

LS4 - Travaux pratiques 5

28 février 2014

1 Premier programme graphique

1. Dans un interpréteur python, importez le module tkinter.
2. Ce module contient une classe Tk. Instanciez un objet fenêtre de cette classe grâce à la commande :

```
fenetre = tkinter.Tk()
```

Vous voyez alors apparaître une fenêtre.

3. Nous allons commencer par construire un bouton pour fermer la fenêtre. Pour cela il faut créer une instance de la classe Button et la placer dans la fenêtre. Tapez la commande suivante :

```
quitter = tkinter.Button(fenetre, text="Quitter", command=fenetre.destroy)
```

4. Le bouton précédemment créé existe mais il n'est pas encore visible. Pour cela il faut le placer dans la fenêtre. Il y a plusieurs façon de faire, la plus simple est d'invoquer la méthode pack sans argument :

```
quitter.pack()
```

Placez le bouton quitter. Notez que si on n'a pas besoin de faire référence au bouton dans la suite du code, on peut écrire directement :

```
tkinter.Button(fenetre, text="Quitter", command=fenetre.destroy).pack()
```

5. Cliquez sur le bouton précédent : selon les cas, la fenêtre se ferme ou rien ne se passe. Si rien ne se passe, pour que le bouton soit actif, mettez la fenêtre en attente d'événements grâce à la méthode mainloop (fenetre.mainloop()) Vous aurez noté que la méthode mainloop ne rend pas la main avant la destruction de la fenêtre, il faut donc bien tout mettre en place avant de lancer mainloop.

q1 :

2 Dessin d'un rectangle

Pour dessiner dans la fenêtre "fenetre" de l'exercice précédent il faut créer un objet de type Canvas associé à "fenetre" par exemple avec :

```
canvas = tkinter.Canvas(fenetre,width=400,height=400,background="white")
```

et :

```
canvas.pack()
```

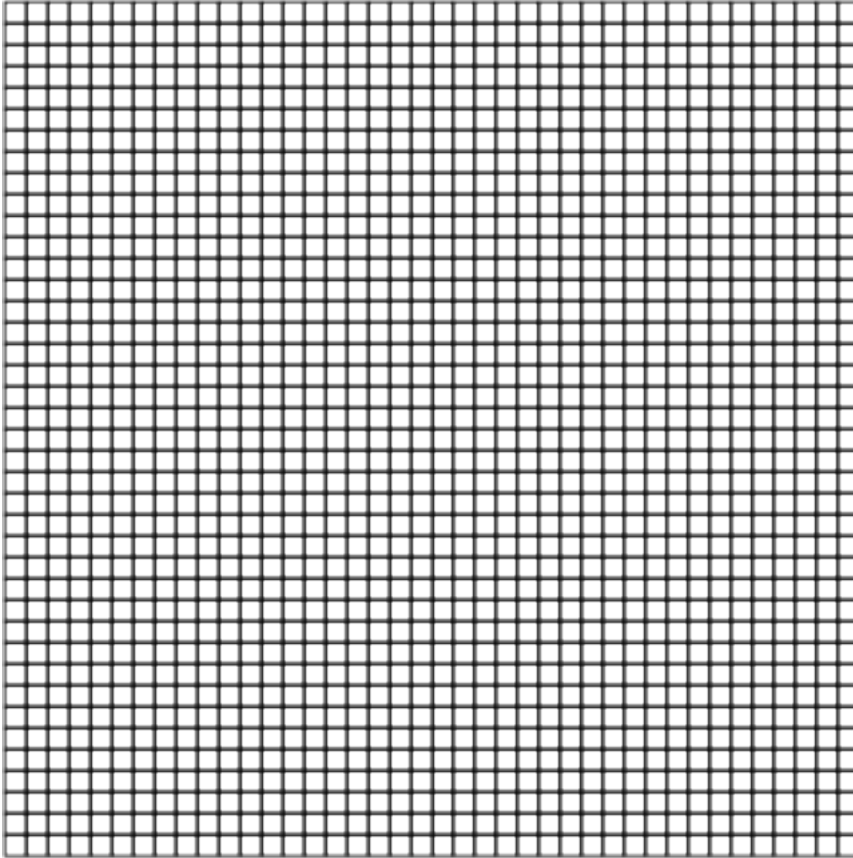
Les coordonnées (0,0) se trouvent en haut à gauche.

