Module random trong Python cung cấp nhiều hàm để tạo số ngẫu nhiên, chọn ngẫu nhiên phần tử từ danh sách, xáo trộn dữ liệu, v.v. Dưới đây là ví dụ cụ thể cho từng hàm phổ biến:

**1. random.random()**

Sinh một số thực ngẫu nhiên trong khoảng [0.0, 1.0).

import random

print(random.random()) # Ví dụ: 0.6374281388607985

**2. random.uniform(a, b)**

Sinh một số thực ngẫu nhiên trong khoảng [a, b].

print(random.uniform(10, 20)) # Ví dụ: 15.743923

**3. random.randint(a, b)**

Sinh số nguyên ngẫu nhiên trong khoảng [a, b].

print(random.randint(1, 100)) # Ví dụ: 42

**4. random.randrange(start, stop, step)**

Sinh số nguyên ngẫu nhiên từ start đến stop với bước nhảy step.

print(random.randrange(1, 10, 2)) # Chỉ có thể ra: 1, 3, 5, 7, 9

**5. random.choice(sequence)**

Chọn một phần tử ngẫu nhiên từ danh sách, tuple hoặc chuỗi.

items = ["apple", "banana", "cherry"]

print(random.choice(items)) # Ví dụ: "banana"

**6. random.choices(sequence, k=n)**

Chọn n phần tử ngẫu nhiên (có thể trùng) từ danh sách.

print(random.choices(items, k=2)) # Ví dụ: ['apple', 'cherry']

**7. random.sample(sequence, k=n)**

Chọn n phần tử ngẫu nhiên (không trùng lặp).

print(random.sample(items, 2)) # Ví dụ: ['cherry', 'banana']

**8. random.shuffle(sequence)**

Xáo trộn danh sách.

numbers = [1, 2, 3, 4, 5]

random.shuffle(numbers)

print(numbers) # Ví dụ: [3, 5, 1, 4, 2]

**9. random.seed(n)**

Đặt hạt giống để tái tạo kết quả ngẫu nhiên giống nhau.

random.seed(10)

print(random.randint(1, 100)) # Luôn cho kết quả như nhau mỗi lần chạy

import random

random.seed(42) # Đặt seed

print("Dùng seed:")

for \_ in range(3):

print(random.randint(1, 100))

🔁 Chạy lần nào cũng ra:

82

15

4

→ Bất kể bạn chạy bao nhiêu lần, kết quả luôn giống nhau.

**✅ Ví dụ so sánh: Không dùng seed vs. Có dùng seed**

**🔹 Không dùng seed:**

python

CopyEdit

import random

print("Không dùng seed:")

for \_ in range(3):

print(random.randint(1, 100))

🔁 **Chạy lần 1:**

67

23

98

🔁 **Chạy lần 2:**

15

72

39

→ Mỗi lần chạy sẽ ra kết quả khác nhau.

**🔍 Khi nào đặc biệt cần seed?**

* 🔁 Muốn **tái lập kết quả** (ví dụ: mô phỏng, chạy thử thuật toán nhiều lần để so sánh).
* 🎓 Trong giảng dạy: giúp học viên thấy cùng kết quả như giảng viên.
* 🤖 Trong **Machine Learning / AI**: để đảm bảo mô hình huấn luyện có thể **so sánh công bằng**.
* 🐞 Trong **debug**: giúp phát hiện lỗi ở một bước ngẫu nhiên cụ thể.
* **📌 Kết luận:**

| **Không dùng seed** | **Dùng seed** |
| --- | --- |
| Kết quả thay đổi mỗi lần chạy | Kết quả cố định nếu cùng seed |
| Phù hợp cho trò chơi, mô phỏng không cần chính xác lại | Phù hợp cho nghiên cứu, kiểm thử, huấn luyện AI |
| Khó debug | Dễ tái hiện lỗi và kiểm chứng |

|  |
| --- |
| 1. Tìm Số Ngẫu Nhiên: Viết một chương trình yêu cầu người dùng nhập một số nguyên dương n, và số nhỏ nhất **a** và lớn nhất **b**. Hãy phát sinh một danh sách có n số ngẫu nhiên nằm trong khoảng [a, b] và in chúng ra màn hình. |

import random

def tao\_danh\_sach\_ngau\_nhien(n, a, b):

if a > b:

print("Lỗi: Giá trị a phải nhỏ hơn hoặc bằng b.")

return

if n <= 0:

print("Lỗi: n phải là số nguyên dương.")

return

danh\_sach\_ngau\_nhien = [random.randint(a, b) for \_ in range(n)]

print("Danh sách số ngẫu nhiên:", danh\_sach\_ngau\_nhien)

# Nhập dữ liệu từ người dùng

try:

n = int(input("Nhập số lượng số ngẫu nhiên (n): "))

a = int(input("Nhập số nhỏ nhất (a): "))

b = int(input("Nhập số lớn nhất (b): "))

tao\_danh\_sach\_ngau\_nhien(n, a, b)

except ValueError:

print("Vui lòng nhập số nguyên hợp lệ.")

