



### Методические рекомендации по теме

«Двумерные массивы. Основные алгоритмы обработки двумерных массивов»

#### Цель:

- дать представление о с двумерными массивами и генераторами в языке Python.

#### Задачи:

- практика применения двумерных массивов и генераторов в Python;
- изучение различных операций при работе с двумерными массивами в Python;
- 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 о создании и основных операциях с двумерными массивами в Python.

*Метапредметные:* обучающиеся получают возможность владения общепредметными понятиями «массив», «список», «двумерный массив», «генератор»; 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.

#### **2. Вводный блок.**

**Тема.**

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия.

**Проблемная дискуссия** по вопросам:

- Что такое двумерный массив?
- Как работает индексация списка при помощи цикла?
- Какие методы работы со списками в Python?
- Как вы думаете, операции и методы, которые мы изучали подходят для работы с двумерными массивами?

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

- Двумерные (сложные) массивы позволяют хранить и структурировать большее количество информации
- Все известные нам функции работают с одномерными массивами, поэтому для работы с двумерным массивом требуется сначала создать цикл для вычленения основной ячейки

*\*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.**

#### **Двумерный массив.**

Новый материал в занятии не выделяется в отдельный теоретический блок. Занятие по своему формату является практикумом и направлено на решение прикладных программных задач. Новый материал изучается в теоретических вставках при постановке и разборе практических заданий.

*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*

## **5. Блок заданий.**

### **«Работа с двумерным массивом».**

**«Работа с двумерным массивом»:** включает *блок практических заданий*, которые позволят провести прикладные разработки типовых задач для работы с двумерным массивом:

- Создание двумерного массива
- Двойной случайный массив через генератор
- Вычисление суммы массива
- Максимум всех ячеек
- Минимум всех ячеек

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

*\*см. сцены 9 – 22 (кроме сцен по теории).*

## **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 23*

## Приложение 1

### Сценарный план видеоролика

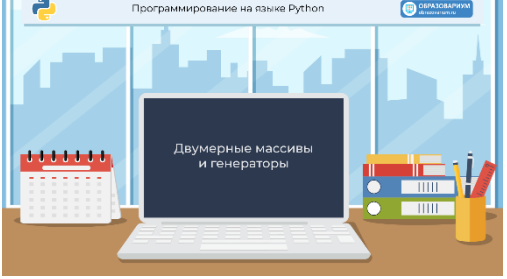

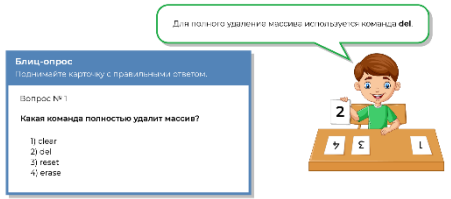
В таблице «Содержание видеоролика» представлены:

- 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
----------------	--------------------------------	--------------------------	--------

<p>Вводный блок.</p> <p><b>Мы узнаем</b></p>	<p><i>Обозначаем ученикам тему и цели урока.</i></p> <p>Двумерные массивы. Основные алгоритмы обработки двумерных массивов</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>Мы знаем, что массив в Python это однотипный список и все методы работы со списком подходят и для него.</p> <p>Теперь речь пойдет о двумерных, или как еще принято говорить, сложных массивах, которые позволяют хранить и структурировать гораздо <i>большее</i> количество информации.</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 полностью удалит массив?</p> <ol style="list-style-type: none"> <li>1) clear</li> <li>2) del</li> <li>3) reset</li> <li>4) erase</li> </ol> <p><i>Ответ 2. Для полного удаление массива используется команда <b>del</b>.</i></p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

**Второй вопрос.** Какая команда может дать дробное число?

- 1) uniform
- 2) round
- 3) float
- 4) любая из них

*Ответ 4. **uniform** – создает случайное дробное число; **round** – округляет до дробного, если указать второй параметр; **float** – создает дробное число.*

**Третий вопрос.** Чем отличается список от массива?

- 1) Скобками
- 2) Типом данных
- 3) Методами
- 4) Названием

*Ответ 2. Массив имеет один тип данных, а список – любой.*

**Четвертый вопрос.** В какой строке имеются числа, которые не могут быть сгенерированы **randint** (-5, 5)

- 1) 1, 2, 3, 4
- 2) 6, 5, 4, 3
- 3) 0, 1, 2, 3
- 4) -1, -2, -3, -4

*Ответ 2. Командой **randint** (-5, 5) могут быть сгенерированы только числа от -5 до 5.*


Давайте узнаем  
Двумерные массивы и генераторы

**Блиц-опрос**  
Поднимайте карту, жу с правильными ответом.

Вопрос № 2  
Какая команда может дать дробное число?

