



# Gra loteria liczbowa

Opracował : Andrzej Nowak

Bibliografia: Python – Kurs programowania na prostych przykładach (K. Jagaciak;Komputer Świat, 2019)  
Python dla każdego. Podstawy programowania (M. Dawson; Helion, 2014)

## Zadanie

Napisz prostą grę wzorowaną na popularnej loterii liczbowej, która polega na podaniu przez gracza zestawu sześciu liczb, tak by trafić jak najwięcej z sześciu wylosowanych.

### 1 Będziemy korzystać z losowania liczb.

Potrzebne do tego polecenia nie znajdują się standardowo w języku Python. Żeby z nich skorzystać, musimy zaimportować moduł `random`.

Od tego zaczynamy skrypt .

```
File Edit Format
import random
```

### 2 Nasza gra będzie pozwalała na wielokrotną rozgrywkę.

Możemy stworzyć zmienną `gramy` i nadać jej wartość `tak`,

```
File Edit Format
import random

gramy = "tak"
```

a samą rozgrywkę umieścić wewnątrz pętli `while`, która będzie się wykonywać, dopóki zmienna `gramy` będzie miała wartość `tak`.

```
import random

gramy = "tak"

while gramy == "tak":
```

### 3 Skupmy się jeszcze na tym, co można zrobić przed pętlą `while`.

Zarówno podane przez użytkownika, jak i wylosowane przez program liczby musimy jakoś przechować.

Wiemy już, że dane możemy przechowywać w zmiennych. Tworzenie teraz dwunastu nowych zmiennych – bo tyle liczb musimy przechować – byłoby rozwiązaniem dość kiepskim. Są lepsze sposoby. Jednym z nich są listy.

Lista to struktura danych, która pozwala na przechowywanie wielu elementów pod tą samą nazwą.

Potrzebujemy dwóch takich dużych pojemników na dane: jednego przeznaczonego na liczby podane przez użytkownika i drugiego – na liczby wylosowane przez program. Jeszcze przed pętlą trzeba zatem zdefiniować dwie listy. Robimy to, przypisując do nazwy listy pusty kwadratowy nawias .

Tworzymy tak listy `podane` i `wylosowane`.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
```

4 Ponieważ gra polega na prawidłowym wytypowaniu sześciu liczb, zarówno losowanie, jak i pytanie gracza o kolejną liczbę powinno wykonać się 6 razy.

Jakiej instrukcji należy użyć, aby coś wykonało się 6 razy?

Oczywiście, potrzebna jest pętla `for`.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
```

5 We wnętrzu tej pętli powinniśmy pytać gracza o kolejne liczby, a ich wartości dodawać do listy `podane`.

W tym celu skorzystamy z polecenia `podane.append()` – to funkcja pozwalająca na dodawanie do listy elementów umieszczonych w nawiasie.

Jeśli w nawiasie umieścimy polecenie `input()`, do listy trafią wartości podane przez gracza.

Możemy jeszcze te wartości rzutować na typ `integer`. Używamy zatem polecenia `podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))`.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
```

Jeśli odpowiednio sformułujemy to, co ma wyświetlić polecenie `input()` i wykorzystamy zmienną `i`, której wartość rośnie z kolejnymi przejściami pętli, program będzie mógł wypisywać, którą z kolei liczbę gracz ma podać.

6 W tej samej pętli możemy dodawać liczby do listy liczb wylosowanych.

W tym celu użyjemy polecenia `wylosowane.append()`

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1, 50))
```

jednak w tym wypadku w nawiasie musimy użyć polecenia do losowania liczb. Znajduje się ono w module `random`, który importowaliśmy wcześniej. Potrzebne nam polecenie nazywa się `randint`, a w nawiasie podajemy zakres, z którego ma pochodzić wylosowana liczba. Powiedzmy, że będzie to liczba od 1 do 50.

