



Le module **tkinter** (tool kit interface) permet de créer des interfaces graphiques.

### Les contrôles

Les **contrôles** permettent de créer la zone graphique et tout ce qu'elle pourra contenir : boutons cliquables, zones de dessins (canevas), conteneur, ...

#### Importation / Syntaxe

En début de programme : **import** tkinter

Utilisation d'un contrôle : **tkinter.nom\_controle**(parent, attribut1=valeur, attribut2=valeur, ...)

**parent** (*obligatoire*) : variable désignant le nom du conteneur

**attributs** (*optionnels*) : liste des attributs avec leur valeur

Remarque : On peut aussi saisir en début de programme : **from** tkinter **import**\*

Ce qui évite de réécrire systématiquement le nom du module **tkinter**. devant le contrôle.

On peut aussi saisir : **import** tkinter **as** tk. Cela permet de raccourcir le nom du module.

#### Contrôles standards / Description / Attributs

**tkinter.Tk()**

Crée une fenêtre parente, racine, primaire et unique (formulaire, cadre de l'application, ..., conteneur principal)



Remarque : Le contrôle **tkinter.Tk()** ne peut être utilisé qu'une seule fois dans un programme. On ne peut donc ouvrir qu'une seule fenêtre principale (ou parente). Pas de parent ou d'attributs dans les parenthèses.

**tkinter.Button**(parent, attribut, ...) Crée un bouton cliquable.

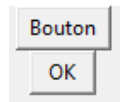
**texte**=valeur' Texte du bouton.

**command**=valeur Action à effectuer lorsque le bouton est cliqué.

**background**=valeur Couleur du bouton ("green", "red", "blue", "cyan", ...).

**foreground**=valeur Couleur du texte ("green", "red", "blue", "cyan", ...).

**height**=valeur, **width**=valeur Hauteur et largeur du bouton en pixels.



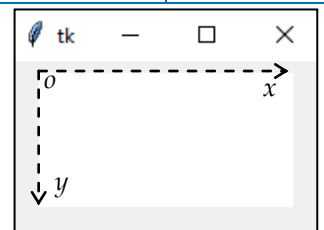
**tkinter.Canvas**(parent, attribut, ...)

Crée une zone de dessin, ou canevas, ou toile avec son système d'axes (O en haut à gauche).

**height**=valeur Hauteur de la toile (ou canevas) en pixels.

**width**=valeur Largeur de la toile (ou canevas) en pixels.

**background**=valeur Couleur de la toile ("green", "red", "blue", "cyan", ...).



#### Exemples

```
Mafenetre = tkinter.Tk()
```

```
Mondessin = tkinter.Canvas(Mafenetre, width=500, height=200, bg='white')
```

Crée une zone de dessin nommé **Mondessin** dans le conteneur nommé **Mafenetre**. Cette zone de dessin a une largeur de 500 pixels, une hauteur de 200 pixels et un fond blanc.

```
Mafenetre = tkinter.Tk()
```

```
btn1 = tkinter.Button(Mafenetre, text="OK", foreground='blue')
```

Crée un bouton nommé **btn1** dans le conteneur nommé **Mafenetre** avec un texte "OK" écrit en bleu.

Remarque : Les contrôles permettent de créer un objet mais ne l'affiche pas. Il faudra pour cela utiliser une méthode **.pack**, **.place** ou **.grid** (Voir plus loin).

## Les méthodes

Les **méthodes** sont les actions qui vont être réalisées sur l'objet créé à l'aide d'un contrôle (bouton cliquable, zone de dessin, ...). Elles permettent d'afficher l'objet créé, de dessiner dans une zone de dessin, ...

Il existe des méthodes communes et des méthodes spécifiques à chaque contrôle (ou objet créé).

### Syntaxe

On applique une méthode à un objet `nom_objet.nom_methode(attribut1=valeur, attribut2=valeur, ... )`  
en écrivant selon la syntaxe ci-contre. `attributs (optionnels)` : liste des attributs avec leur valeur

Quelques méthodes générales :

| Méthodes / Description / Attributs          |  |
|---|--|
| <code>nom_objet.pack(attribut, ...)</code>  | Permet de positionner les contrôles dans le conteneur les uns à la suite des autres.<br><br><code>expand=valeur</code> 'true' : se redimensionne si son conteneur est étiré. 'False' : reste statique.<br><code>fill=valeur</code> 'x' : le contrôle remplit toute la ligne. 'y' : le contrôle remplit toute la colonne.<br><code>side=valeur</code> 'bottom' : en bas 'left' : à gauche 'right' : à droite 'top' : en haut  |
| <code>nom_objet.place(attribut, ...)</code> | Permet de positionner les contrôles dans le conteneur de façon précise et unique.<br><br><code>x=valeur</code> Position de l'abscisse du bord gauche du contrôle en pixels.<br><code>y=valeur</code> Position de l'ordonnée du bord gauche du contrôle en pixels.<br><code>height=valeur</code> Hauteur du contrôle en pixels.<br><code>width=valeur</code> Largeur du contrôle en pixels.<br><code>anchor=valeur</code> Position "boussole" : 'n', 'ne', 'e', 'se', 's', 'sw', 'w', 'nw', 'ns', 'ew', ... |
| <code>nom_objet.grid(attribut, ...)</code>  | Permet de positionner les contrôles dans le conteneur selon une grille.<br><br><code>column=valeur</code> Rang de la colonne où doit être placé le contrôle.<br><code>row=valeur</code> Rang de la ligne où doit être placé le contrôle.   |
| <code>nom_objet.destroy()</code>            | Détruit la fenêtre ou le contrôle précisé.   |
| <code>nom_fenetre.title()</code>            | Modifie le titre de la fenêtre créée par le contrôle <code>tkinter.Tk()</code> .   |

