

## Bibliothèque graphique Tkinter pour Python - fiche simplifiée

Importation de la bibliothèque : `from tkinter import *`

Création d'une fenêtre :

- Constructeur :  
`fenetre = Tk()` *#fenêtre par défaut. Tk() est un objet*
- Méthodes :
  - `fenetre.title("Titre de la fenêtre")` *#titre de la fenêtre*
  - `fenetre.resizable(True, False)` *#redimensionnement de la fenêtre possible en largeur, impossible en hauteur*
  - `fenetre.maxsize(500, 600)` *#largeur maximale = 500 pixels, hauteur maximale = 600 pixels*
  - `fenetre.minsize(200, 100)` *#largeur minimale = 200 pixels, hauteur minimale = 100 pixels*
  - `fenetre.update()` *#force le rafraichissement de l'affichage*
  - `fenetre.mainloop()` *#démarré le traitement des événements*
  - `fenetre.quit()` *#fait sortir de la boucle des événements (mainloop)*

Création d'une zone de dessin blanche (canevas ou *canvas* en anglais) de largeur 500 pixels et de hauteur 600 pixels:

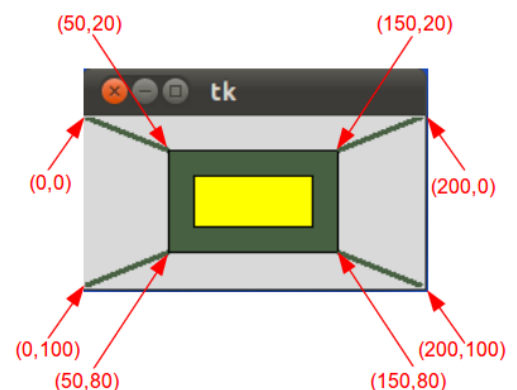
- `canevas = Canvas(fenetre, width=500, height=600, bg='white')` *#bg définit la couleur de fond, c'est un argument optionnel*
- `canevas.pack()` *#ajoute la zone de dessin à la fenêtre*
- `canevas.winfo_height()` *#renvoie la hauteur de la zone de dessin*
- `canevas.winfo_width()` *#renvoie la largeur de la zone de dessin*

Les éléments graphiques qu'on ajoute dans la fenêtre sont appelés des widgets. Cela peut être une image, du texte, une liste déroulante, des boutons à cocher, ...

- `picture = PhotoImage(file='nom_image.gif')`
- `canevas.create_image(2, 5, anchor=NW, image=picture)` *#ajoute une image dans la zone de dessin. Le coin haut-gauche de l'image est décalé de 2 pixels vers la gauche et 5 pixels vers le bas par rapport au coin haut-gauche de la fenêtre (NW = North-West). L'option state='hidden' permet de rendre l'image invisible.*

Cette image est un widget qui peut être stocké dans une variable. Dans ce cas, on écrira :

- `nom_widget = canevas.create_image(2, 5, anchor=NW, image=picture)`
- `canevas.create_text(2, 5, anchor=NW, text='Exemple', font=('Arial', '14', 'italic'), fill='green')` *#ajoute le texte 'Exemple' en vert avec la police indiquée dans la zone de dessin à la position indiquée (idem que pour la méthode create\_image)*
  - `canevas.create_line(0, 0, 50, 20, fill='green')` *#dessine une ligne entre les points de coordonnées (0,0) et (50,20) de couleur verte*
  - `canevas.create_rectangle(50, 20, 150, 80, fill='green', outline='black')` *#dessine un rectangle dont le coin supérieur gauche a pour coordonnées (50,20) et le coin inférieur droit a pour coordonnées (150, 80), de couleur verte avec une bordure noire.*



- `canevas.create_oval(50, 20, 150, 80, fill='green', outline='black')` *#dessine une ellipse (ou un cercle) inscrit dans un rectangle (ou un carré) dont les coordonnées des coins supérieur gauche et inférieur droit sont respectivement aux coordonnées (50,20) et (150,80)*
- `canevas.bind_all('<KeyPress-Down>', action)` *#applique la fonction action lorsque la touche bas du pavé numérique est enfoncée.*
- `canevas.delete(nom_widget)` *#supprime le widget nommé nom\_widget.*
- `canevas.move(nom_widget, 3, 4)` *#déplace le widget nommé nom\_widget de 3 pixels vers la droite et de 4 pixels vers le bas.*
- `canevas.itemconfig(nom_widget, attribut='nouvel valeur de l'attribut')` *#modifie la valeur de l'attribut du widget nommé nom\_widget. Cet attribut peut être l'image importée si le widget est une image, la chaîne de caractères si le widget est une zone de texte.*