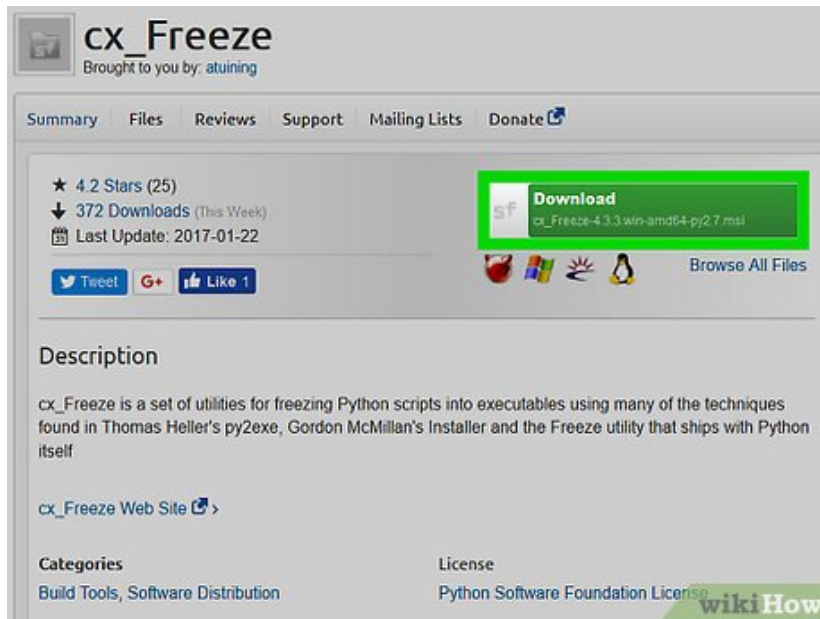


How to Compile Python Script

Python is a very popular language for programming. But what if the person running your program does not want or know how to run a Python script? This article will teach you how to compile a Python script into an executable. Download...

Method 1 of 2:

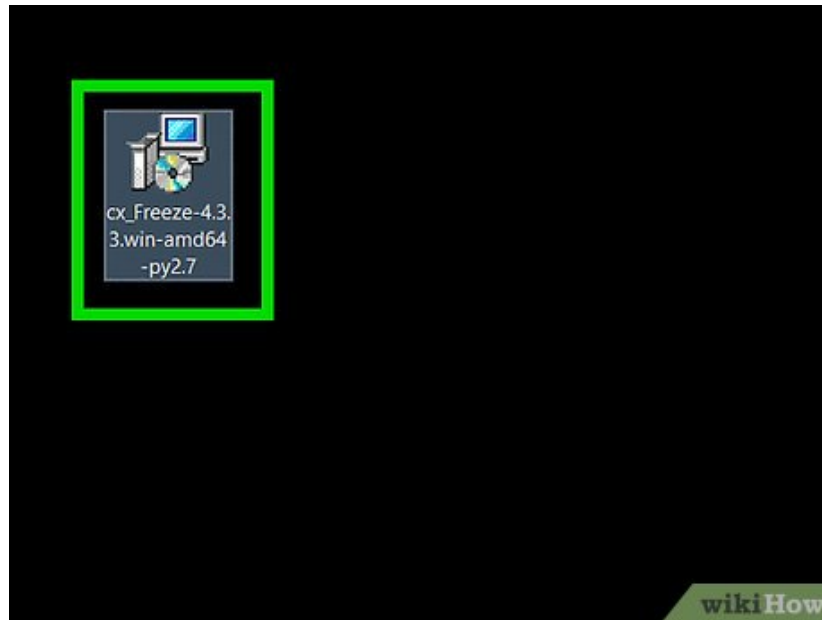
Using CX_Freeze



1.

Download cx_Freeze from Sourceforge. It's a tool for packaging Python scripts into standalone executables.

2.



