

## COMMENT CRÉER UN FICHIER PÝTHON EXÉCUTABLE SOUS WINDOWS ? CÀD UN FICHIER AYANT L' EXTENSION .EXE

### COMMENT CRÉER UN EXÉCUTABLE WINDOWS À PARTIR D'UN SCRIPT PÝTHON

Pour créer et pouvoir exécuter un fichier avec Python, il faut avoir installé l'interpréteur Python sur son ordinateur (Mac, Windows ou Linux)

Mais le script que vous aurez créé et qui fonctionnera parfaitement sur votre ordinateur ne fonctionnera pas sur un autre ordinateur qui n'est pas équipé de cet interpréteur.

Si vous voulez faire bénéficier un ami d'un jeu, par exemple, que vous avez créé avec Python, vous devrez lui en offrir une version dite « compilée ».

Il existe un grand nombre de façons de compiler un script ; une façon particulièrement simple de créer une version compilée qui fonctionnera sur un PC Windows est d'avoir recours à la procédure suivante :

Après avoir installé pip (Rappel ci-dessous) :

1/ Ouvrir un terminal en tapant cmd dans la zone de recherche

2/ Se déplacer dans le répertoire où se trouve le script à compiler, portant l'extension .py (à l'aide de la commande cd)

3/ Puis :

- Taper : pip install pywin32
- Taper : pip install pyinstaller (packages à installer)

4/ Ensuite :

- Taper : pyinstaller --onefile nomduscriptàcompiler.py

Deux sous-répertoires nommés build et dist seront créés, et c'est dans le sous-répertoire dist que se trouvera le fichier compilé portant l'extension .exe que vous pourrez distribuer à toute personne équipée d'un simple PC Windows.

Il suffira de cliquer dessus pour l'exécuter avec succès.

## Rappel : **INSTALLATION DE PIP**

il faut préalablement utiliser le logiciel approprié, qui permet d'importer des modules, soit pip.

La stratégie pour installer ce dernier est la suivante :

- Ouvrir une fenêtre de commande (Alt-^T sous Linux), ou écrire « cmd » dans la zone de recherche (en bas de l'écran) sous Windows

- Taper : python -pip install – upgrade pip

Ceci installera pip s'il n'est pas déjà installé.

## **AUTRE FAÇON DE CRÉER UN EXÉCUTABLE : UTILISER AUTO-PY-TO-EXE POUR GÉNÉRER UN FICHIER QUE L'ON POURRA DISTRIBUER À VOLONTÉ**

Une fois pip installé, il suffira d'ouvrir un terminal (cmd ou Ctrl-alt-T) , puis de taper :

**pip install auto-py-to-exe**

Application :

Soit le code suivant : **main.py** :

```
import tkinter
import mgame
```

```
def open() :
    mgame.run()
```

```
app = tkinter.Tk()
app.title("My jeu")
app.geometry("720x480")
lb = tkinter.Label(app, text = "Cliquer sur le bouton").pack()
bt = tkinter.Button(app, text= "Clique", command= open).pack()

app.mainloop()
```

Avec **mgame.py** :

```
import pygame
```

```
def run() :
```

```
    pygame.init()
```

```
    pygame.display.set_caption("Ma belle fenêtre")
```

```
    screen = pygame.display.set_mode((400,400))
```

```
    boucle = True
```

```
    while boucle :
```

```
        for event in pygame.event.get() :
```

```
            if event.type == pygame.QUIT :
```

```
                boucle = False
```