1) uniform  
2) round  
3) float  
4) любая из них

**uniform** – создает случайное дробное число.  
**round** – округляет до дробного, если указать второй параметр.  
**float** – создает дробное число.



Сцена 4


Давайте узнаем  
Двумерные массивы и генераторы

**Блиц-опрос**  
Поднимайте карту, жу с правильными ответом.

Вопрос № 3  
Чем отличается список от массива?

1) скобками  
2) типом данных  
3) методами  
4) названием

Данные в массиве должны быть одного типа,  
а в списке могут быть различных типов.



Сцена 5


Давайте узнаем  
Двумерные массивы и генераторы

**Блиц-опрос**  
Поднимайте карту, жу с правильными ответом.

Вопрос № 4  
В какой строке имеются числа, которые не могут быть сгенерированы **randint**(-5, 5)?

1) 1, 2, 3, 4  
2) 6, 5, 4, 3  
3) 0, 1, 2, 3  
4) -1, -2, -3, -4

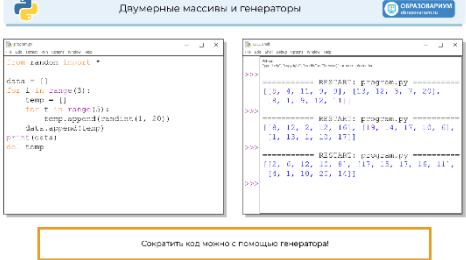
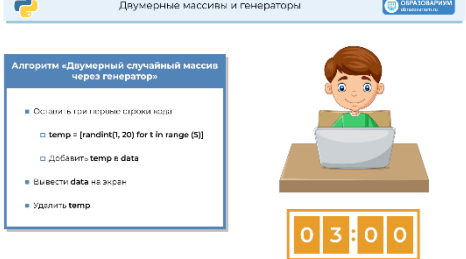
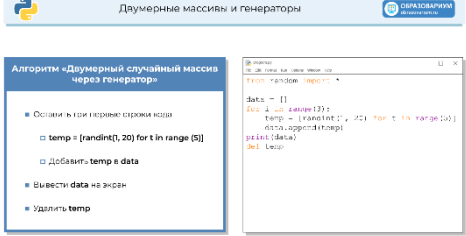
Командой **randint**(-5, 5) могут быть сгенерированы только числа от -5 до 5.



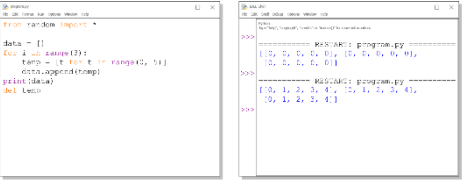





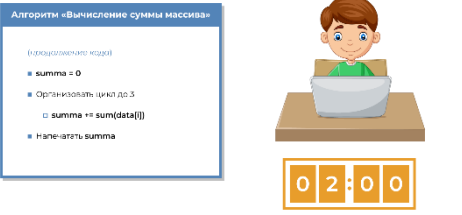


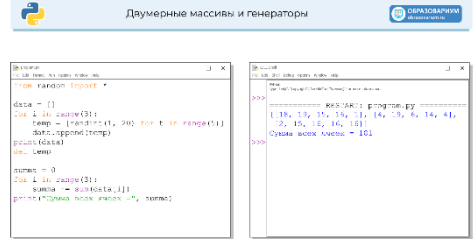
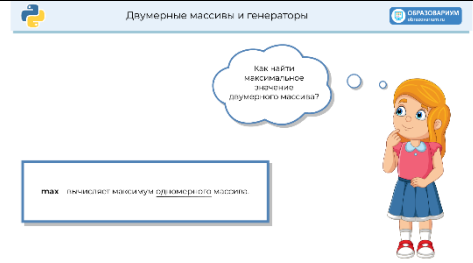
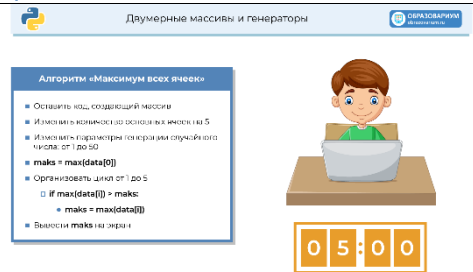
Сцена 6

