# What Is Cython?

Cython is an optimizing static compiler for both the Python programming language and the extended Cython programming language (based on Pyrex). It makes writing C extensions for Python as easy as Python itself.

Many projects use Cython to **speed up some operations** of their code. The result is generally very impressive.

<https://pythonprogramming.net/introduction-and-basics-cython-tutorial/>

<https://towardsdatascience.com/use-cython-to-get-more-than-30x-speedup-on-your-python-code-f6cb337919b6>

<https://en.wikipedia.org/wiki/Cython>

# QAs

***What is Cython best used for?***

It typically used to generate CPython extension modules which can be loaded and used by regular Python code using the import statement, but with significantly less computational overhead at run time. It also facilitates wrapping independent C or C++ code into Python-importable modules.

For how to do that, check <https://cython.readthedocs.io/en/latest/index.html>.

***Can we create a Cython program which is fully standalone without the Python interpreter?***

Yes … but No.

Cython is mostly designed to make compiled Python-importable modules instead of executable files (though it's possible with a lot of work). Thus, in most cases, a Cython program will depend on the Python interpreter and standard libraries. It’s very hard to avoid this!

To make the program fully standalone, the best way is to combine Cython with an executable generator, such as **PyInstaller**. In particular, Cython will be used to compile Python to C code (not all, but some), and PyInstaller will be used to generate executable from Python and C code.

***How to generate executable file with Cython?***

As said, although it’s hard, it's possible for Cython to executable file from code. You can do what you want, but it isn't well supported. More details [here](https://stackoverflow.com/a/59389683).

A trick to compile multiple source file into a single executable file is:

1. Merge all of the source files into one .py file. Then edit the merged code a bit (removing duplicate import statement, renaming duplicate constants/variables, etc.) and make sure it runs correctly.
2. Use Cython to convert the .py or .pyx files into C files. If the converting process fails, check [this guide](https://stackoverflow.com/a/56032204).
3. Use gcc (in Windows, install Mingw-w64), or MSVC, to compile C files and build the executable. Check [this guide](http://masnun.rocks/2016/10/01/creating-an-executable-file-using-cython/) and [this guide](https://stackoverflow.com/a/22513682) for more details.

## Guideline

<https://www.peterspython.com/en/blog/using-pyinstaller-and-cython-to-create-a-python-executable>

<https://nicolas-hug.com/blog/cython_notes#:~:text=Cython%20allows%20you%20to%20release,parts%20that%20release%20the%20GIL>)