Guide to Converting Python Code to a Standalone PC Application (.exe)

The most popular and effective tool for converting Python scripts into standalone executables (like `.exe` files for Windows, or similar applications for macOS and Linux) is **PyInstaller**.

This guide will walk you through the process of installing PyInstaller and using it to package your Python code.

Important Note on Web Applications

Converting a Python script into a **PC application** (executable) is done with tools like PyInstaller.

Converting a Python script into a **Web Application** is a completely different process that involves:

1. Using a web framework (like Flask or Django).
2. Writing HTML, CSS, and JavaScript for the frontend.
3. Deploying the application to a web server (like AWS, Heroku, or a private server).

PyInstaller **cannot** turn a Python script into a web application. This guide focuses on creating a standalone PC application.

Step 1: Install PyInstaller

You must have Python installed on your system. PyInstaller is installed using Python's package manager, `pip`.

Open your terminal or command prompt and run the following command:

Bash

```
pip install pyinstaller
```

Step 2: Prepare Your Python Script

For this example, let's assume you have a simple Python script named `my_app.py`.

Python

```python
# my_app.py
import sys
print("Hello from the standalone application!")
input("Press Enter to exit...")
```

Step 3: Run PyInstaller

Navigate to the directory where your Python script (`my_app.py`) is located in your terminal or command prompt. Then, run the PyInstaller command.

Basic Command (Creates a folder with all files)

This command creates a folder named `dist` containing the executable and all necessary dependencies.

Bash

```
pyinstaller my_app.py
```

Recommended Command (Creates a single, standalone file)

The `--onefile` flag is the most common and useful option, as it bundles everything into a single executable file, making it easier to distribute.

Bash

```
pyinstaller --onefile my_app.py
```

Command for Graphical User Interface (GUI) Apps

If your application uses a GUI (like Tkinter, PyQt, or Kivy) and you do not want the black console window to appear, use the `--windowed` (or `--noconsole`) flag:

Bash

```
pyinstaller --onefile --windowed my_app.py
```

Step 4: Find Your Executable

After PyInstaller finishes, it will create two new folders in your current directory: `build` and `dist` .

Your final executable file will be located in the `dist` folder.

- **Windows:** `dist/my_app.exe`
- **macOS:** `dist/my_app` (a file with no extension)
- **Linux:** `dist/my_app` (a file with no extension)

You can now copy this single file and run it on any computer with the same operating system, even if that computer does not have Python installed.

Summary of Useful Flags

Flag	Description
`--onefile`	Creates a single executable file instead of a folder. **(Highly Recommended)**
`--windowed` or `--noconsole`	Hides the console window for GUI applications.
`--name "MyAppName"`	Assigns a specific name to the executable and the `dist` folder.
`--icon "icon.ico"`	Adds a custom icon to the executable (Windows requires `.ico` format).
`--add-data "source;dest"`	Used to include external files (like images, configuration files, etc.) that your script needs.