



### Методические рекомендации по теме

«Создание списка, сортировка, действия над элементами, циклический просмотр списка»

#### Цель:

- дать представление об использовании списков в языке Python.

#### Задачи:

- практика применения списков в Python;
- анализ программного кода с целью определения, что выведет программа при конкретных исходных данных;
- понятие списка в реальной жизни;
- 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 о понятии «список» в программировании и об использовании списков, в Python.

*Метапредметные:* обучающиеся получают возможность владения общепредметными понятиями «список», «структура», «индекс»; 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: Домашнее 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 начнем знакомство со структурой, которая позволит нам хранить и обрабатывать большое количество различных данных, которые будут располагаться в одном месте, в определенной последовательности; эта структура называется – список).

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

- Как вы понимаете понятие «список»?
- Для чего списки могут быть использованы в программировании и реальной жизни?
- Какие признаки списка вы могли бы назвать?
- Что могут включать в себя списки в программировании?

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

- Список – это структура, которая позволяет хранить и обрабатывать большие объемы данных
- Список в программировании может включать текст, целые и дробные числа, переменные, формулы

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.**

### **Списки и работа с ними.**

Новый материал излагается в сопровождении видеоролика, рекомендуется разместить на доске или флип-чарте изображения объектов псевдографики, в которых используется большое количество повторяющихся символов, при необходимости обращаться к ним для иллюстрации использования циклов.

Обсуждением вопросов по просмотренным материалам:

- Назовите признаки списка в программировании?
- Какие особенности написания имен списков?
- Какие особенности синтаксиса при работе со списками в Python?
- Можно ли вносить изменения в уже созданный список?

*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 – 12*

## **5. Блок заданий.**

**Проекты: «Первый список», «Случайные числа».**

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

**«Первый список»:** включает *практическое задание 1* «Первый список» по созданию списка, его сортировке и вывода списка на экран; функция последовательного вывода ячеек списка.

**После выполнения задания** ученики получают работающий программный продукт – программа позволяет создать и вручную заполнить данные для списка, сортировать элементы списка, выводить список целиком и отдельные ячейки списка.

**«Случайные числа»:** включает *практическое задание 2* для учеников с последующим разбором. Задание представляет собой небольшой программный проект для создания списка путем автоматического заполнения его ячеек, изучение метода **reverse** для переворачивания списка наоборот и вывода отдельного элемента списка по его индексу.

*После выполнения задания* ученики получают работающую программу, которая позволяет не только автоматически заполнять список по заданному количеству ячеек, а также сортировать и переворачивать его элементы, выводить нужный элемент списка, указывая его индекс.

Блок включает теоретические вставки: метод **reverse**, понятие индекс элемента в списке.

Обсуждения по теоретическим вставкам:

- Как работает переменная **name** ?
- Для чего нужна команда **append** ?
- Что такое индекс в списке?
- Как вывести на экран определенную запись в списке?

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

*\*см. сцены 13 – 24.*

## **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 25*

## **Приложение 1**

### **Сценарный план видеоролика**

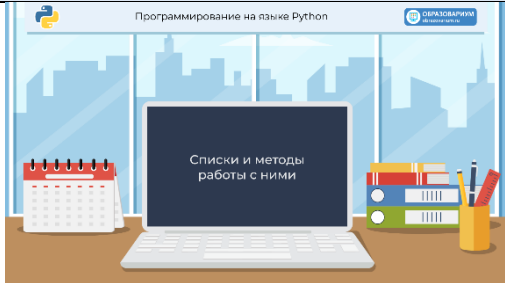
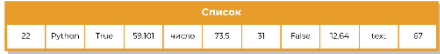
В таблице «Содержание видеоролика» представлены:










