**Генерирование случайных чисел**

В самом начале работы необходимо [импортировать модуль](https://python-scripts.com/import-modules-python" \t "https://python-scripts.com/_blank) random в программу. Только после этого его можно будет полноценно использовать. Оператор для импорта модуля random выглядит следующим образом:

|  |  |
| --- | --- |
| 1 | import random |

Теперь рассмотрим использование самого **модуля random** на простом примере:

|  |  |
| --- | --- |
| 1  2  3  4  5 | import random      print("Вывод случайного числа при помощи использования random.random()")  print(random.random()) |

Вывод:

|  |  |
| --- | --- |
| 1  2 | Вывод случайного числа при помощи использования random.random()  0.9461613475266107 |

## Случайное целое число — randint(a, b) модуль random

* Возвращает случайное целое число Number, такое что a <= Number <= b;
* randint(a,b) работает только с целыми числами;
* Функция randint(a,b) принимает только два параметра, оба обязательны;
* Полученное в результате случайно число больше или равно a, а также меньше или равно b.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | import random      print("Использование random.randint() для генерации случайного целого числа")  print(random.randint(0, 5)) |

## Генерация случайного целого числа — randrange() модуль random

Метод random.randrange() используется для генерации случайного целого числа в пределах заданного промежутка. Скажем, для получения любого числа в диапазоне между 10 и 50.

Шаг показывает разницу между каждым числом заданной последовательности. Шаг по умолчанию равен 1, однако его значение можно изменить.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | import random      print("Генерация случайного числа в пределах заданного промежутка")  print(random.randrange(10, 50, 5)) |

**Задача 1.**

Используя функцию randrange() получите псевдослучайное четное число в пределах от 6 до 12.

**Задача 2.**

Напишите программу, которая запрашивает у пользователя границы диапазона и какое (целое или вещественное) число он хочет получить. Выводит на экран подходящее случайное число.

**Задача 3. Кости**

Даны две игральные кости. Вывести на экран сумму их случайных выпадений. Использовать randint и randrange.

**Задача 4. Датчик настроения**

Сгенерировать рандомные числа от 1 до 3 и вывести соответствующее настроение для числа:

1 - Безудержная радость

2 - Спокойствие и безмятежность

3 - Уныние и грусть

**Цикл while в Python**

Инструкция while в Python повторяет указанный блок кода до тех пор, пока указанное в цикле условие будет оставаться истинным. While - один из самых универсальных циклов в Python, поэтому довольно медленный.

Цикл while записывается следующим образом:

Псевдокод

while <условие>:

<выражение>

При этом выражением может быть как одна так и несколько инструкций. Условием может быть любое истинное или не нулевое значение. Выражение будет повторяться, пока условие будет истинным.

Когда условие становится ложным интерпретатор переводит выполнение программы на строку, следующую за циклом. Рассмотрим следующий пример цикла:

Python

apples = 5

# Запускаем цикл

while(apples > 0):

# Работаем внутри цикла, печатаем сообщение

print('We have', apples, 'apples')

# Уменьшаем счетчик на 1

apples -= 1

# Вышли из цикла

print('We have no apples any more!')

А вот как выглядит результат выполнения:

Терминал - интерактивный режим

>>> count\_apples(5)

We have 5 apples

We have 4 apples

We have 3 apples

We have 2 apples

We have 1 apples

We have no apples any more!

***Бесконечный цикл***

Цикл while становится бесконечным в случае, когда условие цикла никогда не становится ложным. Примером задачи, для реализации которой необходим бесконечный цикл, может быть, например, создание программы "Часы", которая бесконечно будет обновлять и отображать время. Однако, часто бесконечный цикл является ошибкой начинающего программиста, который забыл добавить изменение условия цикла. Например:

num = 1

while num < 10 :

\_\_\_\_print('We are still in cycle!')

Не спешите запускать данный цикл, иначе ваша программа начнет бесконечное выполнение.

**Задача 5 «Битва троллей»**

Исправьте в программе логическую ошибку

# Демонстрирует тот самый ненавистный нам бесконечный цикл

#Героя окружила толпа троллей. Задача программы посчитать сколько троллей

#успеет убить рыцарь перед смертью

рrint ("Baшeгo героя окружила огромная армия троллей.")

рrint ("Смрадными зелеными трупами врагов устланы все окрестные поля.")

рrint ("Одинокий герой достает меч из ножен,готовясь к последней битве в своей жизни.\n")

health=10

trolls= 0

damage= 3

while health!=0 :

trolls=trolls +1

health = health - damage

рrint( "Взмахнув мечом,ваш герой истребляет злобного тролля," \

"но сам получает повреждений на",damage,"очков . \n")

print ( "Ваш герой доблестно сражался и убил",trolls,"троллей.")

print ("Ho увы !Он пал на поле боя . ")

input ("\n\n Haжмитe Enter, чтобы выйти . ")

Трассировка программы (пошаговая симуляция исполнения программы вручную)

|  |  |  |  |
| --- | --- | --- | --- |
| health | trolls | damage | health!=0 |
| 10 | 0 | 3 | True |
| 7 | 1 | 3 | True |
| 4 | 2 | 3 | True |
| 1 | 3 | 3 | True |
| -2 | 4 | 3 | True |
| -5 | 5 | 3 | True |
| -8 | 6 | 3 | True |

Вывод: цикл бесконечный т.к. здоровье никогда не будет равно нулю

Задание: исправьте в программе логическую ошибку

Заменить while health!=0: на while health>0:

**Операторы continue и break**

Оператор continue начинает следующий проход цикла, минуя оставшееся тело цикла ( for или while). Пример:

for letter in 'Стол':

\_\_\_\_if letter == 'т':

\_\_\_\_\_\_\_\_continue

\_\_\_\_print(letter)

Результат выполнения:

С

о

л

Оператор break досрочно прерывает цикл. Пример:

for letter in 'Стол':

\_\_\_\_if letter == 'т':

\_\_\_\_\_\_\_\_break

\_\_\_\_print(letter)

Результат выполнения:

С

**Задача 6 «Считалка»**

Напишите программу, которая считывает целые числа с консоли по одному числу в строке.

Для каждого введённого числа проверить:

* если число меньше 10, то пропускаем это число;
* если число больше 100, то прекращаем считывать числа;

в остальных случаях вывести это число обратно на консоль в отдельной строке.

**Задача 7 «Угадай число»**

Псевдокод игры:

* выбрать случайное число
* до тех пор, пока игрок не отгадает число
* предоставить игроку возможность отгадывать
* поздравить игрока

Доработанный вид алгоритма:

поприветствовать игрока и объяснить ему правила игры

* выбрать случайное число в диапазоне от 1 до 100
* предложить игроку отгадать число
* приравнять порядковый номер попытки к 1
* если число, предположенное игроком, не совпадает с загаданным
  + если оно больше загаданного
    - предложить игроку выбрать число поменьше
  + иначе
    - предложить игроку выбрать число побольше
  + вновь предложить игроку отгадать число
  + увеличить порядковый номер попытки на 1

поздравить игрока с победой

сообщить игроку сколько попыток ему понадобилось на отгадывание числа