

Cython

Cython is a programming language that aims to be a superset of the Python programming language, designed to give C-like performance with code that is written mostly in Python with optional additional C-inspired syntax.^{[5][6]} Cython is a NumFOCUS affiliated project.^[7]

Cython is a compiled language that is typically used to generate CPython extension modules. Annotated Python-like code is compiled to C or C++ then automatically wrapped in interface code, producing extension modules that can be loaded and used by regular Python code using the import statement, but with significantly less computational overhead at run time. Cython also facilitates wrapping independent C or C++ code into python-importable modules.

Cython is written in Python and C and works on Windows, macOS, and Linux, producing source files compatible with CPython 2.6, 2.7, and 3.3 and later versions.

Cython 3.0.0 is in development.^[8]

Contents
<u>Design</u>
<u>History</u>
<u>Example</u>
<u>Using in IPython/Jupyter notebook</u>
<u>Uses</u>
<u>See also</u>
<u>References</u>
<u>External links</u>

Design

Cython works by producing a standard Python module. However, the behavior differs from standard Python in that the module code, originally written in Python, is translated into C. While the resulting code is fast, it makes many calls into the CPython interpreter and CPython standard libraries to perform actual work. Choosing this arrangement saved considerably on Cython's development time, but modules have a dependency on the Python interpreter and standard library.

Cython

	
Developer	Robert Bradshaw, Stefan Behnel, et al.
First appeared	28 July 2007 ^[1]
Stable release	0.29.21 (8 July 2020) [±] (https://en.wikipedia.org/w/index.php?title=Template:Latest_stable_software_release/Cython&action=edit) ^[2]
Preview release	3.0a6 (31 July 2020 ^[3]) [±] (https://en.wikipedia.org/w/index.php?title=Template:Latest_preview_software_release/Cython&action=edit) ^[2]
Implementation language	<u>Python</u>
OS	<u>Windows</u> 、 <u>MacOS</u> 、 <u>Linux</u>
License	<u>Apache License 2.0</u>
Filename extensions	.pyx, .pxd, .pxi ^[4]
Website	<u>cython.org</u> (https://cython.org) 
Influenced by	
<u>C</u> , <u>Python</u>	

Although most of the code is C-based, a small stub loader written in interpreted Python is usually required (unless the goal is to create a loader written entirely in C, which may involve work with the undocumented internals of CPython). However, this is not a major problem due to the presence of the Python interpreter.^[9]

Cython has a foreign function interface for invoking C/C++ routines and the ability to declare the static type of subroutine parameters and results, local variables, and class attributes.

A Cython program that implements the same algorithm as a corresponding Python program may consume fewer computing resources such as core memory and processing cycles due to differences between the CPython and Cython execution models. A basic Python program is loaded and executed by the CPython virtual machine, so both the runtime and the program itself consume computing resources. A Cython program is compiled to C code, which is further compiled to machine code, so the virtual machine is used only briefly when the program is loaded.^{[10][11][12][13]}

Cython employs:

- Optimistic optimizations
- Type inference (optional)
- Low overhead in control structures
- Low function call overhead^{[14][15]}

Performance depends both on what C code is generated by Cython and how that code is compiled by the C compiler.^[16]

History

Cython is a derivative of the Pyrex language, and supports more features and optimizations than Pyrex.^{[17][18]} Cython was forked from Pyrex in 2007 by developers of the Sage computer algebra package, because they were unhappy with Pyrex's limitations and could not get patches accepted by Pyrex's maintainer Greg Ewing, who envisioned a much smaller scope for his tool than the Sage developers had in mind. They then forked Pyrex as SageX. When they found people were downloading Sage just to get SageX, and developers of other packages (including Stefan Behnel, who maintains the XML library LXML) were also maintaining forks of Pyrex, SageX was split off the Sage project and merged with `cython-lxml` to become Cython.^[19]