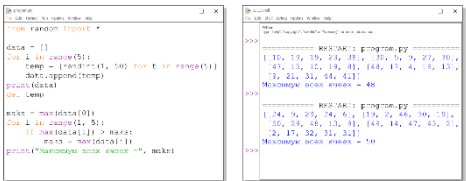
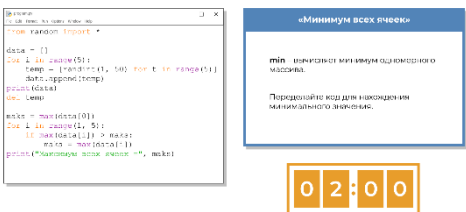
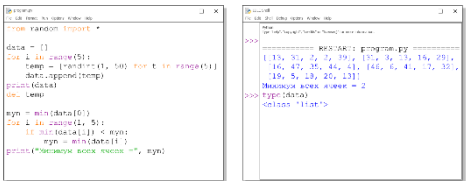
	<p><b>Пятый вопрос.</b> Какая команда из нижеприведенного списка перемешивает список?</p> <ol style="list-style-type: none"> <li>1) shuffle</li> <li>2) uniform</li> <li>3) randint</li> <li>4) choice</li> </ol> <p><i>Ответ 1. Для перемешивания элементов списка в случайном порядке используют команду <b>shuffle</b>.</i></p>	<p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Двумерный массив</b></p>	<p>Для того, чтобы начать работу с двумерным массивом надо научиться его создавать. При помощи цикла мы сможем значительно упростить данный процесс, однако есть свои хитрости. Двумерный массив создается при помощи двух, вложенных друг в друга циклов. Внешний отвечает за основные ячейки, а внутренний – за дополнительные.</p> <p>Также нам понадобится создать не один массив, а два: основной, с которым мы в дальнейшем будем работать и временный, который потом удалим.</p> <p>Ну, а генерацию чисел поручим модулю <b>random</b>.</p>	<p>Сцена 8</p>	8
<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 5 мин.</i></p> <p><b>Задание 1. Алгоритм «Создание двумерного массива»</b></p> <ul style="list-style-type: none"> <li>▪ Подключить random</li> <li>▪ Создать основной массив с именем data</li> <li>▪ Создать цикл до 3 <ul style="list-style-type: none"> <li>○ Создать временный массив temp</li> <li>○ Создать цикл до 5 <ul style="list-style-type: none"> <li>• Добавить во временный массив случайное число</li> </ul> </li> <li>○ data.append(temp)</li> </ul> </li> <li>▪ Вывести основной массив на экран</li> </ul>	<p>Сцена 9</p>	9 10 11 12


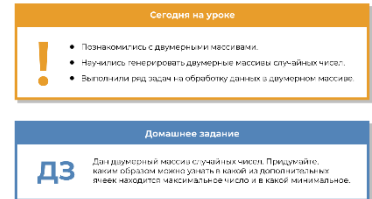


	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.</p> <pre> from random import * data = [] for i in range(3):     temp = []     for t in range(5):         temp.append(randint(1, 20))     data.append(temp) print(data) del temp </pre> <p>Сократить код можно с помощью генератор!</p>	 <p>Сцена 10</p>
	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Задание 2. Алгоритм «Двойной случайный массив через генератор»</b></p> <ul style="list-style-type: none"> <li>Оставить три первые строки кода             <ul style="list-style-type: none"> <li>temp = [randint(1, 20) for t in range (5)]</li> <li>Добавить temp в data</li> </ul> </li> <li>Вывести data на экран</li> <li>Удалить temp</li> </ul>	 <p>Сцена 11</p>
	<p><b>Разбор задания 2.</b> Ваш код может выглядеть так.</p> <pre> from random import * data = [] for i in range(3):     temp = [randint(1, 20) for t in range(5)]     data.append(temp) print(data) del temp </pre>	 <p>Сцена 12</p>