**7** Kolejnym etapem rozwiązywania zadania powinno być sprawdzanie, ile liczb udało się graczowi trafić.

Liczbę tę będziemy przechowywać w zmiennej trafione .

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
```

Tworzymy ją i na początku dajemy jej wartość 0. Potem będziemy tę wartość zwiększać, jeśli któraś z liczb zostanie trafiona przez gracza.

**8** Jak będzie wyglądać logika tego sprawdzania?

Powinniśmy brać liczbę z podanych przez gracza i porównywać ją z kolejnymi liczbami z listy wylosowanych. Jak to zrobić? Użyjemy do tego pętli `for` w nieco innej formie.

Zapisanie `for z in podane`:

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
```

sprawia, że pętla wykona się tyle razy, ile elementów jest na liście `podane`.

Zmienna `z`, której użyliśmy w pętli, nie będzie natomiast przybierała kolejnych rosnących wartości od zera.

Będzie ona przybierała wartości kolejnych elementów z listy `podane`.

**9** Następnie każdą liczbę z podanych przez gracza powinniśmy porównywać z każdą kolejną wylosowaną.

Dlatego w napisanej przez nas pętli tworzymy drugą pętlę `for` „przechodzącą” przez elementy listy `wylosowane`. Robimy to za pomocą polecenia `for j in wylosowane`:

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
```

**10** Kolejnym krokiem powinno być teraz porównanie wartości, jakie zostaną umieszczone w `z` oraz w `j`.

W tym celu w naszym przykładzie użyjemy `if`, czyli `if`.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
```

**11** Jeśli liczby te okażą się takie same, zwiększamy wartość zmiennej `trafione` o 1.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
                trafione = trafione + 1
```

**12** Dalej wychodzimy z pętli `for` i piszemy, co ma wykonywać się w pętli `while`.

Zauważmy, że kolejne linijki mają już tylko jedno wcięcie.

Poprzez `print`

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
                trafione = trafione + 1

    print("Twój wynik to: "+str(trafione))
```

należałoby wpisać, ile było trafionych liczb.

### 13 Można również, korzystając z pętli `for`,

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
                trafione = trafione + 1

    print("Twój wynik to: "+str(trafione))
    print("Wylosowane liczby:")
    for i in wylosowane:
        print(i)
```

wypisać liczby, które wylosował program.

### 14 Aby możliwe było przeprowadzenie kolejnej rozgrywki, trzeba wyrzucić wszystkie elementy z naszych dwóch list.

Robimy to, pisząc nazwę listy, a następnie stawiając kropkę i wywołując dla listy polecenie `clear()`.

```
import random

gramy = "tak"

podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
                trafione = trafione + 1

    print("Twój wynik to: "+str(trafione))
    print("Wylosowane liczby:")
    for i in wylosowane:
        print(i)
    podane.clear()
    wylosowane.clear()
```

**15** Na sam koniec możemy nadać nową wartość zmiennej `gramy`,

```
import random

gramy = "tak"