- 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

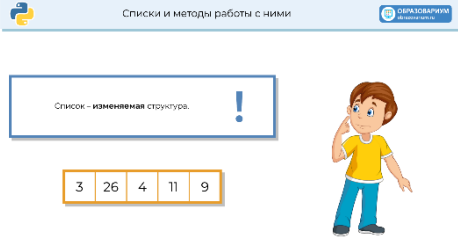
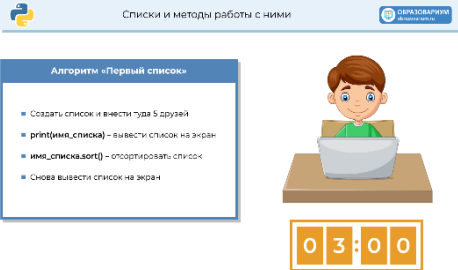
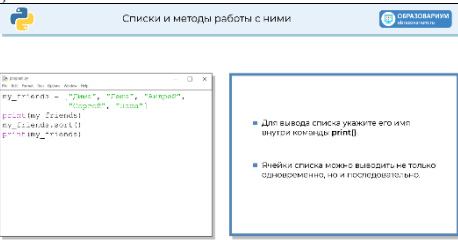
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока.  Создание списка, сортировка, действия над элементами, циклический просмотр списка.	 Сцена 1	1 2
	На этом занятии мы начнем знакомство со структурой, которая позволит нам хранить и обрабатывать большое количество различных данных. И все они будут располагаться в одном месте, в определенной последовательности. Называется эта структура – <b>список</b> .	  Сцена 2	



<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 строка при подключенном модуле random записана неверно?</p> <ol style="list-style-type: none"> <li>1) num = randint(1, 10)</li> <li>2) num = randint(-10, 1)</li> <li>3) num = randint(1, 1)</li> <li>4) num = randint(10, 1)</li> </ol> <p><i>Ответ 4. Для функции randint(a, b) обязательно условие <math>a \leq b</math>.</i></p>	<div> <div>  Списки и методы работы с ними </div> <div>  Блиц-опрос </div> </div> <p>Для функции randint(a, b) обязательно условие <math>a \leq b</math>.</p> <div> <div> <p><b>Блиц-опрос</b> Поднимайте карточку с правильным ответом.</p> <p>Вопрос №1</p> <p>Какая строка при подключенном модуле random записана неверно?</p> <ol style="list-style-type: none"> <li>1) num = randint(1, 10)</li> <li>2) num = randint(-10, 1)</li> <li>3) num = randint(1, 1)</li> <li>4) num = randint(10, 1)</li> </ol> </div> <div>  </div> </div> <p><b>Сцена 3</b></p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p><b>Второй вопрос.</b> Какое последнее значение будет у счетчика? Поднимите карточку с номером ответа.</p> <p><b>for i in range (10, -2, -3)</b></p> <p><i>Ответ 1. Счетчик получит значения 10, 7, 4, 1</i></p>	<div> <div>  Списки и методы работы с ними </div> <div>  Блиц-опрос </div> </div> <p>Счетчик получит значения 10, 7, 4, 1.</p> <div> <div> <p><b>Блиц-опрос</b> Поднимайте карточку с правильным ответом.</p> <p>Вопрос №2</p> <p>Какое последнее значение будет у счетчика? for i in range (10, -2, -3)</p> <p>Поднимите карточку с соответствующим числом.</p> </div> <div>  </div> </div> <p><b>Сцена 4</b></p>	
	<p><b>Третий вопрос.</b> Какая строка содержит не правильную запись увеличения num в 2 раза</p> <ol style="list-style-type: none"> <li>1) num = num + num</li> <li>2) num = num * 2</li> <li>3) num == num + num</li> <li>4) num *= 2</li> </ol> <p><i>Ответ 3. Запись num == num + num неверна, « == » используется при сравнении..</i></p>	<div> <div>  Списки и методы работы с ними </div> <div>  Блиц-опрос </div> </div> <p>Запись num == num + num неверна. « == » используется при сравнении.</p> <div> <div> <p><b>Блиц-опрос</b> Поднимайте карточку с правильным ответом.</p> <p>Вопрос №3</p> <p>Какая строка содержит неправильную запись увеличения num в 2 раза?</p> <ol style="list-style-type: none"> <li>1) num = num + num</li> <li>2) num = num * 2</li> <li>3) num == num + num</li> <li>4) num *= 2</li> </ol> </div> <div>  </div> </div> <p><b>Сцена 5</b></p>	

