



### Методические рекомендации по теме «Множества и методы работы с ними»

#### Цель:

- дать представление об использовании множеств в языке Python.

#### Задачи:

- практика применения множеств 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 понятии «множество» в программировании и об использовании множеств в Python.

*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.

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

- Как вы понимаете понятие «множество»?
- Чем, по-вашему, мнения могут быть множества в программировании?
- Чем по вашему мнению могут различаться списки и множества?

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

- Множество – это структура, которая также позволяет хранить и обрабатывать большие объемы данных
- Элементы множеств расположены без порядка и индексов в отличие от списков

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.

### Проект «5 разных чисел».

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

Обсуждением вопросов по просмотренным материалам:

- Может ли список хранить одинаковую информацию в разных записях?
- Почему это может быть неудобно для работы?

Обсуждением вопросов по просмотренным материалам (блок 2):

- Что такое множество?

- Что означает, что множество содержит уникальную информацию?
- Возможно ли найти элемент множества по индексу?
- Как различается синтаксис списков и множеств в Python?

*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, 11-14, 17, 20-21*

## **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 9 – 27 (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 28*

## Приложение 1

### Сценарный план видеоролика

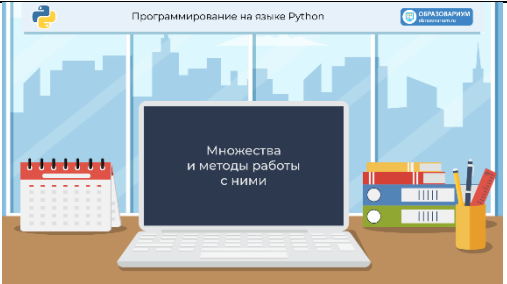

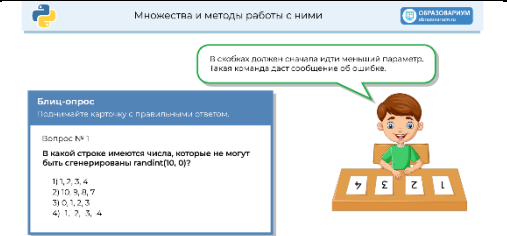
В таблице «Содержание видеоролика» представлены:

- 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. Мы узнаем</p>	<p>Обозначаем ученикам тему и цели урока.</p> <p>Множества и методы работы с ними</p>	 <p>Сцена 1</p>	<p>1 2</p>
	<p>Мы уже знаем, что большое количество данных удобнее записывать не в переменные, а в списке: индексируемой и изменяемой структуре.</p> <p>Однако помимо списка, существуют и другие конструкции для хранения информации. Одной из них является множество.</p> <p>Что это такое и в чем его отличие от списка?</p>	 <p>Сцена 2</p>	
<p>Блок повторения. Блиц-опрос</p>	<p>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</p> <p><b>Первый вопрос.</b> В какой строке имеются числа, которые не могут быть сгенерированы <code>randint(10,0)</code>?</p> <ol style="list-style-type: none"> <li>1) 1, 2, 3, 4</li> <li>2) 10, 9, 8, 7</li> <li>3) 0, 1, 2, 3</li> <li>4) -1, -2, -3, -4</li> </ol> <p>Ответ _. В скобках должен сначала идти меньший параметр. Такая команда даст сообщение об ошибке.</p>	 <p>Сцена 3</p>	<p>3 4 5 6 7</p>

**Второй вопрос.** Какие индексы у буквы "Г"?

`[["A", "Б"], ["В", "Г"], ["Е"]]`

- 1) 4 1
- 2) 2 2
- 3) 1 1
- 4) 0 3

Ответ 3.

`[["A", "Б"], ["В", "Г"], ["Е"]]`  
                  0                  1                  2

**Третий вопрос.** Какая команда считает длину объекта?

- 1) append
- 2) pop
- 3) len
- 4) range

Ответ 3. Команда **len** подсчитывает длину объекта.

