

```

# ===== #
# TD CALCUL NUMERIQUE #
# INTEGRATION PAR LA METHODE DE SIMPSON. #
# Version de janvier 2018 #
# ===== #

## Importations
from math import log
from numpy import linspace

## Fonction de calcul d'une valeur approchée d'une intégrale définie par la
méthode de Simpson
def IntSimp(f,a,b,n):
    if n%2 != 0:
        raise ValueError("Le nombre d'intervalles de la subdivision doit
être pair !")
    else:
        X = linspace(a,b,n+1)
        h = (b - a) / n
        S = f(X[0]) + f(X[n])
        # Somme des images des abscisses d'indices pairs
        SP = 0
        for i in range(1,n//2):
            SP += f(X[2*i])
        S += 2.*SP
        # Somme des images des abscisses d'indices impairs
        SI = 0
        for i in range(1,n//2+1):
            SI += f(X[2*i - 1])
        S += 4.*SI
        # Renvoi de la valeur approchée de l'intégrale
        return(S * h / 3)

## Pour tester
def f1(x):
    return(x**2)

def f2(x):
    return(log(x))

n = 100000
a,b = 0,2

I = IntSimp(f1,a,b,2*n)

print("Une valeur approchée de l'intégrale de la fonction carrée entre
:",a,"et",b,"vaut",I)
print("La valeur exacte vaut :",(b**3 - a**3)/3)

print("\n")

n = 10000
a,b = 1, 4

I = IntSimp(f2,a,b,2*n)

print("Une valeur approchée de l'intégrale de la fonction logarithme entre
:",a,"et",b,"vaut",I)
print("La valeur exacte vaut",b*(log(b)-1) - a*(log(a)-1))

```