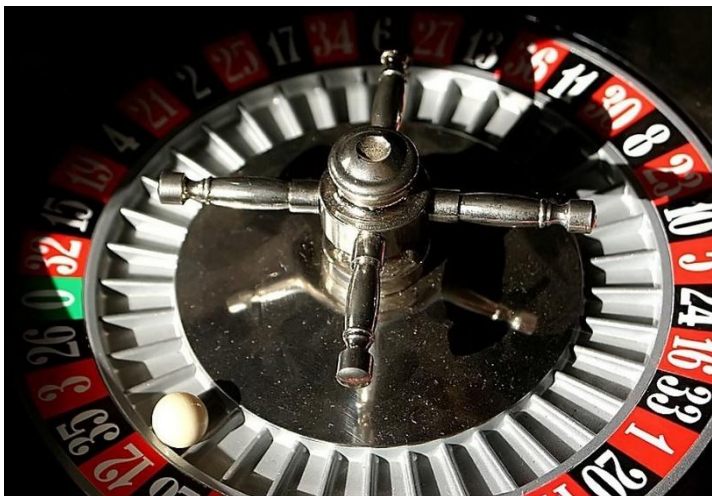




Losowość w Pythonie

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Spis treści

- Losowość i pseudolosowość
- Losowość w Pythonie
- Opcje polecenia random
- Operacje na liczbach losowych
- randrange
- Operacje na listach
- Instrukcje sterujące generatorem liczb

Losowość i pseudolosowość

- Losowość to uzyskanie liczb, sytuacji, zjawisk losowych, czyli nie pozostających ze sobą w związku.
- Prawdziwie losowe zjawiska to np. rozpad promieniotwórczy, promieniowanie kosmiczne, szum cieplny.
- W praktyce w informatyce spotykamy się z procesami pseudolosowymi.
- Są one deterministyczne, ale w praktyce są nieprzewidywalne, co daje efekt losowości.

Losowość w Pythonie

- Moduł **random** umożliwia generowanie losowych liczb czy też losowych wartości z sekwencji.

```
import random  
# Losowa liczba od 0 do 1  
print (random.random())
```

- Generator domyślnie korzysta z aktualnego czasu systemowego.

Opcje polecenia random

- Losowa liczba od 0 do 1 – przedział [0, 1)

```
print (random.random())  
x = random.random()
```

- Zakres liczb losowych – przedział [a,b)

```
print (random.randint(45, 55))
```

- Sekwencja z której zwrócona zostanie losowa wartość

```
import random  
lista = ["Jan", "Ala", "Ola", "Rysiu"]  
print (random.choice(lista))
```

- Wyświetlenie 20 liczb losowych

```
for i in range (20):  
    print (random.random())
```

Operacje na liczbach losowych

- **random()** – zwraca liczbę zmiennoprzecinkową z zakresu 0.00 – 1.00 (zawsze mniejsza od 1.00)

```
print (random.random())
```

- **uniform(a, b)** – zwraca liczbę zmiennoprzecinkową z zakresu a - b

```
print (random.uniform(10, 40))
```

- **triangular(a, b, c)** – zwraca liczbę zmiennoprzecinkową z zakresu a - b, ale zwiększa nieco prawdopodobieństwo wylosowania liczby bliższej c.

```
print ( triangular(10, 40, 30) )
```

- **randint(a, b)** – zwróci liczbę typu całkowitego z zakresu a - b

```
print ( randint(1, 100) )
```

- **randrange(a, b, c)** – zwróci liczbę typu integer od a - b. Ostatni opcjonalny parametr to element z krokiem = c

```
print ( randrange(3, 100, 10) )
```

- **getrandbits(16)** – zwróci liczbę typu integer 16-bitową

```
print ( getrandbits(16) )
```

randrange

- Zakres [0, n)

```
from random import *  
print (randrange(5))  
#losowa wartość spośród 0,1,2,3,4  
x = randrange(5)
```

- Wartość startowa, zakres [x, n)

```
from random import *  
print (randrange(2, 6))  
#losowa wartość spośród 2,3,4,5  
x = randrange(2, 6)
```

- Wartość startowa, zakres, krok

```
from random import *  
print (randrange(2, 15, 3))  
#losowa wartość spośród 2,5,8,11,14  
x = randrange(2, 15, 3)
```

Operacje na listach

- **choice(lista)** – zwraca losowy element z listy

```
lista = ["Jan", "Ala", "Ola", "Rysiu"]  
print (random.choice(lista))
```

- **choices(lista, prawdopodobieństwa)** – zwraca losowy element listy i pozwala na wybranie prawdopodobieństwa wylosowania poszczególnych elementów.

```
lista = ["Jan", "Ala", "Ola", "Rysiu"]  
wagi = [5, 1, 1, 5]  
print (random.choices(lista, wagi))
```

- **shuffle(lista)** – zamienia losowo elementy na liście (ta funkcja zmienia listę, nie zwraca niczego)

```
random.shuffle(lista)  
print(f"Przetaskowana lista: {lista}")
```

- **sample(lista, n)** – zwraca n losowych elementów tablicy

```
print (random.sample(lista, 2))
```


Instrukcje sterujące generatorem liczb

- **Seed** (ziarno czasowe)
- Wskazuje komputerowi na numeryczne miejsce, z którego może wygodnie zacząć generować liczby losowe.
- `random.seed()` – wartość domyślna, czyli aktualny czas systemowy (liczba od początku czasu Uniksa - 1.1.1970)
- `random.seed(50)` – wymusza samodzielną inicjalizację generatora liczb pseudolosowych daną liczbą. Za każdym razem, będzie generował ten sam ciąg liczb pseudolosowych.
- **Zapisywanie i przywracanie stanu generatora**
- **`random.getstate`** - Zapisuje aktualny stan wewnętrzny generatora liczb pseudolosowych, co pozwala na przywrócenie w późniejszym czasie
`stan_przed_zmiana = random.getstate()`
- **`random.setstate()`** – Przywraca zapisany stan generatora
`random.setstate(stan_przed_zmiana)`

Przykład programu

```
import random
random.seed ()
n = int (input ("Ile liczb mam wylosować? "))
x1 = int (input ("Podaj dolny zakres liczb "))
x2 = int (input ("Podaj górny zakres liczb "))
lista = []
for i in range (n):
    lista.append (random.randrange (x1, x2))
print (lista)
for i in range (5):
    random.shuffle(lista)
    print (lista)
print (random.sample(lista, 3))
```

Ćwiczenie

- 1. Rzut monetą.** Napisz program symulujący rzut monetą. Efektem jest *Orzeł* lub *Reszka*.
- 2. Rzut kośćmi.** Napisz program pozwalający na wybór ilości kostek wyrzucających liczbę oczek od 1 do 6.
- 3. Totolotek (Lotto).** Napisz program losujący 6 liczb z zakresu od 1 do 49.
 - a) Liczby nie mogą się powtarzać.
 - b) Dodaj możliwość obstawiania zakładów. Program ma sprawdzać ile z obstawianych liczb wypadło i obliczać ewentualną nagrodę.

Powtórzenie

- W jakim celu stosujemy zjawiska losowe?
- Czym się różni losowość od pseudolosowości
- Jaki moduł zapewnia operacje losowe w Pythonie?
- Omów poszczególne parametry modułu random.
- Co to jest ziarno czasowe?