Cython files have a `.pyx` extension. At its most basic, Cython code looks exactly like Python code. However, whereas standard Python is dynamically typed, in Cython, types can optionally be provided, allowing for improved performance, allowing loops to be converted into C loops where possible. For example:

```
def primes(int kmax): # The argument will be converted to int or raise a TypeError.
    cdef int n, k, i # These variables are declared with C types.
    cdef int p[1000] # Another C type
    result = [] # A Python type
    if kmax > 1000:
        kmax = 1000
    k = 0
    n = 2
    while k < kmax:
        i = 0
        while i < k and n % p[i] != 0:
            i = i + 1
        if i == k:
            p[k] = n
            k = k + 1
```

```

        result.append(n)
    n = n + 1
    return result

```

Example

A sample hello world program for Cython is more complex than in most languages because it interfaces with the Python C API and the `setuptools` extension building facility. At least three files are required for a basic project:

- A `setup.py` file to invoke the `setuptools` build process that generates the extension module
- A main python program to load the extension module
- Cython source file(s)

The following code listings demonstrate the build and launch process:

```

# hello.pyx - Python module, this code will be translated
to C by Cython.
def say_hello():
    print("Hello World!")

```

```

# launch.py - Python stub loader, loads the module that
was made by Cython.

# This code is always interpreted, like normal Python.
# It is not compiled to C.

import hello
hello.say_hello()

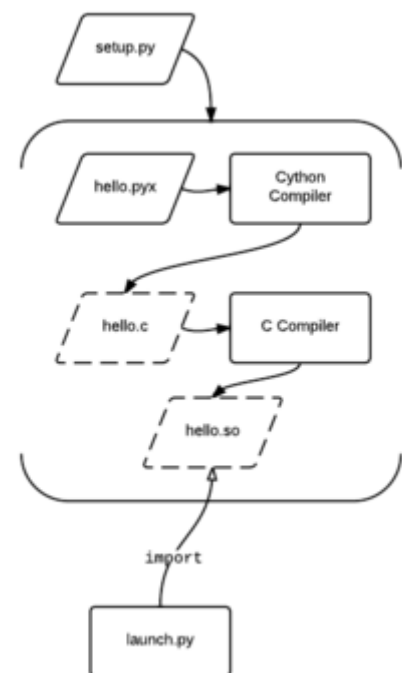
```

```

# setup.py - unnecessary if not redistributing the code, see below
from setuptools import setup
from Cython.Build import cythonize

setup(name = "Hello world app",
      ext_modules = cythonize("*.pyx"))

```



Hello World in Cython

These commands build and launch the program:

```

$ python setup.py build_ext --inplace
$ python launch.py

```

Using in IPython/Jupyter notebook

A more straightforward way to start with Cython is through command-line IPython (or through in-browser python console called Jupyter notebook):

```

In [1]: %load_ext Cython

In [2]: %%cython
...: def f(n):

```

```

...:     a = 0
...:     for i in range(n):
...:         a += i
...:     return a
...:
...: cpdef g(int n):
...:     cdef long a = 0
...:     cdef int i
...:     for i in range(n):
...:         a += i
...:     return a
...:

```

```

In [3]: %timeit f(1000000)
10 loops, best of 3: 26.5 ms per loop

```

```

In [4]: %timeit g(1000000)
1000 loops, best of 3: 279 µs per loop

```

which gives a 95 times improvement over the pure-python version. More details on the subject in the official quickstart page.^[20]

Uses

Cython is particularly popular among scientific users of Python,^{[12][21][22]} where it has "the perfect audience" according to Python creator Guido van Rossum.^[23] Of particular note:

- The free software SageMath computer algebra system depends on Cython, both for performance and to interface with other libraries.^[24]
- Significant parts of the scientific computing libraries SciPy, pandas and scikit-learn are written in Cython.^{[25][26]}
- Some high-traffic websites such as Quora use Cython.^[27]

Cython's domain is not limited to just numerical computing. For example, the lxml XML toolkit is written mostly in Cython, and like its predecessor Pyrex, Cython is used to provide Python bindings for many C and C++ libraries such as the messaging library ZeroMQ.^[28] Cython can also be used to develop parallel programs for multi-core processor machines; this feature makes use of the OpenMP library.

