

---

## WinPython Portable

---

Page de présentation : <http://winpython.github.io/>

Page de téléchargement : <https://github.com/winpython/winpython/releases>

Les versions 0dot :

[Winpython32-3.8.6.0dot.exe \(version 32 bits\)](#)

et

[Winpython64-3.8.6.0dot.exe \(version 64 bits\)](#)

sont des fichiers autoextractibles de WinPython Zéro.

Ces fichiers sont légers. Ils font entre 20 MB et 30 MB. Ils contiennent python sans aucun package.

Leurs avantage est qu'ils vous permettent d'avoir python sans aucun package. A l'aide de Pyzo, vous installez uniquement les packages dont vous avez besoin.

### Mode d'emploi

Supposons que votre clé usb se trouve en D :

Créer un dossier D :\PythonPortable.

Copier le fichier Winpython64-3.8.6.0dot.exe dans ce dossier.

Double cliquez sur ce fichier afin de l'extraire dans ce dossier.

Vous obtenez un dossier D :\PythonPortable\Wpy64-3860 faisant entre 70 MB et 80 MB.

---

## Pyzo Portable

---

Page de présentation : <https://pyzo.org/>

Page de téléchargement : <https://github.com/pyzo/pyzo/releases>

Les fichiers :

[pyzo-4.11.0-win32.zip \(version 32 bits\)](#)

et

[pyzo-4.11.0-win64.zip \(version 64 bits\)](#)

sont des versions portables de Pyzo.

### Mode d'emploi

Supposons que votre clé usb se trouve en D :

Copier le fichier pyzo-4.11.0-win32.zip ou le fichier pyzo-4.11.0-win64.zip dans le dossier

D:\PythonPortable.

Faites un clic droit sur ce fichier puis extraire tout dans le dossier D:\PythonPortable.

Vous obtenez un dossier D:\PythonPortable\pyzo-4.11.0 faisant environ 100 MB.

### **Configuration de Pyzo**

Dans le dossier D:\PythonPortable\pyzo-4.11.0, lancer pyzo.exe

Faites Settings puis Select language French

puis faites File Restart Pyzo

puis faites Outils et sélectionner uniquement l'outil Environnement

puis faites Shell Configuration des shells :

nom : remplacer Shell config 1 par WinPython

exe : écrire : ..\WPy32-3860\python-3.8.6.amd64\python.exe (version 32 bits)

ou

..\WPy64-3860\python-3.8.6.amd64\python.exe (version 64 bits)

ensuite cliquez sur done et faites Fichier Redémarrer Pyzo

Faites ctrl + plusieurs fois pour augmenter la taille de la police jusqu'à ce que cela vous convienne. Pyzo sauvegarde automatiquement votre réglage.

### **Premiers pas avec Python et Pyzo**

Dans la fenêtre de droite, après >>>, taper 0.1 + 0.2 et valider.

Vous n'obtenez pas le bon résultat.

### **Utilisation de Xcas avec Python et Pyzo**

Consulter la rubrique [Logiciels - Xcas Portable](#) du site [gbmaths.free.fr](#) pour avoir des informations sur Xcas.

Lancer Xcas Portable, taper 0.1 + 0.2 et valider.

Vous obtenez le bon résultat.

Il est possible d'utiliser toutes les commandes de Xcas dans python.

### **Installation du module giacpy**

Placez vous dans la fenêtre de droite et taper : **pip install --upgrade giacpy** afin d'installer le module giacpy.

Placez vous dans la fenêtre de gauche et taper : **from giacpy import \*** afin d'importer toutes les commandes du module giacpy puis faites ctr + E afin d'exécuter le contenu de cette fenêtre pour vérifier que le module giacpy a bien été installé.

### **Utilisation de la commande giac du module giacpy**

Remplacer from giacpy import \* par from giacpy import giac.

Dans la fenêtre de droite, après >>>, taper giac(0.1) + giac(0.2) et valider.

Vous obtenez le bon résultat.

Un seul giac suffit. L'astuce que j'ai trouvée est de taper votre calcul normalement puis d'ajouter giac(0) à la fin du calcul.

$0.1 + 0.2 + \text{giac}(0)$  donne le bon résultat.

On peut faire du calcul exact avec des fractions.

Exemple :  $\text{giac}(1/3) + \text{giac}(2/7)$  donne  $13/21$ .

On peut aussi taper  $\text{giac}("1/3") + \text{giac}("2/7")$ .

On peut aussi taper  $\text{giac}(1)/3 + \text{giac}(2)/7$ .

On peut aussi taper  $1/\text{giac}(3) + 2/\text{giac}(7)$ .

Dans le calcul ci-dessus, tous les giac sont nécessaires.

### Un programme avec des commandes du module giacpy

#### Algorithme

```
1 from giacpy import factorial, exp, approx
2
3 n = int(input("n ="))
4 e = 1
5
6 for i in range(1,n+1):
7     e = e + (1/factorial(i))
8
9 print("e =",e)
10 print("e =",approx(e))
11 print("exp(1) =",approx(exp(1)))
```

Pour  $n = 14$ , on obtient le même résultat pour  $e$  et  $\exp(1)$ .

#### Résultat dans la console

```
>>>
n = 14
e = 2.71828182846
exp(1) = 2.71828182846
>>>
```