

Cython for TriPython

Chris Calloway, UNC/RENCI

October 22, 2015

What is Cython?

- **A programming language**
 - **Includes 100 percent of Python**
 - **Plus a very few additional helper keywords**
- **A Cython language compiler**
 - **Creates Python C-API extension modules**
 - **Or creates standalone executables**
 - **Also can wrap C and C++ into an extension module**

Installing Cython

- **Use a scientific Python distribution**
 - **Anaconda**
 - **Canopy**
 - **Pythonxy**
 - **Sage**
- **Or install and configure a C compiler**
- **Followed by pip install cython**

Cython Notes

- **Depends on Python C-API**
 - **Therefore targets CPython**
- **Two step process**
 - **First creates Python C-API source code**
 - **Then compiles the C source**

Cython Notes

- **Improves processor-bound performance**
 - **Function call overhead**
 - **Loops**
 - **Math (operation lookup and data access)**
 - **Object creation (stack vs heap allocation)**

Cython Notes

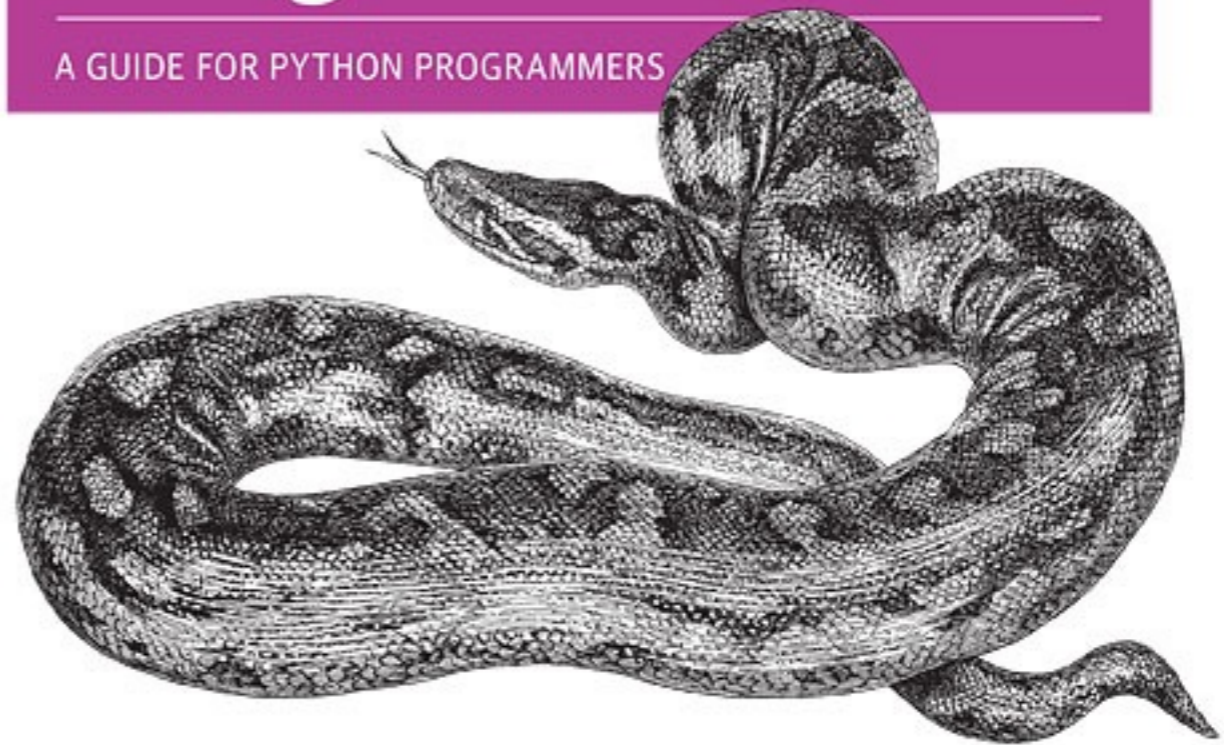
- **Grew out of Pyrex**
 - **Easier than Boost or SWIG**
- **Sage has several hundred KLOC of Cython**
- **Other users: Numpy, Scipy, Pandas, lxml, Scikit**
- **Support for Numpy**
- **Support for parallelization (turns off the GIL!)**