**Четвертый вопрос.** Какая команда из нижеприведенного списка перемешивает список?

- 1) shuffle
- 2) uniform
- 3) randint
- 4) choice

Ответ 1. Команда **shuffle** перемешивает список.


Множества и методы работы с ними

Блиц-опрос  
Подчеркните карту(жу) с правильными ответом.

Вопрос № 2  
Какие индексы у буквы "Г"?  
`[["A", "Б"], ["В", "Г"], ["Е"]]`

1) 4 1  
2) 2 2  
3) 1 1  
4) 0 3

Буква "Г" имеет индексы (1,1).



Сцена 4


Множества и методы работы с ними

Блиц-опрос  
Подчеркните карту(жу) с правильными ответом.

Вопрос № 3  
Какая команда считает длину объекта?

1) append  
2) pop  
3) len  
4) range

Команда **len** подсчитывает длину объекта.



Сцена 5


Множества и методы работы с ними

Блиц-опрос  
Подчеркните карту(жу) с правильными ответом.

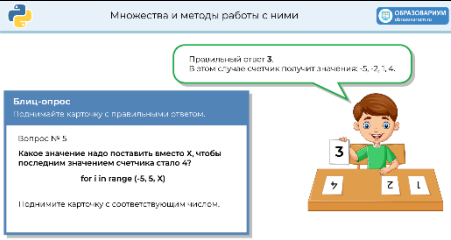
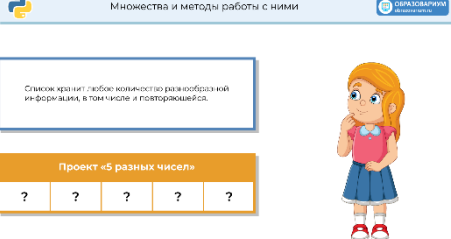
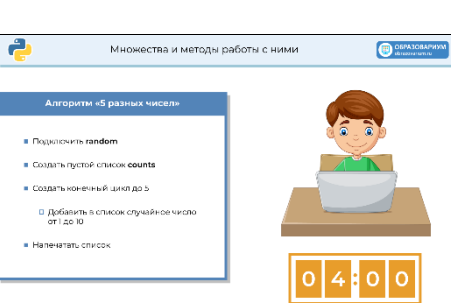
Вопрос № 4  
Какая команда перемешивает список?

1) shuffle  
2) uniform  
3) randint  
4) choice

Команда **shuffle** перемешивает список.


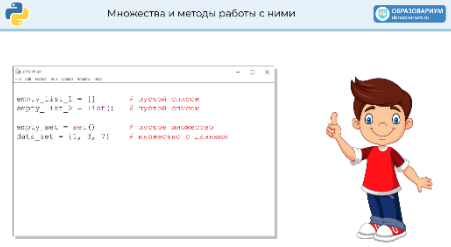
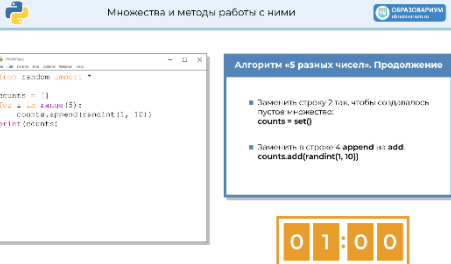


