Introduction to Jython

Frank Wierzbicki
note: some slides borrowed from Jim Baker

Me

- Java developer for 10+ years
- Python/Jython developer for 10+ years
- Jython contributor for the last 5 years
- Jython project lead for the last 3 years
- Sun Microsystems hired me to work full-time on Jython about 4 months ago

What is Python?

- A programming language that is:
  - Elegant and Robust
  - Easy to pick up: readability is at the forefront of the language design
  - Easy to use, yet powerful
  - The fastest growing language of 2007 according to http://www.tiobe.com

What is Jython?

- Jython brings the Python language to the JVM.
- Jython has full and nearly seamless integration into any Java libraries and code.
- Jython can access many of the libraries and frameworks written in Python.

Some code

```
print “hello world”

def hello(name):
print “hello”, name
```

Demo: Jython basics

Project Status

- Jython 2.2.1 is the production version
- Jython 2.5 alpha is out
- 2.5 Unit test compliance
  - 199 passing
  - 58 fails, but most are minor
- Active group of 8 committers

Applications

- Django - Leo Soto
- TurboGears 2 - Ariane Paola
- Zope - Georgy Berdyshev
- Twisted, ipython

And Existing Apps

- Admin - WebSphere, Weblogic, VMware
- Testing tools like PushToTest, Grinder
- Large and small companies

Demo: Jython basics

Jython DB access

- Jython has built in support for DB-API with zxjdbc: a thin wrapper around JDBC
- Provides Python programmers with access to any database with a JDBC driver.
- Provides Java programmers with access to any Python frameworks that are built on DB-API

“with” statement

```
with VAR as EXPR:
    BLOCK

#roughly translates into this:
VAR = EXPR
VAR.enter()
try:
    BLOCK
finally:
    VAR.exit()
```

transactional code

```
db = zxJDBC.connect(
  "jdbc:postgresql://localhost/world",
  'user', 'pass', "org.postgresql.Driver")

with db_transaction(db) as cursor:
    cursor.execute("select name from continent")
    rowdata = [row for row in cursor.fetchall()]
    print rowdata
```

context manager

```
@contextmanager
def db_transaction (connection):
  cursor = connection.cursor()
  try:
    yield cursor
  except:
    connection.rollback()
    raise
  else:
    connection.commit()
```

Jython and Swing

```
from javax.swing import JTable
from javax.swing import JFrame

rowdata = [('bill', 'Bill Williams')]
colnames = ['user name', 'full name']
table = JTable(rowdata, colnames)
frame = JFrame("Table")
frame.getContentPane().add( table )
frame.size = 400, 300
frame.visible = True
```

Demo: DB and Swing

Django

- A Python MVC framework with a web and database bias (similar to Ruby on Rails)
- Makes creating a project very simple
- Comes with a powerful admin tool that can be used to manage the data in your database -- no need to write your own admin tool!

Demo: Django

Java Platform

java.util.concurrent
not interpreted
no GIL

-J for setting options
like one of 20 or so GCs
heap size
etc

- Now takes advantage of Java 5 libraries
- Continue to
  - Compile to Java Bytecode
  - Use a choice of Java garbage collectors
  - Use Java native threads
  - -J flag namespace for JVM config

Python Platform

- _ast to use the same AST - almost there
- CPythonLib unit tests, augmented with Jython specific ones
- Most libraries are imported directly from the standard library (154 out of 191)

Deployment

virtualenv

portable bytecode vm?

of course, PBCs do vary over time, but fine for internal usage!

- setuptools, easy_install (only on trunk!)
- modjy + app servers like GlassFish
- Coming: JAR, WAR packaging
- Looking at: applets, Android
  - Bytecode gen in advance
  - No eval, use reflection to/from Java

Planned Integrations

- sqlite3 via NestedVM
- ctypes, either JNA or using PyPy's work
- PyAMF
- NumPy

PyAMF - probably via BlazeDS

NumPy - multidim array support in Java from UCAR; need also support for ufunc, broadcast/reduce, then ctypes too

Collaboration

spans the spectrum

- Major apps
- Implementations - CPython, PyPy
- Other Java dynamic languages - JRuby
- Tooling support
- Java Virtual Machine development
- Academic research - gradual typing