On peut dessiner un rectangle avec la méthode de Canvas :

```
create_rectangle(x0, y0, x1, y1, option, ...)
```

En appelant la méthode `delete("all")` on peut effacer le canvas. Pour mettre à jour le canvas on utilise la méthode `update()`.

Dessiner des rectangles de taille 10*10 sur tout le canvas pour obtenir une grille comme ici :



q2 :

3 Visualiser un tri

On considère des tableaux avec 20 éléments.

Ces tableaux contiennent des entiers de 0 à 40.

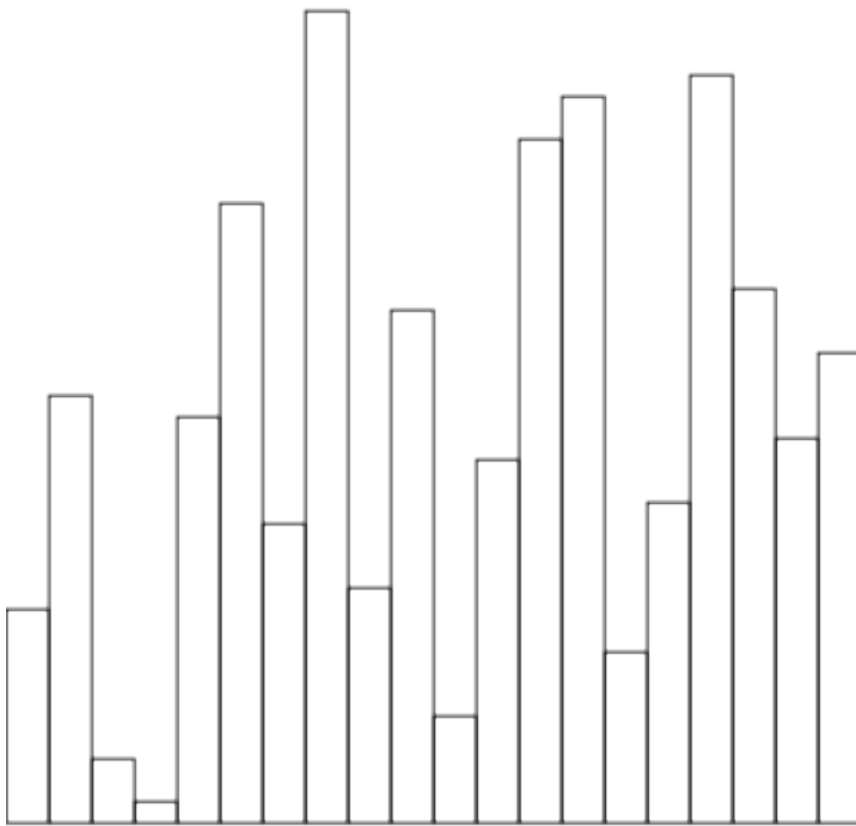
On visualise le contenu d'un tableau avec des histogrammes.

Chaque entier correspond à un rectangle d'une hauteur proportionnelle à la valeur.

Par exemple le tableau

```
T = [10,20,3,1,19,29,14,38,11,24,5,17,32,34,8,15,35,25,18,22]
```

correspond à :



Écrire une méthode `afficher(canvas,tab)` qui affiche sur un canvas le histogramme correspondant à un tableau.

Changer ensuite les méthodes pour les tris pour qu'elles affichent les étapes successives sur le même canvas. On peut effacer à chaque étape le canvas et afficher le tableau après l'étape du tri.

q3 :