

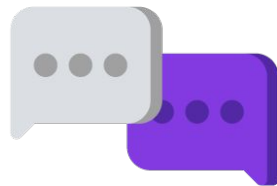
Модуль 6. Урок 2.

Списки

Ссылка на методичку 

Обсуждение:

Игра «Вопрос- ответ»



Данные в игре «Вопрос-ответ»

В прошлый раз вы написали первую программу на Pygame — игру-шутку «Вопрос-ответ».

Ведущий разработчик Костя внимательно пересмотрел код ваших программ и нашёл один существенный недостаток.

Какой недостаток нашёл Костя?



Алёна,
менеджер проектов



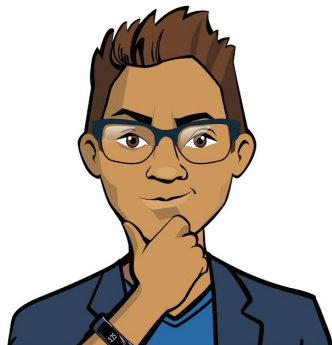
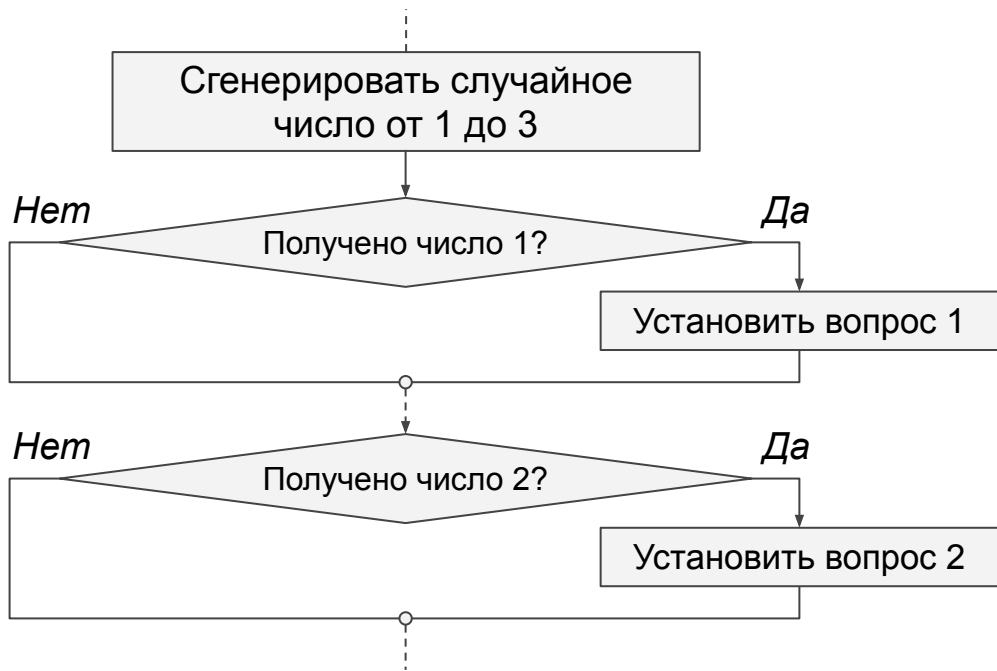
Обсуждение
рабочих задач



Данные в игре «Вопрос-ответ»

Вспомните, как происходил выбор случайного вопроса или ответа.

Необходимо было искать выбранное число.



Обсуждение
рабочих задач



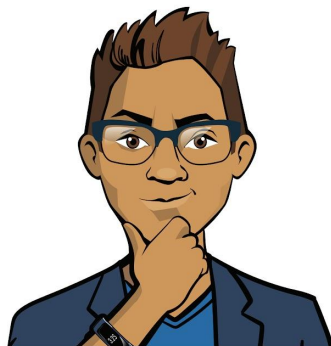
Данные в игре «Вопрос-ответ»

Вспомните, как происходил выбор случайного вопроса или ответа.

Необходимо было искать выбранное число.

```
num = randint(1,3)
if num == 1:
    quest_card.set_text('Что изучаешь в Алгоритмике?', 25)
if num == 2:
    quest_card.set_text('На каком языке говорят во Франции?', 25)
if num == 3:
    quest_card.set_text('Что растёт на яблоне?', 35)
```

*Это код всего лишь для трёх вопросов!
А что если вопросов будет 10 или 100?*



Обсуждение
рабочих задач



Данные в игре «Вопрос-ответ»

Вспомните, как происходил выбор случайного вопроса или ответа.

Необходимо было искать выбранное число.

```
num = randint(1,3)
```

```
if num == 1:
```

```
    quest_card.set_text('Что изучаешь в Алгоритмике?', 25)
```

```
if num == 2:
```

```
    quest_card.set_text('На каком языке говорят во Франции?', 25)
```

```
if num == 3:
```

```
    quest_card.set_text('Что растёт на яблоне?', 35)
```

```
#...
```

```
if num == 100:
```

```
    quest_card.set_text('Что падает с неба во время снегопада?', 25)
```

Чтобы дойти до 100-го вопроса, придётся
выполнить **100 проверок!**

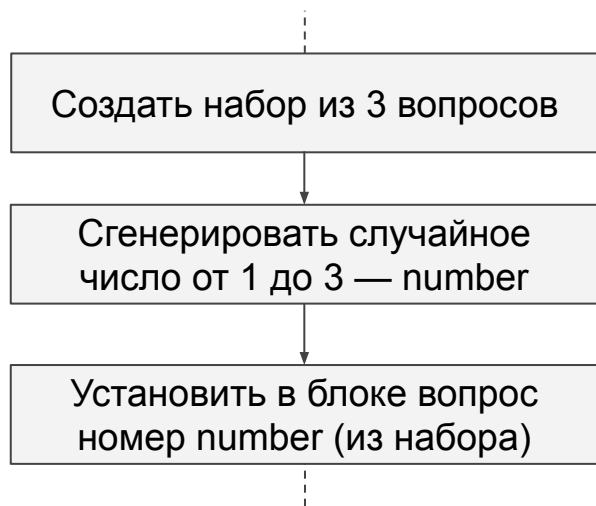


Обсуждение
рабочих задач

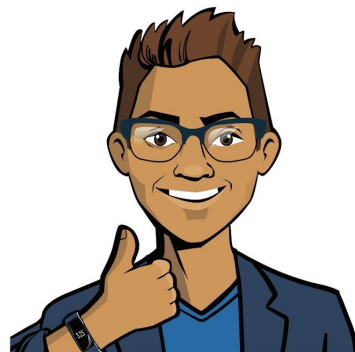


Данные в игре «Вопрос-ответ»

Более удобным было бы хранение всех вопросов вместе и обращение к ним по номеру!



С таким подходом нет необходимости выполнять поиск сгенерированного номера.



Обсуждение
рабочих задач



Данные в игре «Вопрос-ответ»

Структуры, хранящие наборы значений, хорошо знакомы программистам.

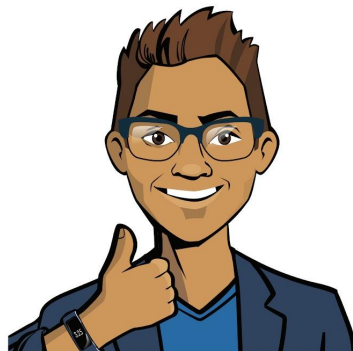
Необходимость хранить данные вместе возникает очень часто, поэтому **структуры данных** включены в стандартную библиотеку Python.

Вопрос1	Вопрос2	Вопрос3
0	1	2

Пронумерованный набор вопросов.

