

Base de données – manip2

Objectifs de la manip :

- Mise en œuvre d'un outil graphique (Tkinter) et de programmation événementielle.
- Mise en œuvre des fonctions
- Utiliser la gestion de fichier en python pour permettre d'encoder des données.
- Formatage du fichier de base et création de fichiers dérivés

Nouvelles notions :

- Mise en œuvre d'une interface graphique Tkinter (introduction)
- Illustration par l'exemple.

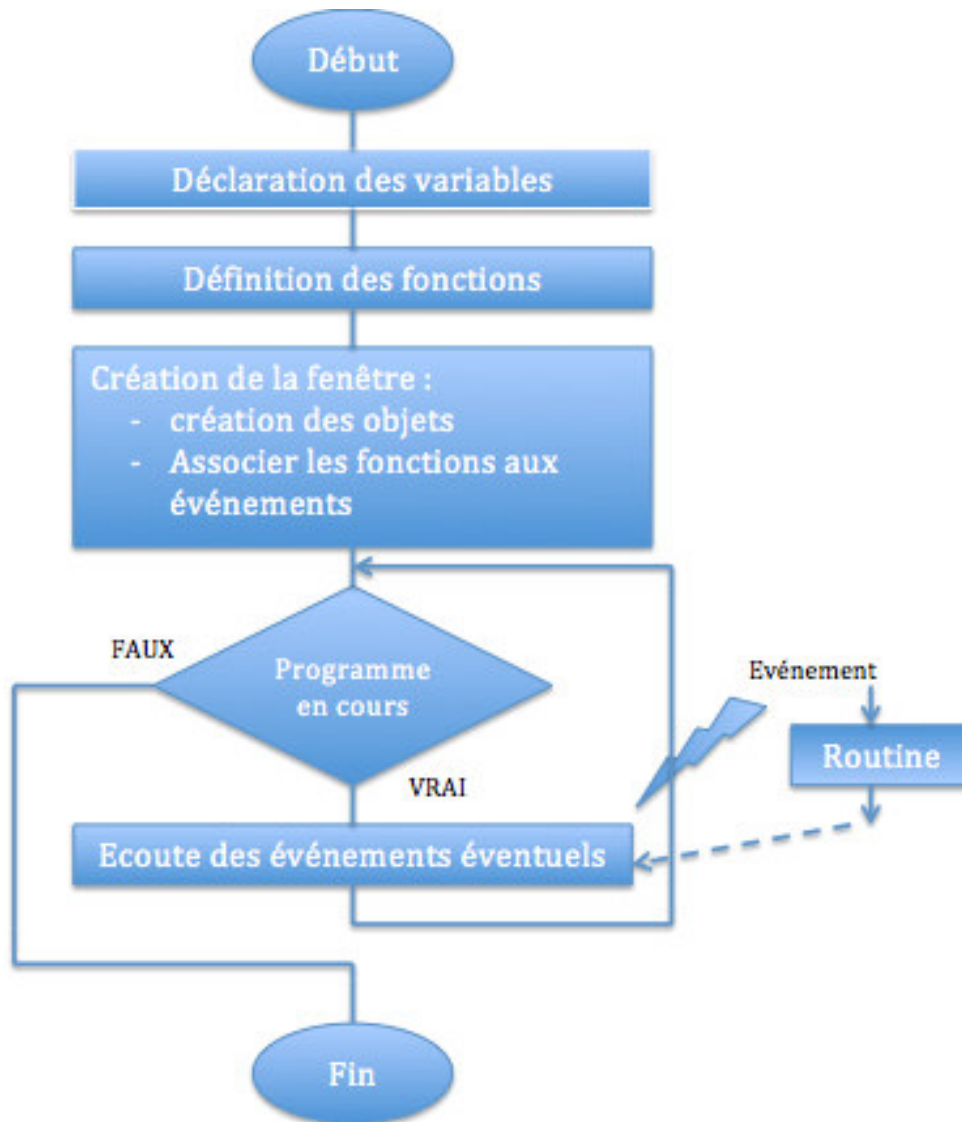
Exemple manip2.demo.py	Explications
<pre>from Tkinter import * nom = '' age = '' def bonjour(): global nom nom = case_nom.get() case_nom.delete(0, END) print "Bonjour", nom def affiche_age(): global age age = case_age.get() case_age.delete(0, END) print "vous avez", age, "ans" my_frame = Tk() my_frame.title("demo") my_frame.minsize(300, 200) Label(text="nom, svp:").pack() case_nom = Entry() case_nom.pack() Button(text = "msg bonjour", command = bonjour).pack() Label(text="\n").pack() Label(text="age svp:").pack() case_age = Entry() case_age.pack() Button(text = "quel age ?", command = affiche_age).pack() my_frame.mainloop()</pre>	<p>Import Tkinter</p> <p>Variables globales</p> <p>Fonction qui affiche bonjour suivi du contenu de la case_nom La case est ensuite effacée</p> <p>Fonction qui affiche un message avec contenu de la case_age La case est ensuite effacée</p> <p>On crée une fenêtre avec un titre et une taille minimale</p> <p>On joint un label à la fenêtre On crée une zone case_nom On crée un bouton et on lui associe la fonction bonjour()</p> <p>Saut de ligne</p> <p>On joint un label à la fenêtre On crée une zone case_age On crée un bouton et on lui associe la fonction affiche_age()</p> <p>On se place en attente d'événements</p>