Quelques méthodes du contrôle Canvas :

| Méthodes / Description / Attributs  |   |
|---|---|
| <code>nom_canevas.create_line(xA, yA, xB, yB, ...)</code>   | Crée une ligne droite (1 seul segment) ou brisée (plusieurs segments) en reliant les points (xA;yA), (xB;yB), ...<br><br><code>fill=valeur</code> Couleur de la ligne : "green", "blue", "red", "light grey", ...<br><code>width=valeur</code> Largeur du trait en pixels.                          |
| <code>nom_canevas.create_rectangle(xG, yH, xD, yB)</code>   | Crée un carré ou un rectangle en saisissant les 4 valeurs ci-contre. (H : haut, B : bas, G : gauche, D : droite)  |
| <code>nom_canevas.create_polygon(xA, yA, xB, yB, ...)</code>  | Crée une figure géométrique, régulière ou non, fermée.  |
| <code>nom_canevas.create_oval(xG, yH, xD, yB)</code>  | Crée un cercle ou ellipse en saisissant les 4 valeurs ci-contre.<br><br><code>fill=valeur</code> Couleur de l'intérieur de la figure : "green", "red", "blue", "light grey", ...)<br><code>outline=valeur</code> Couleur de la ligne.<br><code>width=valeur</code> Largeur de la bordure en pixels. |
| <code>nom_canevas.text(x, y, text='Texte')</code>   | Crée un élément permettant d'insérer un objet texte.<br>x : abscisse bord gauche y : ordonnée bord haut<br><br><code>fill=valeur</code> Couleur du texte.<br><code>size=valeur</code> Taille de la police : 10, 11, 12, ...   |
| <code>nom_image = tkinter.PhotoImage(file="fichier.gif")</code><br><code>nom_canevas.create_image(x, y, image=nom_image)</code> | Crée un cadre permettant d'insérer un objet image (chargée grâce au contrôle <code>tkinter.PhotoImage()</code> )<br>x : abscisse bord gauche y : ordonnée bord haut   |

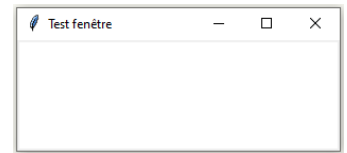
## Exemple : Contrôles, méthodes et attributs

```
Mafenetre = tkinter.Tk()
Mondessin = tkinter.Canvas(Mafenetre, width=300, height=100)
Mafenetre.title('Test fenêtre')
Mondessin.pack()
```

Contrôles

Attributs

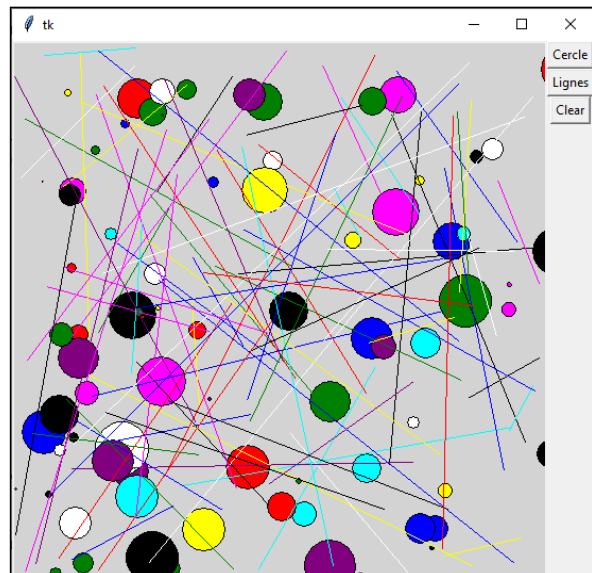
Méthodes



On crée une fenêtre (ligne 1) nommée "Mafenetre" avec le texte "Test Fenêtre" (ligne 3) dans laquelle on affiche une zone de dessin (ligne 2) à l'aide de la méthode `pack()` (ligne 4).

## Exemple : Dessin psychédélique

```
1 import tkinter as tk
2 import random as rd
3
4 def rd_col():
5     return rd.choice(("black", "red", "green", "blue", "yellow", "magenta", "cyan", "white", "purple"))
6
7 def dessine_cercles():
8     for i in range(20):
9         x, y = rd.randint(1, 500), rd.randint(1, 500)
10        diameter = rd.randint(1, 50)
11        canvas.create_oval(x, y, x+diameter, y+diameter, fill=rd_col())
12
13 def dessine_lignes():
14     for i in range(20):
15         x1 = rd.randint(1, 500)
16         x2 = rd.randint(1, 500)
17         y1 = rd.randint(1, 500)
18         y2 = rd.randint(1, 500)
19         canvas.create_line(x1, y1, x2, y2, fill=rd_col())
20
21 def effacer():
22     canvas.create_rectangle(0, 0, 501, 501, fill='light grey')
23
24 fenetre = tk.Tk()
25 canvas = tk.Canvas(height=500, width=500, bg='light grey')
26 canvas.pack(side=tk.LEFT)
27
28 bouton_cercles = tk.Button(text="Cercle", command=dessine_cercles)
29 bouton_lignes = tk.Button(text="Lignes", command=dessine_lignes)
30 bouton_clean = tk.Button(text="Clear", command=effacer)
31 bouton_cercles.pack()
32 bouton_lignes.pack()
33 bouton_clean.pack()
34
35 fenetre.mainloop()
```



Une fois le programme lancé :

Le bouton "Cercle" permet de dessiner 20 disques de tailles et couleurs aléatoires.

Le bouton "Lignes" permet de dessiner 20 segments de longueurs et couleurs aléatoires.

Le bouton "Clear" permet d'effacer le dessin.