Enthought

Product

Consulting

Training

Cython!

Poruri Sai Rahul,
Software Developer,
Enthought Inc.

What?

- What?
  - Cython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

What? Cython!

Cython, not CPython

- Cython, the language
  - A superset of the Python language
- Cython, the library
  - The library that compiles Cython code

Why?

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

Why? - Need for Speed

- Why is Python slower than C/C++ or JavaScript/Julia?
- Compiled (vs) Interpreted (vs) JIT compiled.

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

@rahulporuri

Why? - Need for Speed

- JIT compilers for Python code
  - PyPy, which has it's own JIT compiler in RPython
  - Pyston and Numba, which use LLVM as a JIT compiler
  - Pyjion, which uses CoreCLR as a JIT compiler. Also see, PEP 523
- Cython, which can be used to compile Python code and interoperate with C/C++ code.
- Ctypes, which can be used to load C modules and use C functions.
- Python extension modules, which are C modules that can be used in Python code.

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

Why? - C/C++ interoperability

- Inertia because of existing C/C++ codebases

How?

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Let's start with functions

Let's look at a standard function in Python.

```
def sum_till_num(n):
    total_sum = 0
    for i in range(n):
        total_sum += i
    return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - All Python is valid Cython

The same function, written in Cython, looks like

```
def sum_till_num(n):
    total_sum = 0
    for i in range(n):
        total_sum += i
    return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Adding type information

For real, rewriting the function with help from Cython gives

```
def sum_till_num(int n):
    cdef int total_sum = 0
    cdef int i
    for i in range(n):
        total_sum += i
    return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Using cdef

```
cdef sum_till_num(int n):
    cdef int total_sum = 0
    cdef int i
    for i in range(n):
        total_sum += i
    return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Using cpdef

```
cpdef sum_till_num(int n):
    cdef int total_sum = 0
    cdef int i
    for i in range(n):
        total_sum += i
    return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Understanding Cython

What's the difference?

- `def` - creates Python function objects
- `cdef` - creates functions that are faster but only accessible from Python modules.
- `cpdef` - creates functions that are faster but also accessible from Python modules.

Stop!
Profile time!

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

Stop - Are you sure you need to do this?

- “Premature optimization is the root of all evil” - Donald Knuth.

- Is Python not good enough?
- What if your Python code could be improved, instead of having to write Cython code?
- Profile before moving to Cython! Use timeit, profile or the cProfile modules.

Back to Cython.

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Where to write Cython

Profile your Python code for places that can be improved using Cython.

```
python -m cython -a file.py
```

```
+1: def sum_till_num(n):
+2:     total_sum = 0
+3:     for i in range(n):
+4:         total_sum += i
 5:
+6:     return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Profile Cython code

```
+1: def sum_till_num(int n):
+2:     cdef int total_sum = 0
 3:     cdef int i
+4:     for i in range(n):
+5:         total_sum += i
 6:
+7:     return total_sum
```

@rahulporuri

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Profile Cython code

```
+1: cdef sum_till_num(int n):
+2:     cdef int total_sum = 0
 3:     cdef int i
+4:     for i in range(n):
+5:         total_sum += i
 6:
+7:     return total_sum
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Compiling Cython code

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

(setup.py)

```
from distutils.core import setup

from Cython.Build import cythonize


setup(

    ext_modules = cythonize("module.pyx")

)
```

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

How? - Compiled Cython code

- The pure Python function when cythonized creates a C file that's ~2700 lines long.
- The cythonized functions, using def and cdef, creates files that are ~1900 lines in length.

What now?

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile
  - Use

What now?

- Read Cython docs
- Read Cython code
  - Cython, Pandas, SciPy
- Write Cython code

- What?
  - Cython, not CPython
- Why?
  - Speed
  - Interop
- How?
  - Write
  - Compile

Thank You

- Read Cython docs
- Read Cython code
  - Cython, Pandas, SciPy
- Write Cython code

@rahulporuri