



**JN APMEP 2012 Metz**

**Lundi 29 octobre**

**Agnès BARAQUIN François PREDINAS**



# Pourquoi python ?

Parce que c'est un langage de programmation libre, gratuit et performant :

Exemple : On lance 100 000 fois de suite deux dés cubiques simultanément, et on s'intéresse au nombre de fois où la somme des deux faces vaut 7

```
1 from __future__ import division
2 from lycee import *
3 j=0
4 for i in range (100000):
5     s=randint(1,6)+randint(1,6)
6     if s==7:
7         j=j+1
8 f=j/100000.
9 print "le 7 est sorti",j,"fois donc a une fréquence égale à",f
```

Essayez donc de faire la même chose avec un tableur !

C1489    fx    =NB.SI(A1:A1508;7)

	A	B	C	D	E
481	6				
482	7				
483	7				
484	5				
485	10				
486	8				
487	6				
488	11				
489	6		268		
490	7				
491	9		fréquence de 7 :		0,17771883
492	11				

**100000 lignes !**

Ou avec une calculette



L'équipe: Vincent MAILLE, François PREDINAS,  
Julien POLLET, Agnès BARAQUIN

AmiensPython est une compilation d'outils prêts  
à l'emploi et facilement installable pour faire  
des mathématiques avec Python : avec en  
particulier la bibliothèque lycée (from lycee  
import\*)

Outre la puissance mathématique du langage,  
il permet de faire facilement des sorties  
graphiques.

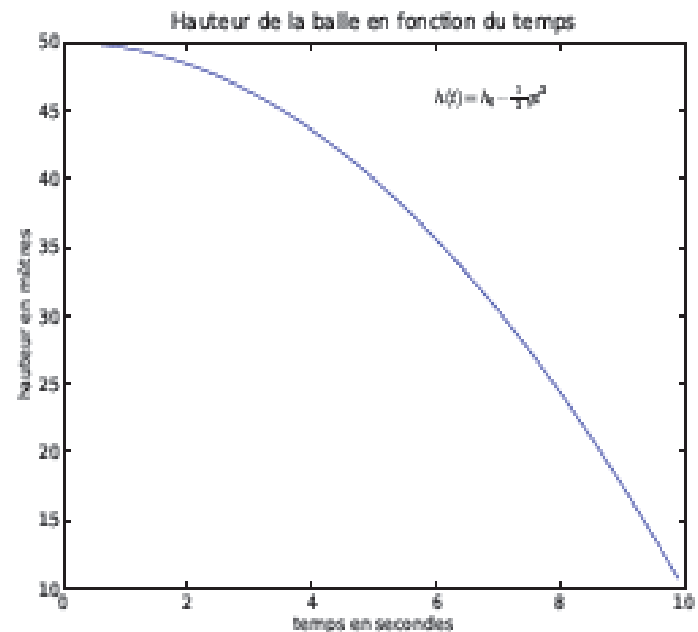
- Graphes de fonctions

Télécharger



Code: *Chute d'une balle.*

```
from lycee import *
t = np.arange(0, 10, 0.1)
repere.plot(t, 50-0.5*0.8*t**2)
repere.title('Hauteur de la balle
              en fonction du temps')
repere.ylabel(u"hauteur en mètres")
repere.xlabel(u"temps en secondes")
repere.text(6, 45, r'$h=h_0-$
              \frac{1}{2}gt^2$')
repere.show()
```



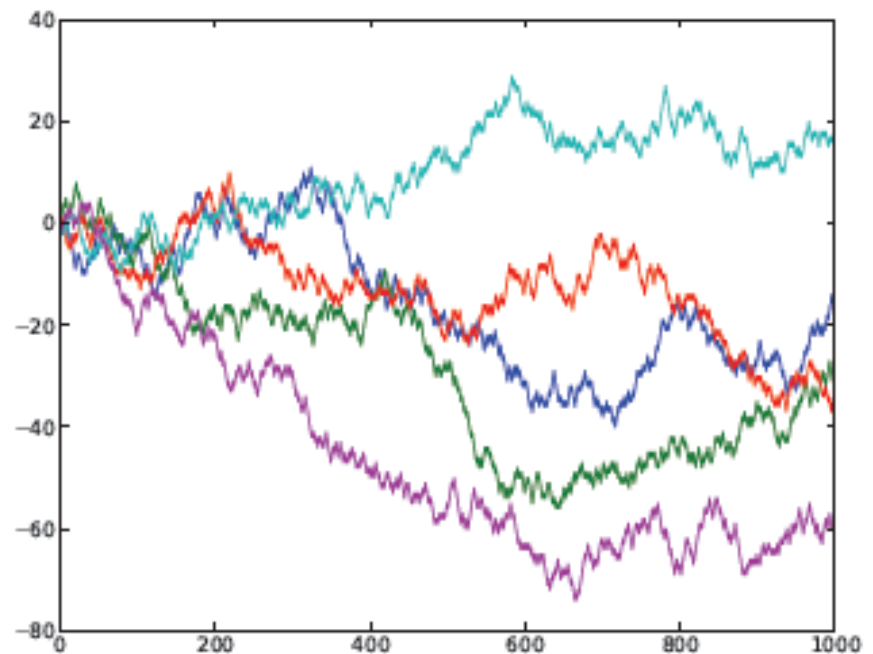
- Diagrammes ....

