

## Exercices d'informatique - Corrigé de la série n° 2

Cours 4INOC01

### 1. La fonction `sleep()`.

La fonction `sleep()` du module `time` permet de suspendre l'exécution du programme pendant un certain intervalle de temps exprimé en secondes, sous la forme d'un réel (`float`). On peut utiliser des très petites valeurs (par exemple 0.0003).

### 2. Animation.

On peut, par exemple, utiliser le script suivant:

---

```
1 import time
2 x="Programmer avec Python"
3 for l in x:
4     time.sleep(0.5)
5     print l
```

---

### 3. Processus légers ou *threads*.

Les classes `job1` et `job2` dérivées de la classe `Thread()` du module `threading` contiennent une méthode `run()`. Le code contenu dans ces méthodes s'exécute au moment de l'appel de la méthode `start()`. Les méthodes sont exécutées en parallèle.

La méthode `getName()` renvoie le nom attribué au *thread* par Python. La méthode `isAlive()` renvoie `True` si le *thread* est actif et `False` sinon. Finalement, la méthode `Thread__stop()` permet de fermer un *thread*.

### 4. Chronomètre.

On peut, par exemple, utiliser le script suivant:

---

```
1 import threading, time
2 class job1(threading.Thread):
3     def run(self):
4         raw_input("Stop: ")
5         j2._Thread__stop()
6         m=n/600
7         s=(n%600)/10
8         d=n%10
9         print m," ",s," ",d
10 class job2(threading.Thread):
11     def run(self):
12         global n
13         while True:
14             time.sleep(0.1)
15             n=n+1
16 n=0
17 raw_input("Start: ")
18 j1=job1()
19 j2=job2()
20 j1.start()
21 j2.start()
```

---

Notons qu'il est fortement recommandé de ne pas inclure le nom d'un objet dans une classe. C'est pourquoi le script suivant est beaucoup plus élégant:

---

```

1 import threading , time
2 class job(threading.Thread):
3     def run(self):
4         self.n=0
5         while True:
6             time.sleep(0.1)
7             self.n=self.n+1
8 raw_input("Start: ")
9 j=job()
10 j.start()
11 raw_input("Stop: ")
12 j._Thread__stop()
13 m=j.n/600
14 s=(j.n%600)/10
15 d=j.n%10
16 print m,"'",s,"'",d

```

---

## 5. La valse des monstres.

On peut, par exemple, utiliser le script suivant:

---

```

1 import threading ,time
2 from math import *
3 class Monstre:
4     def __init__(self ,canevas ,couleur='red' ,coordonnees =[10,10,100,10,100,100,10,100,10,10]):
5         self.coordonnees=coordonnees
6         self.couleur=couleur
7         self.canevas=canevas
8         self.objet=canevas.create_line(self.coordonnees ,width=2,fill=self.couleur ,smooth="true")
9     def deplacement(self ,x=10,y=10):
10        d=[x-self.coordonnees[0] ,y-self.coordonnees[1]]
11        for n in range(0, len(self.coordonnees) ,1):
12            self.coordonnees[n]=self.coordonnees[n]+d[n%2]
13        self.canevas.coords(self.objet ,tuple(self.coordonnees))
14 class Objet(threading.Thread):
15     def run(self):
16        bernard=Monstre(self.cadre ,self.couleur ,self.forme)
17        t=self.init
18        while True:
19            x=200+100*cos(t)
20            y=200+100*sin(t)
21            time.sleep(.1)
22            bernard.deplacement(x,y)
23            t=t+0.1
24 import Tkinter
25 cadre=Tkinter.Tk()
26 cadre.geometry("600x400")
27 canevas=Tkinter.Canvas(width=600,height=400)
28 canevas.place(x=0,y=0)
29 bernard=Objet()
30 bernard.cadre=canevas
31 bernard.couleur='red'
32 bernard.forme=[50,100,10,10,50,50,100,10,50,100]
33 bernard.init=0
34 bernard.start()
35 gustave=Objet()
36 gustave.cadre=canevas
37 gustave.couleur='blue'
38 gustave.forme=[50,10,10,100,50,50,100,100,50,10]
39 gustave.init=pi/4
40 gustave.start()
41 cadre.mainloop()

```

---

ou le script suivant:

