

OS Lab 8

Managing files and folders in Python
Graphical interface using Tkinter

The os module

- This module is a library dedicated to files and records management needs. For all the examples it will be necessary to import this module.

import os

- Each file or folder is associated with a kind of skill that can easily find the error.
- There are two type of path: absolute path from the file system of your root and the relative path from the currently playing file.

Manipulate the paths 1

- The most useful methods:

- `abspath(path)` → Returns an absolute path
- `basename(P)` → Returns the last element of a path
- `commonprefix(list)` → Returns the common path of the longest path list
- `dirname(P)` → Returns the parent folder of the element
- `exists(path)` → test if a path exists
- `getATime(filename)` → Get date of last access to the file [`os.stat()`]
- `getctime(filename)` → Returns the date of the last change metadata of the file
- `getMTime(filename)` → Returns the date of the last modification of the file
- `getsize(filename)` → Returns tailkle a file (in bytes)

Manipulate the paths 2

- `isabs(S)` → test if a path is absolute
- `isdir(S)` → test if the path is a folder
- `isfile(path)` → test if the path is a regular file
- `islink(path)` → test if the path is a symbolic link
- `ismount(path)` → test if the path is a mount point
- `join(path, S)` → Adds an item to the path as a parameter
- `normcase(S)` → Normalize the case of a path
- `Normpath(path)` → Normalize the way, eliminates double slashes, etc.
- `realpath(filename)` → Return the canonical path of the specified filename (removes symbolic links)
- `samefile(F1, f2)` → Test if two paths refer to the same actual file
- `sameopenfile(F1, f2)` → test if two objects of open files refer to the same file
- `split(p)` → Splits a path. returns a tuple

Example

```
import os
path = "C:\\Users\\Iuliana"
print("Nom du répertoire = " + os.path.dirname(path))
print("Nom de base = " + os.path.basename(path))
print("Chemin = " + os.path.join(path, "workspace"))
print(os.path.split(path))
```

Nom du répertoire = C:\Users

Nom de base = Iuliana

Chemin = C:\Users\Iuliana\workspace

('C:\\Users', 'Iuliana')

List the files in a folder

- You can pick from a list all the items in a folder using the method `listdir`:

```
os.listdir("C:\\Users\\Iuliana")
```

```
['.anaconda',  
 '.android',  
 '.AndroidStudio1.5',  
 '.appletviewer',  
 '.astah',  
 '.bash_history',  
 '.conda'...]
```

List items recursively

```
import os

folder_path = "C:\\Users\\Iuliana"

for path, dirs, files in os.walk(folder_path):
    for filename in files:
        print(filename)
```

```
.appletviewer
.bash_history
.condarc
.emulator_console_auth_token
.gitconfig
.neo4j_shell_history
.packettracer
.recently-used.xbel
.scala_history
```

Research items by reason

- * → any sequence of characters
- ? → any character
- [] → any character listed in the brackets
- It is necessary to import the module glob.
 - **import glob**
- here is the methods:
 - glob.glob(Pattern) → List folders and the corresponding files on the grounds
 - glob.iglob(Pattern) Same as → glob but returns a iterator

```
import glob
glob.glob("C:\\Users\\Iuliana\\*.log")
```

```
['C:\\Users\\Iuliana\\3DimViewer.log']
```


Manipulate the items

- `os.makedirs(path)` → Create recursively all files of a path if they do not exist
- `os.mkdir(path)` → Create the last record of a path. If a folder does not exist an error is returned
- `os.remove(path)` → Remove the file / folder specified
- `os.rename(old, new)` → Rename the file / folder specified

Tkinter Graphical Interface

- **Tkinter** is an integrated base module in **Python**.
Normally you do not have to do anything to use it.
- The code for your first **hello world with a Entry**:

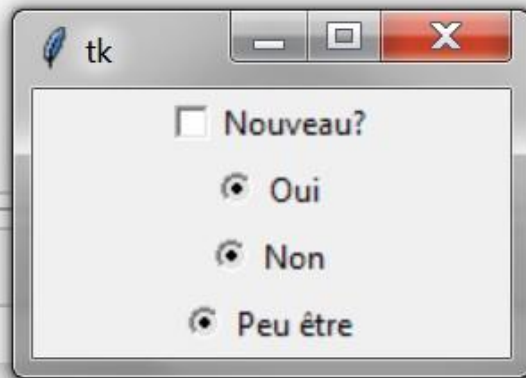
```
from tkinter import *  
import string  
fenetre = Tk()  
  
label = Label(fenetre, text="Hello World")  
label.pack()  
# entrée  
value = StringVar()  
value.set("texte par défaut")  
entree = Entry(fenetre, textvariable=string, width=30)  
entree.pack()  
fenetre.mainloop()
```



Check and radio buttons

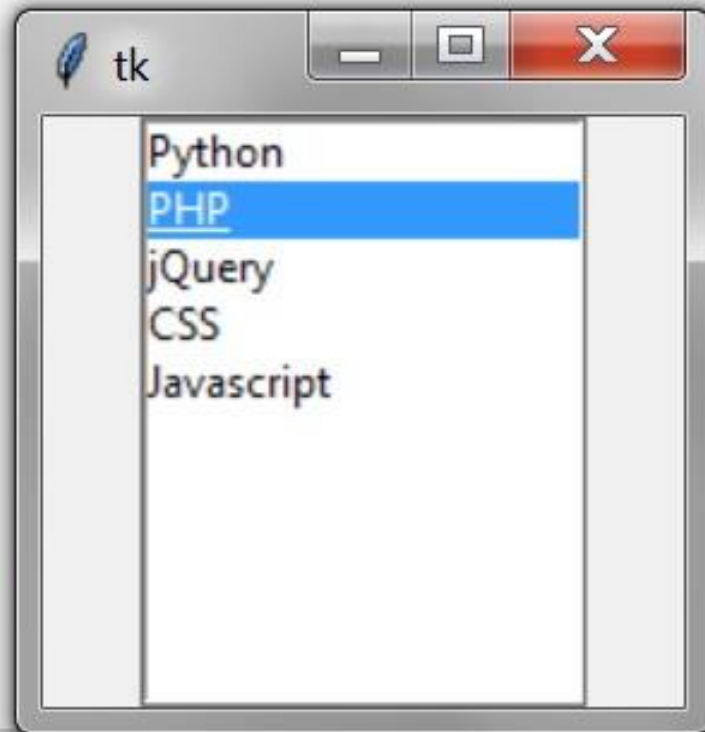
```
from tkinter import *
import string
fenetre = Tk()
# checkbutton
bouton = Checkbutton(fenetre, text="Nouveau?")
bouton.pack()
# radiobutton
value = StringVar()
bouton1 = Radiobutton(fenetre, text="Oui",
                      variable=value, value=1)
bouton2 = Radiobutton(fenetre, text="Non",
                      variable=value, value=2)
bouton3 = Radiobutton(fenetre, text="Peu être",
                      variable=value, value=3)

bouton1.pack()
bouton2.pack()
bouton3.pack()
fenetre.mainloop()
```



Lists

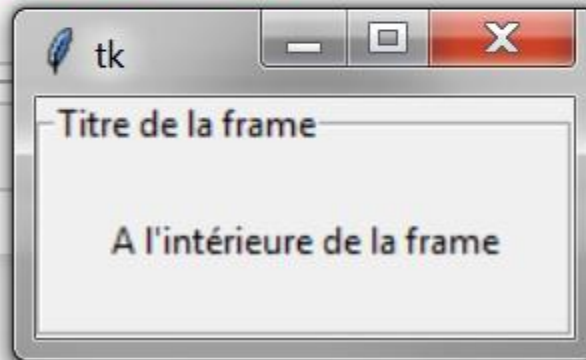
```
from tkinter import *
import string
fenetre = Tk()
# Liste
liste = Listbox(fenetre)
liste.insert(1, "Python")
liste.insert(2, "PHP")
liste.insert(3, "jQuery")
liste.insert(4, "CSS")
liste.insert(5, "Javascript")
liste.pack()
fenetre.mainloop()
```