See also

- PyPy
- Numba


References

1. Behnel, Stefan (2008). "The Cython Compiler for C-Extensions in Python" (<http://www.behnel.de/cythonEP2008/cython-ep2008.html>). *EuroPython* (28 July 2007: official Cython launch). Vilnius/Lietuva.
2. "Releases – cython/cython" (<https://github.com/cython/cython/releases>). Retrieved 16 October 2020 – via GitHub.
3. "cython/cython" (<https://github.com/cython/cython/blob/master/CHANGES.rst#300-alpha-6-2020-07-31>). *GitHub*. Retrieved 7 October 2020.

4. "Language Basics — Cython 3.0.0a9 documentation" (https://cython.readthedocs.io/en/latest/src/userguide/language_basics.html#cython-file-types). *cython.readthedocs.io*. Retrieved 9 September 2021.
5. "Cython - an overview — Cython 0.19.1 documentation" (<http://docs.cython.org/src/quickstart/overview.html>). *Docs.cython.org*. Retrieved 21 July 2013.
6. Smith, Kurt (2015). *Cython: A Guide for Python Programmers* (<http://shop.oreilly.com/product/0636920033431.do>). O'Reilly Media. ISBN 978-1-4919-0155-7.
7. "NumFOCUS Affiliated Projects" (<https://numfocus.org/sponsored-projects/affiliated-projects>). NumFOCUS. Retrieved 25 October 2021.
8. "Support Unicode identifiers · Issue #2601 · cython/cython" (<https://github.com/cython/cython/issues/2601>). *GitHub*. Retrieved 11 October 2019.
9. "Basic Tutorial — Cython 3.0a6 documentation" (https://cython.readthedocs.io/en/latest/src/tutorial/cython_tutorial.html). *cython.readthedocs.io*. Retrieved 11 December 2020.
10. Oliphant, Travis (20 June 2011). "Technical Discovery: Speeding up Python (NumPy, Cython, and Weave)" (<http://technicaldiscovery.blogspot.com/2011/06/speeding-up-python-numpy-cython-and.html>). *Technicaldiscovery.blogspot.com*. Retrieved 21 July 2013.
11. Behnel, Stefan; Bradshaw, Robert; Citro, Craig; Dalcin, Lisandro; Seljebotn, Dag Sverre; Smith, Kurt (2011). "Cython: The Best of Both Worlds" (<http://research.google.com/pubs/pub36727.html>). *Computing in Science and Engineering*. **13** (2): 31–39. Bibcode:2011CSE....13b..31B (<https://ui.adsabs.harvard.edu/abs/2011CSE....13b..31B>). doi:10.1109/MCSE.2010.118 (<https://doi.org/10.1109%2FMCSE.2010.118>). S2CID 14292107 (<https://api.semanticscholar.org/CorpusID:14292107>).
12. Seljebot, Dag Sverre (2009). "Fast numerical computations with Cython" (http://conference.scipy.org/proceedings/SciPy2009/paper_2). *Proceedings of the 8th Python in Science Conference (SciPy 2009)*: 15–22.
13. Wilbers, I.; Langtangen, H. P.; Ødegård, Å. (2009). B. Skallerud; H. I. Andersson (eds.). "Using Cython to Speed up Numerical Python Programs" (http://simula.no/research/sc/publications/Simula.SC.578/simula_pdf_file) (PDF). *Proceedings of MekIT'09*: 495–512. Retrieved 14 June 2011.
14. "wrapper benchmarks for several Python wrapper generators (except Cython)" (<https://web.archive.org/web/20150404154630/http://telecom.inescporto.pt/~gjc/pybindgen-benchmarks/>). Archived from the original (<http://telecom.inescporto.pt/~gjc/pybindgen-benchmarks/>) on 4 April 2015. Retrieved 28 May 2010.
15. "wrapper benchmarks for Cython, Boost.Python and PyBindGen" (<https://web.archive.org/web/20160303224104/http://www.behnel.de/cycppbench/>). Archived from the original (<http://www.behnel.de/cycppbench/>) on 3 March 2016. Retrieved 28 May 2010.
16. "Cython: C-Extensions for Python" (<http://cython.org/index.html>). Retrieved 22 November 2015.
17. "Differences between Cython and Pyrex" (<https://github.com/cython/cython/wiki/DifferencesFromPyrex>).
18. Ewing, Greg (21 March 2011). "Re: VM and Language summit info for those not at Pycon (and those that are!)" (<http://mail.python.org/pipermail/python-dev/2011-March/109642.html>) (Message to the electronic mailing-list python - dev). Retrieved 5 May 2011.
19. Says Sage and Cython developer Robert Bradshaw at the Sage Days 29 conference (22 March 2011). "Cython: Past, Present and Future" (<https://www.youtube.com/watch?v=osjSS2RrvM0>). *youtube.com*. Archived (<https://ghostarchive.org/varchive/youtube/20211221/osjSS2RrvM0>) from the original on 21 December 2021. Retrieved 5 May 2011.
20. "Building Cython code" (<https://cython.readthedocs.io/en/latest/src/quickstart/build.html>). *cython.readthedocs.io*. Retrieved 24 April 2017.