«Выдать 1-й элемент
набора вопросов»

Вопрос2



Обсуждение
рабочих задач



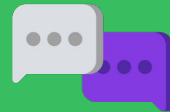
Данные в игре «Вопрос-ответ»

Структурированные наборы данных используются:

- в компьютерных играх;
- в оконных приложениях;
- в мобильных приложениях;
- в создании сайтов

и т. д.

*Изучим работу **со списками** и разнообразим игру «Вопрос-ответ» новыми данными.*



Обсуждение
рабочих задач



Цель рабочего дня —

*изучить работу со списками данных и
улучшить игру «Вопрос-ответ».*

Сегодня вы:

- изучите устройство списков и готовые методы для работы с ними;
- пройдёте практический тренинг по работе со списками;
- примените знания для совершенствования развлекательной игры «Вопрос-ответ».



Обсуждение
рабочих задач



Подтверждение квалификации



**Покажи, что ты готов к
«мозговому штурму»!**

**Продemonстрируй знания
основ создания игр на
Rugame.**



**Подтверждение
квалификации**



Что такое **Pygame**?

Какие возможности Pygame вы знаете?



Подтверждение
квалификации



pygame — это библиотека для создания игр

Раньше мы работали только со стандартной библиотекой Python.

В pygame есть **модули** с готовыми инструментами для:

- обработки внутриигровых событий;
 - обработки событий внешнего мира;
 - настройки игровых таймеров;
 - настройки интерфейса игры и звуковых эффектов
- и др.

Команда	Назначение
<code>import pygame</code>	Подключение всех возможностей библиотеки PyGame



Подтверждение
квалификации



Как создать игровую сцену?

Как задать ей **цветной** фон?



Подтверждение
квалификации



Сцена игры с цветным фоном

Примечание. Цвет можно задать с помощью палитры RGB.

Команда	Назначение
<code>window = pygame.display.set_mode((500, 500))</code>	Создать окно размера: (ширина, длина).
<code>window.fill(<цвет>)</code>	Залить фон указанным цветом.
<code>pygame.display.update()</code>	Обновить наполнение окна игры.
<code>clock = pygame.time.Clock()</code>	Создать игровой таймер.
<code>clock.tick(40)</code>	Установить обновление сцены с частотой ~40 кадров/сек.



Подтверждение
квалификации



Что такое **палитра RGB**?

Каким образом в ней получаются
разные цвета?



Подтверждение
квалификации

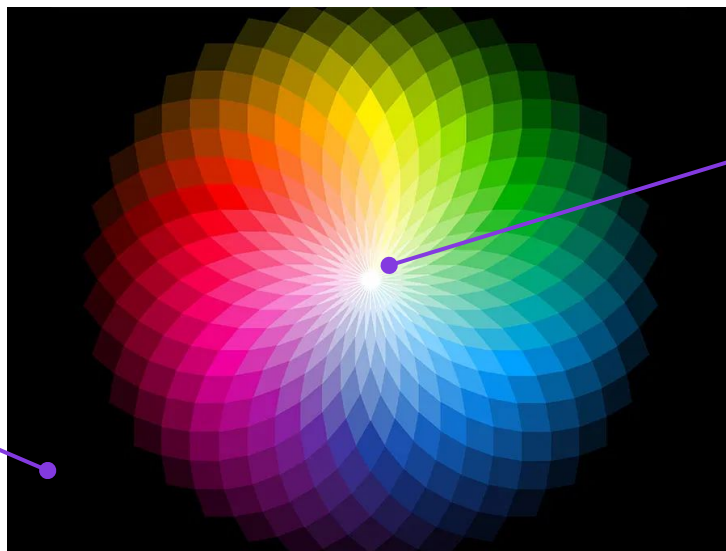


Цветовая палитра RGB (red, green, blue)

Можно использовать цветовую палитру RGB.

Цвета палитры получаются смешением красного, зелёного и синего цветов. Отсутствие цвета задаётся чёрным цветом.

Нет ни одного из трёх базовых цветов — чёрный цвет.



Максимальная насыщенность трёх базовых цветов — белый цвет.

[Ссылка на RGB-калькулятор цветов](#)



Подтверждение
квалификации



Что такое **игровой цикл**?

Как его создать?

Каким будет условие его завершения?

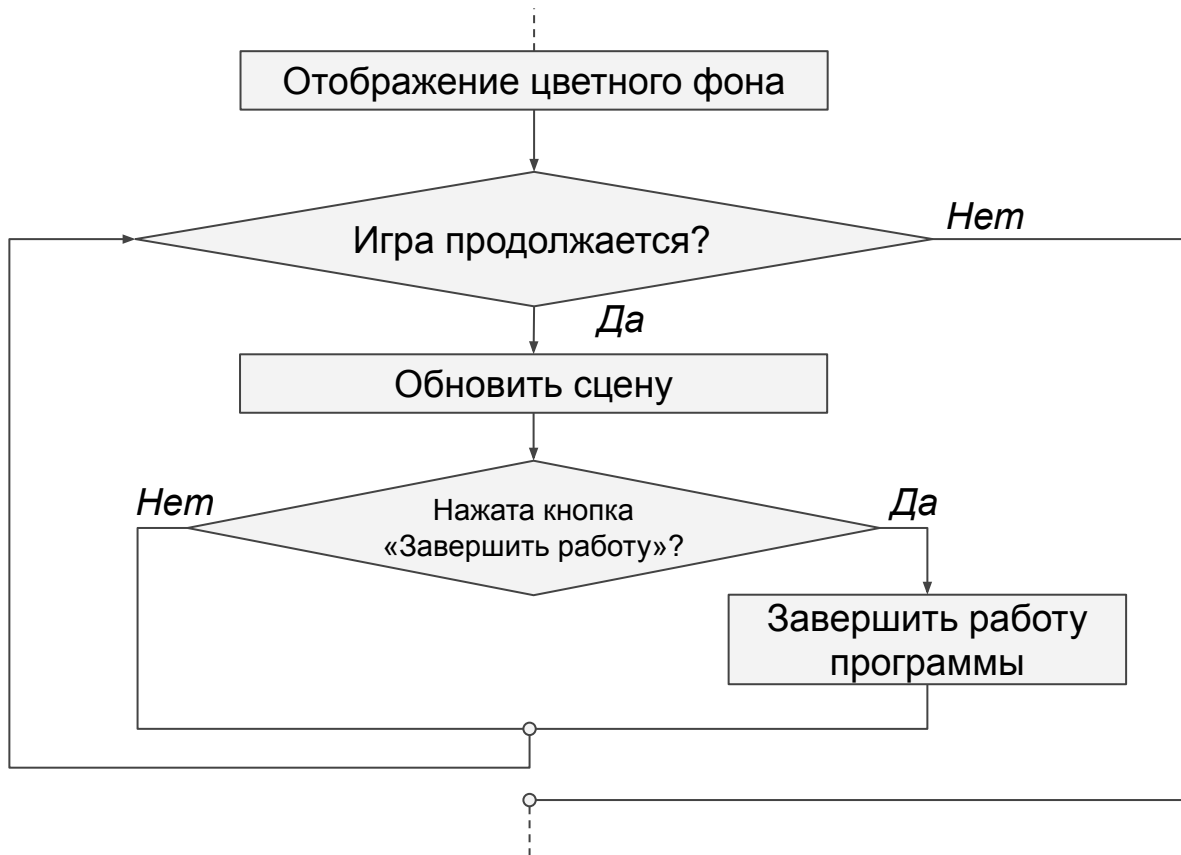


Подтверждение
квалификации



Схема работы игрового цикла

Цикл завершает работу при нажатии на кнопку «Завершить работу»



Подтверждение
квалификации



Игровой цикл для сцены с фоном

Простейшая заготовка игры с отображением сцены без спрайтов:

Подключение модулей Pygame

Создание объекта-сцены

Заливка сцены цветом

← Фон может быть белым

Создание игрового таймера

Игровой цикл:

Установить частоту кадров ~40 кадров/сек

Обновление сцены
(следующий кадр игрового цикла)



Подтверждение
квалификации



Тебе уже известны два типа **готовых объектов из ruyate**. Назови их.