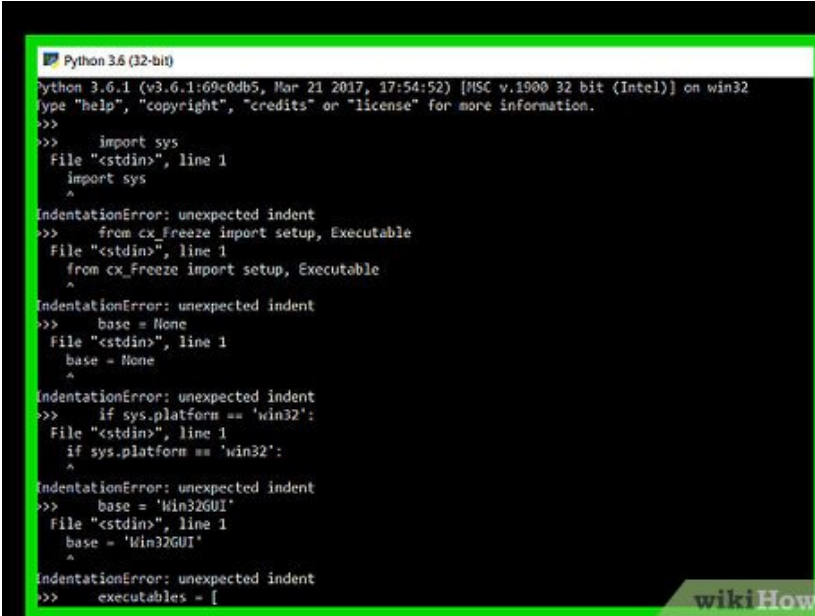
Make sure you are working on the platform you need your executable to run on. For example, if you want to create a Windows executable file, run cx_Freeze on Windows. Same goes for Mac and Linux.

3.



Create a new Python file named setup.py in the directory of the Python program you wish to compile.

4.

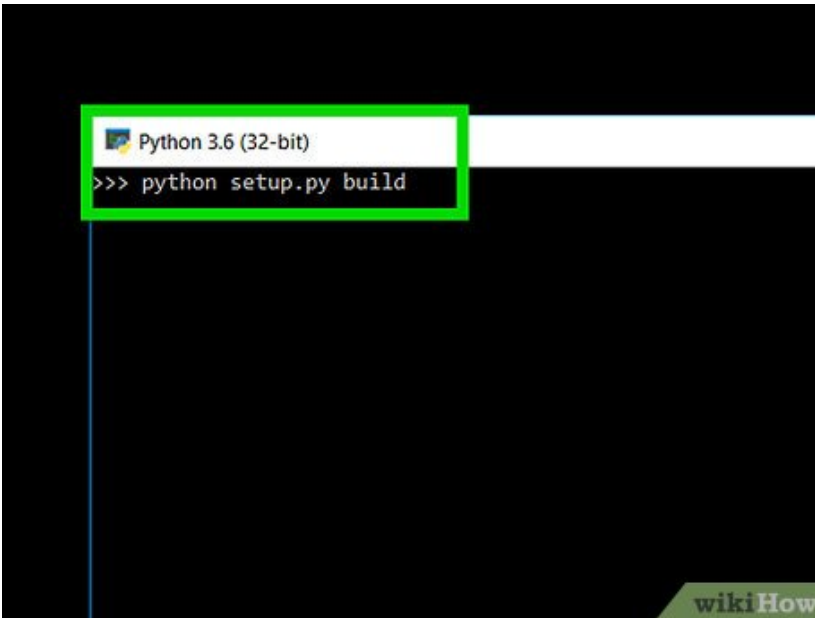


```
Python 3.6 (32-bit)
python 3.6.1 (v3.6.1:69c0db5, Mar 21 2017, 17:54:52) [MSC v.1900 32 bit (Intel)] on win32
type "help", "copyright", "credits" or "license" for more information.
>>>
>>> import sys
File "<stdin>", line 1
import sys
^
IndentationError: unexpected indent
>>> from cx_Freeze import setup, Executable
File "<stdin>", line 1
from cx_Freeze import setup, Executable
^
IndentationError: unexpected indent
>>> base = None
File "<stdin>", line 1
base = None
^
IndentationError: unexpected indent
>>> if sys.platform == 'win32':
File "<stdin>", line 1
if sys.platform == 'win32':
^
IndentationError: unexpected indent
>>> base = 'Win32GUI'
File "<stdin>", line 1
base = 'Win32GUI'
^
IndentationError: unexpected indent
>>> executables = [
```

