Cython
Past, Present, and Future

Robert Bradshaw
Google, Seattle

In the beginning...

- 1991. **Guido van Rossum** releases the first version of **Python**.
- 2002. **Greg Ewing** releases **Pyrex** as a tool to more easily create Python extensions.
- 2004. **William Stein** discovers Pyrex, filling essential need in what would become **Sage**.

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage...

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage... but there were issues:

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage... but there were issues:

- no cross-package cimport (`sage/ext`)

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage... but there were issues:

- no cross-package cimport (`sage/ext`)
- no introspection

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage... but there were issues:

- no cross-package cimport (`sage/ext`)
- no introspection
- no 2.5 support.

Pyrex

By 2006, **thousands** of lines of Pyrex code had been added to Sage... but there were issues:

- no cross-package cimport (`sage/ext`)
- no introspection
- no 2.5 support.

These were **not** considered **bugs**, patches **not accepted** upstream.

SageX

So we started shipping our own **patched** copy of Pyrex, called SageX.

- List Comprehension
- In-place arithmetic
- Conditional expressions
- `Py_ssize_t` type
- `inline` modifier for c functions
- Assignment on declaration
- `bint` (boolean int) type
- ...

and lots of **optimizations**.

SageX

A **new philosophy** starts to take hold

Why not compile all valid Python?

SageX

Outside interest

- People **downloading** all of Sage **just** for **SageX**
- **Stefan Behnel** maintained `cython-lxml` for `lxml`
- Pyrex development continued at a **snail's pace**.

Cython

In 2007, we officially **forked** Pyrex as a new project, **Cython**.

Cython

In 2007, we officially **forked** Pyrex as a new project, **Cython**.

Cython

In 2007, we officially **forked** Pyrex as a new project, **Cython**.

Though the projects and languages have diverged, it's about as **amicable** of a fork as I've ever seen.

Philosophy

Cython aims to be an **optimizing compiler** for the Python langauge, requiring minimal knowledge of the Python/C API to get **significant speedups** with **optional static typing** for **low-level number crunching** and direct linking to **C/C++/Fortran libraries**.

Cython is a very **pragmatic** project, driven by **user needs**.

Philosophy

Pyrex "makes no claim to be a Python compiler; rather, it's a **tool** for **bridging** between the Python and C worlds. As such, strict adherence to Python syntax and semantics, or supporting all Python features, or performing heroic optimisations on pure Python code, are **not design goals**."

and

"...changes at a much more conservative pace!" [1]

[1] Greg Ewing, Python-Dev, March 2011.
http://mail.python.org/pipermail/python-dev/2011-March/109642.html

Cython

Since 2007 we have attracted dozens of **contributors** and many new **features**

- buffer (NumPy) support
- cpdef functions
- annotated html
- type inference
- build, inline
- closures
- Python3, ...

Of special interest to Sage developers

Cython has evolved over the years

- Compiler directives
- profiling
- C++ support
- Cython/Includes
- `pxd` not `pxi`

Present

- Cython **0.14.1** (coming in Sage 4.7)
- About **2500 visitors**/week to our site, thousands of downloads.
- Packaged in several major **linux distrobutions**.
- The majority of our codebase is **new, post-Pyrex** code.
- **Cython Days** 1 in Munich, Germany next week

Future

- Full Python (some specific 2.x) compatibility a **1.0** goal
  - Generators are last big missing piece (mostly implemented).
  - Hopefully by the end of the year (summer?).
- Control flow (guards, many other optimizations)
- IronPython, PyPy?.
- Lots of other improvements (NumPy, C++, type parameterization, signal handling, ctypes, auto-cdef, ...).
- Use in standard library? (Recent thread on python-dev.)

Questions?