

# Interface Homme Machine

## Python-Tkinter

### Travaux Pratiques N° 5

#### Objetif :

- Etudier les trois gestionnaires de positionnement dans un container de tkinter:
  - ✓ place
  - ✓ pack
  - ✓ grid

### Gestion de disposition des widgets dans un container

Lorsque nous concevons l'interface graphique de notre application, nous décidons des widgets que nous allons utiliser et comment nous allons organiser ces widgets dans l'application.

Pour organiser nos widgets, nous utilisons des objets non visibles spécialisés appelés gestionnaires de disposition.

Il existe deux types de widgets : les conteneurs et leurs enfants.

Les containers regroupent leurs enfants selon des schémas appropriés.

Tkinter a trois gestionnaires de disposition intégrés : les gestionnaires "place", "pack" et "grid".

Le gestionnaire "place" positionne les widgets en coordonnées absolues

Le gestionnaire "pack" organise les widgets dans des zones ou boîtes horizontales et verticales. C'est un gestionnaire très simple. Pour créer des mises en page plus complexes, nous pouvons combiner plusieurs cadres (Frame) ayant chacun une mise en page partielle utilisant le gestionnaire "pack".

Le gestionnaire "grid" place les widgets dans une grille à deux dimensions. C'est le gestionnaire de disposition le plus complexe et le plus puissant de Tkinter.

#### I-Positionnement absolu avec "place"

Dans la plupart des cas, les programmeurs doivent utiliser des gestionnaires de disposition.

Il y a quelques situations où nous pourrions utiliser le positionnement absolu.

En positionnement absolu, le programmeur spécifie la position et la taille de chaque widget en pixels.

La taille et la position d'un widget ne change pas si nous redimensionnons une fenêtre

Avec ce type de positionnement, les applications n'ont pas le même look sur différentes plateformes. En plus le changement de la police de caractères peut fausser le positionnement.

Exemple de programme utilisant le gestionnaire "place"

## Gestionnaire de position : place : positionnement absolu



```
from PIL import Image, ImageTk
from tkinter import Tk, BOTH
from tkinter.ttk import Frame, Label, Style

root = Tk()
root.geometry("800x600+30+30")
root.title("Gestionnaire de position 'place'")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

style = Style()
style.configure("TFrame", background="#333")

tennis = Image.open("tennis.jpg")
tennisphoto = ImageTk.PhotoImage(tennis)
label1 = Label(frame, image=tennisphoto)
label1.image = tennisphoto
label1.place(x=20, y=20)

foot = Image.open("football.jpg")
footphoto = ImageTk.PhotoImage(foot)
label2 = Label(frame, image=footphoto)
label2.image = footphoto
label2.place(x=340, y=20)
```

```

basket = Image.open("basket.jpg")
basketphoto = ImageTk.PhotoImage(basket)
label3 = Label(frame, image=basketphoto)
label3.image = basketphoto
label3.place(x=20, y=260)

athletisme = Image.open("mbr.jpg")
athletismephoto = ImageTk.PhotoImage(athletisme)
label4 = Label(frame, image=athletismephoto)
label4.image = athletismephoto
label4.place(x=380, y=260)

root.mainloop()

```

Dans l'exemple, nous plaçons trois images en utilisant le positionnement absolu.

Nous utilisons les classes `Image` et `ImageTk` du module `PIL` (Python Imaging Library)

```
from PIL import Image, ImageTk
```

Nous définissons ensuite un `frame` qui prend toute la surface de la fenêtre parent (`root`). Les quatre étiquettes avec leurs images sont placées sur ce `frame`.

```
frame = Frame(root)
frame.pack(fill=BOTH, expand=True)
```

Nous configurons notre `frame` pour avoir un fond gris foncé.

```
style = Style()
style.configure("TFrame", background="#333")
```

Nous créons un objet `image` et un objet `image photo` à partir d'une image situé dans le dossier courant.

```
tennis = Image.open("tennis.jpg")
tennisphoto = ImageTk.PhotoImage(tennis)
```

Nous créons une étiquette avec une image. Les étiquettes peuvent contenir du texte ou des images.

```
label1 = Label(frame, image=tennisphoto)
```

Nous devons garder la référence à l'image pour éviter que l'image ne soit détruite par le `GarbageCollector`.

```
label1.image = tennisphoto
```

L'étiquette est ensuite placée sur le cadre aux coordonnées `x = 20` et `y = 20`.

```
label1.place(x=20, y=20)
```

