

Python en .exe

installer pyinstaller

Prerequis

Windows

PyInstaller fonctionne sous Windows 10 et versions ultérieures. Il permet de créer des applications graphiques fenêtrées (applications ne nécessitant pas de fenêtre de commande).

macOS

PyInstaller fonctionne sur macOS 10.15 (Catalina) ou version ultérieure. Il permet de créer des applications graphiques fenêtrées (sans fenêtre de terminal). PyInstaller crée des applications compatibles avec la version macOS sur laquelle il est exécuté et les versions suivantes. Il peut créer des binaires x86_64.NET arm64 ou des binaires hybrides universal2 sur les machines macOS des deux architectures. Consultez la prise en charge multi-architecture de macOS pour plus de détails.

GNU/Linux

PyInstaller nécessite que l'application terminal détecte les bibliothèques partagées requises par chaque programme ou bibliothèque partagée. Il se trouve généralement dans le package de distribution glibc ou libc-bin.

L'application terminale doit également objdump extraire les informations des fichiers objets et objcopy ajouter les données au chargeur de démarrage. Ces informations se trouvent généralement dans le package de distribution binutils.

Comment installer PyInstaller

PyInstaller est disponible sous forme de package Python standard. Les archives des sources des versions publiées sont disponibles sur PyPI, mais il est plus simple d'installer la dernière version avec pip :

```
pip install pyinstaller
```

Utilisation de PyInstaller

[Utilisation de pyinstaller EN](#)

La syntaxe de la pyinstallercommande est :

```
pyinstaller[ options ] script [ script ...] | fichier de spécifications
```

Dans le cas le plus simple, définissez le répertoire courant sur l'emplacement de votre programme myscript.py et exécutez :

```
pyinstaller myscript.py
```

PyInstaller analyse myscript.py et :

- Écrit myscript.spec dans le même dossier que le script.
- Crée un dossier build dans le même dossier que le script s'il n'existe pas.
- Écrit certains fichiers journaux et fichiers de travail dans le build dossier.
- Crée un dossier dist dans le même dossier que le script s'il n'existe pas.
- Écrit le myscript dossier exécutable dans le dist dossier.

Exemple avec un fichier calculatrice.py

Dans la console windows 11 :



Aller dans le dossier d'installation de pyinstaller (nom_utilisateur = votre nom utilisateur windows 11)

```
cd
```

```
C:\users\nom_utilisateur\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\Scripts
```



Copier le programme calculatrice.py (qui fonctionne) dans ce repertoire et ensuite taper dans le terminal la commande ci-dessous

```
.\pyinstaller.exe --onefile -w calculatrice.py
```

Explications

.\pyinstaller.exe : on lance le programme pyinstaller

--onefile : on ne creer qu'un seul fichier exe

-w : on enleve l'affichage du terminal

calculatrice.py : Le programme à transformer en .exe



Dans le repertoire\python313\Scripts\dist crée, vous trouverez le programme calculatrice.exe

le programme calculatrice.exe

calculatrice.zip

From:

<https://chanterie37.fr/fablab37110/> - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:

<https://chanterie37.fr/fablab37110/doku.php?id=start:python:pointexe&rev=1744116894>

Last update: **2025/04/08 14:54**

