cppyy is a fully automated, run-time, language bridge between C++ and Python. Full automation is very effective and covers the vast majority of use cases, but strictly bound C++ code sometimes lacks a Python “feel” to it, making it harder to understand and use than necessary. We provide a pythonization API to easily cover common cases and allow further custom detailing where desired.

Pythonizations Supported

- Method decoration/adaptation
- Getter/setters → properties
- Overload additions
- Method/variable renaming
- Memory ownership control
- GIL management
- Transparent smart pointers
- Iteration on STL-like containers
- Exception remapping
- Return type pinning

Or, take full manual control using reflection and JIT!

Python Interpreter

```python
>>> from cppyy.gbl import TGraph, MyClass, throw_it
>>> N = 5
>>> vals = array.array('d', xrange(N))
>>> g = TGraph(N, vals, vals)
>>> print len(g.GetX()) == N
True
```

Full manual control is available by introspecting classes in the callback. C++ helper functions can be generated, JITed, and bound to further augment a Python class.

References

http://root.cern.ch/drupal/content/pyroot
http://root.cern.ch/drupal/content/pypyroot

Wim T.L.P. Lavrijsen (WLavrijsen@lbl.gov)
Toby St Clere Smith (mail@tsmithe.net)