Tooling Support

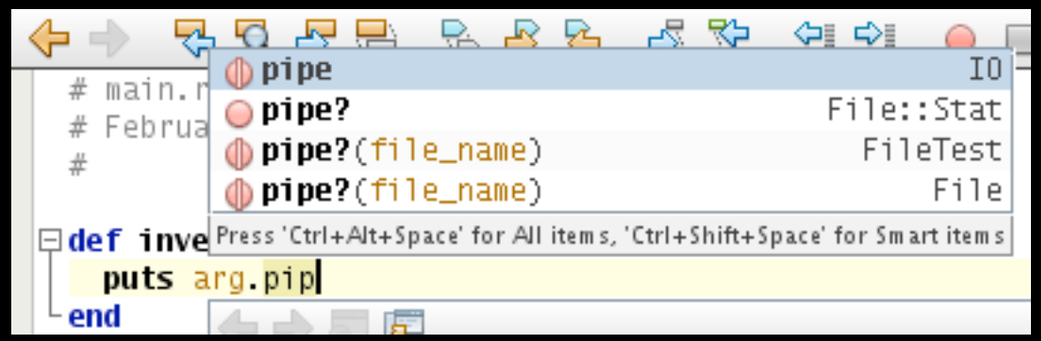
- IDEs
  - Especially integrated packaging - simple, standardized deployment options
  - Code completion
  - Possibilities of refactoring, etc.
  - Typing?

