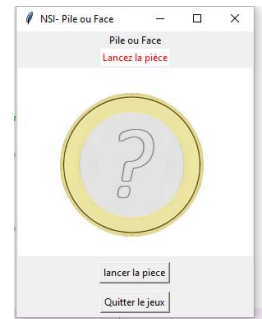


Pile ou Face

Téléchargez les fichiers images disponibles sur
<http://ressource.elec.free.fr>,
rubrique NSI, | TP Pile ou Face |



On se propose de coder un jeu de Pile ou Face

1/ En mode console

Puisqu'il s'agit d'un jeu de hasard, on utilisera le module *random* de Python. Pour rappel, pour affecter à une variable un nombre entier au hasard, le code s'écrit **`a = random.randint(0,2)`** (entier au hasard entre 0 et 2) après avoir pris soin d'importer la bibliothèque par un **`import random`** en début de code.

Ecrire un programme qui :

- ➔ Importe la bibliothèque *random*
- ➔ Affecte à une variable **a** un nombre entier entre 0 et 1
- ➔ Affiche dans la console « PILE » si a vaut 1 et « FACE » si a vaut 0

Testez votre programme. Vous voyez, ce n'était pas très compliqué...

-> transformer votre programme en une fonction **`jouer()`**

2/ En mode graphique

On va profiter de ce programme très simple pour construire une interface graphique. Pour cela, on utilisera la bibliothèque **`tkinter`** (c'est une bibliothèque graphique parmi beaucoup d'autre).

Pour l'importer, **`from tkinter import *`** (qui signifie qu'on va importer tous les modules de la biblio *tkinter*)

Puis, on va créer la fenêtre « racine », qui contiendra tous les éléments. Appelons la *root* (mais on peut l'appeler comme on veut) avec ce code : **`root = Tk()`**

En mode graphique, tout le programme se retrouvera dans une boucle. On « rafraichi » la fenêtre tout le temps. En fin de programme, on mettra cette ligne de code : **`root.mainloop()`**

Normalement, votre programme ressemble à celui ci-contre:

Lancez-le pour vérifier que jusque là tout va bien (une fenêtre vide doit s'ouvrir).

On peut maintenant ajouter les éléments graphique qu'on veut entre la ligne **`root = Tk()`** et **`root.mainloop()`**

a/ Un titre à ma fenêtre :

On utilise la méthode *title* ainsi : **`root.title('NSI- Pile ou Face')`** (rq : si vous avez appelé votre fenêtre racine autrement (racine, par exemple), vous devez écrire **`racine.title('NSI- Pile ou Face')`**)

b/ Un texte dans ma fenêtre :

On appellera cela un Label. Créons un titre avec la fonction *Label(fenêtre parent, text= "... ")* :

`titre = Label(root, text = "Pile ou Face", bg= "white", fg="red")` # pour définir le titre (en rouge sur fond blanc)
`titre.grid()` # pour afficher le titre ; il existe plusieurs façon de placer un élément. Nous n'en parlerons pas aujourd'hui.

- ➔ Essayez cela, et ajouter un second label *texte1* qui donnera, par exemple, votre nom.

```
import random
from tkinter import *

root = Tk()

def jouer():
    a= random.randint(0,1)
    if a==1:
        print ("FACE")
    else :
        print ("PILE")

root.mainloop()
```

c/ Un bouton « jouer » dans ma fenêtre

On peut ajouter toute sorte de choses dans la fenêtre : des boutons, des menus, des cases à cocher, des images etc...
On appelle cela de widgets. Nous allons ajouter un bouton « jouer »

```
Bouton1=Button(root, text='lancer la pièce', command=jouer) # on crée bouton1 dans root. Cliquer dessus lancera la fonction jouer()  
Bouton1.grid(padx=5, pady=5) # on affiche le bouton
```

d/ Changer le texte d'un label

Normalement, à chaque clique sur le bouton jouer, vous avez en console le résultat PILE ou FACE qui s'affiche. Mais c'est nul : on a fait une interface graphique, autant afficher le résultat dans la fenêtre !!

Je vous propose de modifier le label avec votre nom en PILE ou FACE selon le résultat.