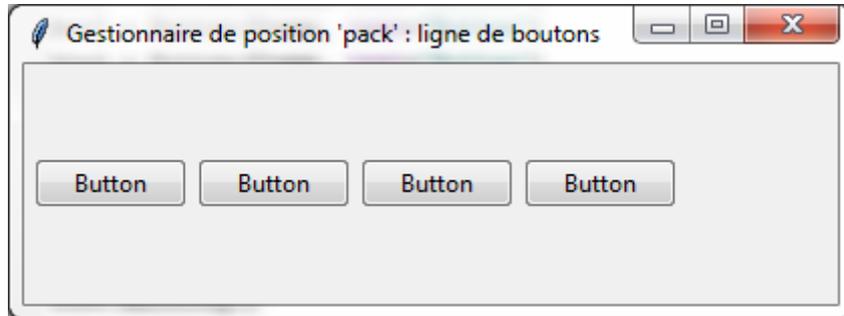
## II-Gestionnaire de positionnement "pack"

Nous allons expliquer l'utilisation de ce gestionnaire par des exemples.

### Exemple1: Création d'une ligne de boutons

Dans l'exemple suivant, nous créons une rangée de boutons avec le gestionnaire de pack. en les séparant par un peu d'espace les boutons.

### Gestionnaire de position pack : Exemple 1 : ligne de boutons



```
from tkinter import Tk, BOTH, LEFT
from tkinter.ttk import Frame, Button

root = Tk()
root.geometry("380x120+300+300")
root.title("Gestionnaire de position 'pack' : ligne de boutons")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

btn1 = Button(frame, text="Button")
btn2 = Button(frame, text="Button")
btn3 = Button(frame, text="Button")
btn4 = Button(frame, text="Button")

btn1.pack(side=LEFT, padx=5)
btn2.pack(side=LEFT)
btn3.pack(side=LEFT, padx=5)
btn4.pack(side=LEFT)

root.mainloop()
```

L'exemple met quatre boutons dans une rangée.

```
btn1.pack(side=LEFT , padx=5)
btn2.pack(side=LEFT)
btn3.pack(side=LEFT , padx=5)
btn4.pack(side=LEFT)
```

Le paramètre "side" de chaque bouton est "LEFT" ; les boutons forment alors une ligne.

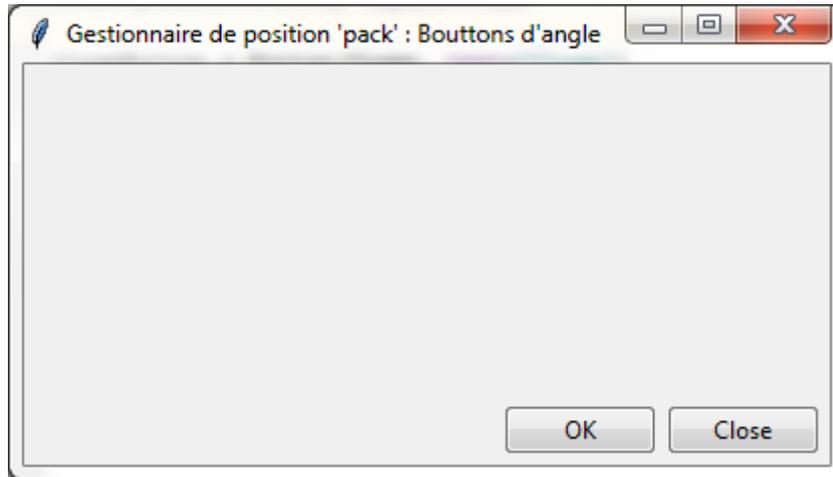
Le paramètre "padx" place un espace horizontal entre les boutons.

### Exemple 2 : Boutons d'angle

L'exemple suivant place deux boutons dans le coin inférieur droit de la fenêtre.

Les boutons sont ancrés au gestionnaire de pack.

