♢ Introduction

Peut-on programmer un ordinateur pour générer des nombres aléatoires ? S'ils sont issus d'un algorithme, ces nombres sont-ils *vraiment* aléatoires ? Et au fait, à quoi ça sert ?

Choisissez dans cette feuille ce qui vous fait le plus envie et lancez-vous ! Et surtout, n'hésitez pas à demander de l'aide dès que vous en avez besoin !

♣ Lois de probabilité et petits jeux

Vous pouvez utiliser la fonction `random()` qui rend un nombre choisi aléatoirement entre 0 et 1.

- Comment faire pour jouer à "pile ou face" ?
- Comment demander à l'ordinateur de simuler un lancer de dé ?
- Comment jouer à "Pierre, papier, ciseaux" contre l'ordinateur ? Êtes-vous plus fort que lui ? Provoquez l'ordinateur du voisin en duel !
- Comment tirer un nombre au hasard dans l'intervalle $[0, 2]$? Dans l'intervalle $[-1, 1]$?
- Comment tirer un nombre au hasard dans le disque unité ?

♠ Applications diverses

- Est-il possible de déterminer une valeur approchée de π à l'aide d'un générateur pseudo-aléatoire ?
- Pour ceux qui ont déjà vu les lois binomiales en cours, comment en simuler une ? Comment en déduire une valeur approchée des coefficients binomiaux ?

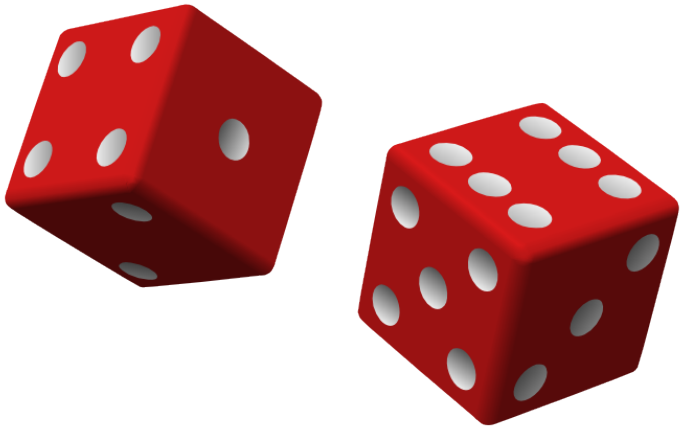
La méthode de Von Neumann

La méthode de Neumann permet de générer des nombres de manière "aléatoire". Elle consiste à partir d'un nombre (appelé *graine*), à l'élever au carré et à prendre les chiffres du milieu en sortie. Par exemple :

- graine : 1234 ;
- $1234^2 = 1522756$;
- on récupère les chiffres du milieu : 2275 et on recommence avec 2275.

Von Neumann utilisait des nombres à 10 chiffres.

- Essayez d'implémenter cette méthode. Comment faire pour récupérer des nombres entre 0 et 1 ?
- Est-ce que vous pouvez penser à une autre méthode pour générer des nombres entre 0 et 1 ?
- Peut-on vérifier graphiquement que les nombres générés sont bien répartis uniformément sur $[0, 1]$?



♡ Premiers pas en Python

Python est un langage de programmation dont la syntaxe est particulièrement simple, il est très puissant et très utilisé (le créateur de ce langage travaille chez Google et la Nasa l'utilise aussi, pour ne citer qu'eux !).

Pour vous familiariser avec le langage Python, essayez ces quelques commandes directement dans la console :

```
a = array([0,1,2,3])
a[2]
a[2] = 1
print "Le nouveau tableau est", a
b = array([2,3,3,4])
plot(a,b,"x")
show()
```

► L'instruction "if"

En Python, pour délimiter les blocs de code, on utilise *l'indentation*. Cela consiste à mettre des espaces devant le code. On pourra utiliser la touche *tabulation* pour le faire. Cette touche se trouve sur la gauche de votre clavier et ressemble à $\leftrightarrows$.

Par exemple, étant donnés deux nombres a et b, si on veut demander à l'ordinateur de stocker le minimum entre a et b, on pourra écrire :

if $a < b$:	Si $a < b$ alors
$z = a$	$z = a$;
else :	sinon,
$z = b$	$z = b$.

► La boucle for

Comme pour l'instruction **if**, le bloc d'instruction de la boucle **for** est délimité par des espaces. Voici un exemple de code :

x = **array**([ ])	x est un tableau vide
for i **in range**(10) :	i vaut 1, 2, ... , 10
x = **append**(x,i)	on ajoute i au tableau

► Pour cet atelier

Vous pouvez écrire votre code dans l'éditeur de texte (sauvez-le avec une extension `.py` pour que le logiciel colorie votre code) et copier/coller votre code dans la console. Pour l'occasion on a écrit une fonction **milieu** qui permet de récupérer les décimales au milieu d'un nombre. Vous pouvez la tester dans la console :

milieu(12345678, 4) #rend en sortie 3456.

Pour élever un nombre x au carré, on écrira : $x{*}{*}2$.

★ Pour aller plus loin

• La page wikipédia sur les générateurs pseudo-aléatoires :
`http://fr.wikipedia.org/wiki/Générateur_de_nombres_pseudo-aléatoires`

• Les librairies Python utilisées aujourd'hui :
  – `numpy` (librairie de calcul numérique) ;
  – `matplotlib` (pour faire des graphiques) ;

• Contacts :
  – `celine.caldini@gmail.com`
  – `alexis.flesch@gmail.com`