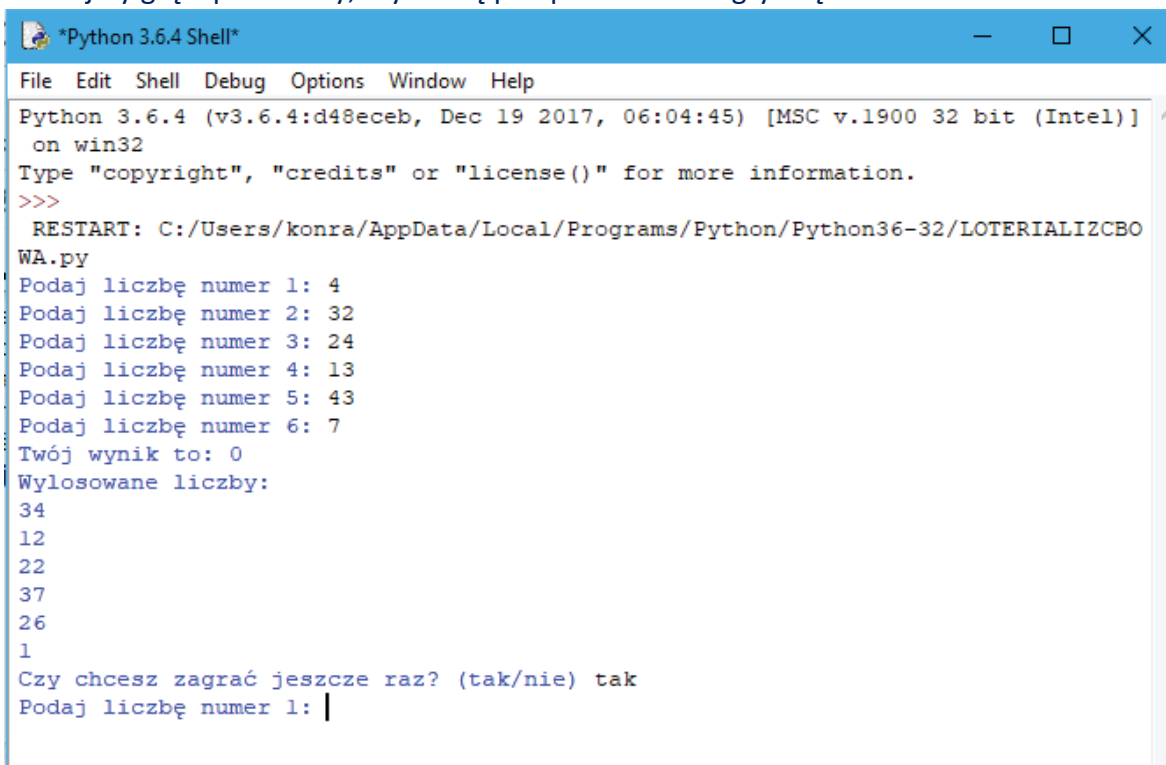
podane = []
wylosowane = []

while gramy == "tak":
    for i in range(6):
        podane.append(int(input("Podaj liczbę numer "+str(i+1)+" : ")))
        wylosowane.append(random.randint(1,50))
    trafione = 0
    for z in podane:
        for j in wylosowane:
            if z == j:
                trafione = trafione + 1

    print("Twój wynik to: "+str(trafione))
    print("Wylosowane liczby:")
    for i in wylosowane:
        print(i)
    podane.clear()
    wylosowane.clear()
    gramy = input("Czy chcesz zagrać jeszcze raz? (tak/nie) ")
```

pytając o nią gracza. Jeśli wpisze on *tak*, program uruchomi kolejną kolejkę rozgrywki. A jeżeli wpisze co innego, gra się zakończy.

**16** Przetestujmy grę. Sprawdźmy, czy da się przeprowadzić rozgrywkę .



```
*Python 3.6.4 Shell*
File Edit Shell Debug Options Window Help
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:04:45) [MSC v.1900 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/konra/AppData/Local/Programs/Python/Python36-32/LOTERIALIZCBO
WA.py
Podaj liczbę numer 1: 4
Podaj liczbę numer 2: 32
Podaj liczbę numer 3: 24
Podaj liczbę numer 4: 13
Podaj liczbę numer 5: 43
Podaj liczbę numer 6: 7
Twój wynik to: 0
Wylosowane liczby:
34
12
22
37
26
1
Czy chcesz zagrać jeszcze raz? (tak/nie) tak
Podaj liczbę numer 1: |
```

## UWAGA:

*Gra nie jest idealnym odzwierciedleniem loterii, w której z bębna wypadają kulki z liczbami.*

*W naszym programie zarówno gracz, jak i program mogą powtarzać liczby.*

*Może to powodować drobne problemy i błędy.*

*Spróbujmy samodzielnie tak zmienić skrypt, aby wylosowanie dwukrotnie tej samej liczby było niemożliwe.*