## Gestionnaire de position pack : Exemple 2 : boutons d'angles (Méthode1)



```
from tkinter import Tk, RIGHT, BOTH, S
from tkinter.ttk import Frame, Button

root = Tk()
root.geometry("400x200+300+300")
root.title("Gestionnaire de position 'pack' : Bouttons d'angle")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

closeButton = Button(frame, text="Close")
closeButton.pack(anchor=S, side=RIGHT, padx=5, pady=5)
okButton = Button(frame, text="OK")
okButton.pack(side=RIGHT, pady=5, anchor=S)

root.mainloop()
```

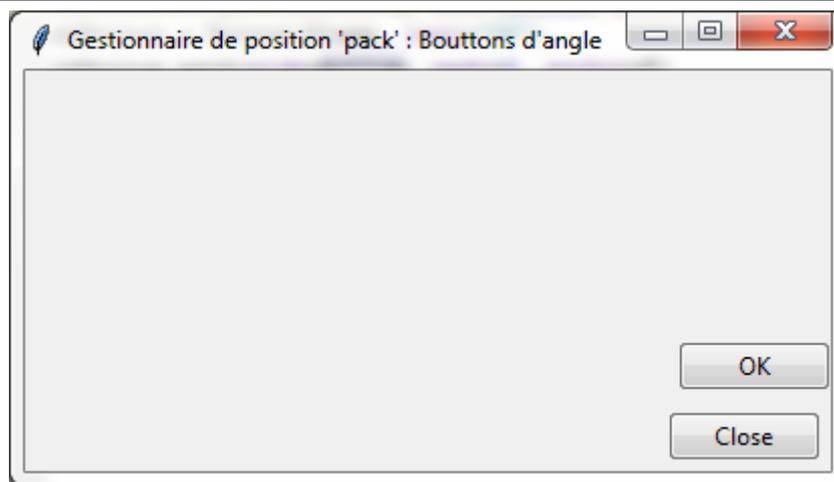
Les boutons sont placés dans le coin à l'aide des paramètres "side" et "anchor" de pack.

```
closeButton.pack(anchor=S, side=RIGHT, padx=5, pady=5)
```

Dans l'exemple ci-dessous, nous avons modifié les valeurs de ces paramètres:

```
closeButton.pack(anchor=E, side=BOTTOM, padx=5, pady=5)
```

## Gestionnaire de position pack : Exemple 2 : boutons d'angles (Méthode2)



```
from tkinter import Tk, RIGHT, BOTH, E, BOTTOM
from tkinter.ttk import Frame, Button

root = Tk()
root.geometry("400x200+300+300")
root.title("Gestionnaire de position 'pack' : Bouttons d'angle")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

closeButton = Button(frame, text="Close")
closeButton.pack(anchor=E, side=BOTTOM, padx=5, pady=5)
okButton = Button(frame, text="OK")
okButton.pack(side=BOTTOM, pady=5, anchor=E)

root.mainloop()
```

### Exemple3: Création de plusieurs lignes de boutons

Le gestionnaire de géométrie de pack convient aux schémas de positionnement simples.

Pour créer des schémas plus compliqués, nous combinons plusieurs Frames ayant chacun un schéma partiel créé avec le gestionnaire de pack.

### Gestionnaire de position pack : Exemple 3 : création de plusieurs lignes de boutons



```

from tkinter import Tk, BOTH, LEFT
from tkinter.ttk import Frame, Button

root = Tk()
root.geometry("380x120+300+300")
root.title("Plusieurs Lignes de Bouttons")

frame = Frame(root) #Création du frame principal
frame.pack(fill=BOTH, expand=True)

frame1 =Frame(frame) #Création de frame1 enfant de frame
frame1.pack(pady=5) #positionnement de frame1 dans root

btn1 = Button(frame1, text="Button")
btn2 = Button(frame1, text="Button")
btn3 = Button(frame1, text="Button")
btn4 = Button(frame1, text="Button")
#positionnement des 4 premiers boutons dans frame1
btn1.pack(side=LEFT, padx=5)
btn2.pack(side=LEFT)
btn3.pack(side=LEFT, padx=5)
btn4.pack(side=LEFT)

frame2 = Frame(frame)
frame2.pack()

btn1 = Button(frame2, text="Button")
btn2 = Button(frame2, text="Button")
btn3 = Button(frame2, text="Button")
btn4 = Button(frame2, text="Button")

btn1.pack(side=LEFT, padx=5)
btn2.pack(side=LEFT)
btn3.pack(side=LEFT, padx=5)
btn4.pack(side=LEFT)

root.mainloop()

```

L'exemple crée deux rangées de boutons. Chaque ligne est placée dans un Frame. Les Frames sont eux placés sur le Frame parent de base.

```

frame1 =Frame(frame) #Création de frame1 enfant de frame
frame1.pack(pady=5) #positionnement de frame1 dans root