Сцена 6

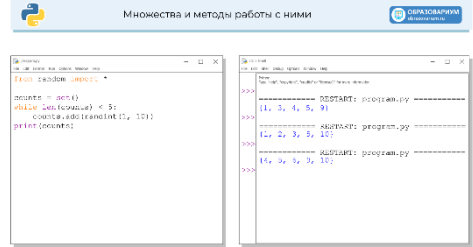
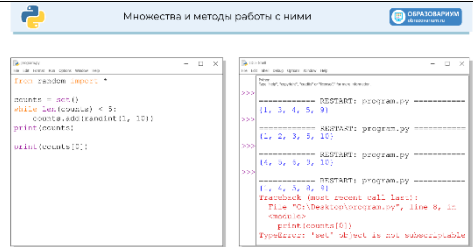
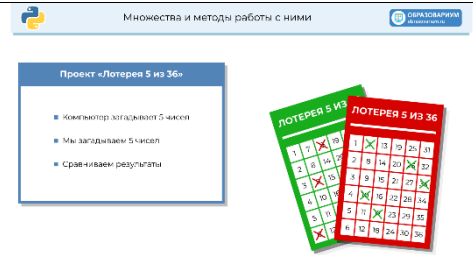
	<p><b>Пятый вопрос.</b> Какое значение надо поставить вместо X, чтобы последним значением счетчика стало 4?</p> <p><b>for i in range (-5, 5, X)</b></p> <p><i>Ответ 3. В этом случае счетчик получит значения: -5, -2, 1, 4.</i></p>	 <p>Множества и методы работы с ними</p> <p>Правильный ответ 3. В этом случае счетчик получит значения: -5, -2, 1, 4.</p> <p>Блиц-опрос Поднимайте карту, жу с правильными ответами.</p> <p>Вопрос № 5 Какое значение надо поставить вместо X, чтобы последним значением счетчика стало 4? for i in range (-5, 5, X)</p> <p>Поднимите карту, жу с соответствующим числом.</p> <p>Сцена 7</p>	
<p>Теоретический блок.</p> <p><b>Проект «5 разных чисел»</b></p>	<p>Перейдем к теме занятия. Мы уже знаем, что список — это структура, которая может хранить любое количество разнообразной информации. В том числе одинаковой. Не всегда это бывает удобно.</p> <p>Давайте рассмотрим ситуацию, при которой нам необходимо сгенерировать пять случайных чисел в диапазоне от одного до десяти.</p>	 <p>Множества и методы работы с ними</p> <p>Список хранит любое количество разнообразной информации, в том числе и повторяющейся.</p> <p>Проект «5 разных чисел»</p> <p>? ? ? ? ?</p> <p>Сцена 8</p>	8
<p>Блок заданий.</p> <p><b>Практические задания:</b></p> <p>Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 4 мин.</i></p> <p><b>Задание 1. Алгоритм «5 разных чисел»</b></p> <ul style="list-style-type: none"> <li>Подключить модуль <b>random</b></li> <li>Подключить random</li> <li>Создать пустой список counts</li> <li>Создать конечный цикл до 5: <ul style="list-style-type: none"> <li>Добавить в список случайное число от 1 до 10</li> </ul> </li> <li>Напечатать список</li> </ul>	 <p>Множества и методы работы с ними</p> <p>Алгоритм «5 разных чисел»</p> <ul style="list-style-type: none"> <li>Подключить random</li> <li>Создать пустой список counts</li> <li>Создать конечный цикл до 5 <ul style="list-style-type: none"> <li>Добавить в список случайное число от 1 до 10</li> </ul> </li> <li>Напечатать список</li> </ul> <p>0 4 0 0</p> <p>Сцена 9</p>	9 10



