

```

# -*- coding: utf-8 -*-
"""
Created on Thu Feb 13 01:38:28 2014

@author: dconduche
"""

"""
Plusieurs façon de coder la formule, au choix :
* une boucle for
* un sum()
* un tableau numpy (plus de place mémoire)
L'essentiel, c'est la formule (et une syntaxe correcte évidemment).

Rmq : dans la fonction numpy,[:-1] permet d'enlever le dernier élément du
tableau.
"""

import numpy as np

def rectangle(f, a, b, n):
    """Calcule l'intégrale de f sur [a,b] par la méthode des rectangles.

    Keyword arguments:
    f -- fonction continue par morceaux définie sur [a,b]
    a -- float représentant le début de l'intervalle
    b -- float représentant la fin de l'intervalle (a<b et f(a)f(b)<0)
    n -- int représentant le nombre de rectangles utilisés
    retourne un float, l'intégrale de f sur [a,b] avec n pas.
    """
    I = 0
    pas = (b-a) / n
    for k in range(n):
        I = I + f(a + k*pas)
    return pas*I

def rectangleBis(f, a, b, n):
    pas = (b-a) / n
    return pas * sum(f(a + k*pas) for k in range(n))

def rectangleNP(f, a, b, n):
    x = np.linspace(a, b, n+1)[: -1]
    pas = (b-a) / n
    return pas * sum(f(x))

f = lambda x: x**2
I = rectangleBis(f, 0, 1, 100000)
Inp = rectangleNP(f, 0, 1, 100000)
print(I, Inp) # Calcul exact : 1/3

```