

Automatismes en Python, semaine du 13 mai

21 avril 2024

Ecrire une fonction qui, étant donné un entier n , renvoie la n -ième somme partielle de la série harmonique alternée.

Ecrire une fonction qui, étant donné un entier n , renvoie la n -ième somme partielle de la série harmonique alternée.

Pour commencer, il faut bien comprendre ce qu'on doit calculer.

Ecrire une fonction qui, étant donné un entier n , renvoie la n -ième somme partielle de la série harmonique alternée.

$$\text{On veut } \text{somme}(n) = \sum_{k=1}^n \frac{(-1)^k}{k}.$$

```
1 from math import *
2
3 def somme(n):
4     # n est un int
5     # renvoie la n-è somme partielle de la série
6     # harmonique alternée
7
8     resultat=
9
10    for k in      :
11        resultat=
12
13    return(resultat)
```

Ecrire une fonction qui, étant donné un entier n , renvoie la n -ième somme partielle de la série harmonique alternée.

$$\text{On veut } \text{somme}(n) = \sum_{k=1}^n \frac{(-1)^k}{k}.$$

```
1 from math import *
2
3 def somme(n):
4     # n est un int
5     # renvoie la n-è somme partielle de la série
6     # harmonique alternée
7
8     resultat=0
9
10    for k in range(1,n+1):
11        resultat=resultat+(-1)**k/k
12
13    return(resultat)
```

Ecrire une fonction qui, étant donné un entier n , renvoie la n -ième somme partielle de la série harmonique alternée.

$$\text{On veut } \text{somme}(n) = \sum_{k=1}^n \frac{(-1)^k}{k}.$$

```
1 from math import *
2
3 def somme(n):
4     # n est un int
5     # renvoie la n-è somme partielle de la série
6     # harmonique alternée
7
8     resultat=0
9
10    for k in range(1,n+1):
11        resultat=resultat+(-1)**k/k
12
13    return(resultat)
```

Une critique à formuler ?

Ecrire une fonction qui, étant donné un réel $a > 0$, renvoie le rang n à partir duquel les sommes partielles de la série harmonique sont supérieurs à a .

Ecrire une fonction qui, étant donné un réel $a > 0$, renvoie le rang n à partir duquel les sommes partielles de la série harmonique sont supérieures à a .

Pour commencer, il faut bien comprendre ce qu'on doit calculer.

Ecrire une fonction qui, étant donné un réel $a > 0$, renvoie le rang n à partir duquel les sommes partielles de la série harmonique sont supérieurs à a .

On veut le premier n tel que $\sum_{k=1}^n \frac{1}{k} > a$.

```
1 def rang(a):
2     # a est un réel >0
3     # renvoie le rang n à partir duquel les sommes
4     # partielles de la série harmonique sont >a
5
6     rang=
7
8
9     while      :
10
11
12
13
14     return(rang)
```

Ecrire une fonction qui, étant donné un réel $a > 0$, renvoie le rang n à partir duquel les sommes partielles de la série harmonique sont supérieures à a .

On veut le premier n tel que $\sum_{k=1}^n \frac{1}{k} > a$.

```
1 def rang(a):
2     # a est un réel >0
3     # renvoie le rang n à partir duquel les sommes
4     # partielles de la série harmonique sont >a
5
6     rang=0
7     somme=0
8
9     while somme<a:
10
11         rang=rang+1
12         somme=somme+1/rang
13
14     return(rang)
```

```
1 from random import randint
2
3 def fonction(x):
4     l=[]
5     for i in range(1,x+1):
6         k=1
7         while randint(1,6)!=6:
8             k+=1
9         l.append(k)
10    return(l,sum(l)/x)
```

Remodeler le code pour le rendre plus clair.

```
1 from random import randint
2
3 def premierSix(n):
4     # n est un entier non nul. On simule des lancers
5     # de dé équilibré jusqu'à l'apparition d'un 6.
6     # On fait n fois la simulation. La fonction renvoie
7     # la liste des rangs d'apparition du premier 6 ainsi
8     # que la moyenne de ces rangs d'apparition.
9
10    liste_tirages=[]
11
12    for i in range(1,n+1): # on fait n fois l'épreuve
13
14        rang=1 # rang d'apparition du premier 6
15        while randint(1,6)!=6: # on lance jusqu'à avoir 6
16            rang+=1
17        liste_tirages.append(rang)
18
19    return(liste_tirages,sum(liste_tirages)/n)
```

```
1 def majeur():  
2     # on demande son âge à l'utilisateur  
3     # renvoie true si la personne est majeure,  
4     # false sinon  
5     age=input('quel est votre âge?\n')  
6     test=true  
7     if age<18  
8         test=false  
9     return(test)
```

```
1 def majeur():  
2     # on demande son âge à l'utilisateur  
3     # renvoie true si la personne est majeure,  
4     # false sinon  
5     age=input('quel est votre âge?\n')  
6     test=true  
7     if age<18  
8         test=false  
9     return(test)
```

Vous avez les 4 erreurs (dont une double) ?

```
1 def majeur():  
2     # on demande son âge à l'utilisateur  
3     # renvoie true si la personne est majeure,  
4     # false sinon  
5     age=input('quel est votre âge?\n')  
6     test=true  
7     if age<18  
8         test=false  
9     return(test)
```