Télécharger




Code: *Marche aléatoire.*

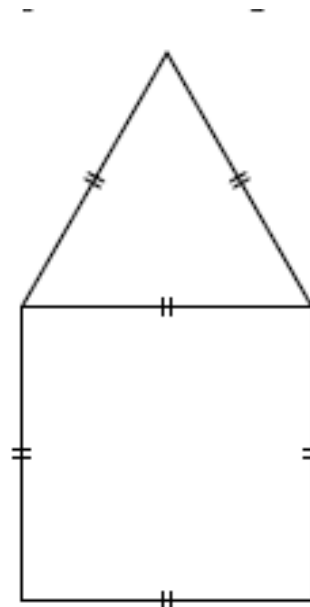
```
from lycee import *
p,x,y=0,[],[]
for i in range(1000):
    x.append(i)
    y.append(p)
    if randint(0,1)==0:
        p=p+1
    else :
        p=p-1
repere.plot(x,y)
repere.show()
```



de travailler également avec les angles à l'aide d'un module type LOGO.

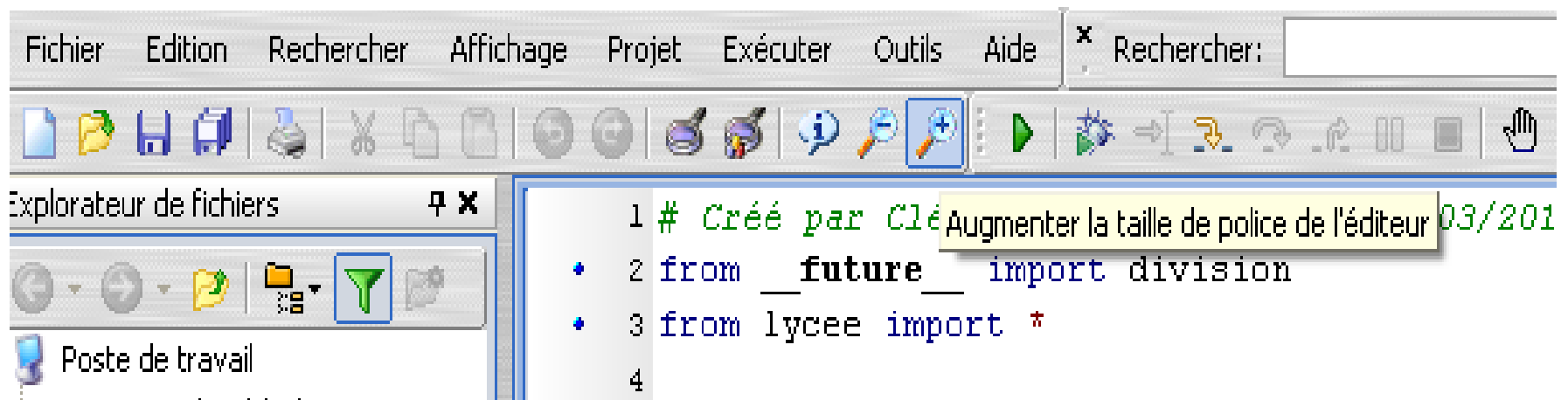
 **Code:** *Une maison.*

```
from lycee import *
l=100
tortue.forward(100)
tortue.right(90)
tortue.forward(100)
tortue.right(90)
tortue.forward(100)
tortue.right(90)
tortue.forward(100)
tortue.right(30)
tortue.forward(100)
tortue.right(120)
tortue.forward(100)
```





L'interface francisée est conviviale et permet également de faire ses premiers pas sans perdre en efficacité.



Pour illustrer la puissance de calcul: Les calculs avec les entiers se font sans arrondis en python. La place est libérée dans la mémoire au besoin.

Deux exemples qui donnent des résultats puissants assez rapidement(compter environ 15 minutes pour la conjecture d'Euler, pratiquement sans attente pour la loi de Benford).

De plus, python est préconisé pour les nouveaux programmes de CPGE.

## Loi de Benford

d'après wikipedia

La **loi de Benford**, également appelée **loi des nombres anormaux**, énonce que dans une liste de [données statistiques](#), le 1<sup>er</sup> chiffre non nul le plus fréquent est 1, pour près du tiers des observations. Puis le 2 est lui-même plus fréquent que 3... et la probabilité d'avoir un 9 comme premier chiffre significatif n'est que de 4,6 %. C'est une loi observée dans les mathématiques des [sciences humaines et sociales](#).