21. "inSClght: The Scientific Computing Podcast" (https://web.archive.org/web/20141010032300/http://inscight.org/2011/03/31/episode_/) (Episode 6). Archived from the original (http://inscight.org/2011/03/31/episode_/) on 10 October 2014. Retrieved 29 May 2011.
22. Millman, Jarrod; Aivazis, Michael (2011). "Python for Scientists and Engineers" (<https://escholarship.org/uc/item/93s2v2s7>). *Computing in Science and Engineering*. **13** (2): 9–12. Bibcode:2011CSE....13b...9M (<https://ui.adsabs.harvard.edu/abs/2011CSE....13b...9M>). doi:10.1109/MCSE.2011.36 (<https://doi.org/10.1109%2FMCSE.2011.36>).
23. Guido Van Rossum (21 March 2011). "Re: VM and Language summit info for those not at Pycon (and those that are!)" (<http://mail.python.org/pipermail/python-dev/2011-March/109634.html>) (Message to the electronic mailing-list python - dev). Retrieved 5 May 2011.
24. Erocál, Burcin; Stein, William (2010). *The Sage Project: Unifying Free Mathematical Software to Create a Viable Alternative to Magma, Maple, Mathematica and MATLAB* (http://wstein.org/papers/icms/icms_2010.pdf) (PDF). *Mathematical Software' ICMS 2010*. Lecture Notes in Computer Science. Vol. 6327. Springer Berlin / Heidelberg. pp. 12–27. CiteSeerX 10.1.1.172.624 (<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.172.624>). doi:10.1007/978-3-642-15582-6_4 (https://doi.org/10.1007%2F978-3-642-15582-6_4). ISBN 978-3-642-15581-9.
25. "SciPy 0.7.2 release notes" (<http://docs.scipy.org/doc/scipy/reference/release.0.7.2.html>).
26. Pedregosa, Fabian; Varoquaux, Gaël; Gramfort, Alexandre; Michel, Vincent; Thirion, Bertrand; Grisel, Olivier; Blondel, Mathieu; Prettenhofer, Peter; Weiss, Ron; Dubourg, Vincent; Vanderplas, Jake; Passos, Alexandre; Cournapeau, David (2011). "Scikit-learn: Machine Learning in Python". *Journal of Machine Learning Research*. **12**: 2825–2830. arXiv:1201.0490 (<https://arxiv.org/abs/1201.0490>).
27. "Is Quora still running on PyPy?" (<https://www.quora.com/Is-Quora-still-running-on-PyPy/answer/Alex-Yakunin>).
28. "ØMQ: Python binding" (<http://www.zeromq.org/bindings:python>).

External links

- [Official website](https://cython.org) (<https://cython.org>) 
 - [Cython](https://github.com/cython) (<https://github.com/cython>) on [GitHub](#)
-

Retrieved from "<https://en.wikipedia.org/w/index.php?title=Cython&oldid=1061421233>"

This page was last edited on 21 December 2021, at 16:34 (UTC).

Text is available under the Creative Commons Attribution-ShareAlike License; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.