- **Great book!**
- **docs.cython.org**
- **SciPy15 tutorial**

O'REILLY®

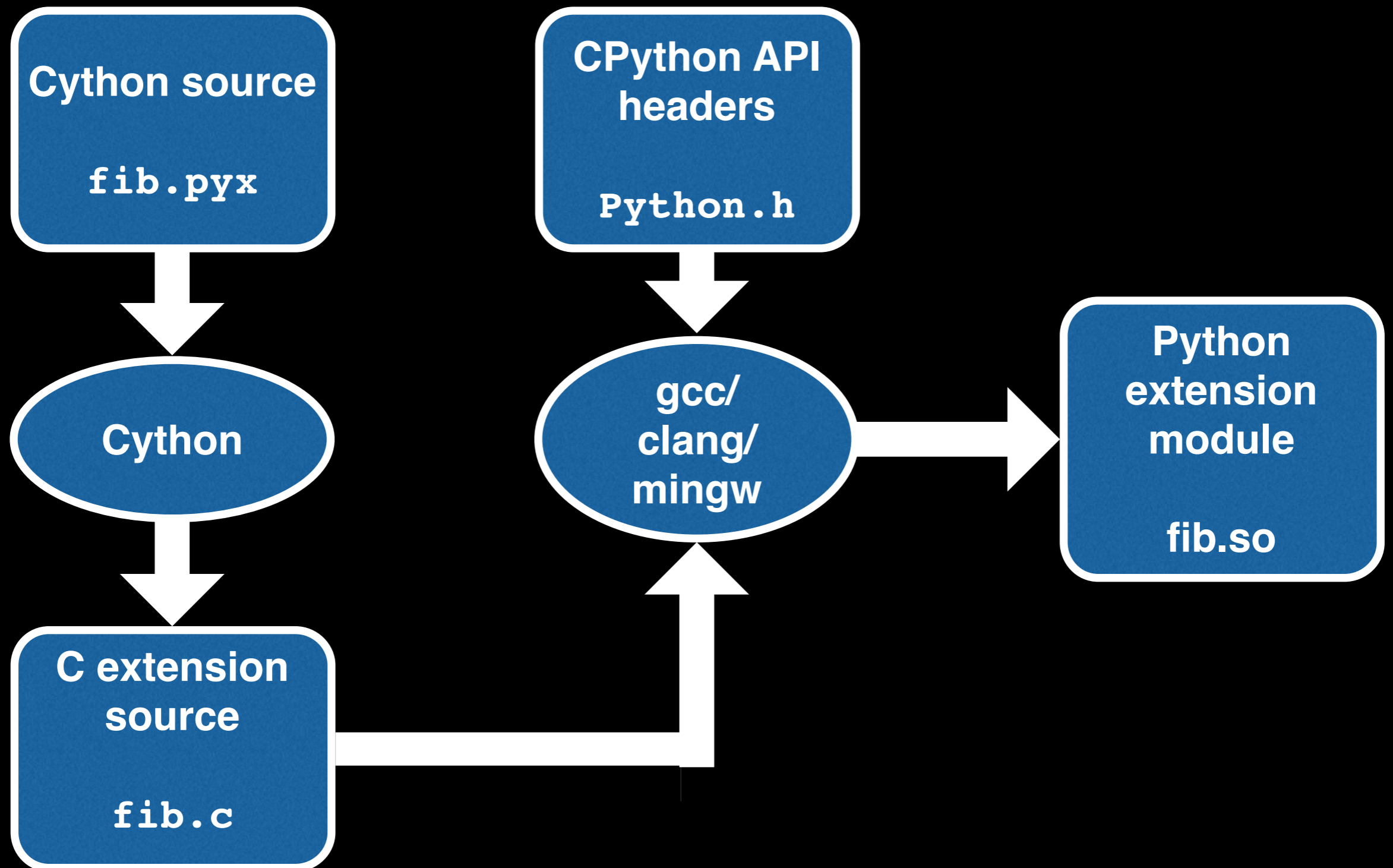
Cython

A GUIDE FOR PYTHON PROGRAMMERS



Kurt W. Smith

How to Cython



fib.py and cyfib.pyx

```
def fib(n):  
    a, b = 0, 1  
    for i in range(n):  
        a, b = a + b, a  
    return a
```

How to Cython

```
$ ipython
```

```
In [1]: import fib
```

```
In [2]: import pyximport
```

```
In [3]: pyximport.install()
```

```
In [4]: import cyfib
```

How to Cython

```
In [5]: fib.__file__
```

```
Out[5]: 'fib.pyc'
```

```
In [6]: cyfib.__file__
```

```
Out[6]: '~/pyxbld/.../cyfib.so'
```