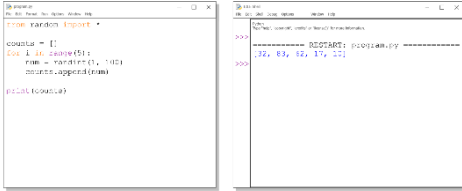

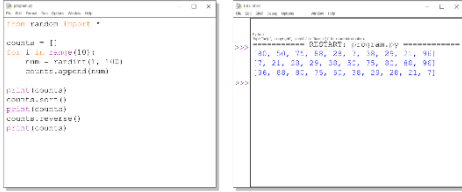
	<p><b>Четвертый вопрос.</b> Какой цикл не сработает ни разу, если <b>count = 1</b>?</p> <ul style="list-style-type: none"><li>1) while "1" != count:</li><li>2) while "1" == count:</li><li>3) while 1 == count:</li><li>4) while count == count:</li></ul> <p><i>Ответ 2. Цикл <b>while "1" == count</b>: не сработает ни разу, поскольку текст не может быть равен числу.</i></p>	<div><div>Списки и методы работы с ними</div><div>Блиц-опрос Поднимайте карту, у с правильными ответом.</div><div>Вопрос № 4 Какой цикл не сработает ни разу, если count = 1?</div><div>1) while "1" == count: 2) while "1" != count: 3) while 1 == count: 4) while count == count:</div><div>Цикл while "1" == count: не сработает ни разу, поскольку текст не может быть равен числу.</div><div>2</div></div> <div>Сцена 6</div>	
	<p><b>Пятый вопрос.</b> Какая команда не выводит целое число?</p> <ul style="list-style-type: none"><li>1) randint</li><li>2) len</li><li>3) uniform</li><li>4) int</li></ul> <p><i>Ответ 3. Команда <b>uniform</b> создает только дробные числа.</i></p>	<div><div>Списки и методы работы с ними</div><div>Блиц-опрос Поднимайте карту, у с правильными ответом.</div><div>Вопрос № 5 Какая команда не выводит целое число?</div><div>1) randint: 2) len 3) uniform: 4) int</div><div>Команда uniform создает только дробные числа.</div><div>3</div></div> <div>Сцена 7</div>	
<p>Теоретический блок.</p> <p><b>Списки и работа с ними</b></p>	<p>Нам уже знакомо понятие ячейки для хранения такой информации как переменная. Но как быть, если надо чтобы программа запомнила, например, пять имен? Или десять, или сто? Есть два варианта:</p> <p>Вариант один – создать необходимое количество переменных.</p> <p>Вариант два – воспользоваться списком</p>	<div><div>Списки и методы работы с ними</div><div>Переменная – ячейка для хранения одной информационно-ной единицы.</div><div>Если информации много</div><div>1 вариант создать много переменных</div><div>2 вариант использовать список</div></div> <div>Сцена 8</div>	<p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p>




	<p><b>Список</b> — это ячейка для хранения практически неограниченного количества информации.</p> <p>Как и переменная, список имеет собственное имя, созданное по тем же правилам. Давайте повторим эти правила</p>	<div data-bbox="1601 231 2056 258">  Списки и методы работы с ними  </div> <div data-bbox="1624 316 1854 435"> <p><b>Список</b></p> <p>Список – ячейка для хранения большого количества информации.</p> <p>Список имеет имя, созданное по правилам переменной.</p> </div>  <p>Сцена 9</p>	
	<p>Имя списка должно быть написано английскими буквами.</p> <p>Первая буква всегда должна быть прописная.</p> <p>Имя списка может содержать цифры.</p> <p>Если необходимо создать список из двух и более слов, то между словами ставится нижнее подчёркивание.</p> <p>Нельзя в качестве имени списка использовать служебную команду</p>	<div data-bbox="1601 630 2056 657">  Списки и методы работы с ними  </div> <div data-bbox="1662 694 1998 869"> <p><b>Правила создания имени</b></p> <ul style="list-style-type: none"> <li>Имя списка пишется английскими буквами.</li> <li>Первая буква должна быть прописная. Пример: <code>name, number</code>.</li> <li>Имя списка может содержать цифры. Пример: <code>number1, text2</code>.</li> <li>Между словами имени списка ставится нижнее подчёркивание. Пример: <code>my_name, my_colors</code>.</li> <li>Нельзя использовать в качестве имени списка служебную команду!</li> </ul> </div> <p>Сцена 10</p>	
	<p>Как же создать список, как занести туда данные и какого типа они должны быть?</p> <p>Список создается при помощи квадратных скобок.</p> <p>Он может быть изначально пустым, а может содержать данные, которые отделяются друг от друга запятой.</p> <p>Если это текстовые данные, то они обязательно пишутся в кавычках, в отличие от чисел. В одном и том же списке можно хранить любые данные</p>	<div data-bbox="1601 1029 2056 1056">  Списки и методы работы с ними  </div> <div data-bbox="1630 1101 1854 1244"> <pre> список = [] список = ["Саша", "Вася", "Таша"] список = [1, 2, 10, 12] список = ["Москва", 1234, True] </pre> </div>  <p>Сцена 11</p>	

