

Chapitre 6 : Nombres aléatoires en informatique

La plupart des nombres aléatoires utilisés en informatique sont des nombres pseudo-aléatoires, au sens où ils sont engendrés de manière prédictible en utilisant une formule mathématique.

Ainsi pour "produire des nombres au hasard", on utilise souvent en informatique une suite de nombres déterminée, appelée *séquence pseudo-aléatoire*, qui contient plusieurs centaines de milliers de nombres différents et satisfait plusieurs propriétés statistiques, ce qui donne à penser, en regardant une partie de ces nombres, que ceux-ci sont des nombres complètement aléatoires.

Remarque : Le principe des séquences pseudo-aléatoires est commun à tous les langages de programmation. Cependant, les langages n'utilisent pas forcément tous la même séquence pseudo-aléatoire.

Génération aléatoire de nombres

La bibliothèque `random` (aléatoire)

Les générateurs aléatoires de nombres se trouvent dans la bibliothèque `random` et sont accessibles grâce à la commande :

```
In [ ]: from random import *
```

Génération aléatoire d'entiers

La fonction `randint(a,b)` retourne des nombres compris entre a (inclus) et b (inclus).

Par exemple, pour afficher un entier entre 0 et 100 (inclus), on utilise l'instruction suivante :

```
In [ ]: from random import *

print("nombre aléatoire entre 0 et 100 (inclus) : " , randint(0, 100))
```

et pour afficher un nombre aléatoire entre -20 (inclus) et 20 (inclus) :

```
In [ ]: print("nombre aléatoire entre -20 et 20 (inclus) :", randint(-20,20))
```

Génération aléatoire de flottants

La fonction `random()` permet d'engendrer un nombre flottant aléatoire dans l'intervalle $[0, 1]$.

Pour tirer un nombre dans l'intervalle $[a, b]$, il faut alors multiplier le résultat renvoyé par `random()` par la taille de l'intervalle (soit $b - a$) et décaler l'intervalle en fonction de la borne inférieure de l'intervalle (soit $+a$).

Ainsi le code suivant affiche 3 nombres flottants aléatoires, le premier compris entre 0 et 1, le second compris entre 0 et 10 et le troisième entre 5 et 20.

```
In [ ]: print("Nombre flottant aléatoire entre 0 et 1 : ", random())
        print("Nombre flottant aléatoire entre 0 et 10 : ", 10.0 * random())
        print("Nombre flottant aléatoire entre 5 et 20 : ", 15.0 * random()+5)
```

On peut aussi définir une fonction permettant d'engendrer un flottant aléatoire de l'intervalle $[a, b]$:

```
In [ ]: def randinter(a,b):
        return random()*(b-a)+a

        print(randinter(15,20))
```

Cette fonction est en fait déjà prédéfinie en python sous le nom de `uniform(a,b)`, mais cela fait plutôt figure d'exception.

Différentes générations de nombres aléatoires

Pour s'assurer de ne pas toujours commencer au même endroit de la séquence pseudo-aléatoire et donc ne pas toujours tirer les mêmes valeurs, on utilise un entier appelé *graine*. Cette graine sert à déterminer le nombre de départ dans la séquence pour la fonction `random()`. Il faut toutefois modifier la graine à chaque exécution du programme. Pour cette raison, la graine est souvent engendrée à partir de l'état du système (par exemple, en lisant la valeur de l'horloge). En Python, cette graine est générée automatiquement lors de l'exécution de programmes utilisant des fonctions de la bibliothèque 'random'.

Vocabulaire

- aléatoire, pseudo-aléatoire, graine