

# Langage Python

Cours 4/5 : Interfaces graphiques

Hubert Godfroy

19 novembre 2015

## La dernière fois...

- ▶ Paradigmes objet
- ▶ Paradigme fonctionnel

## Aujourd'hui

- ▶ Interface graphique
  - ▶ Programmation événementielle
- ⇒ Utilise les deux paradigmes

# Plan

Programmation événementielle

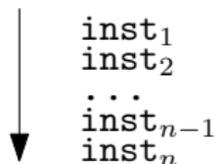
Application : interfaces graphiques

# Plan

Programmation événementielle

Application : interfaces graphiques

# Style habituel



```
inst1  
inst2  
⋮  
instn-1  
instn
```

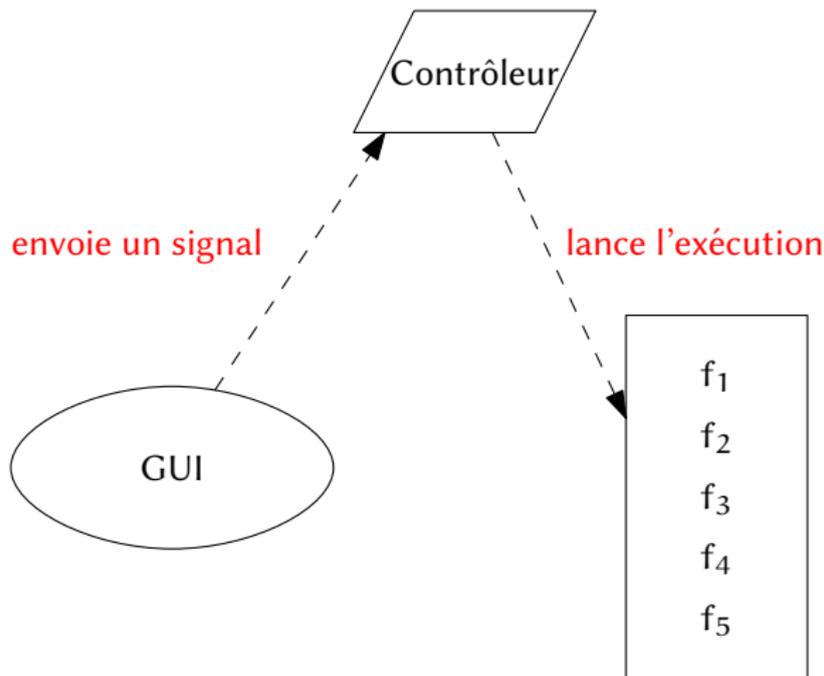
- ▶ Lecture linéaire
- ▶ De haut en bas
- ▶ On voit “tout ce qui se passe”.
- ▶ Rythmé par les appels de fonction/méthodes

## Programmation par événements

- ▶ Envoi de signaux
- ▶ Un contrôleur s'occupe d'envoyer les signaux au bon destinataire

# Programmation par événements

- ▶ Envoi de signaux
- ▶ Un contrôleur s'occupe d'envoyer les signaux au bon destinataire



# Plan

Programmation événementielle

Application : interfaces graphiques

# Modèle - Vue - Contrôleur

## Signaux

Ils sont déclenchés par des instances extérieures au programme (clique de l'utilisateur, branchement d'un périphérique, ...).

# Modèle - Vue - Contrôleur

## Signaux

Ils sont déclenchés par des instances extérieurs au programme (clique de l'utilisateur, branchement d'un périphérique, ...).

## Vue

- ▶ Compose l'interface utilisateur
- ▶ Différents éléments d'interaction (bouton, champs de texte, labels, ...)

# GUI en Python

- ▶ Environnement Tkinter
- ▶ Quatre éléments principaux : Button, Label, Entry et Canvas.

## Exemple

```
fen = Tk()
```

## Exemple

```
fen = Tk()

bouton = Button(fen, text="quitter", command=fen.
               destroy)
lab = Label(fen, text="Trololo")
champs = Entry(fen)
bouton2 = Button(fen, text="the big red button",
                command=update)
can = Canvas(fen, width =200, height =200, bg ='red
            ')
```

## Exemple

```
fen = Tk()

bouton = Button(fen, text="quitter", command=fen.
    destroy)
lab = Label(fen, text="Trololo")
champs = Entry(fen)
bouton2 = Button(fen, text="the big red button",
    command=update)
can = Canvas(fen, width =200, height =200, bg ='red
    ')

can.pack()
bouton.pack()
lab.pack()
champs.pack()
bouton2.pack()
```

## Exemple

```
fen = Tk()

bouton = Button(fen, text="quitter", command=fen.
    destroy)
lab = Label(fen, text="Trololo")
champs = Entry(fen)
bouton2 = Button(fen, text="the big red button",
    command=update)
can = Canvas(fen, width =200, height =200, bg ='red
    ')

can.pack()
bouton.pack()
lab.pack()
champs.pack()
bouton2.pack()

fen.mainloop()
```

## Exemple

```
fen = Tk()

bouton = Button(fen, text="quitter", command=fen.
    destroy)
lab = Label(fen, text="Trololo")
champs = Entry(fen)
bouton2 = Button(fen, text="the big red button",
    command=update)
can = Canvas(fen, width =200, height =200, bg ='red
    ')

can.pack()
bouton.pack()
lab.pack()
champs.pack()
bouton2.pack()

fen.mainloop()

def update():
    lab.config(text="lala")
```

Exécution !