Pour ce, nous allons utiliser la méthode *configure* de *Label*. Cette méthode s'utilise ainsi :

```
texte1.configure(text="FACE") (si votre label s'appelle texte1)
```

Il s'agit donc de remplacer notre vieux `print(« face »)` par `texte1.configure(text="FACE")` et notre vieux `print(« PILE »)` par .. je vous laisse deviner. Faites-le et testez votre programme.

e/ Une image pour faire choli

Il est cool notre petit programme, mais , bon, on devrait pouvoir faire plus beau, en mettant par exemple une image de la pièce...

pour ce, nous avons besoin de mettre ce qu'on appelle un canvas ; sorte de fenêtre dans la fenêtre, qui a un peu le même rôle que nos balises <div> en html. On mettra ensuite l'image dans le canvas. Créons le canvas (en dessous de notre label *texte1* par exemple) :

```
dessin = Canvas(root, bg='white', height=230, width=285) # on crée dessin qui est un canvas de la fenêtre root.  
dessin.grid() # on affiche le canvas dessin
```

Pour ajouter l'image, on crée d'abord une image tkinter avec la fonction **PhotoImage** à partir d'une image png :

```
imgPiece = PhotoImage(file='pile_ou_face.png') # l'image png doit se trouver dans le même répertoire que le programme
```

et on utilise la méthode *create_image* ainsi :

```
piece = dessin.create_image(140,120, image=imgPiece) # 140 et 120 sont les coordonnées du centre de l'image.
```

f/ Changer l'image en fonction du résultat

C'est bien beau mais je voudrais que l'image change en fonction du résultat ! On ne peut pas utiliser la méthode *configure* comme tout à l'heure pour le label, il va falloir utiliser la méthode *itemconfigure*. On utilisera deux images png supplémentaires : *face1.png* et *pile1.png*.

➔ En dessous de `imgPiece = PhotoImage(file='pile_ou_face.png')`, créez les image tkinter *imgPieceFace* et *imgPiecePile*

La méthode *itemconfigure* s'utilise ainsi (pour mettre l'image de la piece coté Face) :

```
dessin.itemconfigure(piece,image=imgPieceFace )
```

➔ Placez ce code dans notre fonction *jouer()*

➔ Ecrire le code à ajouter pour changer l'image lorsque le résultat est PILE

Voilà, notre petit programme commence à ressembler à quelque chose ☺

Vous pouvez ajouter un bouton pour quitter le jeu :

```
bouton2=Button(root, text='Quitter le jeux', command=root.destroy)  
bouton2.grid(padx=5, pady=5)
```

Créer un exécutable de notre programme :

Vous êtes peut-être frustré.e.s d'avoir créé un beau programme Python et de ne pas l'avoir en exécutable sous Windows.

Voilà comment transformer votre fichier .py en fichier .exe :

Nous allons utiliser un programme qui va faire ce travail, et qui s'exécutera en mode console : **PyInstaller**

1/ **ouvrir le prompt anaconda (cherchez « anaconda prompt » dans le menu demarrer)**

1/ **voir si PyInstaller est déjà installé.** Pour ce, taper simplement PyInstaller dans la console. Si la console vous répond des trucs, c'est qu'il est installé. Si elle vous répond « *PyInstaller n'est pas reconnu...* », c'est qu'il n'y est pas et qu'il faut l'installer.

Pour Installer PyInstaller :

Dans le prompt anaconda (anaconda prompt), saisissez : **python -m pip install PyInstaller**
Après téléchargement et installation, c'est bon.

Utiliser PyInstaller : **PyInstaller --noconsole nom_du_fichier.py** depuis le répertoire où se trouve le fichier .py à rendre exécutable.

Deux répertoires se créent alors : dist et build. Dans **dist** se trouve un répertoire du nom de votre fichier py dans lequel se trouve l'exécutable. Rq : il vous faut copier/coller nos trois image png dans le répertoire **dist**

Et voilà ! Votre fichier python exécutable windows est prêt !

Pour en savoir plus sur TKINTER :

Un excellent cours sur l'utilisation de tkinter : <http://s15847115.domainepardefaut.fr/python/tkinter/>

Documentation officielle de tkinter : <https://docs.python.org/fr/3/library/tk.html>