <b>pop</b>. Необходимо указать имя списка, команду и в круглых скобках – основной индекс</p>	 <p>Сцена 15</p>	15
Блок заданий. <b>Практические задания:</b> Задание 1 Продолжение	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Задание 1. Алгоритм «Угадай столицу». Продолжение</b></p> <ul style="list-style-type: none"> <li>Если ответ равен data_base[0][1]             <ul style="list-style-type: none"> <li>Напечатать: «Верно!»</li> <li>Увеличить score на 1</li> <li>data_base.pop(0)</li> </ul> </li> </ul>	 <p>Сцена 16</p>	16 17



Дополнительно	<p>После окончания дикторского текста запускается таймер на 3 мин. Как вы понимаете, чем больше вопросов, тем интереснее игра</p> <p><b>Задание 1. Алгоритм «Угадай столицу». Дополнительно</b></p> <ul style="list-style-type: none"> <li>Внесите в наш сложный список еще несколько пар страна–столица, чтобы игра была более интересной.</li> </ul>	 <p>Сцена 20</p>	20
<p>Блок завершения занятия.</p> <p><b>Рефлексия. 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 узнали:</p> <ul style="list-style-type: none"> <li>Узнали, как удалить ячейку из списка</li> <li>Работали со сложным списком</li> <li>Работали с циклом с условием и условными операторами</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 21</p>	21

## Приложение 2

### Домашнее задание

Предложите еще несколько вариантов игр для которых необходима база данных построенная по принципу вопрос-ответ.

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

## **Практика**

### Проект «Три карты»

Создайте 2 списка: достоинства карт и масти. Сформируйте колоду в виде сложного списка, ячейка которого будет содержать информацию о достоинстве и масти карты.

Случайным образом выберите 3 карты из колоды, выведите их на экран и удалите из колоды.

### Проект «Случайная клетка»

Как известно, для обозначения клеток шахматной доски используется пара «цифра + буква». Сгенерируйте сложный список, ячейкой которого будет служить такая пара.

Выберите случайным образом клетку шахматной доски и выведите ее на экран.

## **Приложение 3**

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

«Работа со списками, перемешивание элементов. Игра «Угадай столицу».

**В начале занятия** необходимо повторить материал по теме «Списки и сложные списки». Чем они отличаются друг от друга, какими свойствами обладают, синтаксис, методы, способы вывод на экран. Также в чем разница в индексации простых и сложных списков. Поинтересуйтесь, какие примеры придумали ребята в качестве домашнего задания. Это может быть вывод варианты по количеству прожитых лет, по длине фамилии. Если добавить имя и отчество, то появится еще несколько вариантов сортировки.

**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 просмотра*** теоретической части задания рекомендуется поставить ролик на паузу и записать на доске несколько вариантов данных для нашего. Ввиду большого объема – проект создается по частям. После каждой части рекомендуется тестировать программу на отсутствие ошибок. Метод **shuffle**, применяемый в проекте, «тасует» список, меняя местами основные ячейки. Поэтому последовательность пар в создание списка не имеет значения. Отдельно обратите внимание на синтаксис вывода слова «Нет» на экран в кавычках.

При желании – можно создать переменную, которая будет подсчитывать не правильные ответы (и приравненные к ним – отказ от ответа). Однако это имеет смысл, если ученики будут тестировать не свой проект (ответы на который они знают), а проект соседа.