---

```

1 import threading , time , Tkinter , random
2 from math import *
3 cadre=Tkinter.Tk()
4 cadre.geometry("800x600")
5 a=Tkinter.Canvas(width=800,height=600)
6 a.place(x=0,y=0)
7 class Monstre(threading.Thread):
8     def run(self):
9         forme=[10,10,100,100,200,50,60,30,10,10]
10        couleur=random.choice(['red','blue','green'])
11        centrex=random.randrange(100,700,1)
12        centrey=random.randrange(100,500,1)
13        rayon=random.randrange(50,200,1)
14        vitesse=random.randrange(0,5,1)
15        posx=centrex+rayon
16        posy=centrey
17        monstre=a.create_line(forme,fill=couleur,smooth='true')
18        n=0
19        while True:
20            t=n/100.
21            posx=centrex+rayon*cos(t)
22            posy=centrey+rayon*sin(t)
23            liste=[10+posx,10+posy,100+posx,100+posy,200+posx,50+posy,60+posx,30+posy,10+posx,10+
24                    posy]
25            a.coords(monstre,tuple(liste))
26            time.sleep(0.01)
27            n=n+1
28        liste=[]
29        for n in range(0,4,1):
30            liste.append(Monstre())
31            liste[n].start()
32        cadre.mainloop()

```

---

## 6. Flocons.

On peut, par exemple, utiliser le script suivant:

---

```

1 import threading , time
2 import Tkinter
3 import random
4 cadre=Tkinter.Tk()
5 cadre.geometry("600x400")
6 a=Tkinter.Canvas(height=400,width=600)
7 a.place(x=0,y=0)
8 class flocon(threading.Thread):
9     def run(self):
10        self.flocon=a.create_oval(self.x,0,self.x+5,5,width=1,fill=self.couleur)
11        for n in range(0,401,1):
12            a.coords(self.flocon,self.x,n,self.x+5,n+5)
13            time.sleep(self.v)
14 class mere(threading.Thread):
15     def run(self):
16        couleurs=['blue','red','green']
17        vitesses=[0.05,0.01,0.005]
18        for n in range(0,100,1):
19            floc=flocon()
20            floc.x=random.choice(range(0,601,1))
21            floc.v=random.choice(vitesses)
22            floc.couleur=random.choice(couleurs)
23            floc.start()
24            time.sleep(0.1)
25 creation=mere()
26 creation.start()
27 cadre.mainloop()

```

---

ou plus simplement

---

```

1 import threading , time
2 import Tkinter
3 import random
4 cadre=Tkinter.Tk()
5 cadre.geometry("600x400")
6 a=Tkinter.Canvas(height=400,width=600)
7 a.place(x=0,y=0)
8 class flocon(threading.Thread):
9     def run(self):
10         couleurs=['blue','red','green']
11         vitesses=[0.05,0.01,0.005]
12         x=random.choice(range(0,601,1))
13         v=random.choice(vitesses)
14         couleur=random.choice(couleurs)
15         albert=a.create_oval(x,0,x+5,5,width=1,fill=couleur)
16         for n in range(0,401,1):
17             a.coords(albert,x,n,x+5,n+5)
18             time.sleep(v)
19 class mere(threading.Thread):
20     def run(self):
21         for n in range(0,100,1):
22             floc=flocon()
23             floc.start()
24             time.sleep(0.1)
25 creation=mere()
26 creation.start()
27 cadre.mainloop()

```

---

## 7. Une particule.

On peut, par exemple, utiliser le script suivant:

---

```

1 import threading , time
2 import Tkinter
3 import random
4 cadre=Tkinter.Tk()
5 cadre.geometry("500x500")
6 a=Tkinter.Canvas(height=400,width=400)
7 a.place(x=50,y=50)
8 class particule(threading.Thread):
9     def run(self):
10         self.x=random.randrange(5,390,5)
11         self.y=random.randrange(5,390,5)
12         dx=random.choice([-5,5])
13         dy=random.choice([-5,5])
14         couleurs=['blue','red','green','yellow','magenta']
15         couleur=random.choice(couleurs)
16         vitesses=[0.01,0.02,0.03]
17         v=random.choice(vitesses)
18         albert=a.create_oval(self.x,self.y,self.x+5,self.y+5,width=1,fill=couleur)
19         while True:
20             if self.stop==0:
21                 #if stop==0:
22                 if self.x==0 or self.x==395:
23                     dx=-dx
24                 if self.y==0 or self.y==395:
25                     dy=-dy
26                 self.x=self.x+dx
27                 self.y=self.y+dy
28                 a.coords(albert,self.x,self.y,self.x+5,self.y+5)
29                 time.sleep(v)
30 #stop=0
31 #particule().start()