Как их создать?

Для чего они используются?



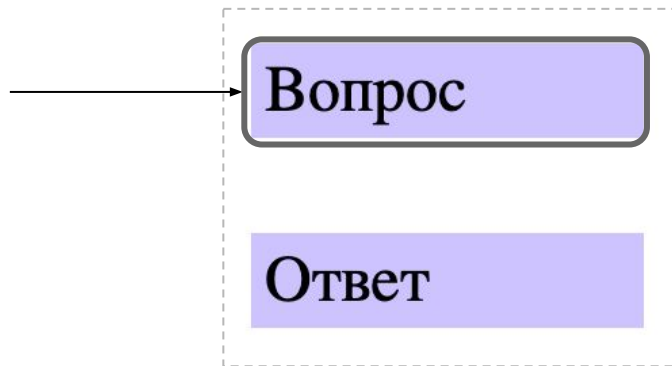
Подтверждение
квалификации



Прямоугольная область Rect

Команда	Назначение
<code>rect = pygame.Rect(x, y, width, height)</code>	Создать прямоугольник в точке (x, y) ширины width и высоты height
<code>pygame.draw.rect(mw, fill_color, rect)</code>	Нарисовать в окне mw прямоугольник rect и залить цветом fill_color

Мы использовали Rect для оформления блоков вопросов и ответов.



Подтверждение
квалификации



Шрифт и текст

Сначала подключается возможность использовать объекты Pygame, затем создаётся объект «шрифт» и после — сама надпись.

Команда	Назначение
<code>pygame.init()</code>	Подключение возможности использовать команды объектов из Pygame.
<code>font1 = font.Font(None, 70)</code>	Установить шрифт / Создание объекта Font с параметрами: шрифт — по умолчанию, кегль — 70.
<code>question = font1.render(text, True, (255, 215, 0))</code>	Создать вопрос с текстом text, нарисованный шрифтом font1, цветом (255, 215, 0).
<code>mw.blit(question, (x, y))</code>	Отобразить в окне mw текст text в точке (x, y).

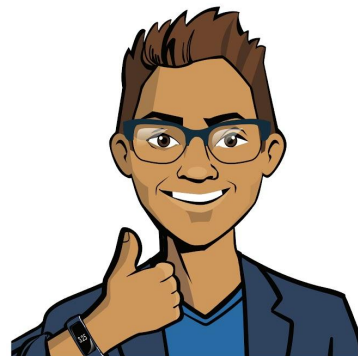


Подтверждение
квалификации



Квалификация подтверждена!

Отлично, вы готовы к «мозговому штурму» и рабочим задачам!



Подтверждение
квалификации



«Мозговой штурм»:

СПИСКИ



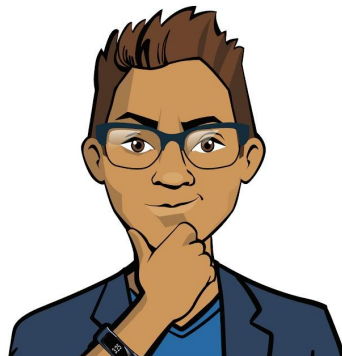
Тренинг по работе со списками

Список — это ещё один тип данных Python. Он используется для хранения упорядоченных наборов данных.

Для работы со списками важно уметь:

1. Создавать список.
2. Управлять содержимым списка (получать значения, добавлять элементы и др.).

Прежде чем приступить к использованию списков в своих проектах, вы пройдёте специальный обучающий тренинг.



«Мозговой
шторм»



Список — это структура для упорядоченного хранения данных разных типов.

Пример. Список результатов турнира по онлайн-игре «SpaceShooter».

```
results = [181, 176, 160, 178, 171, 179, 165]
```

↑
Имя списка

↑
Элементы списка

181	176	160	178	171	179	165
0	1	2	3	4	5	6

↑
Номера элементов списка



«Мозговой
шторм»



Список — это структура для упорядоченного хранения данных разных типов.

Пример. Список результатов турнира по онлайн-игре «SpaceShooter».

```
results = [181, 176, 160, 178, 171, 179, 165]
```

```
print('Лучший результат:', results[0]) → Лучший результат: 181
```

↑
Получение элемента
списка по его номеру
(индексу)



«Мозговой
шторм»



Методы работы со списками

Что требуется сделать?

Какой метод использовать?

Хранение данных и печать
целиком и по частям

Создать список, **получить**
элемент списка, **печатать** список

Добавление новых данных
и удаление ненужных

Добавление и **удаление**
элемента, **очистка** всего списка

Поиск нужных данных в
наборе

Поиск вхождений элемента в
список

Обработка данных для
разных целей

Сортировка элементов списка,
перебор элементов в цикле,
определение **длины** списка



«Мозговой
шторм»

Методы работы со списками

Что требуется сделать?

Какой метод использовать?

Хранение данных и печать
целиком и по частям

Создать список, **получить
элемент** списка, **печатать** список

Добавление новых данных
и удаление ненужных

Добавление и **удаление**
элемента, **очистка** всего списка

Поиск нужных данных в
наборе

Поиск вхождений элемента в
список

Обработка данных для
разных целей

Сортировка элементов списка,
перебор элементов в цикле,
определение **длины** списка

Рассмотрим методы на конкретных задачах.







«Мозговой
шторм»

Работа со списками

Задача 1а. Для участия в онлайн-турнире нужно зарегистрировать свою команду. Напишите программу для записи. Необходимо:

- запрашивать ввод фамилий участников и добавлять их в группу.
- после завершения ввода печатать список группы.

Возможный вид программы:



```
Введите фамилию участника (0 – остановить ввод)
>>> Ермолова
Введите фамилию участника (0 – остановить ввод)
>>> Степанов
Введите фамилию участника (0 – остановить ввод)
>>> Капустина
Введите фамилию участника (0 – остановить ввод)
>>> 0
Группа набрана: ['Ермолова', 'Степанов', 'Капустина']
```

*Что нужно уметь,
чтобы создать такую
программу?*



«Мозговой
шторм»

Работа со списками

Задача 1а. Для участия в онлайн-турнире нужно зарегистрировать свою команду. Напишите программу для записи. Необходимо:

- запрашивать ввод фамилий участников и добавлять их в группу.
- после завершения ввода печатать список группы.