<p>Теоретический блок</p>	<p>Сейчас мы воспользовались, пожалуй, самым сложным генератором.</p> <p>Есть и попроще, правда и функционал у них победнее.</p> <p>Если изменим в проекте генератор на вот такой, то он заполнит все ячейки нулями.</p> <p><b>temp = [0] * 5</b></p> <p>Такой генератор заполнит каждую основную ячейку числами от нуля до четырех.</p> <p><b>temp = [t for t in range(0, 5)]</b></p>	<div data-bbox="1507 228 1966 260">  Двумерные массивы и генераторы  </div> <div data-bbox="1507 292 1966 475">  </div> <p>Сцена 13</p>	<p>13</p>
	<p>Теперь, когда у нас имеется массив с числами, давайте найдем для начала его сумму.</p> <p>Мы знаем, что в Python имеется команда SUM, которая вычисляет сумму массива.</p> <p>Но, к сожалению, только обычного. Поэтому с помощью цикла представим наш двумерный массив в виде нескольких простых</p>	<div data-bbox="1507 651 1966 683">  Двумерные массивы и генераторы  </div> <div data-bbox="1507 699 1966 922">  </div> <p>Сцена 14</p>	<p>14</p>
<p>Блок заданий. <b>Практические задания:</b> Задание 3</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p><b>Задание 3. Алгоритм «Нахождение суммы массива»</b> (продолжение кода)</p> <ul style="list-style-type: none"> <li>■ <code>summa = 0</code></li> <li>■ Создать цикл до 3 <ul style="list-style-type: none"> <li>○ <code>summa += sum(data[i])</code></li> </ul> </li> <li>■ Напечатать <code>summa</code></li> </ul>	<div data-bbox="1507 994 1966 1026">  Двумерные массивы и генераторы  </div> <div data-bbox="1507 1058 1966 1273">  </div> <p>Сцена 15</p>	<p>15 16</p>

	<p><b>Разбор задание 3.</b> Код программы будет выглядеть так:  <i>... дописывается к существующему коду</i></p> <pre> summa = 0 for i in range(3):     summa += sum(data[i]) print("Сумма всех ячеек =", summa) </pre>	 <p>Сцена 16</p>	
Теоретический блок	<p>Дальше будем действовать по той же схеме.          Как найти максимальное значение?</p> <p>Надо создать переменную и сравнивать ее с максимумом каждой основной ячейки.</p> <p>Сделаем это</p>	 <p>Сцена 17</p>	17
Блок заданий. Практические задания: Задание 4	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 4. Алгоритм «Максимум всех ячеек»</b></p> <ul style="list-style-type: none"> <li>Оставить код, создающий массив</li> <li>Изменить количество основных ячеек на 5</li> <li>Изменить параметры генерации случайного числа: от 1 до 50</li> <li><code>maks = max(data[0])</code></li> <li>Цикл от 1 до 5             <ul style="list-style-type: none"> <li><code>if max(data[i]) &gt; maks:</code> <ul style="list-style-type: none"> <li><code>maks = max(data[i])</code></li> </ul> </li> </ul> </li> <li>Вывести maks на экран</li> </ul>	 <p>Сцена 18</p>	18 19

	<p><b>Разбор задание 4.</b> Код программы будет выглядеть так:</p> <pre> from random import * data = [] for i in range(5):     temp = [randint(1, 50) for t in range(5)]     data.append(temp) print(data) del temp maks = max(data[0]) for i in range(1, 5):     if max(data[i]) &gt; maks:         maks = max(data[i]) print("Максимум всех ячеек =", maks) </pre>	 <p>Сцена 19</p>	
<p>Блок заданий. <b>Практические задания:</b> Задание 5</p>	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p><b>Задание 5. Алгоритм «Минимум всех ячеек»</b></p> <p><b>min</b> - вычисляет минимум ОДНОМЕРНОГО массива</p> <p>Переделайте код для нахождения минимального значения.</p>	 <p>Сцена 20</p>	<p>20 21</p>
	<p><b>Разбор задание 5.</b> Код программы будет выглядеть так:</p> <p>... <i>дописывается изменения к существующему коду</i></p> <pre> myn = min(data[0]) for i in range(1, 5):     if min(data[i]) &lt; myn:         myn = min(data[i]) print("Минимум всех ячеек =", myn) </pre>	 <p>Сцена 21</p>	

	<p>Может показаться что все довольно сложно и неудобно, но не надо забывать, что мы все-таки имеем дело не с настоящими массивами, а с числовыми списками.</p> <p>Убедиться в этом легко. Достаточно в терминале ввести команду <b>type</b> вместе с нашим массивом, и увидеть, что это все-таки список – класс <b>list</b></p>		
Теоретический блок	<p>Настоящие массивы сперва нужно импортировать через модуль <b>array</b>. Они более компактны и занимают меньше памяти и места, что делает их более эффективными по сравнению со списками.</p> <p>Если же мы хотим выполнять сложные математические вычисления, то лучше воспользоваться массивами <b>NumPy</b>, импортировав соответствующий модуль.</p> <p>Стоит отметить, что использовать массивы в Python следует только тогда, когда это действительно нужно, ведь списки работают аналогичным образом.</p>	 <p>Сцена 22</p>	22
Блок завершения занятия. Рефлексия. Сообщение домашнего задания	<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>Познакомились с двумерными массивами.</li> <li>Научились генерировать двумерные массивы случайных чисел.</li> <li>Выполнили ряд задач на обработку данных в двумерном массиве.</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 23</p>	23

### Домашнее задание

Дан двумерным массив случайных чисел. Придумайте, каким образом узнать в какой из дополнительных ячеек наибольшая сумма чисел.