```

Les 4 premiers boutons sont placés dans le premier Frame et positionnés avec "pack".

```

btn1 = Button(frame1, text="Button")
btn2 = Button(frame1, text="Button")
btn3 = Button(frame1, text="Button")
btn4 = Button(frame1, text="Button")
#positionnement des 4 premiers boutons dans frame1
btn1.pack(side=LEFT, padx=5)
btn2.pack(side=LEFT)
btn3.pack(side=LEFT, padx=5)
btn4.pack(side=LEFT)

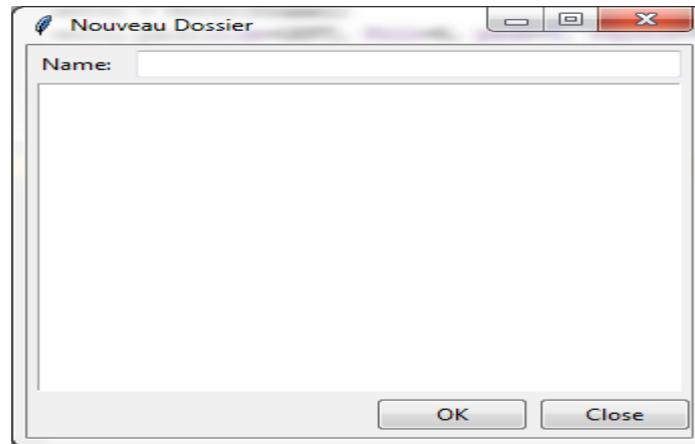
```

On procède de la même manière avec les 4 autres boutons dans le 2<sup>ème</sup> frame.

#### Exemple4: Exemple pratique : Création d'un nouveau dossier

Cet exemple est un exemple pratique d'utilisation du gestionnaire "pack"

#### Gestionnaire de position pack : Exemple 4 : Exemple pratique : Nouveau Dossier



```
from tkinter import Tk, Text, BOTH, LEFT, X, RIGHT
from tkinter.ttk import Frame, Button, Label, Entry

root = Tk()
root.geometry("330x300+300+300")
root.title("Nouveau Dossier")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

frame1 = Frame(frame)
frame1.pack(fill=X)

lbl = Label(frame1, text="Name:")
lbl.pack(side=LEFT, padx=5, pady=5)

entry = Entry(frame1)
entry.pack(side=LEFT, fill=X, padx=5, expand=True)

frame2 = Frame(frame)
frame2.pack(fill=BOTH, expand=True)

txt = Text(frame2, width=20, height=10)
txt.pack(fill=BOTH, expand=True, padx=5)

frame3 = Frame(frame)
frame3.pack(fill=X)

closeBtn = Button(frame3, text="Close")
closeBtn.pack(side=RIGHT, pady=5)
okBtn = Button(frame3, text="OK")
okBtn.pack(side=RIGHT, padx=5)

root.mainloop()
```

Pour créer cette mise en page, nous avons créé quatre Frames. Notez qu'en plus de gérer les widgets enfants à l'intérieur de leurs Frames parents, nous gérons également les Frames dans leur Frame parent de base.

Le Frame de base occupe toute la surface de la fenêtre racine.

```
frame = Frame(root)
frame.pack(fill=BOTH, expand=True)
```

Dans le premier Frame, nous mettons un Label et une Entry. La combinaison des paramètres "fill =X" de remplissage et "expand =True" d'expansion permettent d'étirer le widget Entry horizontalement.

```
frame1 = Frame(frame)
frame1.pack(fill=X)

lbl = Label(frame1, text="Name:")
lbl.pack(side=LEFT, padx=5, pady=5)

entry = Entry(frame1)
entry.pack(side=LEFT, fill=X, padx=5, expand=True)
```

Le deuxième Frame est occupé par un widget Entry. Le Frame et son enfant remplissent leur zone dans les deux sens grâce aux paramètres "fill=BOTH" et "expand=True"

```
frame2 = Frame(frame)
frame2.pack(fill=BOTH, expand=True)

txt = Text(frame2, width=20, height=10)
txt.pack(fill=BOTH, expand=True, padx=5)
```

Le troisième Frame contient deux boutons. Les boutons sont placés à droite de leur conteneur.

