# Руководство по сборке .exe и .app для приложения "Калькулятор измерений"

Это руководство описывает, как собрать исполняемые файлы `.exe` для Windows и `.app` для macOS из Python-скрипта вашего приложения с помощью утилиты \*\*PyInstaller\*\*.

\*\*Важно:\*\*

\* Для сборки `.exe` для Windows, вам \*\*необходимо выполнять команды на Windows машине\*\*.

\* Для сборки `.app` для macOS, вам \*\*необходимо выполнять команды на macOS машине\*\*.

Cross-компиляция (сборка `.exe` на macOS или наоборот) с PyInstaller не поддерживается напрямую и часто бывает проблематичной.

## 1. Подготовка (Общие шаги для обеих платформ)

1. \*\*Активируйте ваше виртуальное окружение:\*\*

Убедитесь, что вы находитесь в том же виртуальном окружении, где разрабатывали и тестировали приложение (`.venv` в вашем случае).

```bash

# Для macOS/Linux

source .venv/bin/activate

# Для Windows CMD

# .venv\Scripts\activate.bat

# Для Windows PowerShell

# .venv\Scripts\Activate.ps1

```

2. \*\*Установите PyInstaller:\*\*

Если PyInstaller еще не установлен в вашем виртуальном окружении, установите его:

```bash

uv pip install pyinstaller

# или pip install pyinstaller

```

3. \*\*Перейдите в директорию проекта:\*\*

Откройте терминал (командную строку) и перейдите в папку, где находится ваш файл `app.py` и другие файлы проекта.

```bash

cd /путь/к/вашему/проекту/CalculatorShSK

```

## 2. Сборка `.app` для macOS

Этот шаг вы можете выполнить на вашей macOS машине.

1. \*\*Запустите PyInstaller:\*\*

Выполните следующую команду в терминале (в директории вашего проекта и с активным виртуальным окружением):

```bash

pyinstaller --onefile --windowed --name "Калькулятор измерений" app.py

```

\* `--onefile`: Создает один исполняемый файл внутри `.app` пакета.

\* `--windowed`: Указывает, что это приложение с графическим интерфейсом и не требует консольного окна. Это важно для корректного создания `.app` пакета.

\* `--name "Калькулятор измерений"`: Задает имя вашему приложению.

\* `app.py`: Ваш главный файл скрипта.

2. \*\*Найдите результат:\*\*

После завершения работы PyInstaller (это может занять некоторое время), в директории вашего проекта появятся две новые папки: `build` и `dist`.

Ваш готовый пакет приложения будет находиться в папке `dist` и называться `Калькулятор измерений.app`.

3. \*\*Иконка (опционально):\*\*

Если у вас есть файл иконки в формате `.icns` (стандартный для macOS), вы можете добавить его командой:

```bash

pyinstaller --onefile --windowed --name "Калькулятор измерений" --icon="путь/к/вашей/иконке.icns" app.py

```

## 3. Сборка `.exe` для Windows 11 (для работы без интернета)

Этот шаг \*\*необходимо выполнить на машине с Windows 11\*\*.

1. \*\*Подготовка на Windows машине:\*\*

\* Скопируйте весь ваш проект (включая `app.py`, `requirements.txt` и т.д.) на Windows машину.

\* Установите Python 3.11 (той же версии, что и при разработке).

\* Создайте и активируйте виртуальное окружение.

\* Установите все зависимости из `requirements.txt` (`pip install -r requirements.txt`).

\* Установите `pyinstaller` (`pip install pyinstaller`).

2. \*\*Запустите PyInstaller на Windows:\*\*

Откройте командную строку (CMD или PowerShell) на Windows, перейдите в директорию проекта и с активным виртуальным окружением выполните команду:

```bash

pyinstaller --onefile --windowed --name "Калькулятор измерений" app.py

```

\* Опции те же: `--onefile` для одного файла, `--windowed` для GUI-приложения без консоли.

3. \*\*Найдите результат:\*\*

Как и на macOS, готовый `.exe` файл будет находиться в папке `dist` внутри директории проекта и называться `Калькулятор измерений.exe`.

4. \*\*Работа без интернета:\*\*

PyInstaller по умолчанию включает все необходимые зависимости (включая сам интерпретатор Python) в сборку. Если ваше приложение само по себе не требует доступа к интернету для своей работы (например, не загружает данные из сети), то собранный `.exe` файл должен работать на Windows 11 без доступа к интернету.

## 4. Важные замечания и возможные проблемы

\* \*\*Зависимости и данные:\*\*

\* PyInstaller обычно хорошо справляется с большинством стандартных библиотек.

\* `ttkbootstrap` темы обычно хорошо упаковываются.

\* `matplotlib` может иногда требовать дополнительной конфигурации для включения своих файлов данных (`mpl-data`), если графики отображаются некорректно или без стилей. У PyInstaller есть встроенные "хуки" (hooks) для `matplotlib`, которые часто решают эту проблему автоматически. Если возникнут проблемы, вам может потребоваться использовать опцию `--add-data` PyInstaller или создать/изменить hook-файл.

\* Пример с `--add-data` (путь к `mpl-data` может отличаться в зависимости от вашей системы и виртуального окружения):

```bash

# Пример для Windows

# pyinstaller --onefile --windowed --add-data "путь\_к\_venv\Lib\site-packages\matplotlib\mpl-data;matplotlib/mpl-data" app.py

# Пример для macOS/Linux

# pyinstaller --onefile --windowed --add-data "путь\_к\_venv/lib/pythonX.Y/site-packages/matplotlib/mpl-data:matplotlib/mpl-data" app.py

```

Замените `путь\_к\_venv` и `pythonX.Y` на актуальные значения. Обратите внимание на разделитель путей (`:` для macOS/Linux, `;` для Windows в опции `--add-data`).

\* \*\*Размер файла:\*\* Исполняемые файлы, созданные с опцией `--onefile`, могут быть довольно большими.

\* \*\*Время сборки:\*\* Первая сборка может занять значительное время.

\* \*\*Антивирусы:\*\* Иногда антивирусные программы могут ложно срабатывать на файлы, созданные PyInstaller.

\* \*\*Отладка:\*\* Если при запуске собранного приложения возникают ошибки:

\* Попробуйте сначала собрать приложение \*\*без\*\* опции `--windowed` (и без `--onefile` для более быстрой отладки). Это создаст консольное приложение, и вы сможете увидеть сообщения об ошибках.

\* PyInstaller создает папку `build`, в которой содержатся логи сборки (например, `warn-YourAppName.txt`), они могут помочь выявить проблемы.

\* \*\*Файл `.spec`:\*\* Для более тонкой настройки сборки PyInstaller использует `.spec` файл. Он генерируется при первом запуске `pyinstaller`. Вы можете редактировать этот файл и затем запускать `pyinstaller your\_app\_name.spec` для сборки с вашими настройками.

## 5. Тестирование

\* Тщательно протестируйте собранный `.exe` на чистой машине с Windows 11 (в идеале, без установленного Python и без доступа к интернету).

\* Тщательно протестируйте собранный `.app` на другой macOS машине.