LabelFrame

- The LabelFrame is a frame with a label.

```
from tkinter import *  
import string  
fenetre = Tk()  
l = LabelFrame(fenetre, text="Titre de la frame",  
               padx=20, pady=20)  
l.pack(fill="both", expand="yes")  
  
Label(l, text="A l'intérieure de la frame").pack()  
fenetre.mainloop()
```

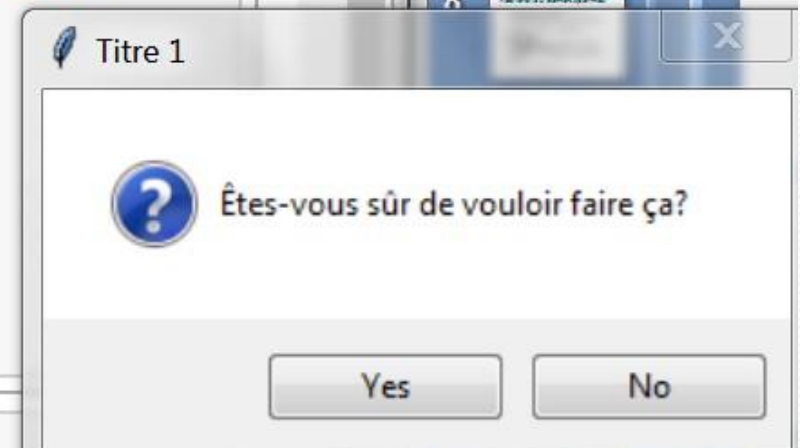
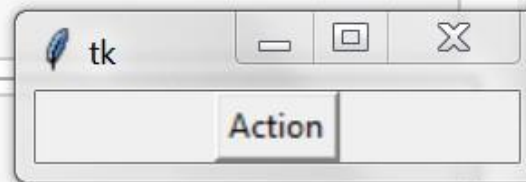


Alerts

```
fenetre.mainloop()
```

```
from tkinter.messagebox import *  
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()  
fenetre.mainloop()
```



Close off menu

```
from tkinter import *
fenetre = Tk()
def alert():
    showinfo("alerte", "Bravo!")

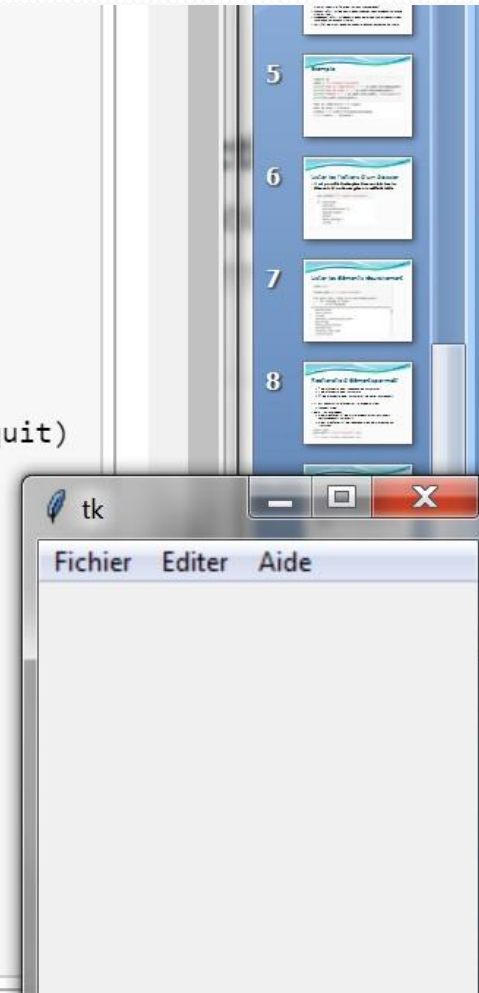
menubar = Menu(fenetre)

menu1 = Menu(menubar, tearoff=0)
menu1.add_command(label="Cr  er", 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)
fenetre.mainloop()
```