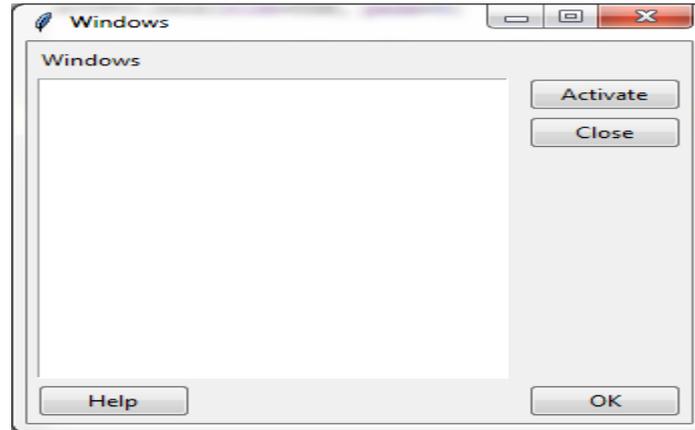
```
frame3 = Frame(frame)
frame3.pack(fill=X)

closeBtn = Button(frame3, text="Close")
closeBtn.pack(side=RIGHT, pady=5)
okBtn = Button(frame3, text="OK")
okBtn.pack(side=RIGHT, padx=5)
```

#### Exemple5: Fenêtre

L'exemple suivant est un autre exemple pratique d'utilisation du gestionnaire pack.

**Gestionnaire de position pack : Exemple 5 : Exemple pratique : Fenêtre**



```

from tkinter import Tk, Text, BOTH, LEFT, X, Y, TOP, RIGHT
from tkinter.ttk import Frame, Button, Label

#====Fenêtre racine====
root = Tk()
root.geometry("330x300+300+300")
root.title("Windows")

#====Frame de base ====
frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

#====frame1 : contient un seul widget Label====
frame1 = Frame(frame)
frame1.pack(fill=X) #par défaut side=TOP

lbl = Label(frame1, text="Windows")
lbl.pack(side=LEFT, padx=5, pady=5)

#====frame2 : contient un widget "Text", et un frame interne frame21==
frame2 = Frame(frame)
frame2.pack(fill=BOTH, expand=True) #par défaut side=TOP

txt = Text(frame2, width=20, height=10)
txt.pack(side=LEFT, fill=BOTH, expand=True, padx=5)

#--frame21 : Frame interne qui contient les boutons Activer et Close--
frame21 = Frame(frame2)
frame21.pack(side=LEFT, fill=Y)

actBtn = Button(frame21, text="Activer")
actBtn.pack(side=TOP, padx=5)

closeBtn = Button(frame21, text="Close")
closeBtn.pack(side=TOP, pady=5)
#-----
#==== frame3 : Contient les boutons Help et Ok====
frame3 = Frame(frame) #par défaut side=TOP
frame3.pack(fill=X)

hlpBtn = Button(frame3, text="Help")
hlpBtn.pack(side=LEFT, pady=5, padx=5)

```

```
okBtn = Button(frame3, text="OK")
okBtn.pack(side=RIGHT, padx=5)

root.mainloop()
```

Dans cet exemple, nous avons utilisé quatre Frames en plus du frame de base.

Le premier Frame frame1 contient un seul widget qui est un Label que nous avons positionné à gauche avec quelques espaces autour de ses frontières.

Le second Frame contient un widget "Text" que nous avons positionné à gauche et qui remplit la majeure partie du frame2.

