

14.1 Задачи на программ...

15 Работа над мини-прое...

15.2 Числовая угадайка

15.3 Магический шар 8

**Q** 

15.1 Модуль random

15.1 Модуль random 14 из 14 шагов пройдено 11 из 11 баллов получено

# Тема урока: модуль random

- 1. Случайные числа
- 2. Модуль random

**Аннотация.** Урок посвящен модулю random, который содержит функции по работе со случайными числами.

# Случайные числа

Случайные числа широко используются в различных задачах программирования:

кубик, нужны случайные числа для представления значений кубика. Программы, которые раскрывают игральные карты, вынимаемые из перетасованной колоды, используют случайные числа для представления достоинства карт; • случайные числа применяются в программах имитационного моделирования. В некоторых симуляциях компьютер должен

• случайные числа используются в играх. Например, компьютерным играм, которые позволяют игроку подбрасывать игральный

- случайным образом решить, как будет вести себя человек, животное, насекомое или другое живое существо. Нередко конструируются формулы, где случайное число используется для определения различных вариантов действий и событий, происходящих в программе; • случайные числа распространены в статистических программах, случайным образом отбирающих данные для анализа;
- случайные числа используются в компьютерной безопасности для шифрования уязвимых данных.
- Python предлагает встроенные функции для работы со случайными числами. Эти функции хранятся в модуле random в стандартной

Модуль random

## Модуль random предоставляет функции для генерации случайных чисел, букв и случайного выбора элементов последовательности

(списка, строки и т.д.). Для использования этих функций в начале программы необходимо подключить модуль, что делается командой **import**:

```
import random
```

Функция randint()

После подключения модуля мы можем использовать его функции.

Функция randint() принимает два обязательных аргумента а и b и возвращает случайное целое число из отрезка  $[a;\ b]$ .

функции random.randint(2, 9) может быть любое число от 2 до 9 включительно.

Следующий код выводит два случайных целых числа: num1 из отрезка [0; 17] и num2 из отрезка [-5; 5].

```
import random
num1 = random.randint(0, 17)
num2 = random.randint(-5, 5)
print(num1)
print(num2)
      Важно: левая и правая граница а и b включаются в диапазон генерируемых случайных чисел. Результатом вызова
```



import random

import random

отбора значений.

библиотеке.

Следующий код выводит 10 случайных целых чисел из диапазона [1; 100]:

```
for _ in range(10):
     print(random.randint(1, 100))
Среди этих чисел возможны повторения, поскольку каждый раз выбирается случайное.
```

Функция randrange()

### Если вы помните, как применять функцию range(), то почувствуете себя непринужденно с функцией randrange(). Функция

randrange() принимает такие же аргументы, что и функция range(). Различие состоит в том, что функция randrange() не возвращает саму последовательность чисел. Вместо этого она возвращает случайно выбранное число из последовательности чисел. Следующий код присваивает переменной  $\mathsf{num}$  случайное число в диапазоне от 0 до 9:

import random

последовательности чисел от 0 до конечного предела, исключая сам предел.

```
num = random.randrange(10)
Аргумент 10 задает конечный предел последовательности значений. Функция возвратит случайно выбранное число из
```

Следующий код задает начальное значение и конечный предел последовательности:

```
num = random.randrange(5, 10)
```

Следующий код задает начальное значение, конечный предел и величину шага:

Таким образом в переменной num будет храниться случайное число в диапазоне от 5 до 9.

```
import random
num = random.randrange(0, 101, 10)
```

70, 80, 90, 100.

Таким образом в переменной num будет храниться случайное число из последовательности чисел: 0, 10, 20, 30, 40, 50, 60,

### Функция random() Функции randint() и randrange() возвращают случайное **целое число**. А вот функция random() возвращает случайное число с

плавающей точкой (вещественное число). В функцию random() никаких аргументов не передается. Функция random() возвращает случайное число с плавающей точкой в диапазоне от 0.0 до 1.0 (исключая 1.0). Следующий код выводит случайное число с плавающей точкой из диапазона  $[0.0;\ 1.0)$ :

import random

```
num = random.random()
print(num)
                                               Функция uniform()
```

Функция uniform() тоже возвращает случайное число с плавающей точкой, но при этом она позволяет задавать диапазон для

Следующий код выводит случайное число с плавающей точкой из диапазона  $[1.5;\ 17.3]$ : import random

```
num = random.uniform(1.5, 17.3)
print(num)
                          Начальные значения случайного числа
```

Числа, генерируемые функциями модуля random, не являются подлинно случайными. Несмотря на то, что обычно их называют случайными числами, это **псевдослучайные числа**, вычисляемые на основе формулы. Формула, генерирующая случайные числа, должна быть инициализирована начальным значением. Оно используется в вычислении, возвращающем следующее случайное число в ряду. Когда модуль random импортируется, он получает системное время из внутреннего генератора тактовых импульсов компьютера и использует его как начальное значение. Системное время - целое число, представляющее текущую дату и время вплоть до одной сотой секунды. Если бы всегда использовалось одно и то же начальное значение, функции генерации случайных чисел всегда возвращали бы один и тот же ряд псевдослучайных чисел. Поскольку системное время меняется каждую сотую долю секунды, можно без опасений утверждать, что всякий раз, когда импортируется модуль random, будет сгенерирована отличающаяся

чисел. При необходимости для этого можно вызвать функцию seed(), задав начальное значение. Следующий код генерирует 10 случайных чисел, и при этом содержит инструкцию, явно устанавливающую начальное значение для

Вместе с тем, могут иметься некоторые программы, где требуется всегда генерировать одну и ту же последовательность случайных

генератора случайных чисел: import random # явно устанавливаем начальное значение для генератора случайных чисел random.seed(17)

```
Результатом выполнения такого кода может быть:
```

print(random.randint(1, 100))

for \_ in range(10):

4602 7 92

207 Комментариев

Шаг 1

от предыдущих последовательность случайных чисел.

```
67
54
39
```

Если выполнить такой код еще раз, то мы получим ту же самую последовательность псевдослучайных чисел. Примечания

Примечание 1. Подключение модуля следующим образом: from random import \*

```
Примечание 2. Функции модуля random на самом деле являются методами одноименного класса random.
Примечание 3. Функция randint() реализована на основе функции randrange() следующим образом:
```

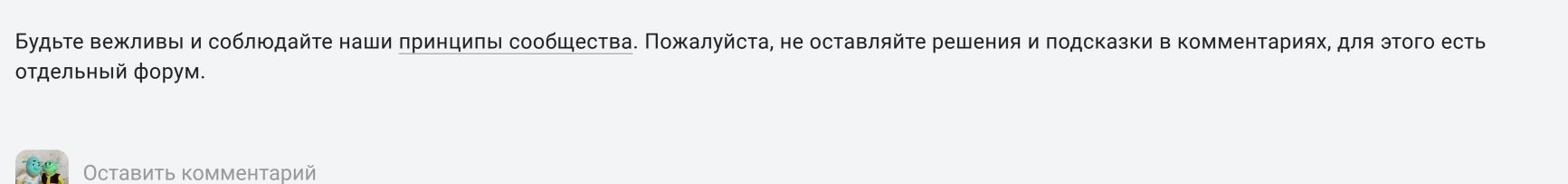
# Return random integer in range [a, b], including both end points. def randint(self, a, b): return self.randrange(a, b + 1)

позволяет в дальнейшем не писать название модуля и символ точки при вызове функций модуля.

```
Happy Pythoning!
```

Следующий шаг 🕻

Самые популярные



```
Показать обсуждения (207)
```