Нам пригодятся функции и методы:

Функция	Назначение
<code>participants = list()</code>	Объявление пустого списка
<code>participants.append('Смирнов')</code>	Добавление элемента в конец списка
<code>print(participants)</code>	Печать списка целиком



«Мозговой
штурм»

Работа со списками

Задача 1а. Для участия в онлайн-турнире нужно зарегистрировать свою команду. Напишите программу для записи. Необходимо:

- запрашивать ввод фамилий участников и добавлять их в группу.
- после завершения ввода печатать список группы.

```
participants = list()
participant = input('Введите фамилию участника (0 - остановить ввод)')
while participant != '0':
    participants.append(participant)
    participant = input('Введите фамилию участника (0 - остановить ввод)')
print('Группа набрана:', participants)
```



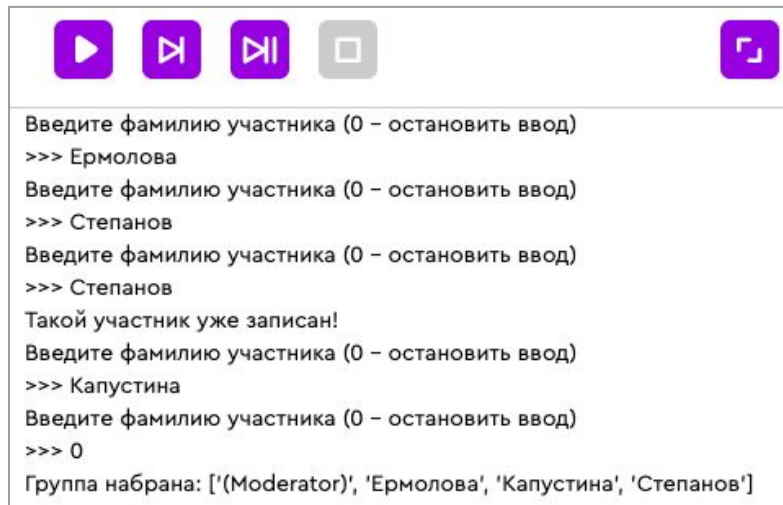
«Мозговой
шторм»

Работа со списками

Задача 16. Организатор турнира попросил доработать программу.

- В списке каждой команды изначально должен быть модератор.
- Защита от невнимательности: если введенная фамилия уже есть в списке, то не добавлять её снова.
- После окончания ввода отсортировать фамилии по алфавиту.

Возможный вид программы:



```
Введите фамилию участника (0 – остановить ввод)
>>> Ермолова
Введите фамилию участника (0 – остановить ввод)
>>> Степанов
Введите фамилию участника (0 – остановить ввод)
>>> Степанов
Такой участник уже записан!
Введите фамилию участника (0 – остановить ввод)
>>> Капустина
Введите фамилию участника (0 – остановить ввод)
>>> 0
Группа набрана: ['(Moderator)', 'Ермолова', 'Капустина', 'Степанов']
```

*Что нужно уметь,
чтобы создать такую
программу?*



«Мозговой
штурм»

Работа со списками

Задача 16. Организатор турнира попросил доработать программу.

- В списке каждой команды изначально должен быть модератор.
- Защита от невнимательности: если введённая фамилия уже есть в списке, то не добавлять её снова.
- После окончания ввода отсортировать фамилии по алфавиту.

Нам пригодятся инструменты:

Команда	Назначение
<code>participants = ['(Moderator)']</code>	Объявление списка с элементами
<code>'Степанов' in participants</code>	Поиск вхождения элемента в список (возвращает True или False)
<code>participants.sort()</code>	Сортировка списка в лексикографическом порядке (по возрастанию чисел и букв алфавита)



«Мозговой
шторм»

Работа со списками

Задача 16. Организатор турнира попросил доработать программу.

- В списке каждой команды изначально должен быть модератор.
- Защита от невнимательности: если введённая фамилия уже есть в списке, то не добавлять её снова.
- После окончания ввода отсортировать фамилии по алфавиту.

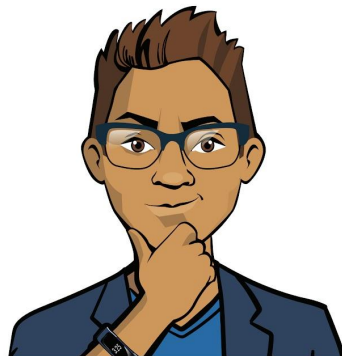
```
participants = ['(Moderator)']
participant = input('Введите фамилию участника (0 - остановить ввод)')
while participant != '0':
    if participant in participants:
        print('Такой участник уже записан!')
    else:
        participants.append(participant)
    participant = input('Введите фамилию участника (0 - остановить ввод)')
participants.sort()
print('Группа набрана:', participants)
```



«Мозговой
шторм»

Перед тем, как перейти к другой задаче:

1. Пользователем были введены фамилии: «Васильков», «Митрофанова», «Илюхин», «Смирнова». Что напечатает программа?
2. Как доработать программу и ограничить возможности ввода десятию фамилиями?



«Мозговой
шторм»

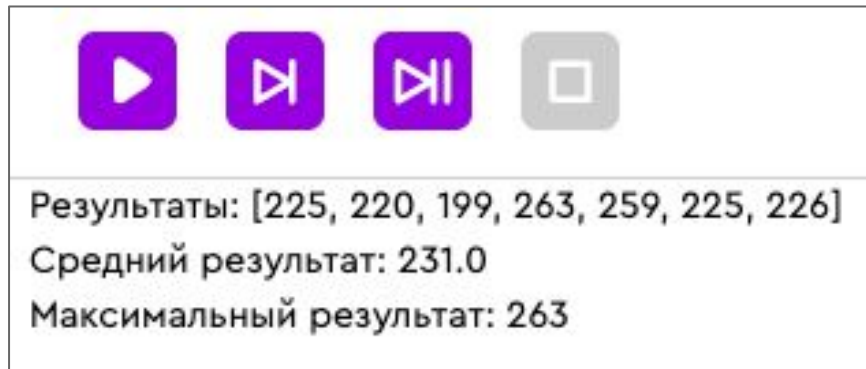


Работа со списками

Задача 2. В конце турнира анализируются результаты всех команд. Группа аналитиков должна:

- подсчитать и напечатать средний результат турнира;
- определить результат команды-победителя.

Возможный вид программы:



*Что нужно уметь,
чтобы создать такую
программу?*



«Мозговой
шторм»



Работа со списками

Задача 2. В конце турнира анализируются результаты всех команд. Группа аналитиков должна:

- подсчитать и напечатать средний результат турнира;
- определить результат команды-победителя.

Нам пригодятся инструменты:

Команда	Назначение
<code>for result in results:</code> Команда1 Команда2	Перебор элементов списка results. “Для каждого элемента (result) списка (results) выполнить Команда1, Команда2”
<code>len(results)</code>	Определение длины списка results



«Мозговой
шторм»

Работа со списками

Задача 2. В конце турнира анализируются результаты всех команд. Группа аналитиков должна:

- подсчитать и напечатать средний результат турнира;
- определить результат команды-победителя.

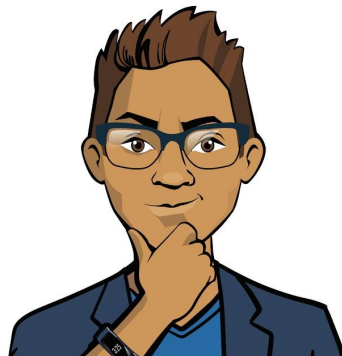
```
results = [225, 220, 199, 263, 259, 225, 226]
average_result = 0
max_result = 0
for result in results:
    average_result += result
    if max_result < result:
        max_result = result
average_result = average_result / len(results)
print('Результаты:', results)
print('Средний результат:', average_result)
print('Максимальный результат:', max_result)
```



«Мозговой
шторм»

Перед тем, как перейти к другой задаче:

1. Что напечатает программа, если загрузить другой набор данных: [200, 204, 202]? Каким будет значение переменной `max_result` на каждом шаге цикла?
2. Как изменить программу, чтобы она печатала не максимальный, а минимальный результат?



«Мозговой
шторм»



Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.



«Мозговой
шторм»



Как написать такую программу? Все необходимые методы вы знаете.

Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.

```
results = list()
amount_200 = 0
result = int(input('Введите результат (0 - стоп):'))
while result != 0:
    if result > 200:
        amount_200 += 1
    results.append(result)
    result = int(input('Введите результат (0 - стоп):'))
print('Результаты раунда:', results)
print('Прошли в следующий раунд:', amount_200)
```



```
Введите результат (0 - стоп):
>>> 220
Введите результат (0 - стоп):
>>> 211
Введите результат (0 - стоп):
>>> 198
Введите результат (0 - стоп):
>>> 185
Введите результат (0 - стоп):
>>> 0
Результаты раунда: [220, 211, 198, 185]
Прошли в следующий раунд: 2
```



«Мозговой
шторм»



Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.

```
results = list()
amount_200 = 0
result = int(input('Введите результат (0 - стоп):'))
while result != 0:
    if result > 200:
        amount_200 += 1
    results.append(result)
    result = int(input('Введите результат (0 - стоп):'))
print('Результаты раунда:', results)
print('Прошли в следующий раунд:', amount_200)
```

Решение верное, но ввод данных можно было бы упростить.

Кстати, метод для упрощения вам известен из другой темы.



«Мозговой
шторм»

Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.

```
results_list = results.split('')
```

Это метод, разбивающий строку на части по указанному разделителю. Из полученных частей составляется список.

Программа	Программа напечатает
<pre>results = '220 211 198 185' results = results.split(' ') print(results)</pre>	<pre>['220', '211', '198', '185']</pre>



«Мозговой
шторм»

Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.

```
results_list = results.split('')
```

— Это метод, разбивающий строку на части по указанному разделителю. Из полученных частей составляется список.

Программа	Программа напечатает
<pre>results = '220 211 198 185' results = results.split(' ') print(results)</pre>	<pre>['220', '211', '198', '185']</pre>

Как переделать прошлую программу?



«Мозговой
шторм»

Планирование рабочих задач

Задача 3. Написать программу, которая запрашивает ввод произвольного числа результатов раунда, сохраняет их в архив (списком) и печатает количество команд, получивших больше 200 очков.

```
results = input('Введите оценки через пробел:')
results = results.split(' ')
amount_200 = 0
for result in results:
    if int(result) > 200:
        amount_200 += 1
print('Результаты раунда:', results)
print('Прошли в следующий раунд:', amount_200)
```



Введите оценки через пробел:

>>> 220 211 198 185

Результаты раунда: ['220', '211', '198', '185']

Прошли в следующий раунд: 2



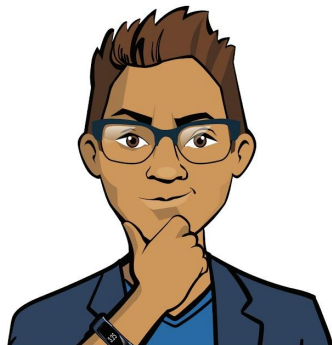
«Мозговой
шторм»



Строки и списки

Вы могли заметить, что строки и списки имеют много общего.

Благодаря интерпретатору Python, стало возможным использовать одни и те же операторы и для работы с несколькими типами данных.



«Мозговой
шторм»



Строки и списки

Работа некоторых методов очень схожа:

Команда	Строки	Списки
<code>marks1 + marks2</code>	Слияние строк в одну	Слияние списков в один
<code>marks * 3</code>	Повтор строки n-раз	Повтор элементов в списке n-раз
<code>len(marks)</code>	Вычисление длины строки (количество символов)	Вычисление длины списка (количество элементов)
<code>marks[i]</code>	Получение символа по номеру	Получение элемента по номеру
<code>marks.find('5')</code>	Поиск вхождения подстроки в строку (возвращает номер входа)	-
<code>'5' in marks</code>	Поиск вхождения символа в строку	Поиск вхождения элемента в список



«Мозговой
шторм»

Строки и списки

Для некоторых методов нет прямых аналогов:

Команда	Строки		Списки	
Добавить элемент	-		<code>marks = [5, 4]</code> <code>marks.append(3)</code>	<code>[5, 4, 3]</code>
Удалить элемент по значению	-		<code>marks = [5, 4, 3]</code> <code>marks.remove(5)</code>	<code>[4, 3]</code>
Заменить элемент другим элементом	<code>marks = '545'</code> <code>m = marks.replace('4', '5')</code>	<code>'555'</code>	-	

↑
Текущая строка не
меняется, создаётся новая.

↑
Меняется текущий список.



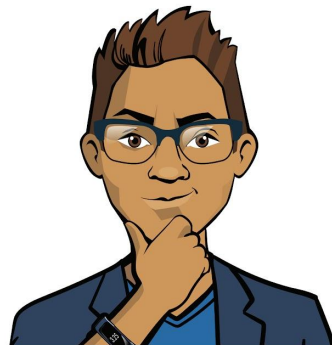
«Мозговой
шторм»



Ваша задача:

- Зайти на платформу и пройти практическую часть тренинга. При необходимости используйте документацию.

Если останется время, изучите дополнительные задачи «Списки: доп. задачи».



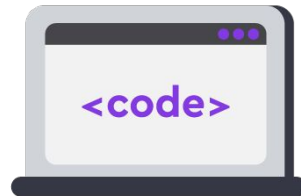
«Мозговой
шторм»



Модуль 6. Урок 2. Списки

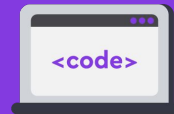
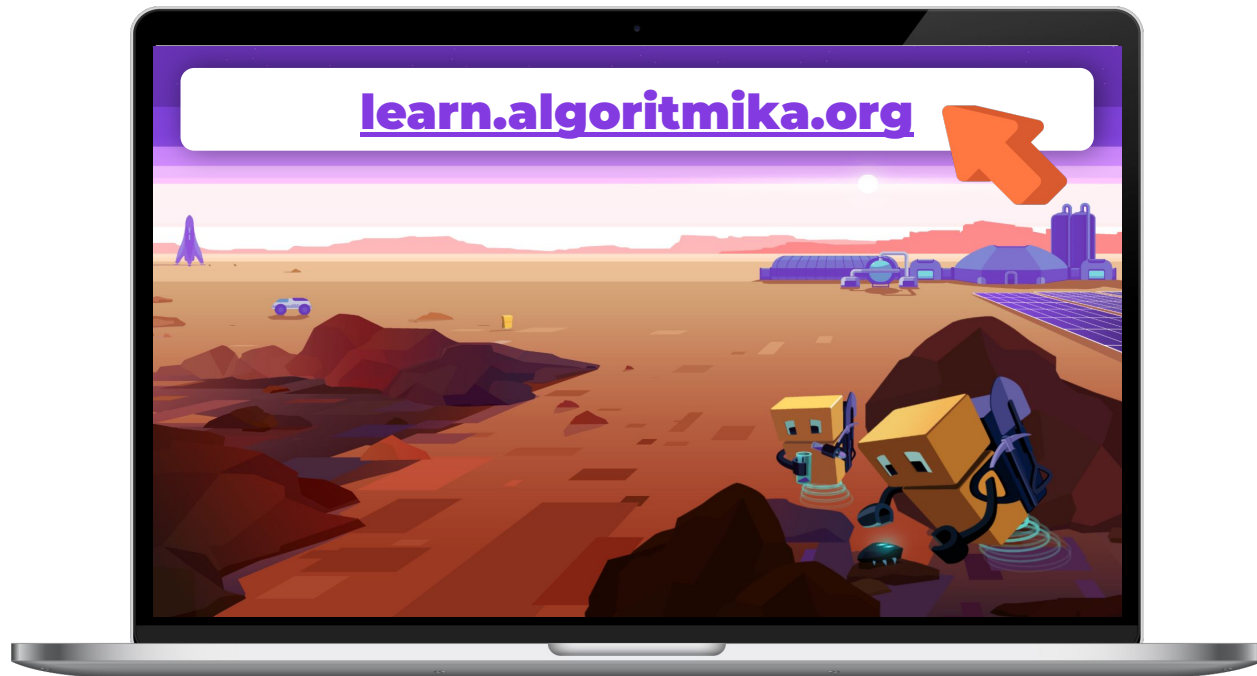
Платформа:

Списки: Тренинг



Выполните задания на платформе

➡ «Списки: тренинг»



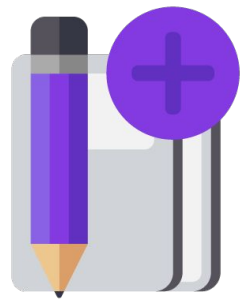
Работаем на
платформе

Перерыв



«Мозговой штурм»:

Хранение данных



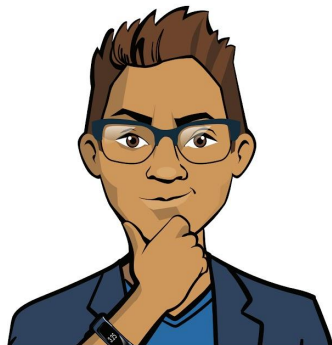
Доработка игры «Вопрос-ответ»

Используем списки, чтобы оптимизировать хранение вопросов и ответов в игре. Необходимо понять:

1. Где будут храниться наборы вопросов и ответов?
2. Как будет происходить отображение случайного вопроса / ответа в игровом цикле?

Решения могут быть разными.

Выберем и разберём внедрение одного из них.



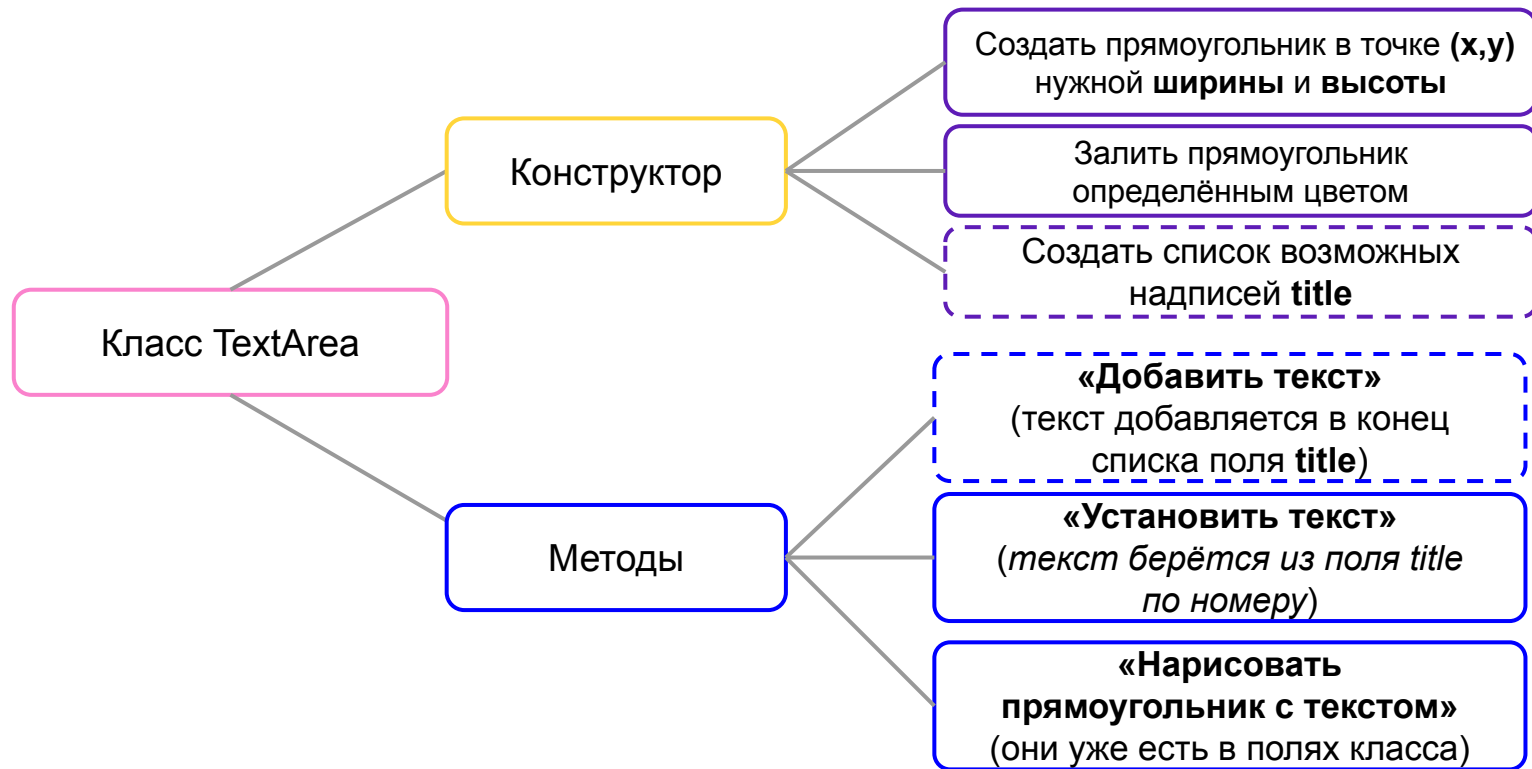
«Мозговой
шторм»



Класс TextArea

Добавим список возможных надписей на блоках в класс.

Для обслуживания списка добавим метод `add_text()` и изменим `set_text()`.

