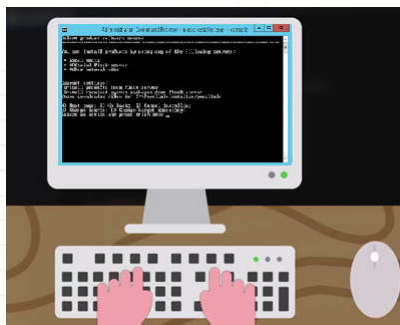


# FORMATION PYTHON

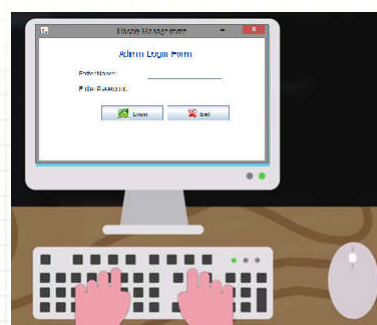
## INTERFACES GRAPHIQUES EN PYTHON

Pr. Mustapha HAIN  
infohain@gmail.com

### INTRODUCTION

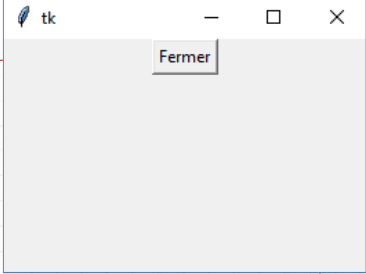


**Application Console**



**Application GUI**

### LES WIDGET TKINTER



```

from tkinter import *

window = Tk()

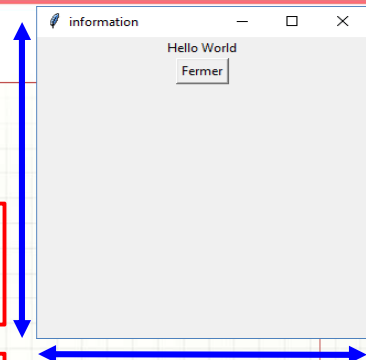
bouton=Button( window, text="Fermer")
bouton.pack()
window.mainloop()

```

Tkinter est un module de base intégré dans Python , l'un des avantages de Tkinter est sa portabilité sur les OS les plus utilisés par le grand public.

3

### LES WIDGET TKINTER



```

from tkinter import *

window = Tk()
window.title('information')
window.geometry("300x300")
window.minsize(100,100)


label = Label(window, text="Hello World")
label.pack()

bouton=Button(window,text="Fermer",command=window.quit)
bouton.pack()
window.mainloop()

```

4

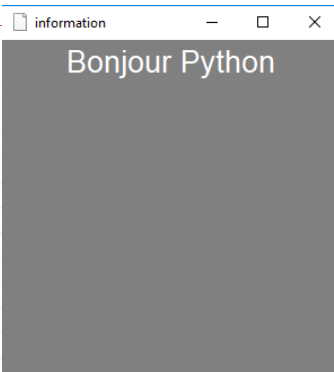
### LES WIDGET TKINTER



```

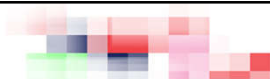
from tkinter import *
window=Tk()
window.title('information')
window.geometry("300x300")
window.minsize(100,100)
window.iconbitmap("icone.png")
window.config(background='grey')
label_title=Label(window,text='Bonjour
Python',font=("courrier",20),bg='grey',fg='white')
label_title.pack()
window.mainloop()

```



5

### LES WIDGET TKINTER

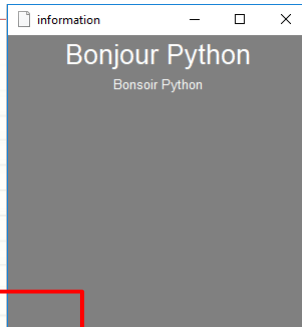


```

from tkinter import *
window=Tk()
window.title('information')
window.geometry("300x300")
window.minsize(100,100)
window.iconbitmap("icone.png")
window.config(background='grey')
label_title=Label(window,text='Bonjour
Python',font=("courrier",20),bg='grey',fg='white')
label_title.pack()
label_s=Label(window,text='Bonsoir
Python',font=("courrier",10),bg='grey',fg='white')
label_s.pack()

window.mainloop()

