

Crear .exe en Python

Cuando realizamos un programa para un cliente, no es lindo pasar el código fuente del mismo, ya sea por temas de contrato o simplemente para facilitarle el uso, para eso existe una librería que nos soluciona la tarea de generar un ejecutable!

PyInstaller

Es una librería para python que permite convertir programas en ejecutables para diferentes plataformas, Windows, macOS y Linux. Esto permite al usuario final utilizar el programa mediante su ejecutable sin la necesidad de haber instalado tanto el intérprete de Python en su PC como los módulos o librerías utilizados.

¿Qué hace y cómo usar PyInstaller?

PyInstaller lee y analiza nuestro código realizado en python, para descubrir que módulos y librerías se han usado y que necesita el código para ejecutarse correctamente. Recolecta esta información y la ubica en un directorio o en un archivo ejecutable según el programador lo especifique.

Facilitando la distribución y ejecución en sistemas que no tengan instalados Python.

Para utilizarlo, debemos abrir la terminal (CMD), es importante tener en cuenta que si desea crear un ejecutable con PyInstaller para windows debe ejecutarlo en Windows, si en cambio desea generar un ejecutable para Linux o macOS debe ejecutarlo en ese sistema operativo.

Para la instalación de PyInstaller usaremos un viejo conocido, ***pip install***, el comando a ejecutar en la terminal es el siguiente:

```
pip install pyinstaller
```

A la hora de ejecutar pyinstaller se pueden incluir los siguientes comandos de configuración:

```
+ -D, --onedir
```

Creará una carpeta de salida que contendrá todos los archivos necesarios para ejecutar, dentro de esta carpeta se incluirá el archivo ejecutable, los recursos y

dependencias requeridas(Si no se especifica, este comando se ejecuta por defecto)

+ -F, --onefile

Generará un solo archivo ejecutable que contendrá toda la aplicacion y sus dependencias, a diferencia del comando anterior que genera una carpeta con multiples archivos, comprime todo en un solo ejecutable, este si es necesario especificarlo a la hora de usarlo.

+ -W, --windowed

Este comando al usarlo le indicaremos a PyInstaller que la aplicacion debe ejecutarse en modo ventana, sin mostrar la consola o terminal. Esto se usa SOLO si la aplicacion se ejecutará con una interfaz grafica.

+ -add-data

Se utiliza para incluir datos o recursos adicionales, como archivos de configuracion, imagenes, etc.

Su sintaxis es --add-data "ruta_origen;ruta_destino"

+ --icon

Como su nombre indica, se utiliza para incorporar una imagen como icono del ejecutable.

¿Pero y como se usa?

Para usarlo, nos tenemos que posicionar con la terminal en la carpeta donde se encuentra nuestro archivo a transformar de .py a .exe, mediante el uso de cd o un truco para posicionarnos directamente en la carpeta, el cual consta de abrir el lugar donde esta nuestro archivo .py con el explorador de archivos y en la barra de direccion escribir cmd.

Una vez posicionado bien, debemos ejecutar el siguiente comando:

pyinstaller nombre_archivo.py

En este caso nos creará dos carpetas "Build" y "Dist" más un .spec ya que al no especificarle que lo queramos en un archivo lo hace con -D por defecto, esto podría traer problemas a la hora de ejecutarlo, ya que pueden borrarse datos o

con simplemente mover nuestro .exe a otro lado dejará de funcionar, por eso una de las mejores forma de hacerlo es colocarle el --onefile al final, tengan en cuenta que el peso del .exe aumentará ya que incluirá todas las dependencias necesarias en un solo archivo.

pyinstaller --onefile nombre_archivo.py

Forzar PyInstaller:

```
python -m PyInstaller charlando_con_ia.py
```

Documentacion de pyinstaller:

<https://pyinstaller.org/en/stable/index.html>

Comandos extras para configurar nuestro ejecutable:

<https://pyinstaller.org/en/stable/usage.html#options>