Enter the following code into your new setup.py file. (As always in Python, correct indentation is important, and unfortunately is not shown here due to formatting difficulties.):

```
import sys from cx_Freeze import setup, Executable base = None if sys.
platform == 'win32': base = 'Win32GUI' executables = [ Executable(
Python program name, base=base) ] setup(name=executable_name, version=
'version', description='desc', executables=executables )
```

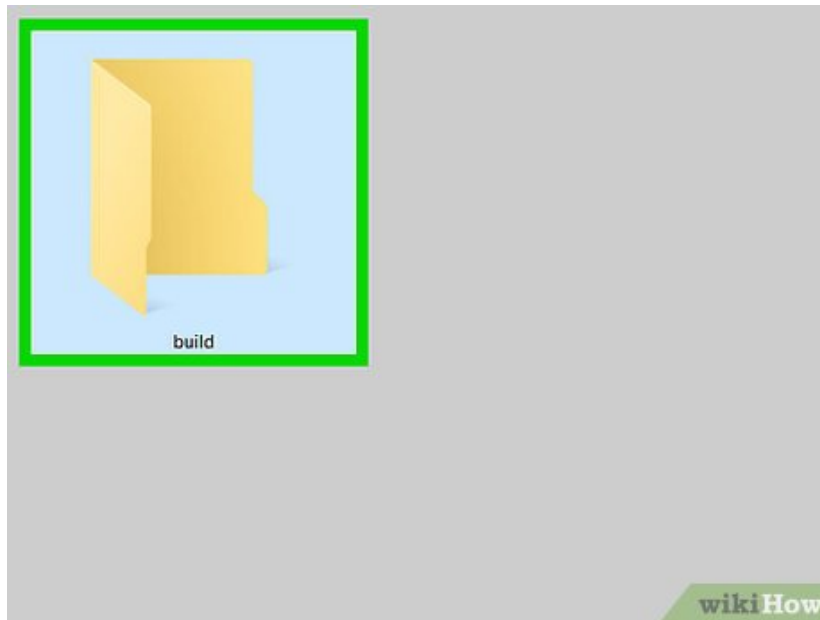
5.



```
Python 3.6 (32-bit)
>>> python setup.py build
```

Run the following commands in your computer's terminal:

```
cd [path to your Python file's directory] python setup.py build
```



6.

Look for a new folder called "build" in the Python program's directory. It should have been created during the previous step. Open that folder and the folder inside it.

1. There's your executable! The other files in that directory are required to run your executable, so be sure to always keep them with the executable.
2. The build can be customized in many ways. See cx-freeze.readthedocs.org for a description of all possible options.

Method 2 of 2:

Using PyInstaller

1.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Windows\System32>pip install pyinstaller
```

Open terminal or command prompt and execute the following code. This will install pyInstaller.

```
pip install pyinstaller
```



2.

Open the directory where the python script is located. On Windows "Right Click" while holding "Shift" and select "open command window here". On linux "Right Click" and select "Open Terminal".

A screenshot of a Windows command prompt window. The title bar reads "Select C:\Windows\system32\cmd.exe". The command prompt shows the command `pyinstaller helloworld.py` being entered and executed. The output is a series of informational messages from PyInstaller, including details about the Python version (3.6.4), platform (windows-8.1-6.3.9600-SP0), and the process of building an executable. The command prompt is dark-themed with white text. A yellow box highlights the command `pyinstaller helloworld.py` in the input line.

