**asos para crear un ejecutable multiplataforma con PyInstaller:**

1. **Instalar PyInstaller:**
2. Bash
3. pip install pyinstaller
4. Use code [with caution.](file:///C:/faq#coding)
5. **Ir al directorio de tu proyecto:**
6. Bash
7. cd mi\_proyecto
8. Use code [with caution.](file:///C:/faq#coding)
9. **Generar el ejecutable:**
10. Bash
11. pyinstaller --onefile --windowed mi\_script.py
12. Use code [with caution.](file:///C:/faq#coding)

* **--onefile:** Crea un solo archivo ejecutable.
* **--windowed:** Oculta la ventana de consola en Windows (si es una aplicación gráfica).
* **mi\_script.py:** El nombre de tu archivo Python principal.

**Explicación de las opciones:**

* **--onefile:** Como su nombre indica, crea un solo archivo ejecutable. Esto es ideal para la portabilidad, ya que solo necesitarás copiar un archivo.
* **--windowed:** Si tu aplicación tiene una interfaz gráfica, esta opción oculta la ventana de consola que suele aparecer al ejecutar scripts de Python.
* **--onedir:** Crea una carpeta con todos los archivos necesarios para ejecutar la aplicación. Esta opción puede ser útil si necesitas más control sobre la estructura del ejecutable.

**Ejecutando el ejecutable:**

Una vez generado el ejecutable, lo encontrarás en la carpeta dist dentro de tu proyecto. Simplemente copia esta carpeta o el archivo ejecutable a tu dispositivo USB y ejecútalo desde allí.

**Consideraciones importantes:**

* **Dependencias:** Asegúrate de que todas las dependencias de tu proyecto estén instaladas correctamente antes de generar el ejecutable. PyInstaller intentará incluirlas automáticamente, pero puede haber casos en los que necesites configurar manualmente algunas dependencias.
* **Sistemas operativos:** PyInstaller puede generar ejecutables para Windows, macOS y Linux. Sin embargo, es recomendable probar el ejecutable en diferentes sistemas para asegurarte de que funciona correctamente.
* **Tamaño del ejecutable:** La opción --onefile crea un solo archivo ejecutable, pero puede ser más grande que una carpeta con todos los archivos. Si el tamaño es un problema, puedes considerar usar la opción --onedir.
* **Personalización:** PyInstaller ofrece muchas otras opciones para personalizar el ejecutable, como cambiar el icono, incluir archivos de datos adicionales, etc. Consulta la documentación oficial para obtener más información.

**Ejemplo de personalización:**

Bash

pyinstaller --onefile --windowed --icon=mi\_icono.ico --add-data "data;data" mi\_script.py

Use code [with caution.](file:///C:/faq#coding)

* --icon=mi\_icono.ico: Establece un icono para el ejecutable.
* --add-data "data;data": Incluye una carpeta llamada "data" dentro del directorio del ejecutable.

**Conclusión:**

PyInstaller es una herramienta muy útil para crear ejecutables Python multiplataforma y portables. Con esta guía, podrás empaquetar tu aplicación y llevarla contigo a cualquier lugar