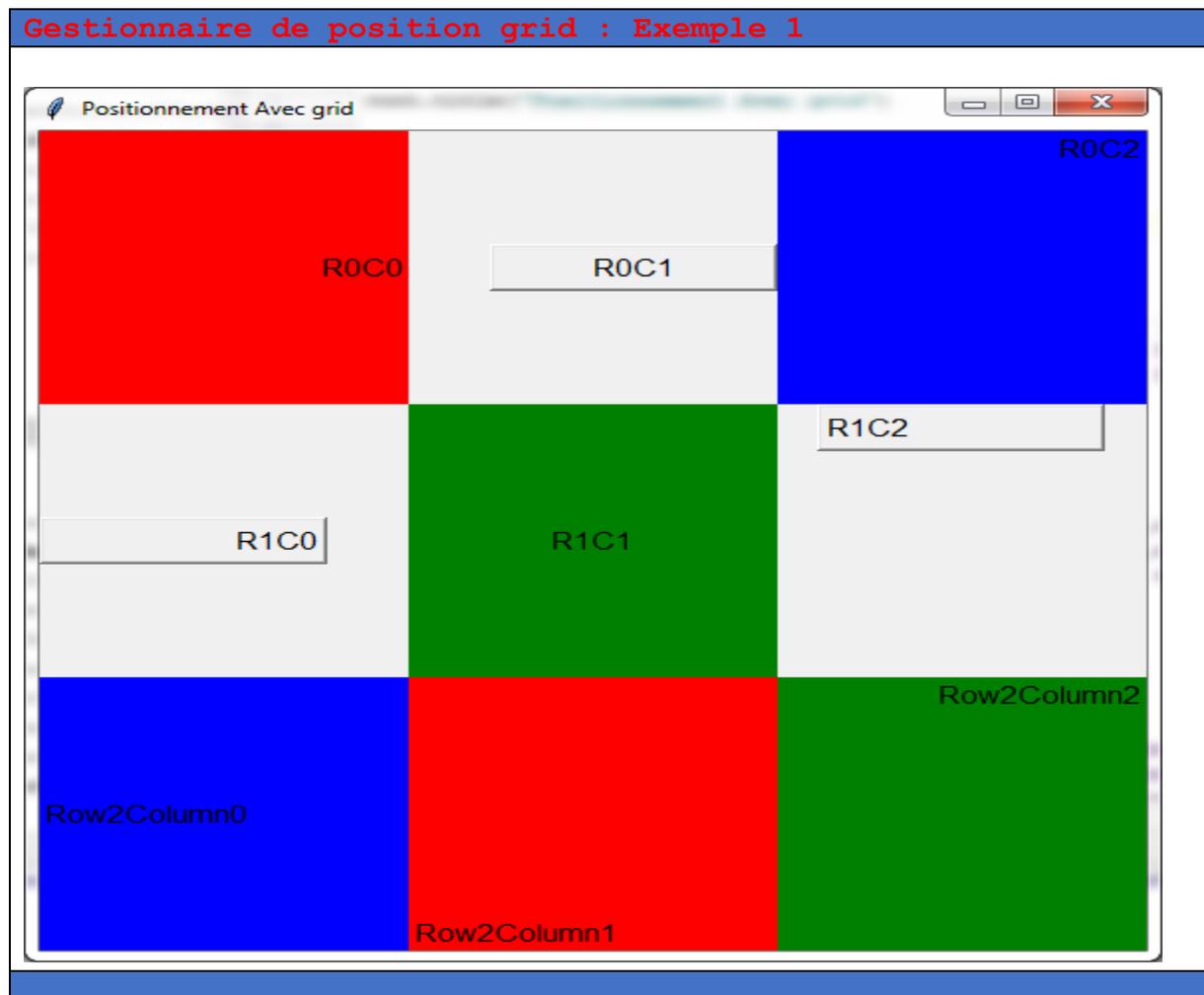
Le Frame interne frame21 contient deux boutons et sont positionnés en haut.

Enfin, le dernier Frame frame3 contient les deux boutons "Help" qui est placé à gauche et "Ok" qui est placé à droite.

### III-Gestionnaire de positionnement "grid"

Le gestionnaire "grid" est le plus complexe et le plus puissant. Nous allons l'expliquer par les exemples qui suivent

#### Exemple1 : Quelques options de grid



```

from tkinter import *

root = Tk()
root.title("Positionnement Avec grid")

frprincipal = Frame(root)
frprincipal.pack()

frgrid = Frame(frprincipal)
frgrid.pack()

#Label
lbr0c0 = Label(frgrid, text="R0C0", height=10, width=20, font=(14),
bg="red", anchor="e")
lbr1c1 = Label(frgrid, text="R1C1", height=10, width=20, font=(14),
bg="green", anchor="center")
lbr0c2 = Label(frgrid, text="R0C2", height=10, width=20, font = (14),
bg="blue", anchor="ne")
lbr0c0.grid(row=0, column=0)
lbr1c1.grid(row=1, column=1)
lbr0c2.grid(row=0, column=2)

#button
btn01 = Button(frgrid, text="R0C1", width=15, font = (14), anchor="center")
btn10 = Button(frgrid, text="R1C0", width=15, font = (14), anchor="e")
btn12 = Button(frgrid, text="R1C2", width=15, font=(14), anchor="w")

btn01.grid(row=0, column=1, sticky="e")
btn10.grid(row=1, column=0, sticky = "w")
btn12.grid(row=1, column=2, sticky="n")

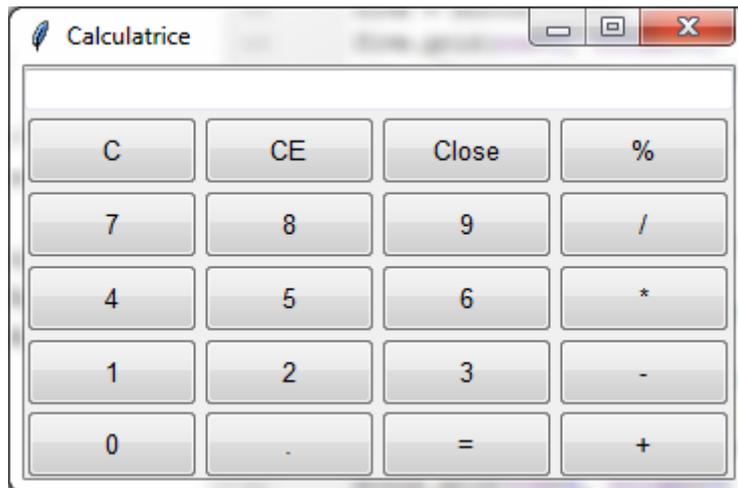
#Labels
lbr2c0 = Label(frgrid, text="Row2Column0", height=10, width=20, font
=(14),bg="blue", anchor="w")
lbr2c1 = Label(frgrid, text="Row2Column1", height=10, width=20, font
=(14),bg="red", anchor="sw")
lbr2c2 = Label(frgrid, text="Row2Column2", height=10, width=20, font
=(14),bg="green", anchor="ne")
lbr2c0.grid(row=2, column=0)
lbr2c1.grid(row=2, column=1)
lbr2c2.grid(row=2, column=2)
root.mainloop()