```
E:\Coding\pygrams\ExampleCode\Hello world>pyinstaller helloworld.py
406 INFO: PyInstaller: 3.3.1
406 INFO: Python: 3.6.4
421 INFO: Platform: windows-8.1-6.3.9600-SP0
421 INFO: wrote E:\Coding\pygrams\ExampleCode\Hello world\helloworld.spec
421 INFO: UPX is not available.
421 INFO: Extending PYTHONPATH with paths
['E:\Coding\pygrams\ExampleCode\Hello world',
'E:\Coding\pygrams\ExampleCode\Hello world']
421 INFO: checking Analysis
421 INFO: Building Analysis because out00-Analysis.toc is non existent
437 INFO: Initializing module dependency graph...
437 INFO: Initializing module graph hooks...
437 INFO: Analyzing base_library.zip ...
6875 INFO: running Analysis out00-Analysis.toc
6875 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
required by c:\python36\python.exe
8344 INFO: Caching module hooks...
8359 INFO: Analyzing E:\Coding\pygrams\ExampleCode\Hello world\helloworld.py
8375 INFO: Loading module hooks...
8375 INFO: Loading module hook "hook-encodings.py"...
8516 INFO: Loading module hook "hook-pydoc.py"...
8531 INFO: Loading module hook "hook-xml.py"...
9125 INFO: Looking for ctypes DLLs
9125 INFO: Analyzing run-time hooks ...
9141 INFO: Looking for dynamic libraries
9312 INFO: Looking for eggs
9312 INFO: Using Python library c:\python36\python36.dll
9312 INFO: Found binding redirects:
[]
9328 INFO: Warnings written to E:\Coding\pygrams\ExampleCode\Hello world\build\helloworld\warnhelloworld.txt
9422 INFO: Graph cross-reference written to E:\Coding\pygrams\ExampleCode\Hello world\build\helloworld\xref-helloworld.html
9469 INFO: checking PYZ
9469 INFO: Building PYZ because out00-PYZ.toc is non existent
9469 INFO: Building PYZ (ZlibArchive) E:\Coding\pygrams\ExampleCode\Hello world\build\helloworld\out00-PYZ.pyz
10734 INFO: Building PYZ (ZlibArchive) E:\Coding\pygrams\ExampleCode\Hello world\build\helloworld\out00-PYZ.pyz completed successfully.
10766 INFO: checking PKG
```

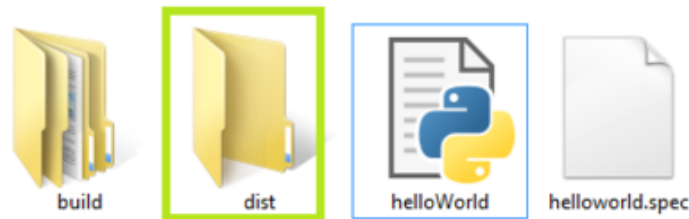
3.

Type this command to compile your script. Wait for the command to finish.

```
pyInstaller script_name.py
```

4.

Name	Date modified	Type	Size
api-ms-win-core-util-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	3 KB
api-ms-win-crt-conio-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	13 KB
api-ms-win-crt-convert-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	16 KB
api-ms-win-crt-environment-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	12 KB
api-ms-win-crt-filestream-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	14 KB
api-ms-win-crt-heap-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	12 KB
api-ms-win-crt-locale-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	12 KB
api-ms-win-crt-math-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	21 KB
api-ms-win-crt-process-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	13 KB
api-ms-win-crt-runtime-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	16 KB
api-ms-win-crt-stdio-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	18 KB
api-ms-win-crt-string-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	18 KB
api-ms-win-crt-time-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	14 KB
api-ms-win-crt-utility-l1-1-0.dll	8/28/2018 3:49 PM	Application extens...	12 KB
base_library	8/29/2018 8:29 AM	Compressed (zipp...	716 KB
helloworld	8/29/2018 8:29 AM	Application	1,406 KB
helloworld.exe.manifest	8/29/2018 8:29 AM	MANIFEST File	2 KB
pyexpat	8/28/2018 3:49 PM	Python Extension ...	186 KB
python36.dll	8/28/2018 3:49 PM	Application extens...	3,519 KB
select	8/28/2018 3:49 PM	Python Extension ...	20 KB
ucrtbase.dll	8/28/2018 3:49 PM	Application extens...	976 KB
unicodedata	8/28/2018 3:49 PM	Python Extension ...	878 KB
VCRUNTIME140.dll	8/28/2018 3:49 PM	Application extens...	86 KB



Move into the newly created "dist" directory. Your compiled project will be there.

You finished reading the article "**How to Compile Python Script**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.