

# TKINTER

Tkinter (Tk interface) est un module intégré à la bibliothèque standard de Python, bien qu'il ne soit pas maintenu directement par les développeurs de Python. Il offre un moyen de créer des interfaces graphiques via Python.

Nous allons commencer par voir le code minimal pour créer une fenêtre avec Tkinter. Petit à petit, nous allons apprendre à rajouter des choses, mais commençons par voir la base de code que l'on retrouve d'une interface Tkinter à l'autre.

Étant en Python, ce code minimal est plutôt court :

```
"""Premier exemple avec Tkinter.

On crée une fenêtre simple qui souhaite la bienvenue à l'utilisateur.

"""

# On importe Tkinter
from tkinter import *

# On crée une fenêtre, racine de notre interface
fenetre = Tk()
# permet de modifier la taille de la fenêtre
fenetre.geometry("400x400")
# On crée un label (ligne de texte) souhaitant la bienvenue
# Note : le premier paramètre passé au constructeur de Label est notre
# fenêtre
champ_label = Label(fenetre, text="Salut les NSI !")

# On affiche le label dans la fenêtre
champ_label.pack()

# On démarre la boucle Tkinter qui s'interrompt quand on ferme la fenêtre
fenetre.mainloop()
```

Essayez ce code, il affichera une fenêtre.

## DE NOMBREUX WIDGETS

Tkinter définit un grand nombre de widgets pouvant être utilisés dans notre fenêtre. Nous allons en voir ici quelques-uns.

## LES WIDGETS LES PLUS COMMUNS

### LES LABELS

C'est le premier widget que nous avons vu, hormis notre fenêtre principale qui en est un également. On s'en sert pour afficher du texte dans notre fenêtre, du texte qui ne sera pas modifié par l'utilisateur.

```
champ_label = Label(fenetre, text="contenu de notre champ label")
champ_label.pack()
```

N'oubliez pas que, pour qu'un widget apparaisse, il faut :

- qu'il prenne, en premier paramètre de la fonction, la fenêtre principale ;
- qu'il fasse appel à la méthode **pack**.

La méthode `pack` permet de positionner un objet dans une fenêtre ou dans un cadre, nous verrons plus loin quelques-uns de ses paramètres optionnels.

---

## LES BOUTONS

Les boutons sont des widgets sur lesquels on peut cliquer et qui peuvent déclencher des actions ou **commandes** comme nous le verrons ultérieurement plus en détail.

```
bouton_quitter = Button(fenetre, text="Quitter", command=fenetre.quit)
bouton_quitter.pack()
```

J'imagine que vous vous posez des questions sur le dernier paramètre passé à notre constructeur de `Button`. Il s'agit de l'action liée à un clic sur le bouton. Ici, c'est la méthode `quit` de notre fenêtre racine qui est appelée.

Ainsi, quand vous cliquez sur le bouton `Quitter`, la fenêtre se ferme.

---

## UNE LIGNE DE SAISIE / TEXTBOX

Le widget que nous allons voir à présent est une zone de texte dans laquelle l'utilisateur peut écrire. En fait de zone, il s'agit d'une ligne simple.

On préférera créer une variable Tkinter associée au champ de texte. Regardez le code qui suit :

```
var_texte = StringVar()

ligne_texte = Entry(fenetre, textvariable=var_texte, width=30)
ligne_texte.pack()
```

À la ligne 1, nous créons une variable Tkinter. En résumé, c'est une variable qui va ici contenir le texte de notre `Entry`. Il est possible de lier cette variable à une méthode de telle sorte que la méthode soit appelée quand la variable est modifiée (l'utilisateur écrit dans le champ **Entry**).

Exemple :

```

from tkinter import *

# On crée une fenêtre, racine de notre interface
fenetre = Tk()

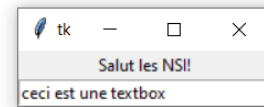
# On crée un label (ligne de texte) souhaitant la bienvenue
champ_label = Label(fenetre, text="Salut les NSI!")

# On affiche le label dans la fenêtre
champ_label.pack()

#création d'une champs de saisie
var_texte = StringVar()
var_texte.set("ceci est une textbox")
ligne_texte = Entry(fenetre, textvariable=var_texte, width=30)
ligne_texte.pack()

# On démarre la boucle Tkinter qui s'interrompt quand on ferme la fenêtre
fenetre.mainloop()

```



Il existe bien d'autre composants Tkinter, telle que des listes déroulantes, des cases à cocher...etc.

## PLACER DES COMPOSANTS

Il existe plusieurs manière de placer des composants. La méthode `.pack()` mais surtout la méthode `.place()` qui est un peu plus simple.

Il suffit de mettre le nom de notre composant, puis en argument de la fonction place, la position en x, la position en y, la largeur et la hauteur souhaitées.

`unWidget.place(x=20, y=10, width=100, height=25)`

**Exemple :**

```

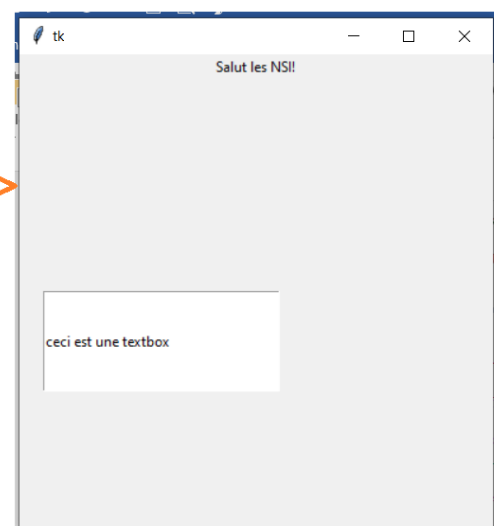
from tkinter import *

# On crée une fenêtre, racine de notre interface
fenetre = Tk()
#permet de modifier la taille de la fenêtre.
fenetre.geometry("400x400")
# On crée un label (ligne de texte) souhaitant la bienvenue
champ_label = Label(fenetre, text="Salut les NSI!")

# On affiche le label dans la fenêtre
champ_label.pack()

#création d'une champs de saisie
var_texte = StringVar()
var_texte.set("ceci est une textbox")
ligne_texte = Entry(fenetre, textvariable=var_texte, width=30)
#placement et affichage de la textbox
ligne_texte.place(x=20, y=200, width=200, height=85)
# On démarre la boucle Tkinter qui s'interrompt quand on ferme la fenêtre
fenetre.mainloop()

```



Voici les bases de Tkinter, pour plus de détails, vous pouvez rechercher sur internet.