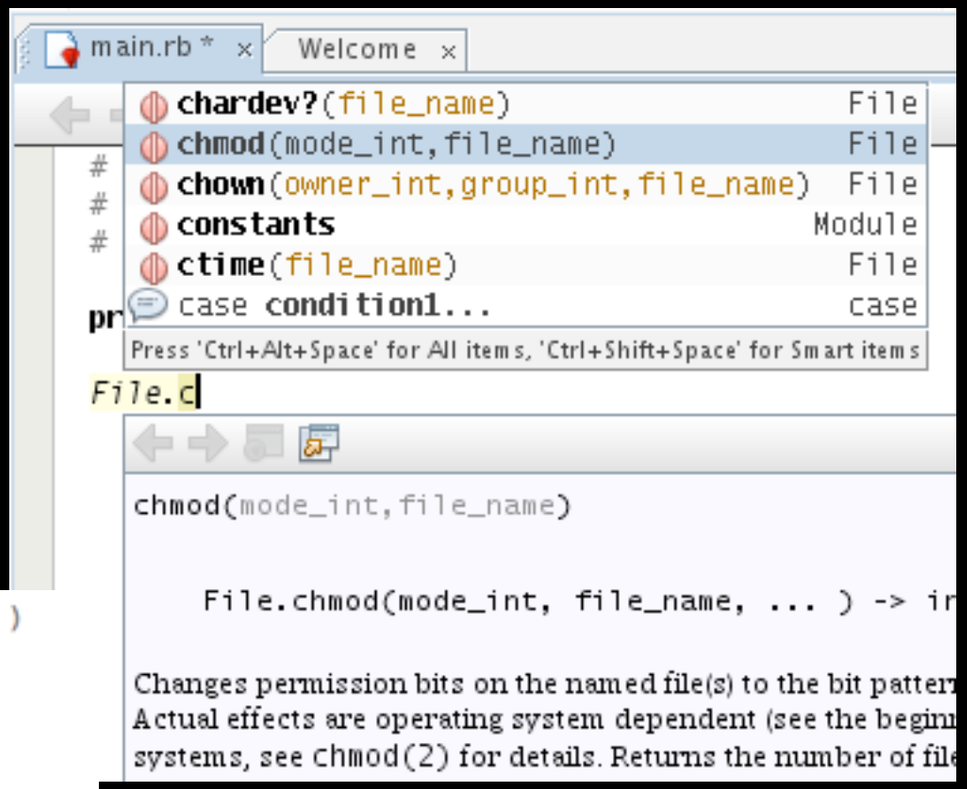
Tooling Support

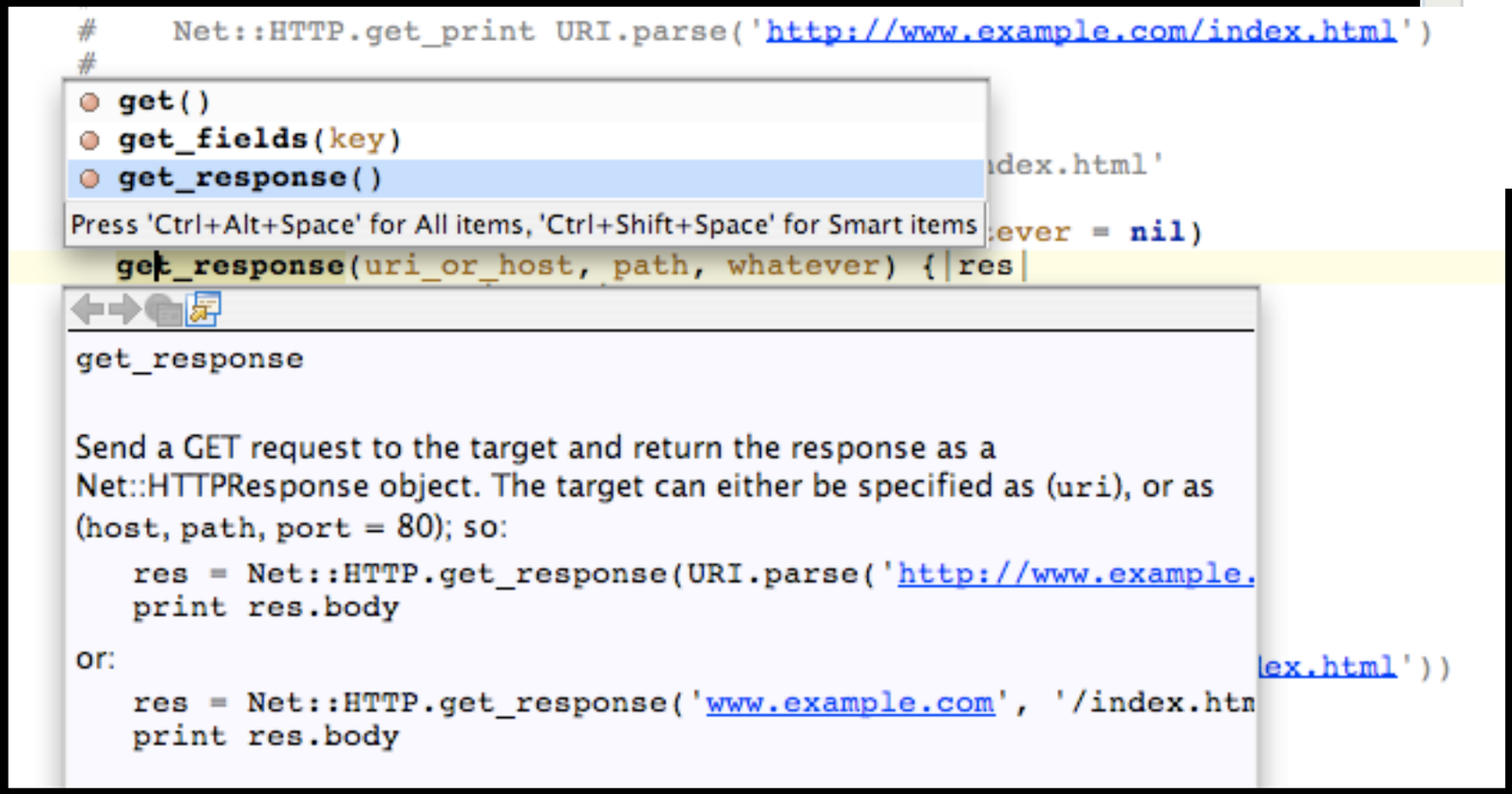
JRuby support in NB

- IDEs
  - Especially integrated packaging - simple, standardized deployment options
  - Code completion
  - Possibilities of refactoring, etc.
  - Typing?

JRuby in NB


pipe IO
pipe? File::Stat
pipe?(file_name) FileTest
pipe?(file_name) File
Press 'Ctrl+Alt+Space' for All items, 'Ctrl+Shift+Space' for Smart items
def inve
puts arg.pip
end


main.rb *
Welcome
chardev?(file_name) File
chmod(mode_int,file_name) File
chown(owner_int,group_int,file_name) File
constants Module
ctime(file_name) File
case condition1... case
Press 'Ctrl+Alt+Space' for All items, 'Ctrl+Shift+Space' for Smart items
File.c
chmod(mode_int,file_name)
File.chmod(mode_int, file_name, ...) -> in
Changes permission bits on the named file(s) to the bit patter
Actual effects are operating system dependent (see the begin
systems, see chmod(2) for details. Returns the number of fil


Net::HTTP.get_print URI.parse('http://www.example.com/index.html')
get()
get_fields(key)
get_response()
Press 'Ctrl+Alt+Space' for All items, 'Ctrl+Shift+Space' for Smart items
get_response(uri_or_host, path, whatever) {|res|
get_response
Send a GET request to the target and return the response as a
Net::HTTPResponse object. The target can either be specified as (uri), or as
(host, path, port = 80); so:
res = Net::HTTP.get_response(URI.parse('http://www.example.
print res.body
or:
res = Net::HTTP.get_response('www.example.com', '/index.htm
print res.body

Java 7 Support

one-shots get us greenlets

- invokedynamic and related functionality
- Relevant because of backported support
- Possible
  - One-shot continuations

More Demos

- Alice
- pydoclet

Where to find out more

- http://www.jython.org
- http://wiki.python.org/jython
- http://fwierzbicki.blogspot.com
- Twitter: fwierzbicki
- http://www.python.org
- http://nbpython.dev.java.net