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

### Практика

#### Проект «Максимальная ячейка»

Сгенерируйте двумерный массив случайных чисел и выведите его на экран.

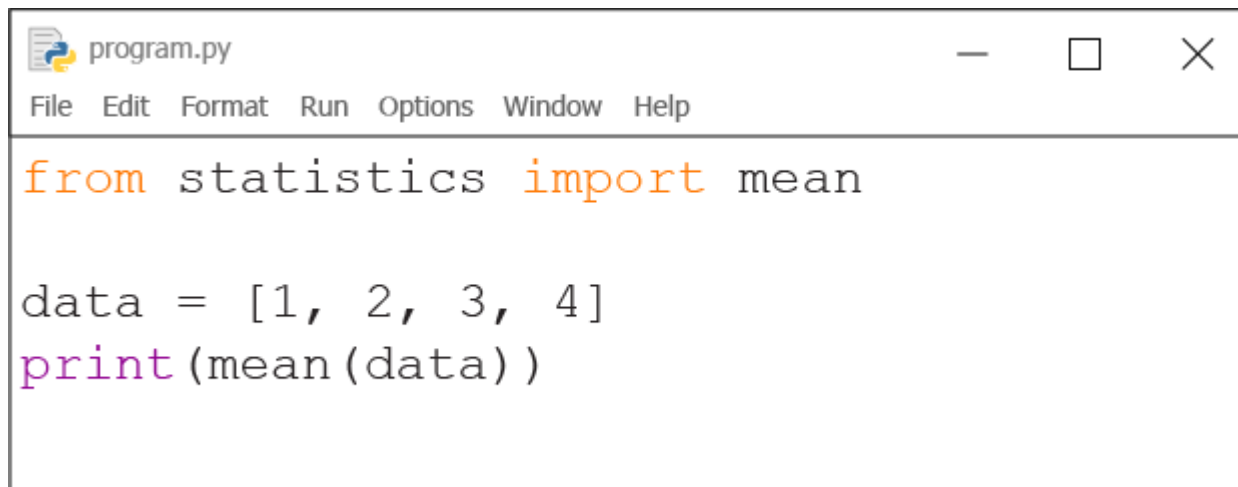
Определите основную ячейку, в которой находится одномерный массив с максимальной суммой. Выведите на экран номер этой ячейки и сумму.

#### Проект «Среднее в ячейках»

Сгенерируйте двумерный массив случайных чисел и выведите его на экран.

Для каждой основной ячейки выведите на экран ее содержимое, сумму и среднее значение.

Для определения среднего значения можно использовать метод **mean()** из модуля **statistics**.



```
from statistics import mean

data = [1, 2, 3, 4]
print(mean(data))
```

Такой код выведет 2.5.

### Приложение 3

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

«Двумерные массивы. Основные алгоритмы обработки двумерных массивов».

**В начале занятия** традиционно повторить материал по теме «списки» и методы работы с ними. Также можно лишний раз затронуть индексацию при помощи цикла, поскольку все задания урока будут связаны именно с этим инструментом.

**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.

***Во всех проектах*** используется единый код, создающий случайным образом числовой массив. Параметры массива и random можно настраивать и самостоятельно.

Для начала рассматривается вариант создания двумерного массива через циклы, а потом – через генератор. Также в теоретическом блоке приведены еще два примера генераторов, для создания массива чисел. Можете проставить ролик на паузу и проработать их самостоятельно, или просто принять к сведению как дополнительную информацию.

Все вычисления основаны на уже известных нам функциях. Однако желательно лишний раз подчеркнуть, что работают они только с одномерными массивами. Поэтому все проекты построены по единому сценарию: создание цикла для вычленения основной ячейки и работа с ней, как с одномерным массивом.

***Если имеется желание*** выполнять более сложные математические вычисления, то лучше воспользоваться массивами NumPy, импортировав соответствующий модуль.