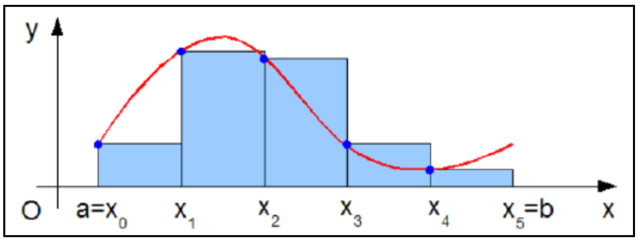
1) Ecrire une fonction **moyenne()** qui renvoie la moyenne des éléments d'une liste **L**.

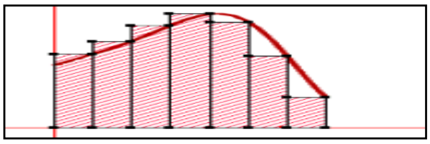



```
def moyenne(L):
    n = len(L)
    som=0
    for i in range(n):
        som=som+L[i]
    return som/n
```

Méthode des rectangles

```
def rectangleg(f,a,b,N):
    """
    paramètres d'entrées :
        *f* : fonction à intégrer (cette fonction renvoie un nombre)
        *a* et *b* : début et fin de l'intervalle d'intégration (flottants ou entiers)
        *N* : nombre de partitions de l'intervalle d'intégration (entier)
    sortie : intégrale numérique par la méthode des rectangles
    """
    h=(b-a)/N               # pas des partitions
    s=0                     # initialisation de la somme
    for k in range(N):      # somme de l'aire des rectangles de toutes les partitions
        s=s+h*f(a+k*h)
    return s
```

2) Indiquer les points communs entre les fonctions **moyenne()** et **rectangleg()**.



points communs : initialisation de s + modification N fois

points différents : l'équation de la modification de s

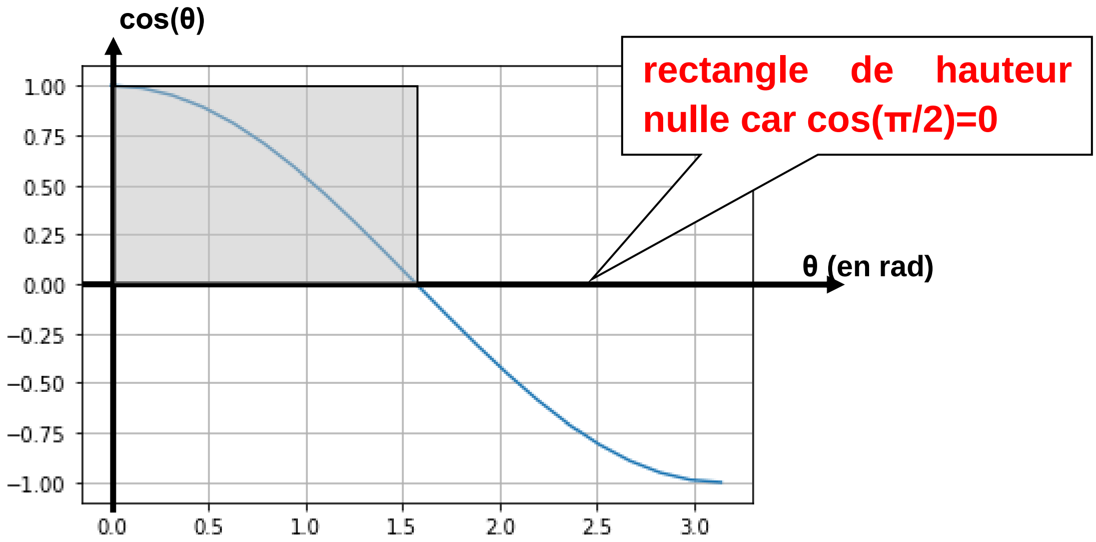
3) Ecrire une fonction **rectanglem()** qui effectue un calcul similaire à la fonction **rectangleg()** mais en prenant la valeur de la fonction au milieu de chaque partition.

```
def rectanglem(f,a,b,N):
    h=(b-a)/N
    S=0
    for k in range(N):
        s=s+h*(a+h/2+k*h)
    return s
```

4) Compléter le tableau d'évolution des variables **k**, **f(a+k*h)** et **s** si on appelle la fonction **rectangleg()** avec la fonction **cos** pour l'intervalle [0 ; π] avec 2 partitions de l'intervalle (remarque : analytiquement on obtient 0).

h= pi/2		
k	0	1
f(a+k*h)	cos(0) = 1	cos(0+pi/2) = 0
s	s = 0+h*1 = pi/2	s = pi/2+h*0 = pi/2

5) Sur la figure suivante, tracer les rectangles relatifs à l'intégration de la question précédente et justifier par une interprétation graphique que la valeur analytique de cette intégrale soit nulle



L'aire sous la courbe réelle positive de 0 à π/2 est égale à l'aire sous la courbe négative de π/2 à π donc l'intégrale est nulle.

6) Calculer $E = \max_{a\leq x\leq b}\left(\left|f^{'}(x)\right|\right) * \frac{(b-a)^2}{2N}$ et conclure quant à la majoration de l'erreur faite à la question précédente par cette expression.

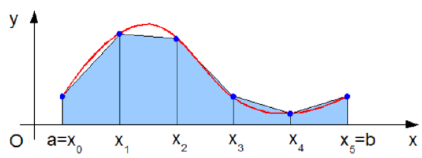
erreur maximum : E = max(|sin(x)|)*(pi)^2/(2*2)=pi^2/4 =2.46

erreur faite sur l'intégrale (sensée être nulle) : pi/2=1.57

(l'erreur effective est correctement majorée)

7) Ecrire une fonction **trapeze()** qui permet d'obtenir l'intégrale de **f** entre **a** et **b** pour **N** partitions.

```
def trapeze(f,a,b,N):
    s = 0
    h = (b-a)/N
    for k in range(0,N):
        s = s + h * (f(a+k*h)+f(a+(k+1)*h))/2
    return s
```



8) Quel est l'intérêt et l'inconvénient de la fonction **trapeze()** par rapport à la fonction **rectangleg()** ?

 # Intérêt de trapèze : plus précis pour même N

 # Inconvénient de trapèze : nécessite plus d'évaluation de f.

9) Adapter la fonction **trapeze()**, en définissant une fonction **trapeze2()** dans laquelle le nombre d'appels à la fonction **f** sera égale à **N** en utilisant la même méthode d'intégration.

```
def trapeze2(f,a,b,N):
    s = 0
    h = (b-a)/N
    yd = f(a)
    for k in range(1,N):
        yg = yd
        yd = f(a+k*h)
        s = s + h * (yg+yd)/2
    return s
```