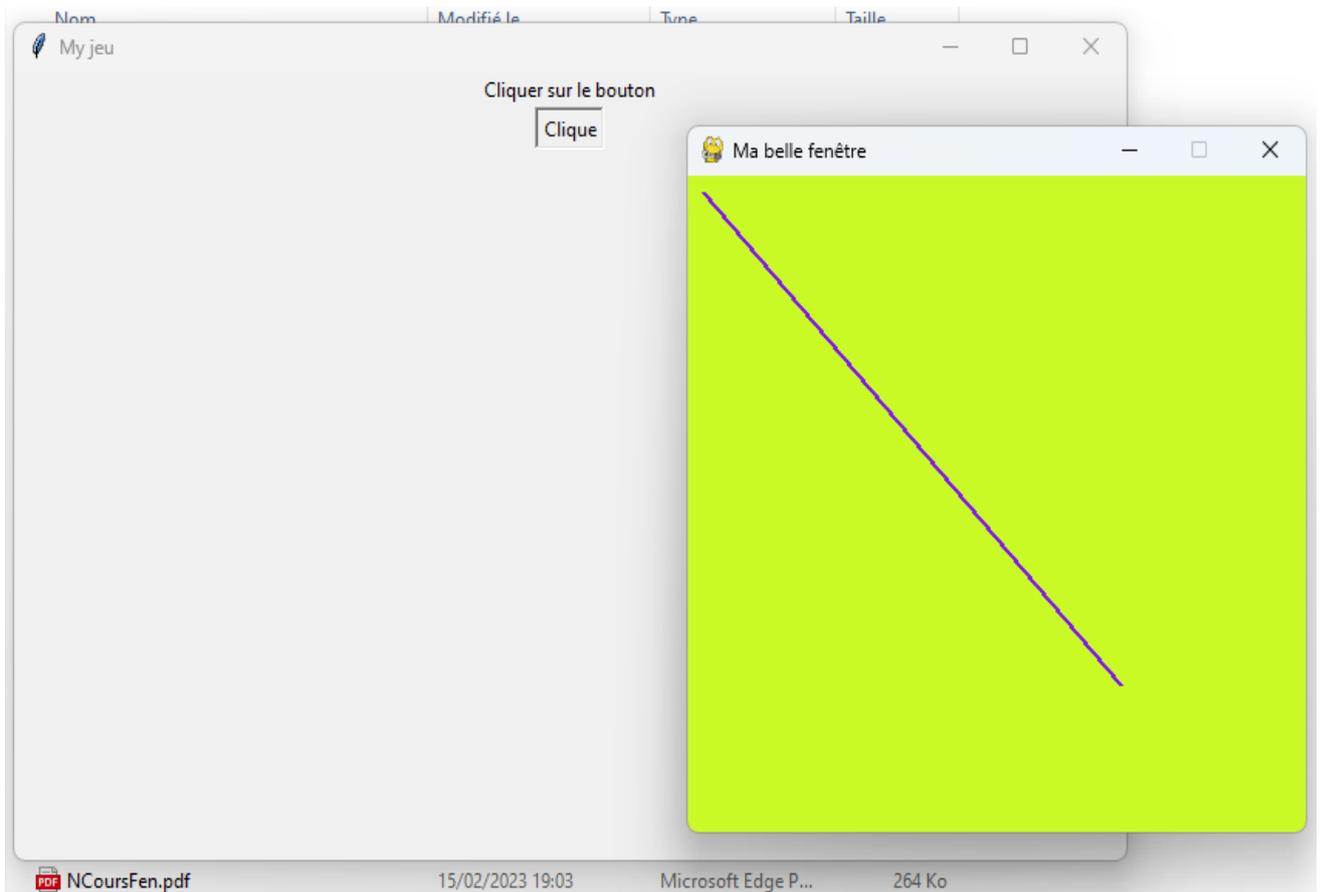
```
        screen.fill((202,250,38))
```

```
        pygame.draw.line(screen,(128,25,240),[10,10],[280,310],3)
```

```
        pygame.display.flip()
```