```

### Exemple2 : Calculatrice

Cet exemple présente comment réaliser l'interface d'une application qui effectue les tâches d'une calculatrice. Nous avons intégré des commentaires lorsque le besoin se présente.

**Gestionnaire de position pack : Exemple 6 : Calculatrice**



```

from tkinter import Tk, W, E
from tkinter.ttk import Frame, Button, Entry, Style

#====Fenêtre racine====
root = Tk()
root.title("Calculatrice")

#====Frame de base====
frame = Frame(root)

#====Personnalisation du style des boutons ====
Style().configure("TButton", padding=(0, 5, 0, 5), font='serif 10')

#====Configuration des colonnes 0, 1, 2 et 3 du frame avec pad=3====
frame.columnconfigure(0, pad=3)
frame.columnconfigure(1, pad=3)
frame.columnconfigure(2, pad=3)
frame.columnconfigure(3, pad=3)

#====Configuration des lignes 0, 1, 2, 3 et 4 du frame avec pad=3====
frame.rowconfigure(0, pad=3)
frame.rowconfigure(1, pad=3)
frame.rowconfigure(2, pad=3)
frame.rowconfigure(3, pad=3)
frame.rowconfigure(4, pad=3)

#====Définition de la zone de saisie : ligne 0 ====
#====et s'étend sur les colonnes 0, 1, 2 et 3 (4colonnes)====
entry = Entry(frame)
entry.grid(row=0, column=0, columnspan=4, sticky=W+E)

cls = Button(frame, text="C")
cls.grid(row=1, column=0)

back = Button(frame, text="CE")
back.grid(row=1, column=1)

close = Button(frame, text="Close")
close.grid(row=1, column=2)

quotient = Button(frame, text="%")
quotient.grid(row=1, column=3)

```

```

seven = Button(frame, text="7")
seven.grid(row=2, column=0)
eight = Button(frame, text="8")
eight.grid(row=2, column=1)
nine = Button(frame, text="9")
nine.grid(row=2, column=2)

div = Button(frame, text="/")
div.grid(row=2, column=3)

four = Button(frame, text="4")
four.grid(row=3, column=0)
five = Button(frame, text="5")
five.grid(row=3, column=1)
six = Button(frame, text="6")
six.grid(row=3, column=2)

mul = Button(frame, text="*")
mul.grid(row=3, column=3)

one = Button(frame, text="1")
one.grid(row=4, column=0)
two = Button(frame, text="2")
two.grid(row=4, column=1)
three = Button(frame, text="3")
three.grid(row=4, column=2)

minus = Button(frame, text="-")
minus.grid(row=4, column=3)

zero = Button(frame, text="0")
zero.grid(row=5, column=0)

dot = Button(frame, text=".")
dot.grid(row=5, column=1)

equal = Button(frame, text="=")
equal.grid(row=5, column=2)

pls = Button(frame, text="+")
pls.grid(row=5, column=3)

frame.pack()

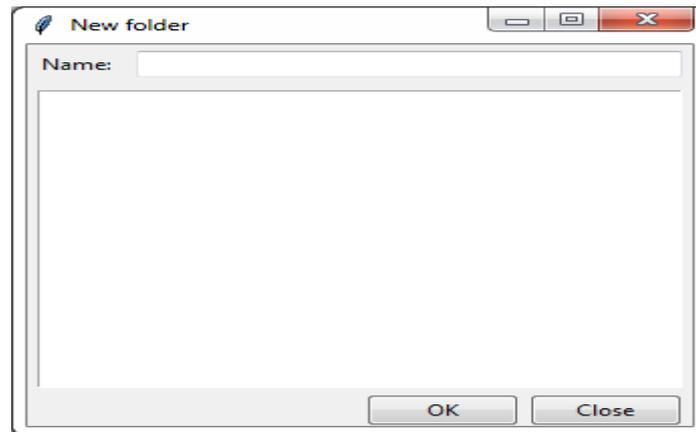
root.mainloop()