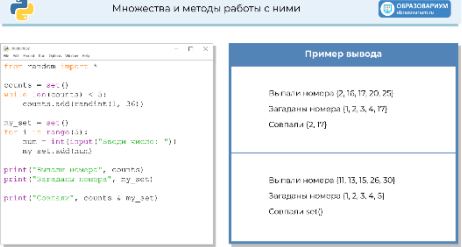
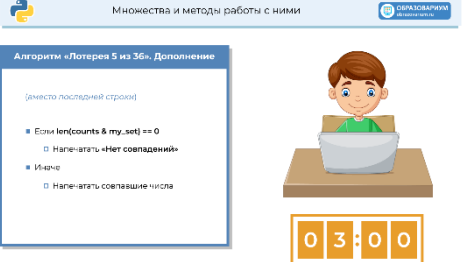
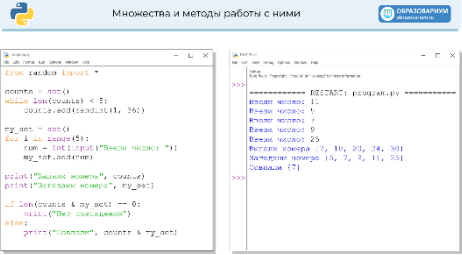
	<p><b>Разбор задания 1.</b> Ваш код может выглядеть так.</p> <pre>from random import * counts = [] for i in range(5):     counts.append(randint(1, 10)) print(counts)</pre> <p><i>Есть вероятность что в списке будут одинаковые числа. То есть принцип уникальности будет нарушен. Произойдет это потому, что могут быть сгенерированы одинаковые числа, поскольку список разрешает хранить в ячейках идентичные данные. стран.</i></p>	<p>Сцена 10</p>	
Теоретический блок.	<p>Как же решить данную проблему? Нам поможет множество.</p> <p><b>Множество</b> — это неупорядоченная изменяемая структура, хранящая уникальную информацию</p> <p><b>Изменяемая</b> — значит в нее можно добавлять или удалять элементы</p> <p><b>Уникальная</b> — это значит, что в нём не может быть одинаковых элементов.</p> <p>А что с ними произойдет? Множество самостоятельно их удалит</p>	<p>Сцена 11</p>	11 12 13 14
	<p>Чем еще множество отличается от списка?</p> <p><b>Множество</b> — это неупорядоченная структура, то есть элементы в нем расположены случайно и найти элемент по индексу невозможно.</p>	<p>Сцена 12</p>	

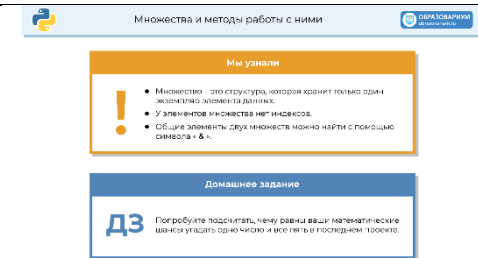
	<p>Представьте себе группу учеников. Мы можем точно знать, сколько учеников в группе, и кто именно в ней находится. В группу можно добавить учеников или кто-то может выйти из группы. Но мы не можем знать в каком порядке ученики построятся, если их попросить стать в ряд.</p>	 <p>Множества и методы работы с ними</p> <p>Сцена 13</p>	
	<p>Есть еще отличие множества от списка – внешнее. Список создается при помощи квадратных скобок или команды <b>list</b>,  А множество – с помощью команды <b>set</b>. И когда оно будет создано, то информация будет находиться в фигурных скобках</p>	 <p>Множества и методы работы с ними</p> <p>Сцена 14</p>	
<p>Блок заданий. <b>Практические задания:</b> Задание 1</p>	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i> <b>Задание 1. Алгоритм «5 разных чисел».</b></p> <ul style="list-style-type: none"> <li>• Заменить строку 2 так, чтобы создавалось пустое множество: <code>counts = set()</code></li> <li>• Заменить в строке 4 <b>append</b> на <b>add</b>: <code>counts.add(randint(1,10))</code></li> </ul>	 <p>Множества и методы работы с ними</p> <p>Сцена 15</p>	<p>15 16</p>

	<p><b>Разбор задание 1.</b> Код программы будет выглядеть так:</p> <pre> from random import * counts = set() for i in range(5):     counts.add(randint(1, 10)) print(counts) </pre>	<p>Сцена 16</p>	
<p>Теоретический блок.</p>	<p>Как же быть в данной ситуации?  Прежде всего надо понять, что конечный цикл нам не поможет.  Ведь мы не знаем сколько попыток понадобится <b>random</b> чтобы сгенерировать пять разных чисел.  А значит нам нужен цикл, который будет работать до тех пор, пока количество ячеек во множестве не станет равно пяти.  И у нас есть такой – это цикл с условием</p>	<p>Сцена 17</p>	17
<p>Блок заданий.  <b>Практические задания:</b>  Задание 1  Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Задание 1. Алгоритм «Угадай столицу». Продолжение</b></p> <ul style="list-style-type: none"> <li>Заменить конечный цикл на цикл с условием:  Работать пока <b>len(counts) &lt; 5</b></li> </ul> <p><i>Измените строку номер три, чтобы вместо конечного цикла, был цикл с условием «работать пока длинна множества меньше пяти». А узнать длину множества, как и списка, как и текстовой переменной, нам поможет команда <b>len</b>.</i></p>	<p>Сцена 18</p>	18 19

