

# TP2 : Tkinter

21 juin 2019

## 1 Présentation Tkinter

Tkinter est un outil de représentation graphique facile à prendre en main et disponible dans la librairie standard Python.

L'objet `Tk` est celui qui crée la fenêtre principale de l'application.

Dans la plupart des objets tkinter on peut ajouter de nombreux *Widgets*, ce sont des objets comme des boutons (`Button`), des étiquettes (`Label`) ou des champs de saisie (`Entry`). Une fois ces *Widgets* créés, il faut le placer dans la fenêtre avec l'opérateur de placement `.pack()`

Pour tout renseignement supplémentaire, vous pouvez utiliser `help()` dans la console ou aller sur le site <http://tkinter.fdex.eu>.

## 2 Ma première fenêtre

Pour créer votre première fenêtre, il vous suffit d'importer tkinter avec la commande :

```
import tkinter as tk
```

Il vous suffit alors de créer une nouvelle application en créant un objet de type `tk.Tk` et de le lancer avec la méthode `.mainloop()`.

/!\ Prenez bien note de lancer la `.mainloop` en dernier, car il est bloquant.

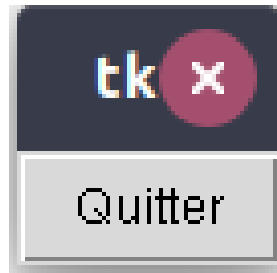
Vous pouvez ensuite commencer à modifier les paramètres de la fenêtre

- `.title(str)` : Change le titre de la fenêtre
- `.geometry(str)` : Change la taille de la fenêtre et son emplacement sur l'écran

Pour ajouter un bouton, il suffit de créer un objet `Button` avec comme premier paramètre l'objet `Tk` pour qu'il lui soit rattaché. On peut modifier le texte du bouton avec l'option *text* à l'initialisation (exemple : `Button(fenetre, text="Texte")`).

On donne également la fonction à exécuter avec l'option *command* dans l'initialisation. Il faut passer la fonction sans les parenthèses, ce qui limite les fonctions possibles aux fonctions sans paramètres. La fonction à utiliser pour fermer une fenêtre est `fenetre.destroy()`.

Vous pouvez maintenant créer une fenêtre possédant un bouton pour pouvoir la fermer. Elle devrait ressembler à celle-ci :



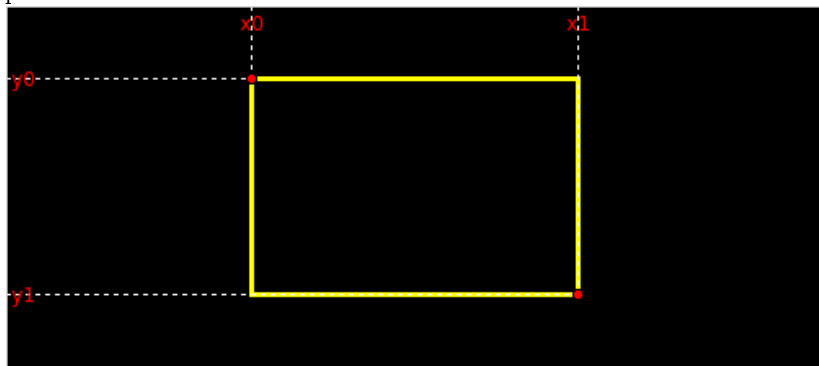
### 3 Les canvas

Le **Canvas** est un *Widget* servant principalement à contenir des dessins. Pour dessiner dans le canvas vous pouvez utiliser les commandes suivantes :

- `create_arc()` : permet de dessiner un arc de cercle
- `create_line()` : permet de dessiner une ligne
- `create_oval()` : permet de dessiner un cercle ou un ovale
- `create_rectangle()` : permet de dessiner un rectangle
- `create_polygon()` : permet de dessiner un polygone

La plupart de ces fonctions demandent la zone dans laquelle la forme doit être dessinée. Les informations demandées sont le point en haut à gauche (`x0`, `y0`), le point en bas à droite (`x1`, `y1`), et des options additionnelles pouvant être spécifiques à chaque forme.

\\!\\ Notez que le point en haut à gauche de la fenêtre est de coordonnées (0,0) et que l'axe des  $x$  est sur la hauteur



Chaque élément dessiné dans le Canvas pourra ensuite être déplacé ou supprimé du Canvas grâce aux méthodes `move` et `delete`. Pour pouvoir utiliser ces méthodes, il vous faudra avoir récupéré l'identifiant de ces éléments qui est donné lorsqu'ils sont créés.

## 4 Le labyrinthe

À partir du tableau donné dans le fichier *labyrinthe.py*, créer une fenêtre tkinter où les -1 sont représentés des carrés noirs et les 0 des carrés blancs. Notez bien que le tableau est haut de 31 ligne et large de 28 colonnes.

Pour diminuer le travail à réaliser, vous pouvez entrer l'option *bg* à l'initialisation du *Canvas* pour spécifier la couleur de son fond (couleur à donner en anglais, entre guillemets).

Votre travail devrait ressembler à ceci :