	<p>Список – это изменяемая структура.          Это значит, что в уже созданный список можно</p> <ul style="list-style-type: none"> <li>• добавлять данные,</li> <li>• удалять,</li> <li>• заменять,</li> <li>• сортировать.</li> </ul>	 <p>Сцена 12</p>	
<p>Блок заданий.  <b>Практические задания:</b>          Задание 1          Задание 2</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Продолжим и организуем проверку.</p> <p><b>Задание 1. Алгоритм «Первый список»</b></p> <ul style="list-style-type: none"> <li>• Создайте список и внесите туда 5 друзей</li> <li>• <b>print (имя списка)</b> – вывести список на экран</li> <li>• <b>имя списка.sort()</b> – отсортировать список</li> <li>• Снова вывести список на экран</li> </ul>	 <p>Сцена 13</p>	<p>13 14 15 16 17 18 19 20 21 22 23</p>
	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.</p> <pre>my_friends = ["Дима", "Леша", "Андрей", "Сергей", "Паша"] print(my_friends) my_friends.sort() print(my_friends)</pre> <p>Для вывода списка укажите его имя внутри команды <b>print()</b>.          Ячейки списка можно выводить поочередно, а не одновременно</p>	 <p>Сцена 14</p>	

	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Давайте переделаем наш проект. Оставим в нем строчку с самим списком и строчку с сортировкой. Создадим цикл с переменной <b>name</b>, а работать он будет столько раз, сколько ячеек в нашем списке. Python это вычислит сам.</p> <p><b>name</b> будет переменной счетчик, но получает она не число, а содержимое очередной ячейки. Это называется вывод ячеек списка по имени.</p> <p>Перепишите код с экрана и протестируйте его.</p> <pre>my_friends = ["Дима", "Леша", "Андрей", "Сергей", "Паша"] my_friends.sort() for name in my_friends:     print(name)</pre>	<div><div>Списки и методы работы с ними</div><div><div><pre>my_friends = ["Дима", "Леша", "Андрей", "Сергей", "Паша"] my_friends.sort() for name in my_friends:     print(name)</pre></div><div><div>02:00</div></div></div></div> <p>Сцена 15</p>
	<p><i>Ещё немного теории:</i></p> <p>Мы составляли список вручную. Но Python позволяет делать это и программным способом, что гораздо интереснее.</p> <p>Для этого нам понадобится команда <b>append</b></p> <p>Давайте создадим новый проект, в котором данные будут добавляться в автоматическом режиме.</p>	<div><div>Списки и методы работы с ними</div><div><div>Создание списка</div><div><div>Вручную</div><div>при создании списка</div><div>Пример: my_friends = ["Дима", "Леша", "Андрей"]</div></div><div><div>Программно</div><div>append (добавляет элемент в список)</div><div>Пример: my_friends.append("Сергей")</div></div></div></div> <p>Сцена 16</p>
	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 2. Алгоритм «Случайные числа»</b></p> <ul style="list-style-type: none"><li>Подключить модуль <b>random</b></li><li>Создать пустой список <b>counts</b></li><li>Организовать цикл до 5<ul style="list-style-type: none"><li><b>num</b> = случайное число от 1 до 100</li><li><b>counts.append(num)</b></li></ul></li><li>Вывести список на экран</li></ul>	<div><div>Списки и методы работы с ними</div><div><div>Алгоритм «Случайные числа»</div><div><ul style="list-style-type: none"><li>Подключить модуль <b>random</b></li><li>Создать пустой список <b>counts</b></li><li>Организовать цикл до 5<ul style="list-style-type: none"><li><b>num</b> = случайное число от 1 до 100</li><li><b>counts.append(num)</b></li></ul></li><li>Вывести список на экран</li></ul></div></div><div><div>05:00</div></div></div> <p>Сцена 17</p>