Ci-dessous la partie création de la fenêtre et son aperçu :

```
my_frame = Tk()  
my_frame.title("demo")  
my_frame.minsize(300, 200)  
  
Label(text="nom, svp:").pack()  
case_nom = Entry()  
case_nom.pack()  
Button(text = "msg bonjour", command = bonjour).pack()  
  
Label(text="\n").pack()  
  
Label(text="age svp:").pack()  
case_age = Entry()  
case_age.pack()  
Button(text = "quel age ?", command = affiche_age).pack()
```



- Introduction à la programmation événementielle



Le programme passe la majorité de son temps à attendre.
Lorsqu'un événement extérieur survient (click de souris, click sur un bouton, enfoncement d'une touche,...) le programme exécute la routine qui lui est associée.

Quelques rappels :

```
import os
os.chdir("c:\directory")
---->directory est le chemin ou doit être créé le fichier
fichier=open("nom_fichier","argument")
---->nom_fichier est le nom du fichier que l'on veut utiliser
---->l'argument peut être w=write, r=read, a=append
fichier.write("texte") ou fichier.write(variable)
---->variable doit être un string
fichier.close()
---->ferme le fichier
```

Base de données, énoncé de la manip 2 :

Créez sur votre ordinateur un nouveau dossier avec votre nom suivi de database2. Ce dossier contiendra tout ce qui concerne cette manip (tous les points).

- 1.) Lancez python 2.7, créez un nouveau fichier nommé tkinter1.py que vous sauvez dans le dossier que vous venez de créer.

Complétez votre programme pour que son exécution fasse apparaître :

Une fenêtre avec une zone pour l'ID, une zone pour le prénom, une zone pour le nom et une zone pour l'âge. Il doit également figurer un bouton avec l'inscription « Affiche_dernier » qui se charge d'afficher sur la console l'ID, le prénom, le nom et l'âge du dernier record formaté comme indiqué ici :

Exemple avec 1 record	Caractéristiques des données dans le fichier
#123# Jo fugtniol 65	Un record commence par l'ID mis entre # et # Un record se termine par un saut de ligne

Après l'affichage les cases avec les données introduites doivent être vides.

- 2.) Toujours dans votre dossier de cette manip, créez une copie de votre programme tkinter1.py, sauvez le sous encodageTk1.py et complétez-le d'un bouton supplémentaire avec l'inscription « Encode ».

Suite à un click sur ce bouton, il faut qu'il apparaisse dans votre dossier courant un dossier « data » qui contient le fichier « data_base.txt » avec les données qui viennent d'être encodées.

Le fichier doit être formaté de la même manière, doit pouvoir être complété à chaque nouvel encodage et montre un saut de ligne après chaque encodage.

- 3.) Partez de votre programme précédent que vous sauvez sous encodageTk2.py.

Le programme contient les mêmes cases mais il y a maintenant 4 boutons :

- Un bouton avec l'inscription « Affiche »

Le contenu des cases est conservé, ce bouton a maintenant pour but d'afficher le record courant sur la console. Rien n'est ajouté à notre fichier.

- Un bouton avec l'inscription « efface »

Ici le bouton a pour unique but d'effacer le contenu des cases, rien ne doit s'afficher et rien ne doit s'ajouter à notre fichier.

- Un bouton avec l'inscription « encode »

Ici le record sera ajouté à notre fichier « data_base.txt », rien ne doit s'afficher sur la console mais les cases doivent être effacées afin de permettre l'introduction d'un nouveau record.

- Un bouton avec l'inscription « lire fichier »

Ce bouton se charge simplement d'afficher sur la console le contenu du fichier « data_base.txt »

- 4.) Imaginez une extension du point 3 que vous nommez affiche_id.py et qui contient un bouton supplémentaire avec l'inscription « affiche id ». Un click sur ce bouton doit afficher l'ensemble des ID du fichier, si vous savez le faire, ajoutez l'option « afficher les ID triés ».