

02. Bibliothèques Python

Nous utiliserons dans ce TP le langage de programmation *Python* au moyen de l'environnement de développement *IDLE*.

Les *bibliothèques* Python regroupent des fonctions utiles que l'on peut utiliser sans avoir à les redéfinir.

1 Bibliothèque *math*

Une fois chargée, la bibliothèque *math* permet d'utiliser les principales constantes et fonctions mathématiques.

```
>>>from math import*
>>>e,pi,sqrt(2)
(2.718281828459045,3.14159265389793,1.4142135623730951)
```

Exercice 1. Calculer une valeur approchée de $e^{2\ln 5}$ et de $\cos\left(\frac{\pi}{9}\right)\cos\left(\frac{2\pi}{9}\right)\cos\left(\frac{4\pi}{9}\right)$.

2 Bibliothèque *fractions*

La bibliothèque *fractions* permet le calcul avec des fractions.

```
>>>from fractions import*
>>>Fraction(1,3)-Fraction(1,4)
Fraction(1,12)
```

Exercice 2. Calculer la valeur exacte de $\frac{1}{2 + \frac{1}{2 + \frac{1}{2 + \frac{1}{2}}}}$.

3 Bibliothèque *cmath*

La bibliothèque *cmath* permet le calcul avec des nombres complexes.

```
>>>from cmath import*
>>>z=complex(3,4)
>>>z.real,z.imag,abs(z),phase(z)
(3.0,4.0,5.0,0.9272952180016122)
```

Exercice 3. Calculer une valeur approchée de $\mathcal{I}m(e^{i\pi})$.

Certaines fonctions pouvant être définies dans plusieurs bibliothèques, il est prudent de préciser la définition que l'on souhaite utiliser.

```
>>>import math,cmath
>>>math.sqrt(4),cmath.sqrt(4)
(2.0,(2+0j))
```

Exercice 4. Comparer les résultats fournis par les bibliothèques *math* et *cmath* pour le calcul de $\cos(\pi)$.

4 Bibliothèque *random*

La bibliothèque *random* permet de générer des nombres pseudo-aléatoires.

```
>>>from random import*
>>>random()
0.30346339068560835
>>>randint(1,6)
2
```

Exercice 5. Simuler une suite de 10 lancers au jeu de Pile ou Face avec une pièce équilibrée.

5 Bibliothèque *sympy*

La bibliothèque *sympy* permet le calcul formel.

```
>>> from sympy import*
>>> x=symbols('x')
>>> y=symbols('y')
>>> expand((x+y)**3)
x**3 + 3*x**2*y + 3*x*y**2 + y**3
>>> factor(x**3-y**3)
(x - y)*(x**2 + x*y + y**2)
>>> simplify(x**2/(1+x)-1/(1+x))
x - 1
```

Exercice 6. Factoriser $(ac - bd)^2 + (ad + bc)^2$.

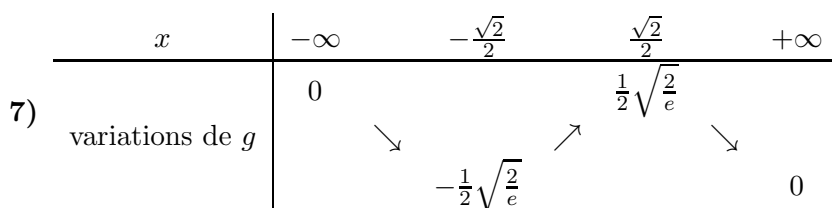
```
>>> f=exp(x)/(2*x**2+1)
>>> plot(f, (x,-5,5))
<sympy.plotting.plot.Plot object at 0x000000009D20588>
>>> limit(f,x,-oo)
0
>>> limit(f,x,+oo)
oo
>>> df=diff(f,x)
>>> factor(df)
(2*x**2 - 4*x + 1)*exp(x)/(2*x**2 + 1)**2
>>> solve(2*x**2-4*x+1>0,x)
And(Or(re(x) < -sqrt(2)/2 + 1, re(x) > sqrt(2)/2 + 1), im(x) == 0)
```

Exercice 7. Étudier les variations de la fonction $g : x \mapsto xe^{-x^2}$.

Réponses

```

1) >>> exp(2*log(5))
24.999999999999996
>>> cos(pi/9)*cos(2*pi/9)*cos(4*pi/9)
0.12500000000000006
2) >>> Fraction(1,2+Fraction(1,2+Fraction(1,2+Fraction(1,2))))
Fraction(12, 29)
3) >>> exp(complex(0,1)*pi).imag
1.2246467991473532e-16
4) >>> math.cos(math.pi), cmath.cos(cmath.pi)
(-1.0, (-1-0j))
5) >>> randint(0,1),randint(0,1),randint(0,1),randint(0,1),randint(0,1),
randint(0,1),randint(0,1),randint(0,1),randint(0,1),randint(0,1)
(0, 1, 0, 0, 1, 1, 1, 1, 1)
6)  $(ac - bd)^2 + (ad + bc)^2 = (a^2 + b^2)(c^2 + d^2)$ 
>>> a,b,c,d=symbols('a'),symbols('b'),symbols('c'),symbols('d')
>>> factor((a*c-b*d)**2+(a*d+b*c)**2)
(a**2 + b**2)*(c**2 + d**2)
    
```



```

>>> g=x*exp(-x**2)
>>> limit(g,x,-oo)
0
>>> limit(g,x,+oo)
0
>>> dg=diff(g,x)
>>> factor(dg)
-(2*x**2 - 1)*exp(-x**2)
>>> solve(-(2*x**2 - 1)>0,x)
And(-sqrt(2)/2 < re(x), im(x) == 0, re(x) < sqrt(2)/2)
>>> g.subs(x,-sqrt(2)/2)
-sqrt(2)*exp(-1/2)/2
>>> g.subs(x,sqrt(2)/2)
sqrt(2)*exp(-1/2)/2
    
```