	<p><b>Разбор задание 2.</b> Код программы может выглядеть так:</p> <pre>from random import * counts = [] for i in range(5):     num = randint(1, 100)     counts.append(num) print(counts)</pre>	<div data-bbox="1597 228 2056 478"><p>Списки и методы работы с ними</p></div> <p>Сцена 18</p>
	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Задание 2. Алгоритм «Случайные числа» (продолжение)</b></p> <ul style="list-style-type: none"><li>• Изменить количество повторов цикла до 10</li><li>• <b>counts.sort()</b></li><li>• вывести список на экран</li><li>• <b>counts.reverse()</b></li><li>• вывести список на экран</li></ul> <p><i>Метод <b>reverse</b> разворачивает список - наоборот. Так, что первый элемент становится последним.</i></p>	<div data-bbox="1597 544 2056 821"><p>Списки и методы работы с ними</p><div data-bbox="1597 603 1832 762"><p>Алгоритм «Случайные числа». Продолжение</p><ul style="list-style-type: none"><li>■ Изменить количество повторов цикла до 10</li><li>■ <b>counts.sort()</b></li><li>■ Вывести список на экран</li><li>■ <b>counts.reverse()</b></li><li>■ Вывести список на экран</li></ul></div></div> <p>Сцена 19</p>
	<p><b>Разбор задание 2 (продолжение).</b> Код программы может выглядеть так:</p> <pre>from random import * counts = [] for i in range(10):     num = randint(1, 100)     counts.append(num) print(counts) counts.sort() print(counts) counts.reverse() print(counts)</pre>	<div data-bbox="1597 914 2056 1165"><p>Списки и методы работы с ними</p></div> <p>Сцена 20</p>

	<p><i>Ещё немного теории:</i></p> <p>Часто у нас возникает необходимость вывести не весь список, а только какой-то его элемент.</p> <p>В этом нам поможет <b>индекс</b>.</p> <p>Как и в срезах, он обозначает порядковый номер и также начинается с нуля</p>	<div><div><div>Python</div><div>Списки и методы работы с ними</div><div>СДЕЛОВАЛИ</div></div><div><div>Как вывести один элемент?</div><div>По индексу!</div><div>Индекс – это порядковый номер, начинается с 0.</div></div><div></div></div> <p>Сцена 21</p>															
	<p>Если надо вывести самый первый элемент списка на экран, то надо указать имя списка и в квадратных скобках – индекс ноль</p> <p>Помним, что индекс может быть и отрицательным и начинаться с минус один.</p> <p>А если попробовать задать несуществующий индекс?</p> <p>Мы получим сообщение об ошибке.</p> <p>Допишите еще три строчки проекта. Пусть нам выведут первый элемент списка, последний элемент списка. А также – сколько всего элементов. Это нам посчитает уже знакомая команда <b>len</b>.</p>	<div><div><div>Python</div><div>Списки и методы работы с ними</div><div>СДЕЛОВАЛИ</div></div><div><pre>l = [38, 7, 21, 56, 75] print(l[0]) print(l[-1]) print(len(l))</pre><div>↑ выведет первый элемент ↑ выведет последний элемент ↑ выведет количество элементов</div></div><div><table><tr><td>-5</td><td>-4</td><td>-3</td><td>-2</td><td>-1</td></tr><tr><td>38</td><td>7</td><td>21</td><td>56</td><td>75</td></tr><tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr></table></div><p>Сцена 22</p></div>	-5	-4	-3	-2	-1	38	7	21	56	75	0	1	2	3	4
-5	-4	-3	-2	-1													
38	7	21	56	75													
0	1	2	3	4													


После окончания дикторского текста запускается таймер на 2 мин.

