

Le module **random** en Python est utilisé pour générer des nombres pseudo-aléatoires et effectuer des opérations aléatoires. Voici un aperçu des fonctions les plus courantes :

## 1. Génération de nombres aléatoires

- **random.random()**

Renvoie un nombre flottant aléatoire entre 0 (inclus) et 1 (exclus).

```
import random
print(random.random()) # Exemple : 0.678945
```

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

Renvoie un entier aléatoire entre **a** et **b** (inclus).

```
print(random.randint(1, 10)) # Exemple : 7
```

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

Renvoie un nombre flottant aléatoire entre **a** et **b**.

```
print(random.uniform(1.5, 5.5)) # Exemple : 3.14159
```

## 2. Sélection aléatoire dans une liste

- **random.choice(sequence)**

Renvoie un élément aléatoire d'une séquence (liste, chaîne, etc.).

```
fruits = ['pomme', 'banane', 'cerise']
print(random.choice(fruits)) # Exemple : 'banane'
```

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

Renvoie une liste de **n** éléments choisis aléatoirement avec remise.

```
print(random.choices(fruits, k=2)) # Exemple : ['cerise', 'pomme']
```

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

Renvoie une liste de **n** éléments choisis aléatoirement sans remise.

```
print(random.sample(fruits, k=2)) # Exemple : ['pomme', 'cerise']
```

### 3. Mélanger une liste

- **random.shuffle(sequence)**

Mélange les éléments d'une liste sur place.

```
random.shuffle(fruits)
print(fruits) # Exemple : ['cerise', 'pomme', 'banane']
```

### 4. Génération selon des distributions

- **random.gauss(mu, sigma)**

Génère un nombre selon une distribution normale (gaussienne) avec une moyenne **mu** et un écart-type **sigma**.

```
print(random.gauss(0, 1)) # Exemple : -0.2345
```

- **random.expovariate(lambd)**

Génère un nombre selon une distribution exponentielle avec un paramètre **lambd**.

```
print(random.expovariate(1.5)) # Exemple : 0.4567
```

Le module **random** est très pratique pour les simulations, les jeux, ou tout autre besoin de génération aléatoire. Pour des besoins de sécurité (comme les mots de passe), utilisez plutôt le module **secrets**.