```

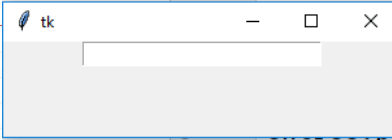


6

### LES WIDGET TKINTER

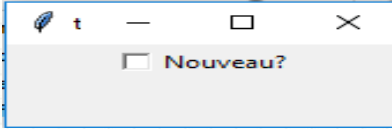
**Entrée / input**

```
entree = Entry(window, textvariable='string', width=30)
entree.pack()
```



**Case à cocher**

```
bouton = Checkbutton(window, text="Nouveau?")
bouton.pack()
```



7

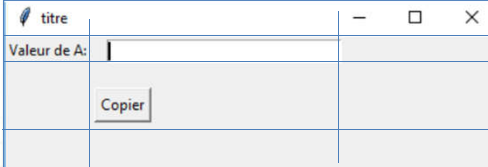
### LES WIDGET TKINTER

```
from tkinter import *
fenetre = Tk()

fenetre.title("titre")
fenetre.geometry("400x300")
name_LabelA = Label(fenetre, text='Valeur de A: ')
name_LabelA.grid(column=0, row=0, sticky='w')
v1 = StringVar()

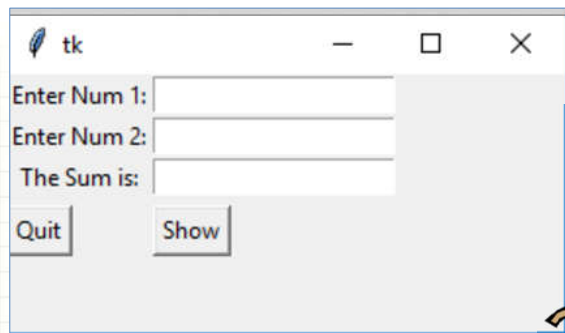
name_entryA = Entry(fenetre, textvariable=v1, width=31)
name_entryA.focus_set()
name_entryA.grid(column=1, row=0, sticky='sw', columnspan=1, padx=10)

send_button = Button(fenetre, text="Copier", pady=2)
send_button.grid(column=1, row=1, sticky='sw', pady=20)
fenetre.mainloop()
```



8

## LES WIDGET TKINTER



**Travaillons ensemble**

9

## LES WIDGET TKINTER

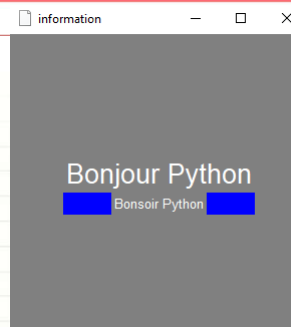
```
window.config(background='grey')
```

```
frame=Frame(window, bg='blue')
```

```
label_title=Label(frame,text='Bonjour_Python',  
font=("courrier",20),bg='grey',fg='white')  
label_title.pack()
```

```
label_s=Label(frame,text='Bonsoir_Python',font=("courrier",10),  
bg='grey',fg='white')  
label_s.pack()
```

```
frame.pack(expand=YES)  
window.mainloop()
```



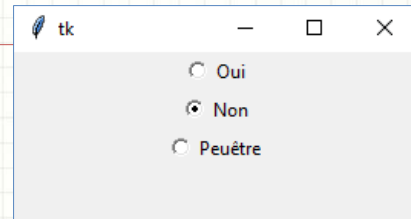
10

## LES WIDGET TKINTER

### Boutons radio

Les boutons radio sont des cases à cocher qui sont dans un groupe et dans ce groupe seul un élément peut être sélectionné.

```
value = StringVar()
bouton1 = Radiobutton(window, text="Oui", variable=value, value=1)
bouton2 = Radiobutton(window, text="Non", variable=value, value=2)
bouton3 = Radiobutton(window, text="Peu-être", variable=value, value=3)
bouton1.pack()
bouton2.pack()
bouton3.pack()
```



11

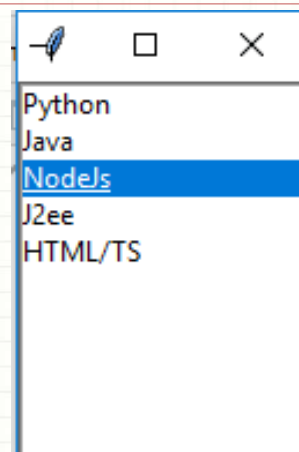
## LES WIDGET TKINTER

### Les listes

Les listes permettent de récupérer une valeur sélectionnée par l'utilisateur.

```
# liste
liste = Listbox(window)
liste.insert(1, "Python")
liste.insert(2, "Java")
liste.insert(3, "NodeJs")
liste.insert(4, "J2ee")
liste.insert(5, "HTML/TS")
```

```
liste.pack()
```