	<p><b>Разбор задание 1 (продолжение).</b> Код программы будет выглядеть так:</p> <pre> from random import * counts = set() while len(counts) &lt; 5:     counts.add(randint(1, 10)) print(counts) </pre>	 <p>Сцена 19</p>	
Теоретический блок.	<p>Может показаться, что элементы в нем расположены по возрастанию. Это заблуждение. Просто программе надо как-то вывести их на экран, вот она выбирает чаще всего такую последовательность.</p> <p>Но у множества нет порядка и номеров, и если мы решим вывести, как в списке, какой ни будь элемент по индексу, то получим ошибку.</p> <p>Traceback (most recent call last):  File "C:\Desktop\programm.py", line 8, in &lt;module&gt;  <b>print(counts[0])</b>  TypeError: 'set' object is not subscriptable</p>	 <p>Сцена 20</p>	20 21
	<p>А теперь давайте на основе уже имеющегося кода сделаем лотерею. Компьютер загадает пять чисел в диапазоне от одного до 36-ти. Тоже самое сделаем и мы. Потом сравним, какие числа совпали.</p> <p><b>Проект «Лотерея 5 из 36»</b></p> <ul style="list-style-type: none"> <li>• Компьютер загадывает 5 чисел</li> <li>• Мы загадываем 5 чисел</li> <li>• Сравниваем результаты</li> </ul>	 <p>Сцена 21</p>	

<p>Блок заданий. <b>Практические задания:</b> Задание 1 Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 5 мин.</i></p> <p><b>Задание 1. Алгоритм «Лотерея 5 из 36». Продолжение</b></p> <ul style="list-style-type: none"> <li>■ Изменить <b>random</b> на (1, 36)</li> <li>■ Убрать строку с выводом информации</li> <li>■ Создать пустое множество <b>my_set</b></li> <li>■ Создать цикл до 5 <ul style="list-style-type: none"> <li>○ Запросить целое число и занести его в переменную <b>num</b></li> <li>○ Добавить <b>num</b> в <b>my_set</b></li> </ul> </li> <li>■ Вывести на экран множество <b>counts</b></li> <li>■ Вывести на экран множество <b>my_set</b></li> </ul>	<div data-bbox="1507 228 1966 499"> </div> <p>Сцена 22</p>	<p>22 23</p>
	<p><b>Разбор задание 1 (продолжение).</b> Код программы будет выглядеть так:</p> <pre> from random import * counts = set() while len(counts) &lt; 5:     counts.add(randint(1, 36)) my_set = set() for i in range(5):     num = int(input("Введи число: "))     my_set.add(num) print("Выпали номера", counts) print("Загаданы номера", my_set) </pre>	<div data-bbox="1507 635 1966 898"> </div> <p>Сцена 23</p>	
<p>Блок заданий. <b>Практические задания:</b> Задание 1 Продолжение</p>	<p><i>После окончания дикторского текста запускается таймер на 1 мин.</i></p> <p><b>Задание 1. Алгоритм «Лотерея 5 из 36». Продолжение</b></p> <ul style="list-style-type: none"> <li>■ Добавить строчку для проверки совпадений: <b>print("Совпали", counts &amp; my_set)</b></li> </ul> <p><i>Теперь нам нужно сравнить наши множества на наличие совпадений. Для этого существует специальный знак, называемый по-научному «амперсанд».</i></p> <p><i>Для нахождения общих элементов между двумя множествами используется знак «&amp;»</i></p> <p><i>Как его использовать: достаточно вывести на экран наши множества, разделенные этим символом, и мы получим числа, которые совпали</i></p>	<div data-bbox="1507 1042 1966 1313"> </div> <p>Сцена 24</p>	<p>24</p>