```

### Exemple3 : Nouveau dossier avec le gestionnaire grid

L'exemple suivant présente une disposition de nouveau dossier créée avec le gestionnaire de grille.

**Gestionnaire de position grid : Exemple 3 : Nouveau Dossier**



```
from tkinter import Tk, Text, BOTH, E, W, S, N
from tkinter.ttk import Frame, Button, Label, Entry

root = Tk()
root.geometry("330x300+300+300")
root.title("New folder")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

#Lorsqu'on redimensionne la fenêtre principale
#la colonne 1 est redimensionné mais pas le colonne0
frame.columnconfigure(1, weight=1)
#Lorsqu'on redimensionne la fenêtre principale
#la ligne 1 est redimensionné mais pas le ligne0
frame.rowconfigure(1, weight=1)
frame.rowconfigure(0, pad=10)

lbl = Label(frame, text="Name:")
lbl.grid(row=0, column=0, padx=5)

entry = Entry(frame)
entry.grid(row=0, column=1, colspan=4, padx=5, sticky=W+E)

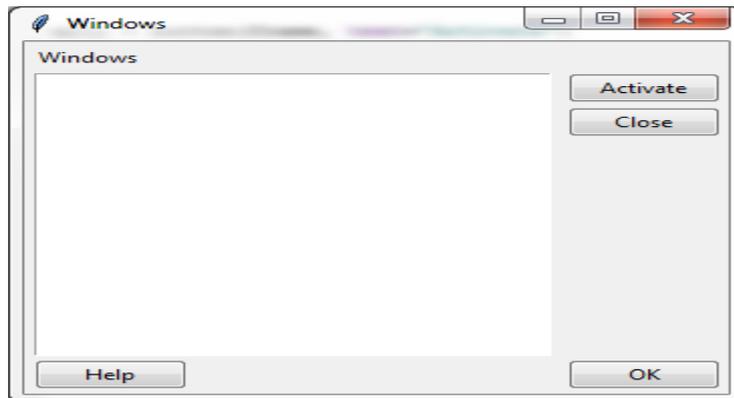
txt = Text(frame, width=20, height=10)
txt.grid(row=1, column=0, colspan=5, padx=5, pady=5, sticky=E+W+N+S)

okBtn = Button(frame, text="OK")
okBtn.grid(row=3, column=3, sticky=E)
closeBtn = Button(frame, text="Close")
closeBtn.grid(row=3, column=4, padx=5, sticky=E)

root.mainloop()
```

**Exemple4 : Fenêtre avec le gestionnaire grid**

**Gestionnaire de position grid : Exemple 4 : Fenêtre**



```
from tkinter import Tk, Text, BOTH, W, N, E, S
from tkinter.ttk import Frame, Button, Label, Style

root = Tk()
root.geometry("350x300+300+300")
root.title("Windows")

frame = Frame(root)
frame.pack(fill=BOTH, expand=True)

frame.columnconfigure(1, weight=1)
frame.columnconfigure(3, pad=7)
frame.rowconfigure(3, weight=1)
frame.rowconfigure(5, pad=7)

lbl = Label(frame, text="Windows")
lbl.grid(sticky=W, pady=4, padx=5)

area = Text(frame)
area.grid(row=1, column=0, columnspan=2, rowspan=4, padx=5,
sticky=E+W+S+N)

abtn = Button(frame, text="Activate")
abtn.grid(row=1, column=3)

cbtn = Button(frame, text="Close")
cbtn.grid(row=2, column=3, pady=4)

hbtn = Button(frame, text="Help")
hbtn.grid(row=5, column=0, padx=5)

obtn = Button(frame, text="OK")
obtn.grid(row=5, column=3)

root.mainloop()
```