

1 Calcul de π (1)

```
1 # -*- coding: utf-8 -*-
2
3 from math import *
4
5 #-----
6 def calculPi(n):
7     """
8     y = calculPi(n)
9     calcul de pi à l'ordre n
10
11     >>> from math import fabs, pi
12     >>> fabs(pi - calculPi(1)) < 1.
13     True
14     >>> fabs(pi - calculPi(1000000)) < 1.e-6
15     True
16     """
17     assert type(n) is int and n >= 0
18
19     y = 0
20     for k in range(1,n+1):
21         u = 1./(k*k)
22         y = y + u
23     return sqrt(6*y)
24
25 #-----
26 if __name__ == "__main__":
27     import doctest
28     doctest.testmod()
```

2 Conversion base $b \rightarrow$ décimal

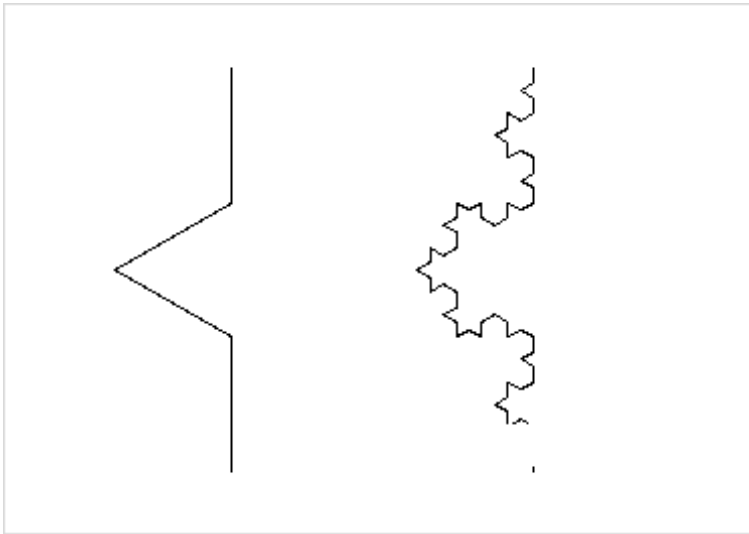
```
1 # -*- coding: utf-8 -*-
2
3 def conversion(code,b=2):
4     """
5     n = conversion(code,b)
6     entier décimal qui représente le code en base b
7
8     >>> conversion([0,0,1,0,1,1,1],2)
9     23
10    >>> conversion([0, 0, 0, 4, 3],5)
11    23
12    >>> conversion([1,2],21)
13    23
14    >>> conversion([0,0,0,0,0,23],25)
15    23
16    """
17    assert type(b) is int and b > 1
18    assert type(code) is list
19
20    n = 0
21    for i in range(len(code)):
22        n = n + (b**i)*code[len(code)-1-i]
23
```

```

24     return n
25
26 #-----
27 if __name__ == "__main__":
28     import doctest
29     doctest.testmod()

```

3 Courbes fractales



4 Portée des variables

```

>>> x = 2
>>> print(x)
2

```

```

>>> y = f(x)
>>> print(x)
f 6
2

```

```

>>> z = g(x)
>>> print(x)
f 6
g 18
2

```

```

>>> t = h(x)
>>> print(x)
f 6
f 18
g 54
h 162
2

```

```

>>> x = 2
>>> print(x)
2

```

```

>>> x = f(x)
>>> print(x)
f 6
6

```

```

>>> x = g(x)
>>> print(x)
f 18
g 54
54

```

```

>>> x = h(x)
>>> print(x)
f 162
f 486
g 1458
h 4374
4374

```

5 Calcul de π (2)

```
1. >>> for n in range(7):  
        print(n,':',end=' ')  
        for m in range(n+1):  
            print(g(n,m),end=' ')  
        print()
```

```
0 : 1  
1 : 0 1  
2 : 0 1 1  
3 : 0 1 2 2  
4 : 0 2 4 5 5  
5 : 0 5 10 14 16 16  
6 : 0 16 32 46 56 61 61
```

```
2. >>> 12*g(5,5)/g(6,6)  
3.1475409836065573
```