

Formation Python

Lycée Carcouët

12 juin 2017

Résumé

1 Modules utiles au lycée

- Le module math
- Nombres complexes
- Le module random
- numpy
- scipy
- matplotlib

2 Spyder

3 Jupyter

Résumé

- 1 Modules utiles au lycée
 - Le module math
 - Nombres complexes
 - Le module random
 - numpy
 - scipy
 - matplotlib
- 2 Spyder
- 3 Jupyter

Les fonctions sqrt, exp, log, log10, sin, cos, tan, asin, acos, atan, floor ... ainsi que pi, e, doivent être importées.

En ligne de commande :

```
from math import *
```

Dans un programme, il vaut mieux importer chaque fonction :

```
from math import exp , pi
```

Certains recommandent (pour une meilleure lisibilité des programmes et au cas où des fonctions de modules différents portent des noms identiques) :

```
import math  
math.e  
math.sin(math.pi/3)
```

Résumé

- 1 Modules utiles au lycée
 - Le module math
 - **Nombres complexes**
 - Le module random
 - numpy
 - scipy
 - matplotlib
- 2 Spyder
- 3 Jupyter

Pour forme polaire, argument, exponentielle complexe, importer le module cmath.

```
from cmath import polar , phase , exp
from math import degrees
z=complex(input(" Entrer un complexe a+bj\n"))
print(" module = ", abs(z))
print(" partie réelle = ", z.real)
print(" partie imaginaire = ", z.imag)
print(" conjugué = ", z.conjugate())
print(" argument : ", phase(z), " radians")
print(" argument : ", degrees(phase(z)), " degrees")
print(" forme polaire : ", polar(z))
print(" exp(z) = ", exp(z))
```

Voir le notebook NombresComplexes.ipynb
NB :

- pour voir le contenu d'un module : `dir(cmath)`
- pour obtenir de l'aide sur une méthode : `help(cmath.exp)`
- `cmath.exp` n'est pas le même que `math.exp`

Résumé

1 Modules utiles au lycée

- Le module math
- Nombres complexes
- **Le module random**
- numpy
- scipy
- matplotlib

2 Spyder

3 Jupyter

génération de nombres pseudo-aléatoires : un réel entre 0 et 1 ou un entier entre 1 et 6

```
>>> import random
>>> random.random()
0.6656199812165009
>>> random.randint(1,6)
2
```

Un programme

```
from random import randint
n=1
de=randint(1,6)
while (de != 6) :
    n=n+1
    de=randint(1,6)
print (n)
```

Une ébauche d'activité : notebook
simulationjeudepetitschevaux.ipynb

Résumé

1 Modules utiles au lycée

- Le module math
- Nombres complexes
- Le module random
- **numpy**
- scipy
- matplotlib

2 Spyder

3 Jupyter

Pour le calcul matriciel (voir le notebook matrices.ipynb)

Résumé

1 Modules utiles au lycée

- Le module math
- Nombres complexes
- Le module random
- numpy
- **scipy**
- matplotlib

2 Spyder

3 Jupyter

Exemples : loi binomiale, normale, intégration
Plein de choses intéressantes sur [scipy.org](https://www.scipy.org).

Loi binomiale

```
from scipy.stats import binom
print("loi binomiale de parametres n et p")
print("X est le nombre de succes")
n=int(input("saisir n\n"))
p=float(input("saisir p\n"))
k=int(input("saisir le nombre de succes k\n"))
print("P(X=k) = ", binom.pmf(k, n, p))
print("P(X<=k) = ", binom.cdf(k, n, p))
print("esperance=", binom.stats(n, p, moments='m'))
```


Calculs sur la loi binomiale et diagramme en bâtons dans le notebook loibinomiale.ipynb

Loi normale

```
from scipy.stats import norm
>>> norm.cdf(3,loc=0,scale=1)
0.9986501019683699
>>> norm.cdf(2,loc=0,scale=1)
0.97724986805182079
>>> norm.cdf(1,loc=0,scale=1)
0.84134474606854293
>>> norm.cdf(10,loc=10,scale=30)
0.5
>>> norm.cdf(15,loc=10,scale=30)
0.56618383261090366
```

Si X suit une loi normale, on calcule ainsi $P(X \leq a)$. Le premier argument est a . loc est l'espérance. $scale$ est l'écart-type.

Intégration

```
>>> import scipy.integrate as integrate
```

```
>>> def carre(x):  
    return x*x
```

```
>>> integrate.quad(carre, 0, 3)  
(9.0000000000000002, 9.992007221626411e-14)
```

Résumé

1 Modules utiles au lycée

- Le module math
- Nombres complexes
- Le module random
- numpy
- scipy
- **matplotlib**

2 Spyder

3 Jupyter

Tracé de représentations graphiques de fonctions ou de suites.
Voir le notebook graphiques.ipynb et simulations.ipynb
Plein d'autres choses [ici](#).

Résumé

- 1 Modules utiles au lycée
 - Le module math
 - Nombres complexes
 - Le module random
 - numpy
 - scipy
 - matplotlib
- 2 Spyder
- 3 Jupyter

Spyder est présent dans la distribution Anaconda

- un éditeur
- une console
- un explorateur de variables (menu Affichage)

Résumé

- 1 Modules utiles au lycée
 - Le module math
 - Nombres complexes
 - Le module random
 - numpy
 - scipy
 - matplotlib
- 2 Spyder
- 3 Jupyter

Jupyter est présent dans la distribution Anaconda

- Pour créer des notebooks
- Utilisable en ligne : try.jupyter.org
sur pc, téléphone, tablette : on peut y programmer en Python.