Get a text file and display it

```
from tkinter.filedialog import *
fenetre = Tk()
filename = askopenfilename(title="Ouvrir votre
    document",filetypes=[('txt files','*.txt'),('all files','*.*)'])
fichier = open(filename, "r")
content = fichier.read()
fichier.close()

Label(fenetre, text=content).pack(padx=10, pady=10)
fenetre.mainloop()
```

Example

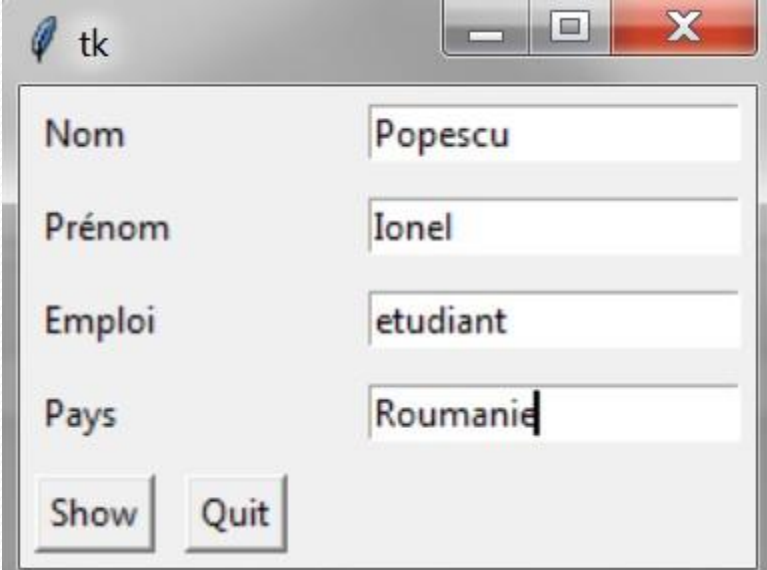
```
from tkinter import *
fields = 'Nom', 'Prénom', 'Emploi', 'Pays'

def fetch(entries):
    for entry in entries:
        field = entry[0]
        text = entry[1].get()
        print('%s: "%s"' % (field, text))

def makeform(root, fields):
    entries = []
    for field in fields:
        row = Frame(root)
        lab = Label(row, width=15, text=field, anchor='w')
        ent = Entry(row)
        row.pack(side=TOP, fill=X, padx=5, pady=5)
        lab.pack(side=LEFT)
        ent.pack(side=RIGHT, expand=YES, fill=X)
        entries.append((field, ent))
    return entries

if __name__ == '__main__':
    root = Tk()
    ents = makeform(root, fields)
    root.bind('<Return>', (lambda event, e=ents: fetch(e)))
    b1 = Button(root, text='Show',
                command=(lambda e=ents: fetch(e)))
    b1.pack(side=LEFT, padx=5, pady=5)
    b2 = Button(root, text='Quit', command=root.quit)
    b2.pack(side=LEFT, padx=5, pady=5)
    root.mainloop()
```

Nom: "Popescu"
Prénom: "Ionel"
Emploi: "etudiant"
Pays: "Roumanie"



Nom	Popescu
Prénom	Ionel
Emploi	etudiant
Pays	Roumanie

Show Quit

Exercises

- Create a window where you can give the path of a folder and that it will show the files that are inside.
- Test if a given path is a directory.
- Create a window where the first button will calculate $n!$ and the second button calculates n to the power m .