How to Cython

```
In [7]: %timeit fib.fib(1000)
```

```
10000 loops, best of 3: 118  $\mu$ s per loop
```

```
In [8]: %timeit cyfib.fib(1000)
```

```
10000 loops, best of 3: 53.7  $\mu$ s per loop
```

cyfib1.pyx

```
def fib(int n):  
    cdef int i  
    cdef double a = 0.0, b = 1.0  
    for i in range(n):  
        a, b = a + b, a  
    return a
```

How to Cython

```
In [9]: import cyfib1
```

```
In [10]: %timeit cyfib1.fib(1000)
```

```
10000000 loops, best of 3: 1.08  $\mu$ s per loop
```

setup.py

```
from distutils.core import setup
from Cython.Build import cythonize

setup(ext_modules=
      cythonize("cyfib1.pyx"))
```

How to Cython

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

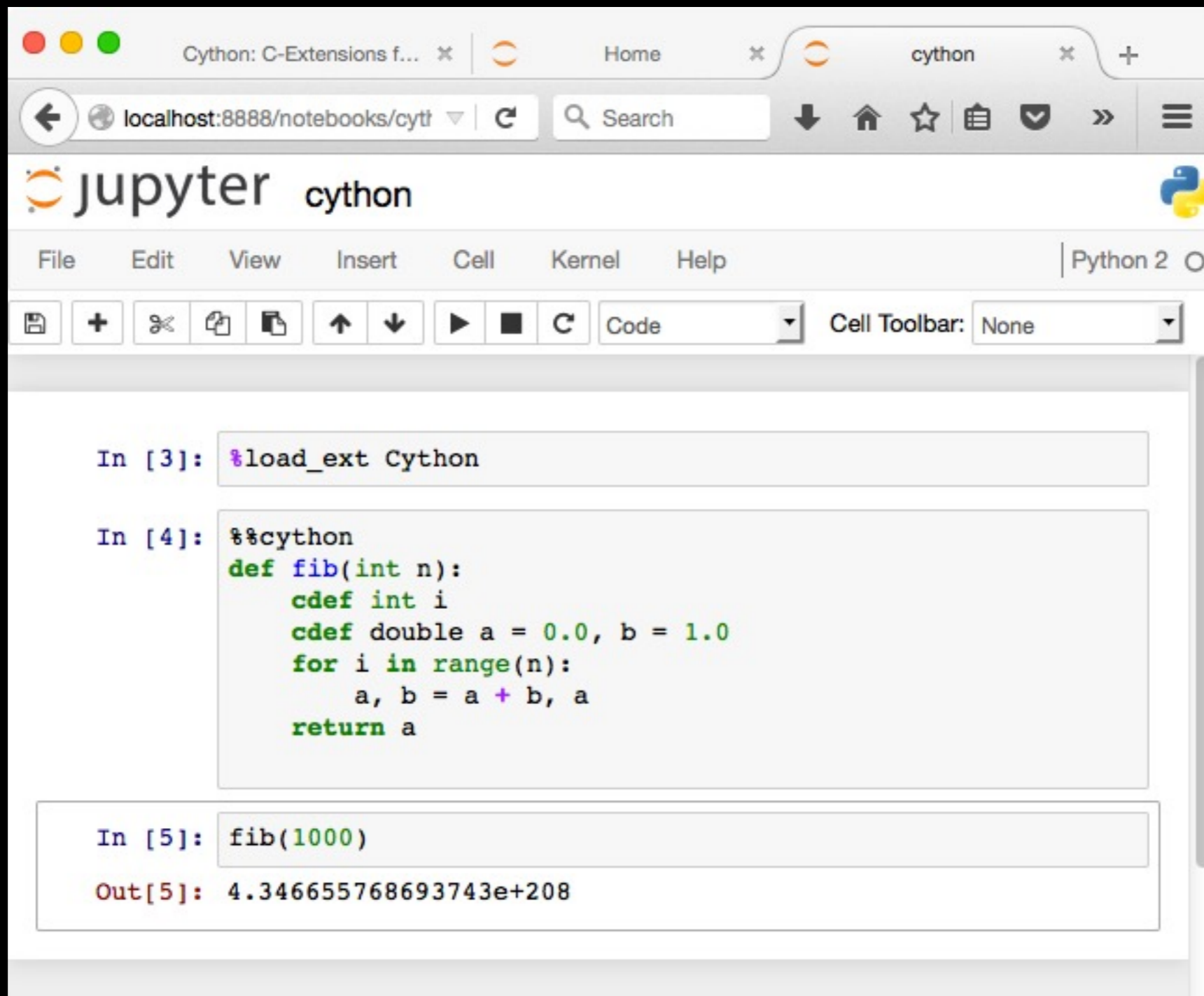
```
$ python
```

```
>>> import cyfib1
```

```
>>> cyfib1.fib(1000)
```

```
4.346655768693743e+208
```


