PyCon 2012

CHINA

# PYTHON 产品构建与发布指南

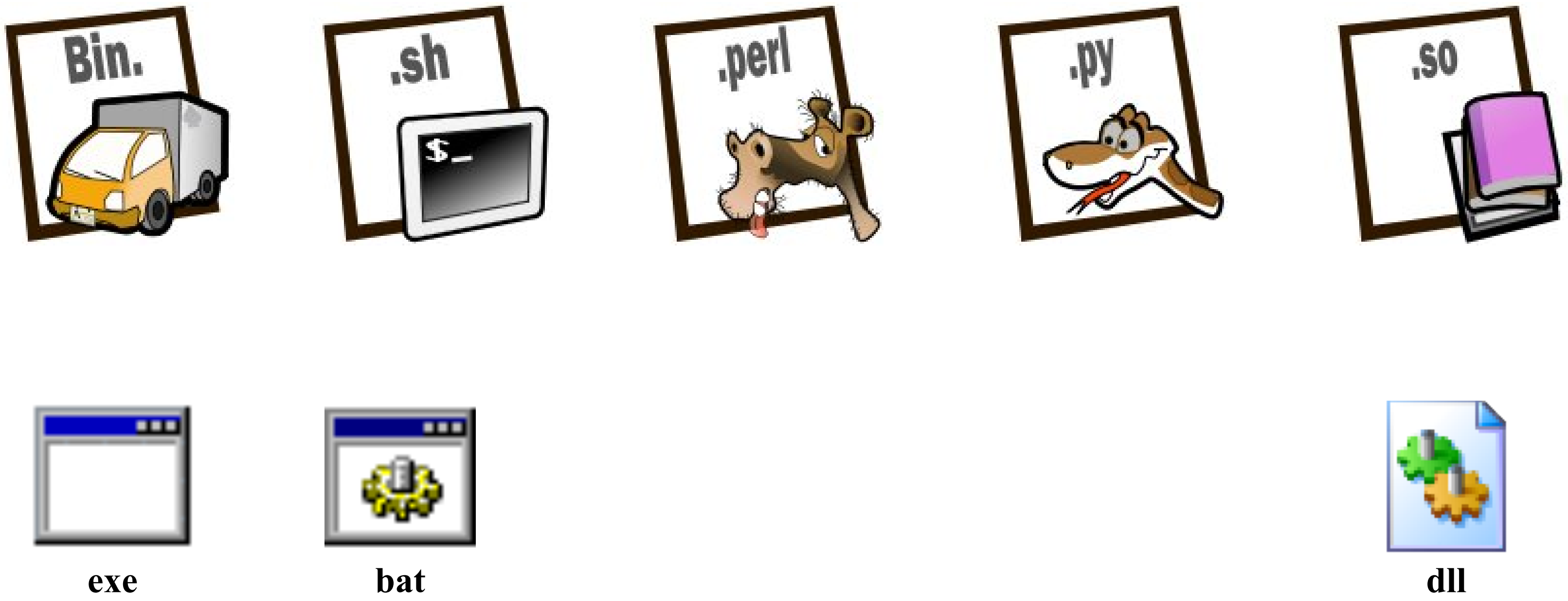
UNIX®
Celebrating 40 years uptime

iOS

# 一个完整的应用

Overview

| UNIX | Windows |
|---|---|
| /sbin | C:\Program Files |
| /usr/bin | |
| /etc | |
| /usr/lib | |
| /usr/include | |
| /usr/share | |
| /var | |


Bin.
.sh
$_
.perl
.py
.so
exe
bat
dll

## 编译为 “可执行程序”

Creating Executables

## Sample

```
#!/usr/bin/python2.7
# demo.py
def main():
    print 'hello world!'

if '__main__' == __name__:
    main()
```

py2exe

Cython

## C 语言版本

```
#include <stdio.h>

int main(int argc, char **argv)
{
    printf("hello world!\n");
    return 0;
}
```

## Cython = C + Python

```
# demo.pyx
cdef public int main(int argc, char **argv):
    print 'hello world!'
```

# demo.pyx（完整版）

```
cdef extern from 'Python.h':
    void Py_Initialize()
    void Py_Finalize()
    void PySys_SetArgv(int argv, char **argc)

cdef extern from 'demo.h':
    ctypedef struct PyMODINIT_FUNC:
        pass
    PyMODINIT_FUNC initdemo()

cdef public int main(int argc, char **argv):
    Py_Initialize()
    PySys_SetArgv(argc, argv)
    initdemo()
    print 'hello world!'
    Py_Finalize()
```