De façon générale, la loi donne la valeur théorique  $f$  de la fréquence d'apparition du premier chiffre d'un nombre  $d$  d'un résultat de mesure exprimé dans une [base](#)  $b$  donnée au moyen d'une unité :

$$f = \log_b \left( 1 + \frac{1}{d} \right)$$

En particulier, pour le [système décimal](#) (base 10), on aboutit au tableau de résultats suivants :

d	1	2	3	4	5	6	7	8	9
f	0,301	0,176	0,125	0,097	0,079	0,067	0,058	0,051	0,046

Dans sa conférence d'hier, Françoise VALETTE-DUCHENE a signalé que la loi de Benford était aussi vérifiée sur certaines suites mathématiques:

- Les puissances de 2
- La suite de Fibonacci

...

```

1 # Créé par AgneS, le 28/10/2012
2 from __future__ import division
3 from lycee import *
4 nombre=1000
5 freqApparition=[0 for i in range(10)]
6
7 for i in range(nombre+1):
8     valeur=2**i
9     exposant=int(ln(valeur)/ln(10))
10    premierchiffre=int(valeur/10**exposant)
11    print premierchiffre, valeur
12    freqApparition[premierchiffre]=freqApparition[premierchiffre]+1/nombre
13 for i in range(10):
14     print i, freqApparition[i]
15 baton(range(10),freqApparition)
16

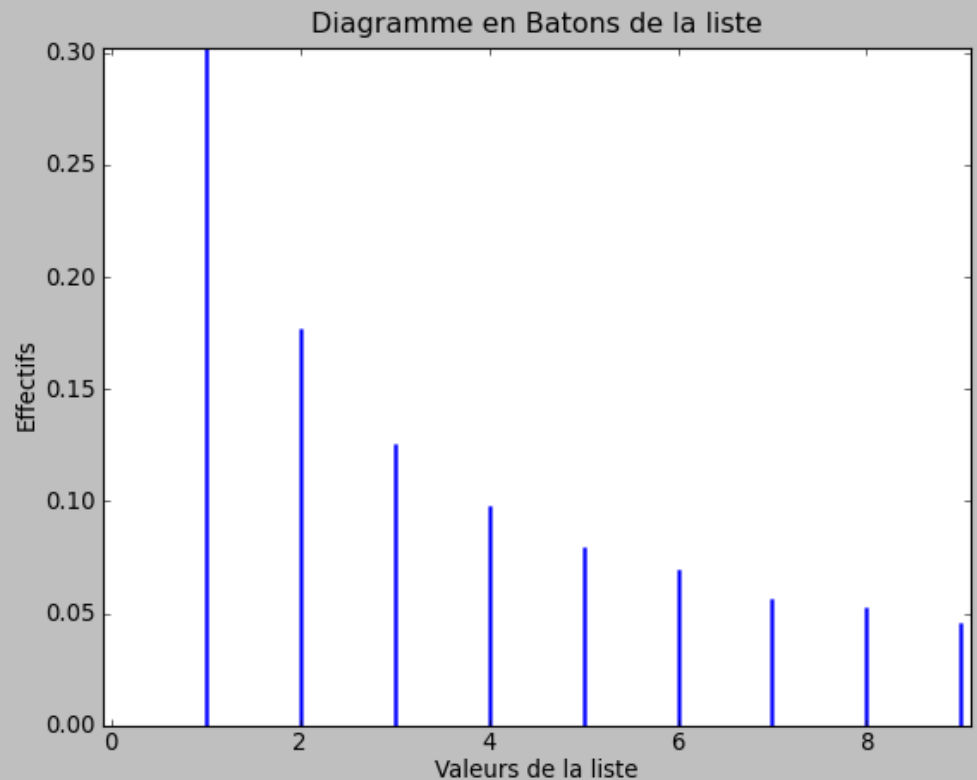
```

#### Console Python

```

0 0
1 0.302
2 0.176
3 0.125
4 0.097
5 0.079
6 0.069
7 0.056
8 0.052
9 0.045
>>>

```



La conjecture d'Euler (cas  $n = 5$ ) : Elle fut énoncée par le mathématicien suisse Leonhard Euler en 1769 et peut dans le cas  $n = 5$  s'énoncer ainsi :

***« On ne peut pas trouver 4 nombres entiers tels que la somme de chacun de ses nombres à la puissance 5 soit à son tour la puissance de 5 d'un nombre entier. »***