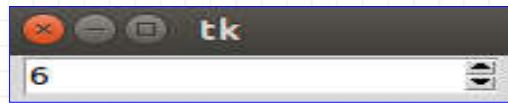
12

## LES WIDGET TKINTER

### Spinbox

La spinbox propose à l'utilisateur de choisir un nombre

```
s = Spinbox(window, from_=0, to=10)
s.pack()
```



13

## LES WIDGET TKINTER

### Les alertes

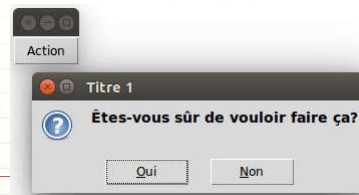
Pour pouvoir utiliser les alertes de votre os, vous pouvez importer le module tkMessageBox (Python 2).

```
from tkinter.messagebox import *
```

Exemple d'utilisation:

```
def callback():
    if askyesno('Titre 1', 'Êtes-vous sûr de vouloir faire ça?'):
        showwarning('Titre 2', 'Tant pis...')
    else:
        showinfo('Titre 3', 'Vous avez peur!')
        showerror('Titre 4', 'Aha')
```

```
Button(text='Action', command=callback).pack()
```



14

## LES WIDGET TKINTER

### Les alertes

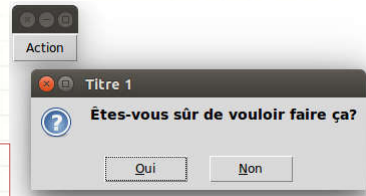
Pour pouvoir utiliser les alertes de votre os, vous pouvez importer le module tkMessageBox (Python 2).

```
from tkinter.messagebox import *
```

#### Exemple d'utilisation:

Voici les alertes possibles:

```
showinfo()
showwarning()
showerror()
askquestion()
askokcancel()
askyesno()
askretrycancel()
```



15

## LES WIDGET TKINTER

### Intégrer une image

Pour intégrer une image vous pouvez créer un canevas et l'ajouter à l'intérieur comme ceci:

```
from tkinter import *
window = Tk()
canvas = Canvas(window,width=350, height=200)
canvas.pack()
photo = PhotoImage(file="python.gif")
canvas.create_image(0, 0, anchor=NW, image=photo)
mainloop()
```



16



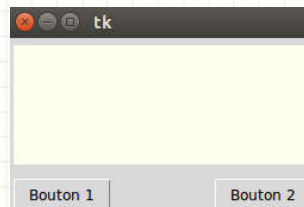
## LES WIDGET TKINTER

### Placer des widgets

Il est possible de placer les widgets à l'aide du paramètre side :

side=TOP : haut  
 side=LEFT : gauche  
 side=BOTTOM : bas  
 side=RIGHT : droite

```
Canvas(window,width=250,height=100, bg='ivory').pack( side=TOP,
  padx=5, pady=5)
Button(window, text ='Bouton 1').pack(side=LEFT, padx=5, pady=5)
Button(window, text='Bouton2').pack(side=RIGHT, padx=5, pady=5)
```



17

## LES WIDGET TKINTER

### Placer des widgets

Il est possible de placer les widgets à l'aide du paramètre side :

side=TOP : haut  
 side=LEFT : gauche  
 side=BOTTOM : bas  
 side=RIGHT : droite

#### Autre exemple:

```
Canvas(window, width=250, height=50, bg='ivory').pack(side=LEFT,
  padx=5, pady=5)
Button(window, text ='Bouton 1').pack(side=TOP, padx=5, pady=5)
Button(window, text ='Bouton 2').pack(side=BOTTOM, padx=5,
  pady=5)
```