How to Cython



The screenshot shows a Jupyter Notebook interface in a web browser. The browser tabs include 'Cython: C-Extensions f...', 'Home', and 'cython'. The address bar shows 'localhost:8888/notebooks/cytl'. The Jupyter logo and 'cython' are visible in the top left. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. The toolbar shows various icons for file operations and execution. The notebook content consists of three input cells and one output cell.

```
In [3]: %load_ext Cython
```

```
In [4]: %%cython
def fib(int n):
    cdef int i
    cdef double a = 0.0, b = 1.0
    for i in range(n):
        a, b = a + b, a
    return a
```

```
In [5]: fib(1000)
```

```
Out[5]: 4.346655768693743e+208
```

cfib.c

```
double cfib(int n) {  
    int i;  
    double a=0.0, b=1.0, tmp;  
    for (i=0; i<n; i++) {  
        tmp = a;  
        a = a+b;  
        b = tmp;  
    }  
    return a;  
}
```

cfib.h

```
double cfib(int n);
```

cyfib2.pyx

```
cdef extern from "cfib.h":  
    double cfib(int n)  
  
def fib(int n):  
    return cfib(n)
```

setup.py

```
from distutils.core import setup, \
    Extension
from Cython.Build import cythonize

ex = Extension(name="cyfib2",
               sources=["cyfib2.pyx",
                       "cfib.c", ])

setup(ext_modules=cythonize([ex, ]))
```

How to Wrap C

```
In [11]: import cyfib2
```

```
In [12]: %timeit cyfib2.fib(1000)
```

```
1000000 loops, best of 3: 1.07  $\mu$ s per loop
```

Standalone Executables

```
$ cython -embed fib.py
```

```
$ gcc $(python-config --includes) \  
      $(python-config --ldflags) \  
      -o fib ./fib.c
```

Standalone Executables

```
$ fib 9
```

```
34
```

```
$ fib 1000
```

```
434665576869374564356885276750406258025646  
605173717804024817290895365554179490518904  
038798400792551692959225930803226347752096  
896232398733224711616429964409065331879382  
98969649928516003704476137795166849228875
```


Standalone Executables

- **Hat trick**
- **Not statically linked**
- **Need a statically linked Python**
- **Need to statically link stdlib packages used**
- **static-python and nuitka**

More Cython

This screenshot shows the GitHub interface for the repository 'kwmsmith / scipy-2015-cython-tutorial'. At the top, there's a search bar and navigation links for 'Pull requests', 'Issues', and 'Gist'. The repository name is displayed with a 'Watch' button and a notification count of '3'. Below this, a link is provided for the SciPy 2015 Cython tutorial content. A summary bar indicates '24 commits', '1 branch', '3 releases', and '2 contributors'. The current branch is 'master'. A recent merge pull request is noted. A file list shows folders for environment checks, exercises, and a README, along with a PDF of slide material. The main content area displays the title 'SciPy 2015 Cython Tutorial'.

This repository Search

Pull requests Issues Gist

kwmsmith / **scipy-2015-cython-tutorial** Watch 3

Content for the SciPy 2015 Cython tutorial: <http://www.scipy2015.scipy.org/ehome/115969/289057/?&>

24 commits 1 branch 3 releases 2 contributors

Branch: master **scipy-2015-cython-tutorial** / +

kwmsmith Merge pull request #1 from tacaswell/py3k_fixes Latest commit 3ef458b on Jul 6

check-env-linux-osx	Rename directories.	4 months ago
check-env-windows	Rename directories.	4 months ago
exercises	MNT: python3 compat	4 months ago
README.md	first commit	4 months ago
cython-scipy-2015-kurt-smith.pdf	Add PDF of slide material.	4 months ago

README.md

SciPy 2015 Cython Tutorial

More Cython

YouTube

SciPy2015

Scientific Computing with Python
Austin, Texas • July 6-12, 2015

0:02 / 3:42:01

Cython: Blend the Best of Python and C++ | SciPy 2015 Tutorial | Kurt Smith

Enthought
Subscribe 7,226

2,421

+ Add to Share ... More

27 0

Thank You!

Questions?