

16-17 - Portée des variables globales (Programmation)



I – Portée des variables globales

1.1 Locales ou globales ?

On nomme variables **locales** les variables déclarées à l'intérieur des fonctions.

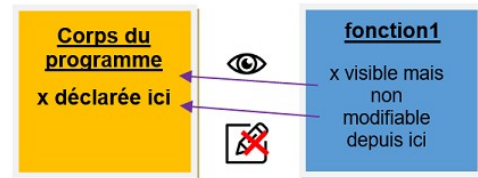
On nomme variables **globales** les variables déclarées en dehors des fonctions. Elles sont donc directement définies dans le programme principal.

Bonnes pratiques :

On évite autant que possible d'utiliser les variables globales directement depuis l'intérieur des fonctions. En effet, cela rend la correction des fonctions beaucoup plus complexe.

1.2 Portée des variables globales

- Une fonction **peut lire** une variable globale.
- Une fonction **ne peut pas** modifier une variable globale.



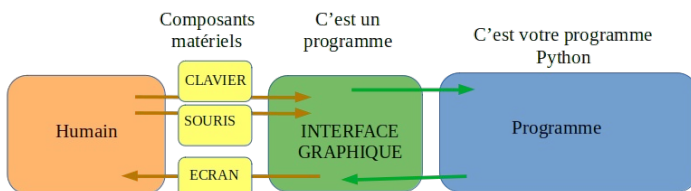
Utilise une affectation dans une fonction en utilisant un nom de variable globale ne fait que créer une variable locale portant le même nom.

II – Application aux interfaces graphiques

L'interface graphique est un programme permettant d'effectuer des interactions entre votre programme et le monde extérieur.

Les interactions se font dans les deux sens :

- L'utilisateur peut transmettre des informations au programme en passant par l'interface
- Le programme peut transmettre des informations à l'utilisateur en passant par l'interface.



Définitions

- Un **capteur** est un composant capable de transmettre une information issue du monde physique en signal électronique.
- Un **actionneur** est un composant capable de provoquer à partir d'un signal électronique une action réelle dans le monde physique

III – Mémo Tkinter

```
# 1 - Importations
import tkinter as tk

# 2 - Déclaration des constantes (globales)
COULEUR_0 = "#666666"
COULEUR_1 = "#448800"
COULEUR_2 = "#884400"
FOND = "black"

# 3 - Déclaration des fonctions
def modifier(event):
    '''Fonction événementielle :
    event se remplit automatiquement'''
    widget_clic = event.widget
    couleur_actuelle = widget_clic.cget('bg')
    if couleur_actuelle == COULEUR_0:
        widget_clic.configure(bg=COULEUR_1)
    elif couleur_actuelle == COULEUR_1:
        widget_clic.configure(bg=COULEUR_2)

def deplacer(event):
    '''Fonction événementielle :
    event se remplit automatiquement'''
    for i in range( len(tableau_widgets) ):
        widget_en_cours = tableau_widgets[i]
```

```
cx = widget_en_cours.winfo_x()
cy = widget_en_cours.winfo_y()
widget_en_cours.place(x=cx+10, y=cy)

# 4 - Programme
application = tk.Tk()
application.geometry("600x400")
application.title("Démonstration")
application.configure(bg=FOND)

wL1C1 = tk.Label(application, text="L1C1",
fg="white", bg=COULEUR_0, width=10, height=5)
wL1C2 = ...
...

wL1C1.place(x=25, y=25)
wL1C2.place(x=125, y=25)
...

tableau_widgets = [wL1C1, wL1C2, wL1C3, wL2C1,
wL2C2, wL2C3, wL3C1, wL3C2, wL3C3]

for i in range( len(tableau_widgets) ):
    tableau_widgets[i].bind('<Button-3>', deplacer)

application.mainloop()
```