	<p>Теперь программа будет выводить номера, которые имеются в обоих множествах.</p> <p>Обратите внимание на скобки – у нас получилось новое, третье множество без имени, которое хранит совпадающие элементы.</p> <p>Однако если таковых нет – будет напечатано следующее... Просто пустое множество</p>	 <p>Множества и методы работы с ними</p> <p>Пример вывода</p> <p>Выпали номера (2, 16, 17, 20, 25) Загаданы номера (1, 2, 3, 4, 17) Совпали (2, 17)</p> <p>Выпали номера (11, 13, 15, 26, 30) Загаданы номера (1, 2, 3, 4, 5) Совпали нет</p> <p>Сцена 25</p>	25
	<p>После окончания дикторского текста запускается таймер на 3 мин.</p> <p><b>Задание 1. Алгоритм «Лотерея 5 из 36». Дополнение</b></p> <p>Дополнение вместо последней строки</p> <ul style="list-style-type: none"> <li>Если <code>len(counts &amp; my_set) == 0</code> Напечатать «Нет совпадений»</li> <li>Иначе Напечатать совпавшие числа</li> </ul>	 <p>Множества и методы работы с ними</p> <p>Алгоритм «Лотерея 5 из 36». Дополнение</p> <p>(вместо последней строки)</p> <ul style="list-style-type: none"> <li>Если <code>len(counts &amp; my_set) == 0</code> Напечатать «Нет совпадений»</li> <li>Иначе Напечатать совпавшие числа</li> </ul> <p>03:00</p> <p>Сцена 26</p>	26
	<p><b>Разбор задание 1 (продолжение).</b> Код программы будет выглядеть так:</p> <pre> from random import * counts = set() while len(counts) &lt; 5:     counts.add(randint(1, 36)) my_set = set() for i in range(5):     num = int(input("Введи число: "))     my_set.add(num) print("Выпали номера", counts) print("Загаданы номера", my_set) if len(counts &amp; my_set) == 0:     print("Нет совпадений") else: </pre>	 <p>Множества и методы работы с ними</p> <p>Сцена 27</p>	27

	<code>print("Совпали", counts &amp; my_set)</code>		
Блок завершения занятия. <b>Рефлексия.</b> <b>Сообщение домашнего задания</b>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p><b>Подведем итоги.</b></p> <p>Мы узнали:</p> <ul style="list-style-type: none"> <li>▪ Множество – это структура, которая хранит только один экземпляр элемента данных.</li> <li>▪ У элементов множества нет индексов.</li> <li>▪ Общие элементы двух множеств можно найти с помощью символа « &amp; ».</li> </ul> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	 <p>Сцена 28</p>	28

Попробуйте подсчитать, чему равны ваши математические шансы угадать одно число или все пять чисел в лотерею «5 из 36»

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

## Практика

Проект «Спортивные секции»