Cette conjecture fut infirmée par L. J. Lander et T. R. Parkin en 1966 grâce au contre-exemple suivant :

$$275^5 + 845^5 + 1105^5 + 1335^5 = 1445^5.$$

(Attention, si vous exécutez le programme c'est un peu long).

```

3 from lycee import *
4 L=[puissance(max,5) for max in range(1,1000)]
5 max=5
6 trouve=0
7 while trouve==0:
8     ind1=0;
9     while ind1<max and trouve==0:
10        ind2=ind1
11        while ind2<max and trouve==0:
12            ind3=ind2
13            while ind3<max and trouve==0 :
14                ind4=ind3
15                while ind4 <max and trouve==0:
16                    if L[ind1]+L[ind2]+L[ind3]+L[ind4]==L[max-1]:
17                        print ind1+1,"^5+",ind2+1,"^5+",ind3+1,"^5+",ind4+1,"^5=",max,"^5"
18                        print L[ind1],"+",L[ind2],"+",L[ind3],"+",L[ind4],"=",L[max-1]
19                        trouve=1
20                    ind4=ind4+1
21                ind3=ind3+1
22            ind2=ind2+1
23        ind1=ind1+1
24    max=max+1
25    print "max=",max

```

```

max= 140
max= 141
max= 142
max= 143
max= 144
27 ^5+ 84 ^5+ 110 ^5+ 133 ^5= 144 ^5
14348907 + 4182119424 + 16105100000 + 41615795893 = 61917364224
max= 145
>>>

```

<http://amienspython.tuxfamily.org>

La documentation illustrée d'exemples pour les enseignants, comprenant un index des instructions.

Des fiches pour les élèves.

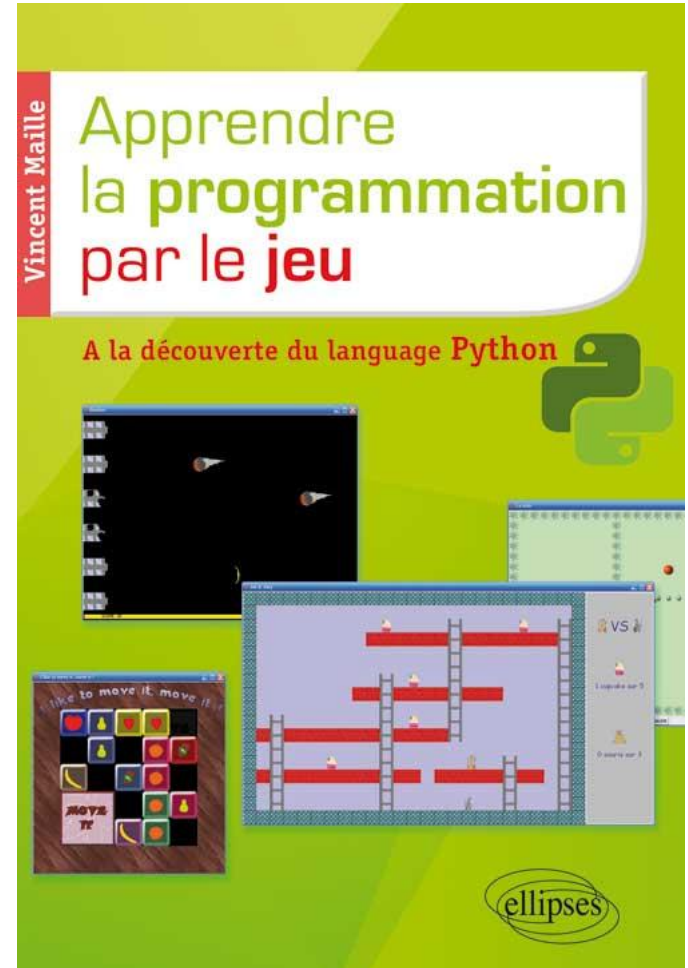
Des vidéos réalisées par l'IREM: des programmes simples réalisés pas à pas pour débuter.



# Pour aller plus loin:

- Le livre de Vincent MAILLE  
« **Apprendre la programmation  
par le jeu. A la découverte du  
langage Python** » ellipses

**Bientôt la nouvelle version  
EDUPython basée sur python 3**





# Télécharger AmiensPython

<http://amienspython.tuxfamily.org>

Attention : Vous obtiendrez une erreur si un nom de dossier ou de fichier comporte des caractères spéciaux (accent, cédille ...)

## Téléchargement

Nouvelles fonctions dans la bibliothèque *lycee*

barre : Génère et affiche un diagramme en barre.

histop : Génère et affiche un histogramme.

factorial : Calcule la factorielle d'un entier.

Ajout de nouveau module dans le pack :

Pyserial : Communiquer avec le port série

Association des fichiers .py avec l'éditeur lors d'une installation non portable

AmiensPython est distribué sous licence **CeCILL**

**Il est dérivé de PortablePython et PyScripteur**

Les textes de la documentation sont sous licence CC