```

```

32 part=particule()
33 part.stop=0
34 part.start()
35 def action(evt):
36     #global stop
37     if evt.char=='s':
38         #stop=1
39         part.stop=1
40     else:
41         #stop=0
42         part.stop=0
43 cadre.bind('<Key>', action)
44 cadre.mainloop()

```

---

## 8. Particules.

On peut, par exemple, utiliser le script suivant

---

```

1 import threading , time
2 import Tkinter
3 import random
4 cadre=Tkinter.Tk()
5 cadre.geometry("600x500")
6 a=Tkinter.Canvas(height=400,width=400,bg='white', bd =2, relief='solid')
7 a.place(x=100,y=50)
8 stop=0
9 class particule(threading.Thread):
10     def run(self):
11         self.x=random.randrange(5,390,5)
12         self.y=random.randrange(5,390,5)
13         dx=random.choice([-5,5])
14         dy=random.choice([-5,5])
15         couleurs=['blue','red','green','yellow','magenta']
16         couleur=random.choice(couleurs)
17         vitesses=[0.03,0.01,0.02]
18         v=random.choice(vitesses)
19         albert=a.create_oval(self.x,self.y,self.x+5,self.y+5,width=1,fill=couleur)
20         vie=1
21         while vie==1:
22             if stop==0:
23                 if self.x==0 or self.x==395:
24                     dx=-dx
25                 if self.y==0 or self.y==395:
26                     dy=-dy
27                 self.x=self.x+dx
28                 self.y=self.y+dy
29                 a.coords(albert,self.x,self.y,self.x+5,self.y+5)
30                 time.sleep(v)
31 class creation(threading.Thread):
32     def run(self):
33         j=0
34         self.liste=[]
35         while j<100:
36             if stop==0:
37                 j=j+1
38                 np=particule()
39                 np.creation=self
40                 np.start()
41                 self.liste.append(np)
42                 time.sleep(0.1)
43 creation().start()
44 def action(evt):
45     global stop
46     if evt.char=='s':
47         stop=1

```

```

48     else :
49         stop=0
50     cadre.bind('<Key>', action)
51     cadre.mainloop()

```

---

## 9. Particules.

On peut, par exemple, utiliser le script suivant

---

```

1  import threading , time
2  import Tkinter
3  import random
4  cadre=Tkinter.Tk()
5  cadre.geometry("600x500")
6  a=Tkinter.Canvas(height=400,width=400,bg='white', bd =2, relief='solid')
7  a.place(x=100,y=50)
8  stop=0
9  class particule(threading.Thread):
10     def run(self):
11         self.x=random.randrange(5,390,5)
12         self.y=random.randrange(5,390,5)
13         dx=random.choice([-5,5])
14         dy=random.choice([-5,5])
15         couleurs=['blue','red','green','yellow','magenta']
16         couleur=random.choice(couleurs)
17         vitesses=[0.03,0.01,0.02]
18         v=random.choice(vitesses)
19         albert=a.create_oval(self.x,self.y,self.x+5,self.y+5,width=1,fill=couleur)
20         vie=1
21         while vie==1:
22             for part in self.creation.liste:
23                 if part!=self and abs(part.x-self.x)<10 and abs(part.y-self.y)<10:
24                     vie=0
25                     break
26             else:
27                 if stop==0:
28                     if self.x==0 or self.x==395:
29                         dx=-dx
30                     if self.y==0 or self.y==395:
31                         dy=-dy
32                     self.x=self.x+dx
33                     self.y=self.y+dy
34                     a.coords(albert,self.x,self.y,self.x+5,self.y+5)
35                     time.sleep(v)
36 class creation(threading.Thread):
37     def run(self):
38         j=0
39         self.liste=[]
40         while j<100:
41             if stop==0:
42                 j=j+1
43                 np=particule()
44                 np.creation=self
45                 np.start()
46                 self.liste.append(np)
47                 time.sleep(0.1)
48 creation().start()
49 def action(evt):
50     global stop
51     if evt.char=='s':
52         stop=1
53     else:
54         stop=0
55 cadre.bind('<Key>', action)
56 cadre.mainloop()

```

---