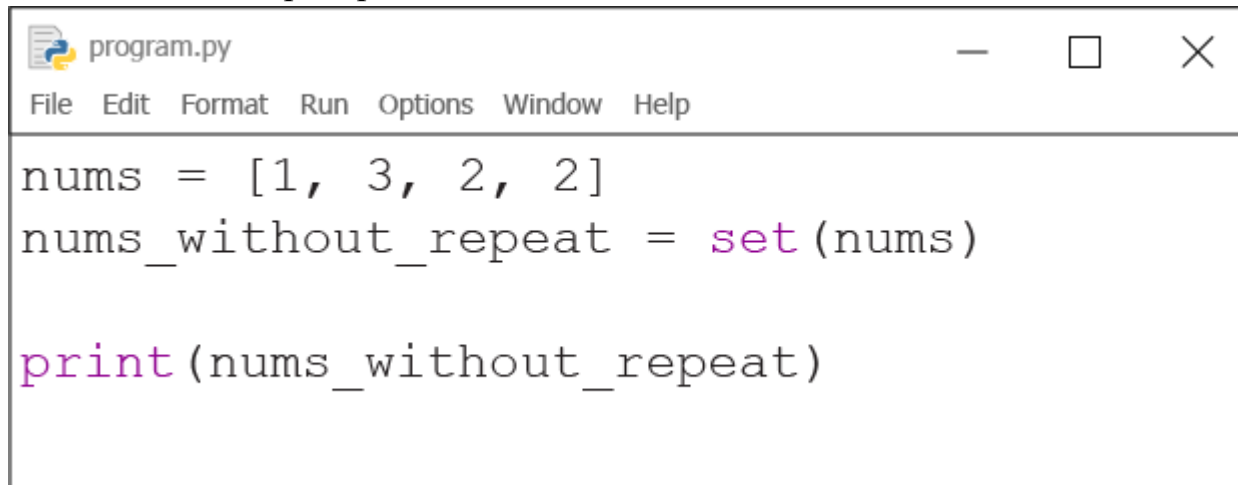
Представьте, что учителю принесли 3 списка:

- учеников его класса,
- тех, кто занимается в секции по футболу
- тех, кто занимается в секции по волейболу

Составьте список ребят, которые не ходят ни в одну секцию.

Примечание: при желании вы можете использовать уже подготовленные списки (скачать файл).

Для того, чтобы преобразовать список в множество, можно использовать команду `set()`.



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

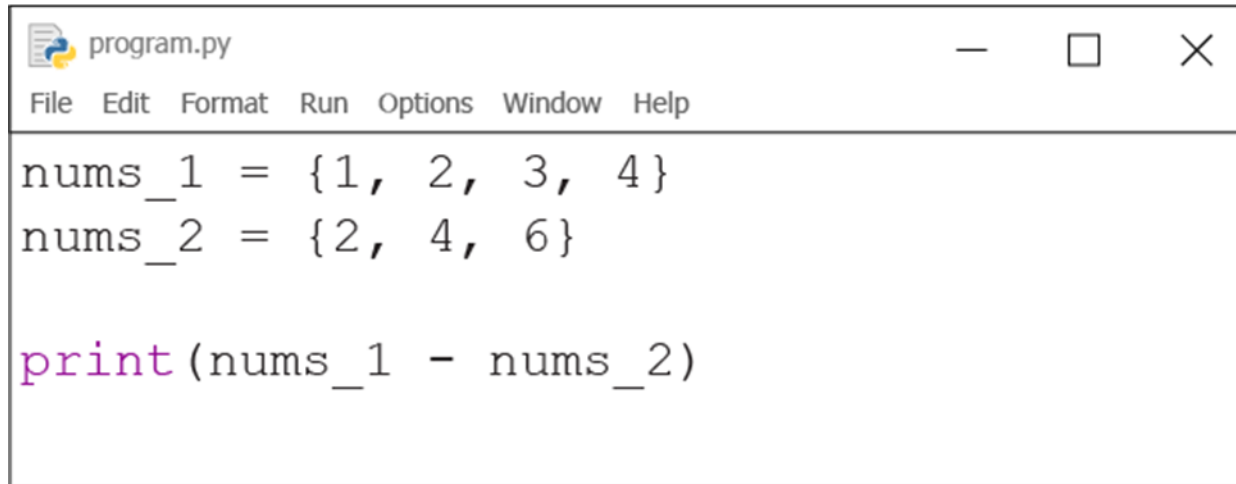
nums = [1, 3, 2, 2]
nums_without_repeat = set(nums)

print(nums_without_repeat)
```



Такой код выведет {1, 2, 3}.

Для того, чтобы получить элементы одного множества без элементов другого, можно использовать оператор « - ».

A screenshot of a Python IDE window titled "program.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code inside the editor is:

```
nums_1 = {1, 2, 3, 4}
nums_2 = {2, 4, 6}

print(nums_1 - nums_2)
```

Такой код выведет {1, 3}.

### Проект «Изучение языков»

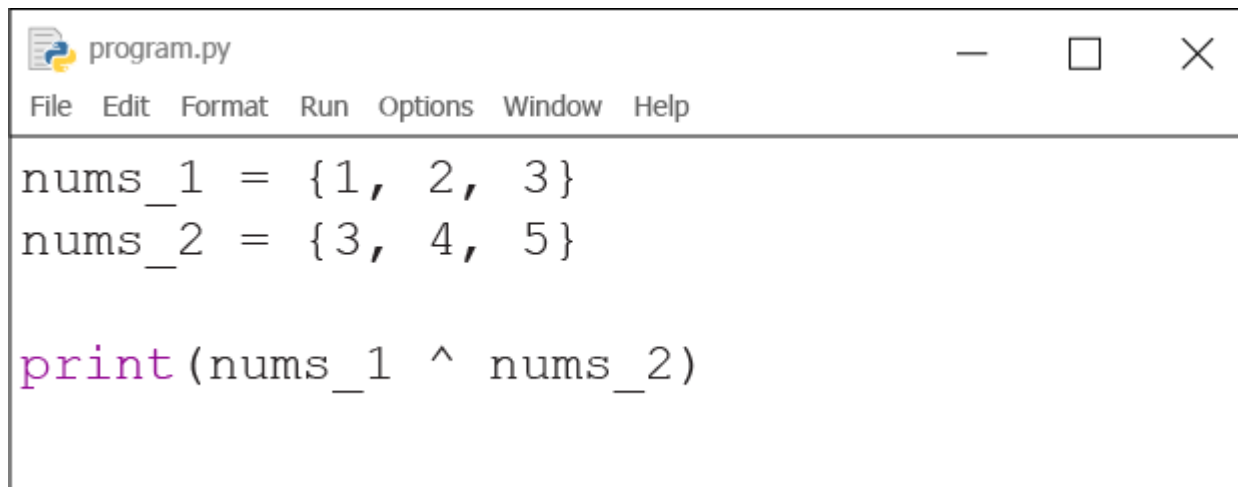
В школе организовали 2 группы для изучения языков: английского и французского. К вам попали списки учеников каждой группы.

Выведите на экран

- количество ребят, которые изучают только один язык.
- имена ребят, которые изучают сразу два языка, и сколько всего таких учеников.

Примечание: при желании вы можете использовать уже подготовленные списки (скачать файл).

Для того чтобы получить элементы, которые входят только в одно из множеств, можно использовать оператор « ^ »



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

nums_1 = {1, 2, 3}
nums_2 = {3, 4, 5}

print(nums_1 ^ nums_2)
```

Такой код выведет {1, 2, 4, 5}.

### Приложение 3

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

«Множества и методы работы с ними».

**В начале занятия** можно подчеркнуть, что будем знакомиться с множествами, которые имеют как общие, так и различные свойства со списками. Поэтому на протяжении урока можно делать записи на доске в формате сравнения этих структур. А начать урок можно с повторения свойств, методов и синтаксиса списка.

**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.

***В ходе урока*** мы будем трансформировать код из одного проекта в другой. Поэтому после каждого практического блока его необходимо тестировать на отсутствие ошибок. В процессе просмотра, ролик можно поставить на паузу чтобы объяснить ребятам: почему вероятность пяти разных чисел в списке при **random** от 1 до 10 близка 50 %. Если позволяет время – можно сделать несколько контрольных запусков чтобы убедиться в этом. Наш проект использует принцип пересечения множеств. Также, если останется время, можно рассказать и о других свойствах множеств (объединение, разность, симметричная разность, подмножество и надмножество) или дать этот материал в качестве домашнего задания. При объяснении рекомендую использовать наглядность.