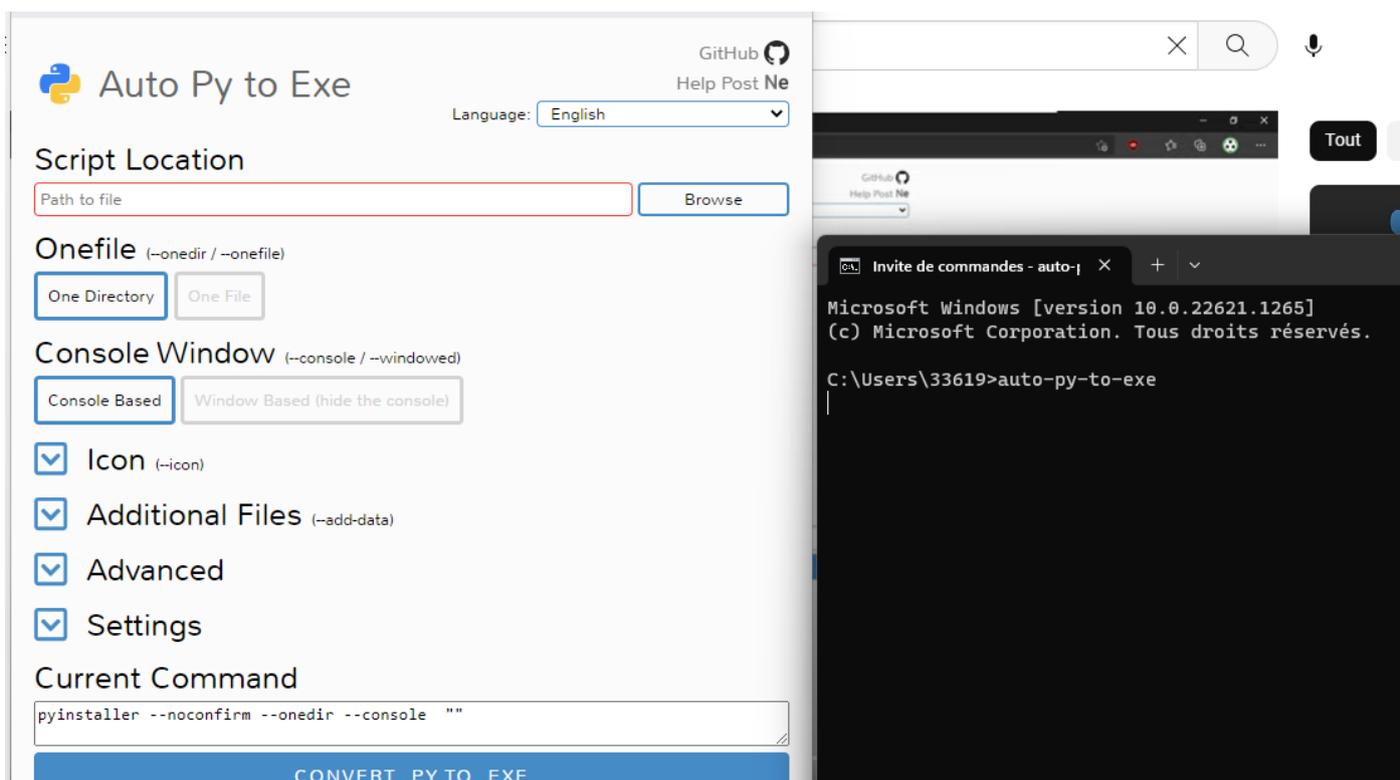
```
    pygame.quit()
```

Voici le résultat de l'exécution de main.app, après avoir cliqué sur le bouton  
Clique :



On ouvre alors un terminal (cmd ou Ctrl-alt-T) , puis on tape :  
**auto-py-to-exe :**

**On obtient ceci :**



On paramètre :

E:/Python/package/main.py dans Script Location

et pour Output Directory dans Settings :

E:/Python/package

On peut éventuellement fournir un icône (fichier .ico)

On clique : **CONVERT PY TO EXE**

Des fichiers de dépendance sont générés et l'on obtient un dossier main (du nom du fichier fourni ) dans :

E:/Python/package

Dans ce dossier main se trouve le fichier main.exe à cliquer, parmi les autres dépendances (fichiers annexes nécessaires)

Mais pour distribuer ou publier il faudra fournir tout le répertoire main créé ; en fait il suffira d'en créer une archive .zip