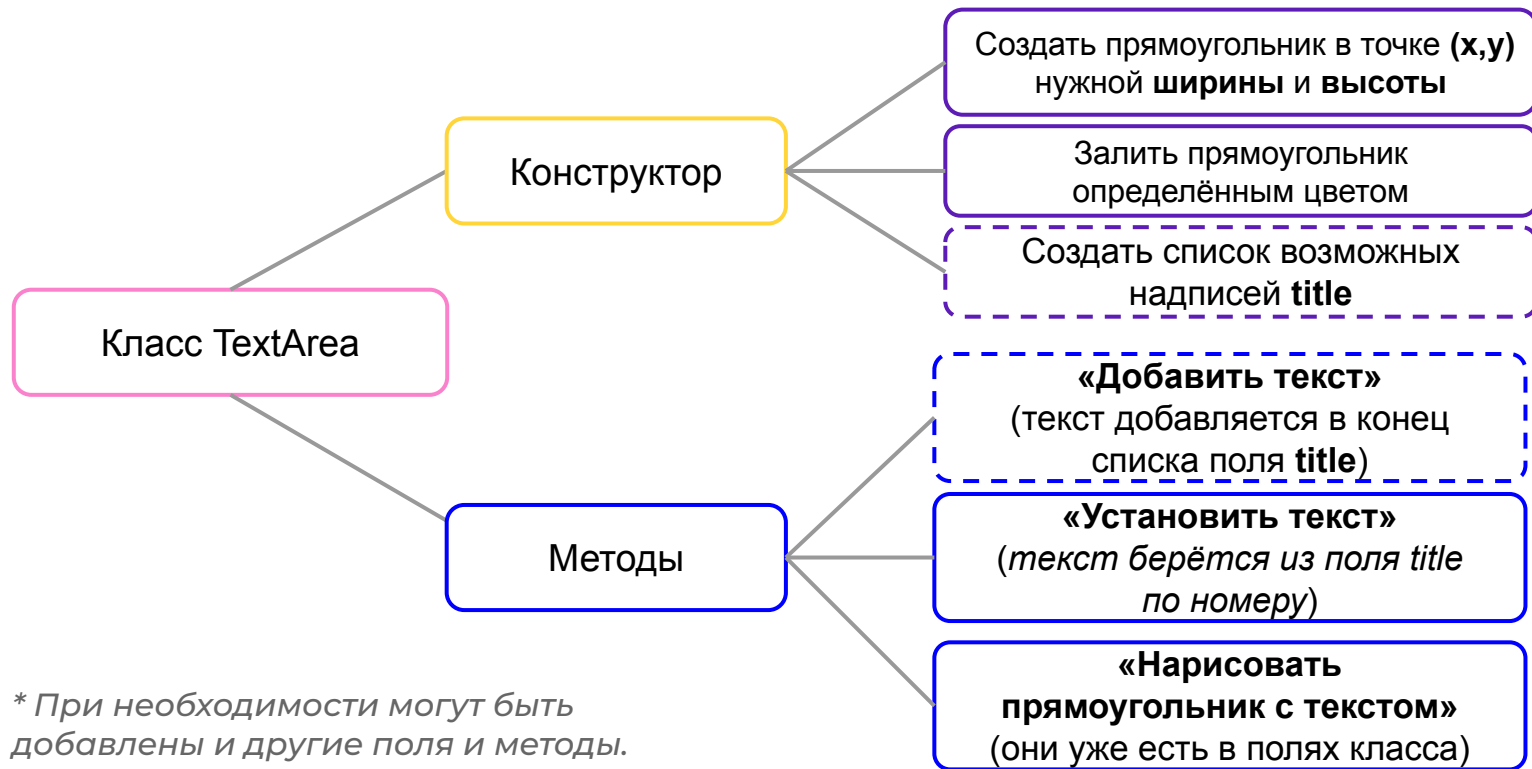


«Мозговой
шторм»

Класс TextArea

Добавим список возможных надписей на блоках в класс.

Для обслуживания списка добавим метод `add_text()` и изменим `set_text()`.



* При необходимости могут быть добавлены и другие поля и методы.



«Мозговой
шторм»

1. Заполнение списка формулировок

При создании экземпляра TextArea для блока с текстом можно сразу добавить возможные формулировки:

Создание экземпляра TextArea

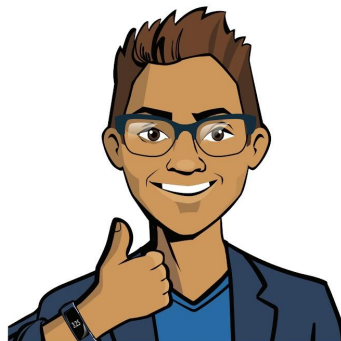
Добавление формулировки 1

Добавление формулировки 2

...

Добавление формулировки n

→ Пусть на 0-м месте лежит
стартовая заготовка «Вопрос».



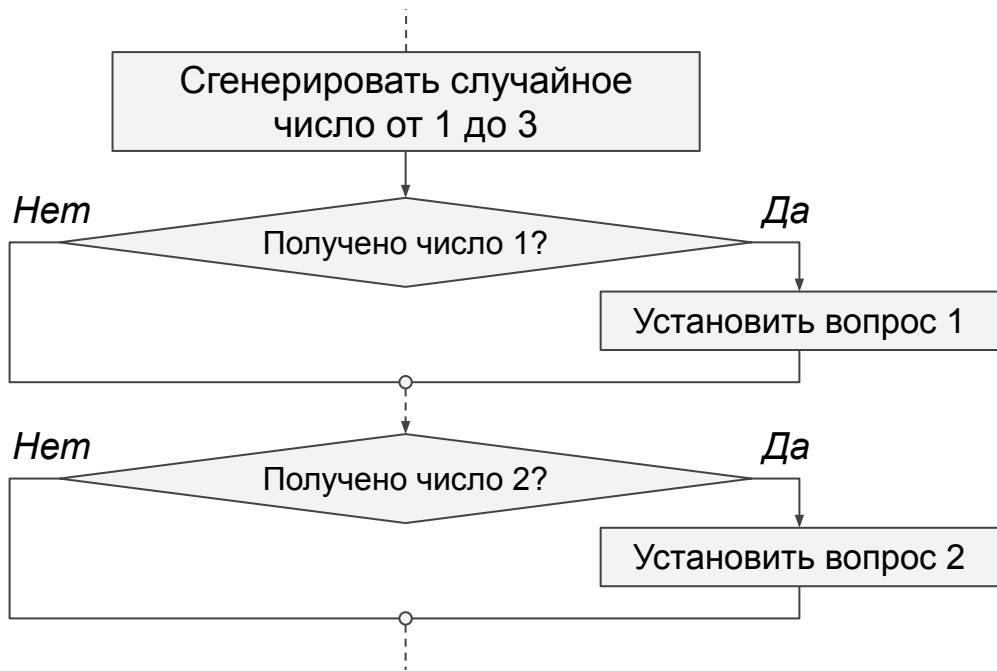
«Мозговой
шторм»



2. Отображение случайной формулировки

Пусть нажата клавиша Q — необходимо показать вопрос.

Раньше случайный вопрос отображался так:

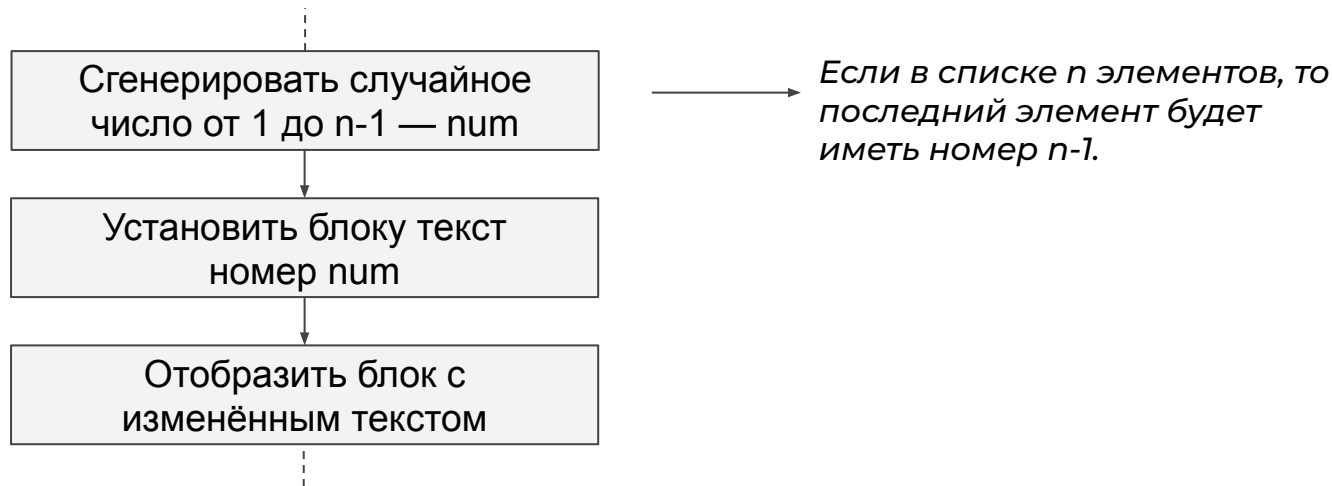


«Мозговой
шторм»

2. Отображение случайной формулировки

Пусть нажата клавиша Q — необходимо показать вопрос.

Теперь можно сразу обратиться к элементу списка.



Как сгенерировать число от 1 до $n-1$, где n — длина списка?

