Introduction to Cython - Week 3

Richard Killam

Monday 2nd November, 2015

Outline

Exercise

sum_nums.pyx

Exercise

sum_nums.pyx

```
1  cdef unsigned long long sum_nums(unsigned long n):
2      cdef unsigned long long s = 0
3      cdef unsigned long i
4      for i in range(1, n+1):
5          s += i
6      return s
```

run_sum_nums.py

run_sum_nums.py

```
1  import sum_nums as sn
2
3  print(sn.sum_nums(10))
```

run_sum_nums.py

```
1  import sum_nums as sn
2
3  print(sn.sum_nums(10))
```

python setup.py build_ext –inplace
python run_sum_nums.py

run_sum_nums.py

```
1  import sum_nums as sn
2
3  print(sn.sum_nums(10))
```

python setup.py build_ext –inplace
python run_sum_nums.py
>> AttributeError: 'module' object has no attribute 'sum_nums'

The Fix

Add to sum_nums.pyx

```
1  def sum_nums_wrapper(n):
2      return sum_nums(int(n))
```

The Fix

Add to sum_nums.pyx

```
1 def sum_nums_wrapper(n):
2     return sum_nums(int(n))
```

run_sum_nums.py

```
1 import sum_nums as sn
2
3 print(sn.sum_nums_wrapper(10))
```

Testing The Fix

```
python setup.py build_ext –inplace
python rum_sum_nums.py
```

Testing The Fix

```
python setup.py build_ext –inplace
python rum_sum_nums.py
>> 55
```

The Sugar

sum_nums.pyx

```
1  cpdef unsigned long long sum_nums(unsigned long n):
2      cdef unsigned long long s = 0
3      cdef unsigned long i
4      for i in range(1, n+1):
5          s += i
6      return s
```

Using The Sugar

run_sum_nums.py

```
1  import sum_nums as sn
2
3  print(sn.sum_nums(10))
```

Using The Sugar

run_sum_nums.py

```
1 import sum_nums as sn
2
3 print(sn.sum_nums(10))
```

python setup.py build_ext –inplace
python run_sum_nums.py

Using The Sugar

run_sum_nums.py

```
1 import sum_nums as sn
2
3 print(sn.sum_nums(10))
```

python setup.py build_ext –inplace
python run_sum_nums.py
>> 55

Exercise

sum_logs.pyx (The log function can be found in the math module)

Exercise

sum_logs.pyx (The log function can be found in the math module)

```
1  import math
2
3  cpdef double log_loop(double n, unsigned long iters):
4      cdef unsigned long i
5      cdef double s = 0
6      for i in range(iters):
7          s += math.log(n)
8
9      return s
```

Running sum_logs.pyx

setup.py

```
1  from distutils.core import setup
2  from Cython.Build import cythonize
3
4  setup(
5      ext_modules=cythonize('*.pyx'),
6  )
```

Running sum_logs.pyx

setup.py

```
1  from distutils.core import setup
2  from Cython.Build import cythonize
3
4  setup(
5      ext_modules=cythonize('*.pyx'),
6  )
```

python setup.py build_ext –inplace
python run_sum_logs.py

Log Loops

clog.pyx

```
1  cimport libc.math as cmath
2
3  cpdef double log_loop(double n, unsigned long iters):
4      cdef unsigned long i
5      cdef double s = 0
6      for i in range(iters):
7          s += cmath.log(n)
8
9      return s
```

Speed Test

run_log_loops.py (Summarized)

```
1 import clog
2 print(clog.log_loop(100,000,000, 10,000,000))
3
4 import pylog
5 print(pylog.log_loop(100,000,000, 10,000,000))
```

Speed Test

run_log_loops.py (Summarized)

```
1 import clog
2 print(clog.log_loop(100,000,000, 10,000,000))
3
4 import pylog
5 print(pylog.log_loop(100,000,000, 10,000,000))
```

clog: 184,206,807.418
clog took: 0.77s

pylog: 184,206,807.418
pylog took: 8.05s

External Definition

clog_hard.pyx

```
1  cdef extern from "math.h":
2      float log(float n)
3
4  cpdef double log_loop(double n, unsigned long iters):
5      cdef unsigned long i
6      cdef double s = 0
7      for i in range(iters):
8          s += log(n)
9
10     return s
```

Testing the Hard Way

clog_hard: 184,206,807.418
clog_hard took: 0.77s

pylog: 184,206,807.418
pylog took: 8.08s

Comparison

cimport

- more generic
- simpler

cdef extern

- more explicit
- works for everything

Comparison

cimport

- more generic
- simpler

cdef extern

- more explicit
- works for everything
- can be moved to a cython header file