*Nhập số lượng số ngẫu nhiên (n): 4*

*Nhập số nhỏ nhất (a): 1*

*Nhập số lớn nhất (b): 90*

*Danh sách số ngẫu nhiên: [7, 80, 84, 89]*

|  |
| --- |
| 1. Sắp Xếp Dãy Số Ngẫu Nhiên: Viết một chương trình phát sinh một dãy số ngẫu nhiên, sau đó sắp xếp dãy số đó theo thứ tự tăng dần và in ra màn hình. |

import random

def tao\_danh\_sach\_ngau\_nhien(n, a, b):

if a > b:

print("Lỗi: Giá trị a phải nhỏ hơn hoặc bằng b.")

return

if n <= 0:

print("Lỗi: n phải là số nguyên dương.")

return

danh\_sach\_ngau\_nhien = [random.randint(a, b) for \_ in range(n)]

print("Danh sách số ngẫu nhiên ban đầu:", danh\_sach\_ngau\_nhien)

danh\_sach\_ngau\_nhien.sort()

print("Danh sách sau khi sắp xếp tăng dần:", danh\_sach\_ngau\_nhien)

# Nhập dữ liệu từ người dùng

try:

n = int(input("Nhập số lượng số ngẫu nhiên (n): "))

a = int(input("Nhập số nhỏ nhất (a): "))

b = int(input("Nhập số lớn nhất (b): "))

tao\_danh\_sach\_ngau\_nhien(n, a, b)

except ValueError:

print("Vui lòng nhập số nguyên hợp lệ.")

**Bài tập 1**

**🎯 Đề bài: Trò chơi Đoán số từ 1 đến 10**

**📌 Mô tả:**

Viết một chương trình Python đơn giản, cho phép người chơi tham gia trò chơi đoán số. Mỗi lần chơi, chương trình sẽ **tạo ra một số ngẫu nhiên từ 1 đến 10**, và yêu cầu người chơi **nhập một con số để đoán**.

* Nếu người chơi **đoán đúng số** được tạo ra, thông báo **"Bạn đã đoán đúng! Bạn thắng!"**.
* Nếu đoán sai, thông báo **"Sai rồi! Số đúng là X"**.
* Người chơi có thể **chơi nhiều lần**, hoặc **dừng trò chơi bằng cách nhập số 0**.

**🔄 Luật chơi:**

1. Máy sẽ **tạo một số ngẫu nhiên từ 1 đến 10**.
2. Người chơi **nhập số từ 1 đến 10** để đoán.
3. Nếu đoán đúng → hiển thị thông báo thắng.
4. Nếu đoán sai → hiển thị số đúng và cho chơi lại.
5. Nếu người chơi nhập số **0**, chương trình kết thúc và hiển thị **số lần chơi và số lần thắng**.

**🧠 Yêu cầu chức năng:**

1. ✅ Tạo số ngẫu nhiên từ 1 đến 10.
2. ✅ Nhập số người chơi đoán.
3. ✅ So sánh với số đúng và thông báo kết quả.
4. ✅ Cho phép chơi lại nếu nhập sai.
5. ✅ Nếu nhập 0, kết thúc chương trình.
6. ✅ Thống kê: số lần chơi, số lần thắng.

**Bài tập 2**

**🎮 Đề bài: Trò chơi "Tung Đồng Xu May Mắn"**

**📌 Mô tả:**

Viết một chương trình mô phỏng trò chơi tung đồng xu. Mỗi lần chơi, máy sẽ **random tung 1 đồng xu**, có thể là:

* **"Ngửa"**
* **"Sấp"**

Người chơi sẽ **dự đoán kết quả** trước khi đồng xu được tung. Nếu đoán đúng thì thắng, nếu sai thì thua.

Người chơi có thể **chơi nhiều lần** cho đến khi chọn thoát.

**🎯 Yêu cầu chức năng:**

1. ✅ Mỗi lượt chơi, yêu cầu người dùng **nhập dự đoán**: "Ngửa" hoặc "Sấp".
2. ✅ Máy **random** kết quả tung đồng xu (random.choice()).
3. ✅ So sánh với dự đoán, in ra **kết quả thắng hoặc thua**.
4. ✅ Ghi lại **số lượt chơi** và **số lượt thắng**.
5. ✅ Sau mỗi lượt, hỏi người chơi có muốn chơi tiếp không (Y/N).
6. ✅ Nếu không, in ra thống kê tổng kết: số lượt chơi, thắng, thua.

**🧪 Gợi ý kỹ thuật sử dụng:**

* import random
* random.choice(['Ngửa', 'Sấp'])
* while loop để cho phép chơi nhiều lần
* So sánh chuỗi đơn giản (lower() hoặc capitalize() để tránh sai chính tả)

**📤 Ví dụ đầu ra:**

|  |
| --- |
| **===== Trò chơi Tung Đồng Xu =====**  **Dự đoán của bạn (Ngửa/Sấp): Sấp**  **Kết quả: Ngửa → Bạn đã thua!**  **Chơi tiếp không? (Y/N): y**  **Dự đoán của bạn (Ngửa/Sấp): Ngửa**  **Kết quả: Ngửa → Bạn thắng!**  **Chơi tiếp không? (Y/N): n**  **===== Thống kê kết quả =====**  **Tổng số lượt chơi: 2**  **Số lượt thắng: 1**  **Số lượt thua: 1** |