

Horloge Tkinter

par [Guillaume Durlaud](#)

Date de publication : 28/01/2007

Dernière mise à jour : 30/01/2007

Exemple de la création d'une horloge en Tkinter

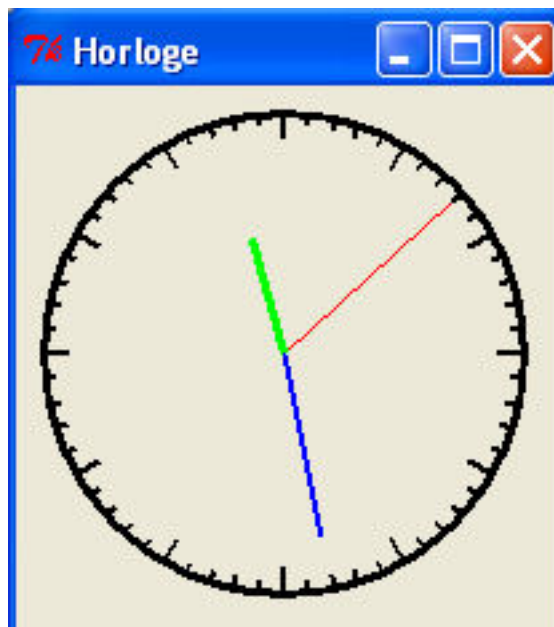
- I - Prérequis
- II - Introduction
- III - Explication du code
- IV - Source
- V - Téléchargement

I - Prérequis

Langage: Python

II - Introduction

Ce petit programme a simplement pour but la création d'un composant Tkinter représentant une horloge très simple comprenant les 3 aiguilles des secondes, minutes, heures.



III - Explication du code

Le programme est composé de 2 classe: **MyTimer** et **Horloge**. La classe **MyTimer** permet de gérer un traitement périodique (ici toutes les secondes). L'horloge est constituée d'un simple **Tk.Canvas** dans lequel on trace un cercle (**create_oval**) pour la pendule et des lignes pour les aiguilles (**create_line**).

Toutes les secondes, la fonction **turn** est appelée pour faire tourner les aiguilles; un petit calcul sur le cercle trigonométrique permet de positionner correctement les aiguilles sachant que l'aiguille des heures est positionnée à partir de l'heure en cours mais aussi des minutes et la position de l'aiguille des minutes dépend des minutes en cours mais aussi des secondes.

IV - Source

TkHorloge.pyw

```

# -*- coding: cp1252 -*-
import threading
import time
import Tkinter as Tk
import math

__version__ = (1, 0, 1)
__build__ = (0, 0)
__date__ = (2007, 1, 30)
__author__ = ('Guillaume', 'Duriaud')

## Classe permettant de gérer un Timer
class MyTimer:
    def __init__(self, tempo, target, args= [], kwargs={}):
        self._target = target
        self._args = args
        self._kwargs = kwargs
        self._tempo = tempo

    def _run(self):
        self._timer = threading.Timer(self._tempo, self._run)
        self._timer.start()
        self._target(*self._args, **self._kwargs)

    def start(self):
        self._timer = threading.Timer(self._tempo, self._run)
        self._timer.start()

    def stop(self):
        self._timer.cancel()

class Horloge(MyTimer, Tk.Canvas):
    def __init__(self, parent):
        MyTimer.__init__(self, 1.0, self.turn)
        Tk.Canvas.__init__(self, parent, width=200, height=200)

        self.create_oval(10,10,190,190, width = 3)
        for i in range(60):
            self.create_line(100 + math.cos(i*math.pi/30-math.pi/2)*(80+(i%5 != 0)*5),
                            100 + math.sin(i*math.pi/30-math.pi/2)*(80+(i%5 != 0)*5),
                            100 + math.cos(i*math.pi/30-math.pi/2)*90, 100 +
math.sin(i*math.pi/30-math.pi/2)*90, width = 2)
        self._second = self.create_line(0, 0, 0, 0, fill = "red")
        self._minute = self.create_line(0, 0, 0, 0, width = 2, fill = "blue")
        self._hour = self.create_line(0, 0, 0, 0, width = 3, fill = "green")

    def turn(self):
        gmtime = time.gmtime()
        self.coords(self._hour, 100, 100, 100 +
math.cos(gmtime[3]*math.pi/12+gmtime[4]*math.pi/360-math.pi/2)*45,
              100 + math.sin(gmtime[3]*math.pi/12+gmtime[4]*math.pi/360-math.pi/2)*45)
        self.coords(self._minute, 100, 100, 100 +
math.cos(gmtime[4]*math.pi/30+gmtime[5]*math.pi/1800-math.pi/2)*70,
              100 + math.sin(gmtime[4]*math.pi/30+gmtime[5]*math.pi/1800-math.pi/2)*70),
        self.coords(self._second, 100, 100, 100 + math.cos(gmtime[5]*math.pi/30-math.pi/2)*85,
              100 + math.sin(gmtime[5]*math.pi/30-math.pi/2)*85)
        self.update()

root = Tk.Tk()
root.title('Horloge')
h = Horloge(root)
h.pack()
btn = Tk.Button(root, text = 'Démarrer', command = h.start)
btn.pack()
root.mainloop()

```

V - Téléchargement

Version	Date	Taille	Mode FTP	Mode HTTP de secours
1.0.1.0.0	2007.01.30	882 o	TkHorloge.zip	TkHorloge.zip