## Задание 2. Алгоритм «Случайные числа» (окончание)


- Вывести на экран первый элемент
- Вывести на экран последний элемент
- `print («Всего элементов», len(counts))`

**Разбор задание 2 (окончание).** Код программы может выглядеть так:  
... дописывается к программному коду

```
print("Первый элемент", counts[0])
print("Последний элемент", counts[-1])
print("Всего элементов", len(counts))
```




## Списки и методы работы с ними



СЕРГЕЙ КОЗЛОВ  
 преподаватель

### Алгоритм «Случайные числа». Окончание

- Ввести на экран «первый» элемент
- Ввести на экран «последний элемент»
- `print("Сколько элементов", len(counts))`



0

2

:

0

0

## Сцена 23

The screenshot shows a Jupyter Notebook interface. The code cell contains the following Python code:

```
from random import *

count = []

for i in range(100):
    num = randint(1, 100)
    count.append(num)

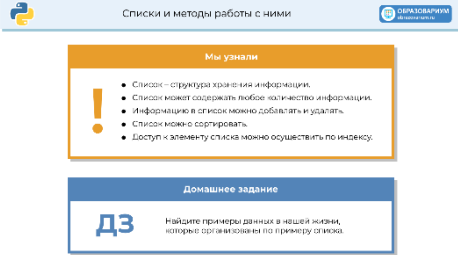
print(count)
count.sort()
print(count)
count.reverse()
print(count)

print("Среднее значение", sum(count)/len(count))
print("Максимальное значение", max(count))
print("Минимальное значение", min(count))
```

The output of the code is displayed in the cell below:

```
139, 44, 19, 77, 64, 37, 43, 25, 12, 32]
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100]
Среднее значение 75
Максимальное значение 100
Минимальное значение 1
```

Сцена 24

<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 узнали:</p> <ul style="list-style-type: none"> <li>▪ Список – структура хранения информации.</li> <li>▪ Список может содержать любое количество информации.</li> <li>▪ Информацию в список можно добавлять и удалять.</li> <li>▪ Список можно сортировать.</li> <li>▪ Доступ к элементу списка можно осуществить по индексу.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 25</p>	<p>25</p>
--	---	---	-----------

## Приложение 2

### Домашнее задание

Создайте дополнение к проекту занятия «Анимация лица», добавив анимацию тела, рук и ног персонажа.

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

### Практика

Проект «Числовой список»

Запросите у пользователя пять целых чисел и занесите их в список. Создайте копию списка и замените в ней все отрицательные числа на их модули. Выведите на экран первоначальный список и тот, что получился после замены чисел.

Проект «Сортировка по длине имени»

Создайте список и поместите в него несколько имен разной длины. При желании вы можете запросить данные для списка у пользователя.

Выведите на экран исходный список.

Далее необходимо отсортировать значения в списке по длине имени.

Для этого организуйте бесконечный цикл и внутри него создайте переменную «метка» со значением 0.

Внутри бесконечного цикла организуйте конечный цикл с количеством повторов на 1 меньше, чем длина списка.

Сравните длину имен в ячейке с индексом, равным счетчику, и в следующей ячейке. Поменяйте значения ячеек при необходимости. Если произошла замена, измените значение переменной «метка» на 1.

Условием выхода из бесконечного цикла будет отсутствие замен в конечном цикле.

После выхода из бесконечного цикла выведите отсортированный список на экран

### *Приложение 3*

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

«Рисование символами: вывод строк с помощью цикла с ограниченным количеством повторений».

**В начале занятия** можно поинтересоваться, какие картинки в стиле псевдографика ребятам с элементами цикличности или математических закономерностей ребятам удалось найти. Если есть техническая возможность – можно их показать и предложить проанализировать. Также необходимо повторить синтаксис и принцип работы конечного цикла, а заодно рассказать о вложенных циклах. Как пример – минуты внутри часа, час внутри суток. Данная информация пригодится при выполнении первого задания.



***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.

***При создании первого проекта*** можно менять вводные данные как в текстовой переменной, так и в конечных циклах. Аналогичным образом можно поступить и в задании № 2 Если у ребят возникнут свои идеи по анимации, то это можно только приветствовать.

***Если останется время*** – можно расширить второй проект, за счет увеличения количества картинок. Также можно проработать дополнения к проекту – например другие части тела.

***Рекомендуем*** заострить внимание на схожести подключения разных модулей, когда меняется только название самого модуля.