18

### LES WIDGET TKINTER

#### Frames

Les frames (cadres) sont des conteneurs pour séparer des éléments.  
fenetre['bg']='white'



```

# frame 1
Frame1 = Frame(fenetre, borderwidth=2, relief=GROOVE)
Frame1.pack(side=LEFT, padx=30, pady=30)

# frame 2
Frame2 = Frame(fenetre, borderwidth=2, relief=GROOVE)
Frame2.pack(side=LEFT, padx=10, pady=10)

# frame 3 dans frame 2
Frame3 = Frame(Frame2, bg="white", borderwidth=2, relief=GROOVE)
Frame3.pack(side=RIGHT, padx=5, pady=5)

# Ajout de labels
Label(Frame1, text="Frame 1").pack(padx=10, pady=10)
Label(Frame2, text="Frame 2").pack(padx=10, pady=10)
Label(Frame3, text="Frame 3",bg="white").pack(padx=10, pady=10)

```

19

### LES WIDGET TKINTER

#### Barre de menu

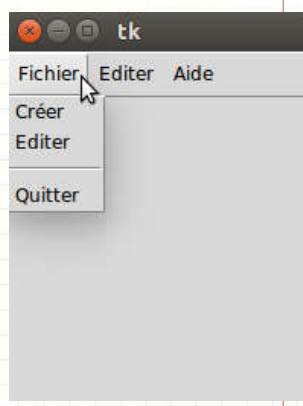
Il est possible de créer une barre de menu comme-cest:

```

def alert():
    showinfo("alerte", "Bravo!")
menubar = Menu(fenetre)
menu1 = Menu(menubar, tearoff=0)
menu1.add_command(label="Créer", command=alert)
menu1.add_command(label="Editer", command=alert)
menu1.add_separator()
menu1.add_command(label="Quitter", command=fenetre.quit)
menubar.add_cascade(label="Fichier", menu=menu1)
menu2 = Menu(menubar, tearoff=0)
menu2.add_command(label="Couper", command=alert)
menu2.add_command(label="Copier", command=alert)
menu2.add_command(label="Coller", command=alert)
menubar.add_cascade(label="Editer", menu=menu2)
menu3 = Menu(menubar, tearoff=0)
menu3.add_command(label="A propos", command=alert)
menubar.add_cascade(label="Aide", menu=menu3)

fenetre.config(menu=menubar)

```



20

## LES WIDGET TKINTER

### Récupérer une valeur

Récupérer la valeur d'un input

Pour récupérer la valeur d'un input, nous utilisons la méthode `get()` :

```
def recupere():
    showinfo("Alerte", entree.get())

value = StringVar()
value.set("Valeur")
entree = Entry(fenetre, textvariable=value, width=30)
entree.pack()

bouton = Button(fenetre, text="Valider", command=recupere)
bouton.pack()
```

21

## LES WIDGET TKINTER

### Les évènements

Pour chaque widget, vous pouvez binder (lier en français) un évènement, par exemple dire lorsque l'utilisateur appuie sur telle touche, faire cela. Voici un exemple qui récupère les touches appuyées par l'utilisateur:

```
def clavier(event):
    touche = event.keysym
    print(touche)

canvas = Canvas(fenetre, width=500, height=500)
canvas.focus_set()
canvas.bind("<Key>", clavier)
canvas.pack()
```

On remarque que l' évènement est encadré par des chevrons .

22

## LES WIDGET TKINTER

### Les évènements

D'autres évènements existent:

<Button-1>	: Click gauche
<Button-2>	: Click milieu
<Button-3>	: Click droit
<Double-Button-1>	: Double click droit
<Double-Button-2>	: Double click gauche
<KeyPress>	: Pression sur une touche
<KeyPress-a>	: Pression sur la touche A (minuscule)
<KeyPress-A>	: Pression sur la touche A (majuscule)
<Return>	: Pression sur la touche entrée
<Escape>	: Touche Echap
<Up>	: Pression sur la flèche directionnelle haut
<Down>	: Pression sur la flèche directionnelle bas
<ButtonRelease>	: Lorsque qu'on relache le click
<Motion>	: Mouvement de la souris
<B1-Motion>	: Mouvement de la souris avec click gauche
<Enter>	: Entrée du curseur dans un widget
<Leave>	: Sortie du curseur dans un widget
<Configure>	: Redimensionnement de la fenêtre
<Map> <Unmap>	: Ouverture et iconification de la fenêtre
<MouseWheel>	: Utilisation de la roulette

23

## Atelier 6: INTERFACES GRAPHIQUES EN PYTHON



**Travaillons ensemble**