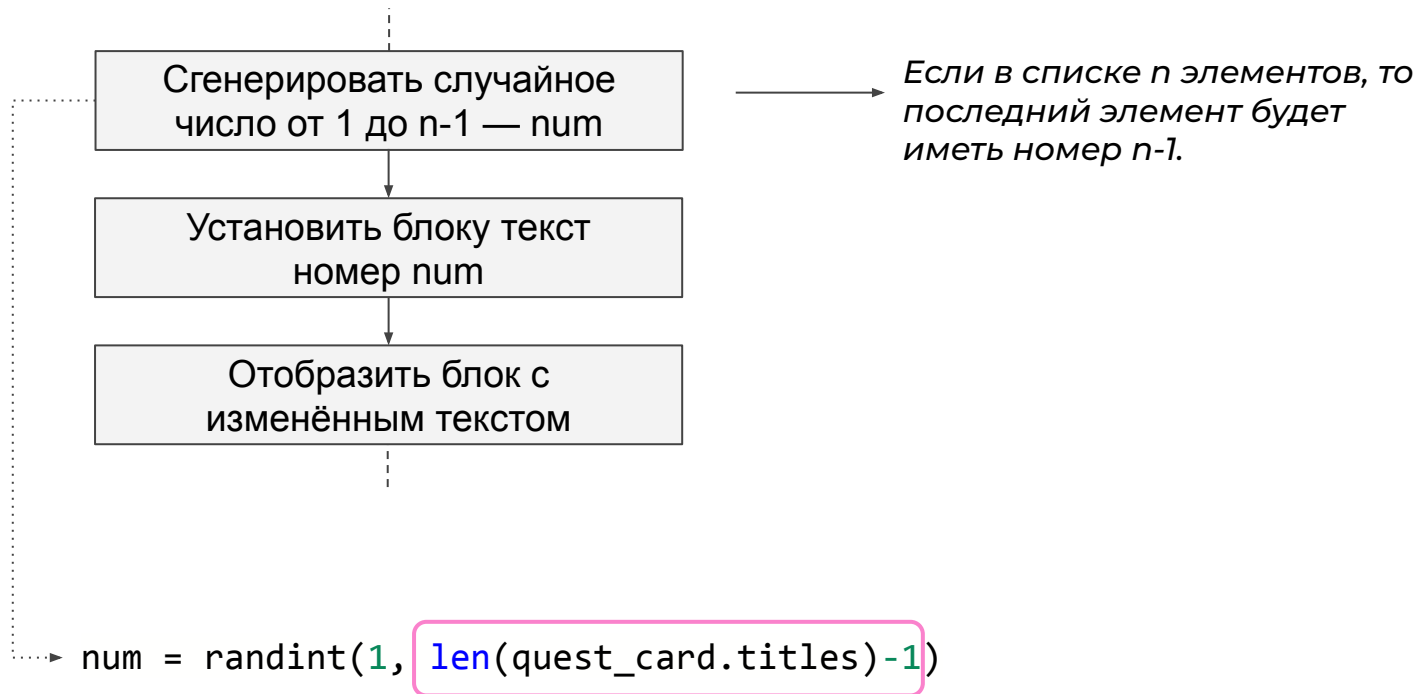


«Мозговой
шторм»

2. Отображение случайной формулировки

Пусть нажата клавиша Q — необходимо показать вопрос.

Теперь можно сразу обратиться к элементу списка.



«Мозговой
шторм»

Ваши задачи:

- Изменить способ хранения данных в игре «Вопрос-ответ». Для этого добавить в класс `TextArea` поле со списком и методы работы с ним.
- Ввести в игру не менее 5 формулировок вопросов и ответов. *Будьте оригинальными!*
- Запустить и протестировать игру.



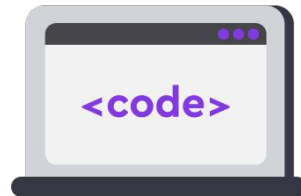
«Мозговой
шторм»



Модуль 6. Урок 2. Списки

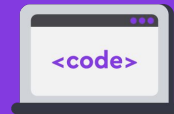
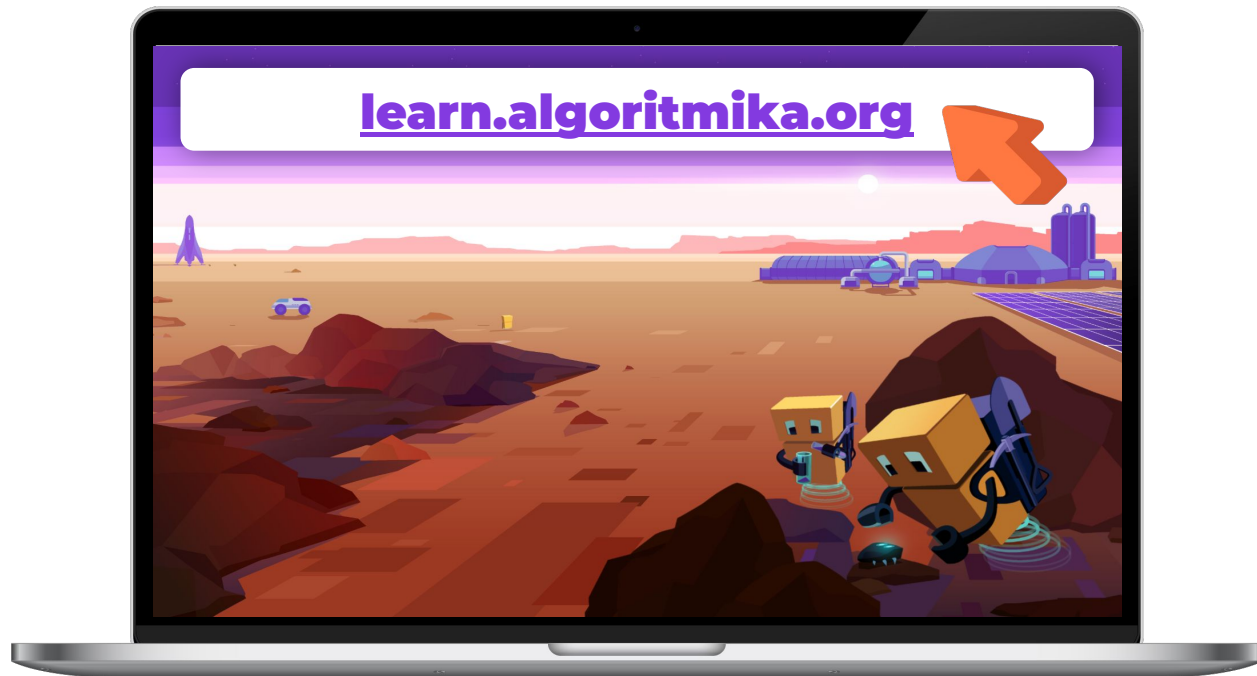
Платформа:

Rudate: игра со списками



Выполните задания на платформе

➡ «Pygame: игра со списком»



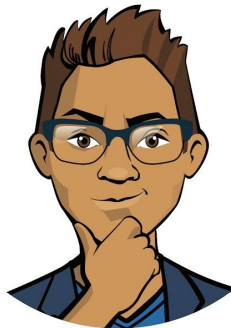
Работаем на
платформе

Завершение рабочего дня



Для завершения пройдите техническое интервью:

1. Что такое список? Для чего он используется?
2. Какие методы работы со списками вы знаете?
3. Может ли список быть полем класса? Как в этом случае добавить в список новый элемент?



Костя,
старший разработчик



Алёна,
менеджер проектов



Подведение итогов
рабочего дня

Отличная работа!

Дорогие коллеги!

Сегодня вы изучили новый мощный тип данных — список — и усовершенствовали игру «Вопрос-ответ».

Поделитесь с коллегами:

1. Что у вас получилось лучше всего?
2. Что получилось не так, как хотелось?
3. Что нужно сделать, чтобы в следующий раз избежать неудачи?



Подведение итогов
рабочего дня