**Building the Application**

**Using Nuitka**

1. **Navigate to script**
2. **Run Bash:**

nuitka --standalone --enable-plugin=pyqt6 --mingw64 --windows-console-mode=disable --nofollow-import-to=tkinter --include-data-dir=resources=resources --windows-icon-from-ico=Turtle\_Icon.ico TURTLE\_AppV3.6.3.py

* **Purpose:**  
  Compiles TURTLE\_AppV3.6.3.py into optimized C code, then builds a standalone distribution (without needing Python installed).
* **Details:**
  + --standalone: Creates a fully self-contained folder with all dependencies.
  + --enable-plugin=pyqt6: Ensures PyQt6 support is properly handled.
  + --mingw64: Uses the 64-bit MinGW compiler for C code compilation.
  + --windows-console-mode=disable: Disables the console window (for a GUI app).
  + --nofollow-import-to=tkinter: Excludes tkinter to avoid unnecessary size or warnings.
  + --include-data-dir=resources=resources: Bundles the resources directory into the output.
  + --windows-icon-from-ico=Turtle\_Icon.ico: Sets a custom icon for the executable.

**Using PyInstaller**

1. **Navigate to script**
2. **Run Bash:**

pyinstaller --onefile --windowed --add-data="resources;resources" TURTLE\_AppV3.6.2.py

* **Purpose:**  
  Packages TURTLE\_AppV3.6.2.py directly into a single .exe file.
* **Details:**
  + --onefile: Creates a single .exe file instead of a folder.
  + --windowed: Suppresses the console window (for a GUI app).
  + --add-data="resources;resources": Includes the resources directory (where icons are located)

**Summary:**

* **Nuitka** gives you a faster, more secure, but slightly larger and more complex output (folder-based distribution).
* **PyInstaller** gives you a single .exe, quicker to make, but less efficient and less secure.