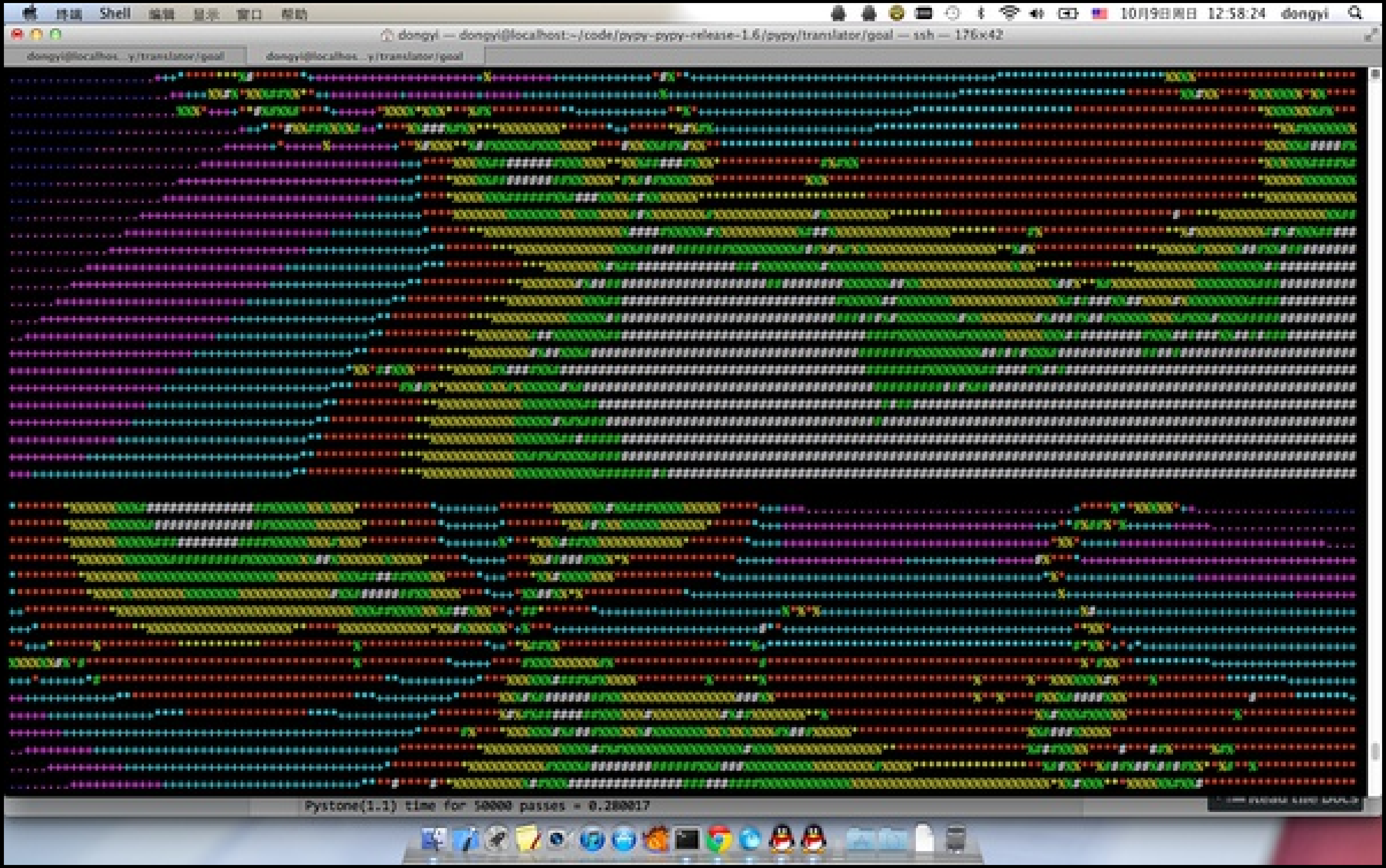
```
$ cython demo.pyx
$ ls
demo.c  demo.h  demo.pyx
$ gcc -I/usr/include/python2.7 -lpython2.7 demo.c -o demo
$ ./demo
hello world!
```

pypy

## demo.py（PyPy版）

```
def main(argv):
    print 'hello world!'
    return 0

def target(drv, args):
    return main, None

if '__main__' == __name__:
    import sys
    main(sys.argv)
```

```
$ python demo.py
hello world!
$ python pypy/translator/goal/translate.py demo.py
$ ./demo-c
hello world!
```


终端 Shell 编辑 显示 窗口 帮助
10月9日周日 12:58:24 dongyi
dongyi — dongyi@localhost:~/code/pypy-pypy-release-1.6/pypy/translator/goal — ssh — 176×42
dongyi@localhos…y/translator/goal
dongyi@localhos…y/translator/goal

```
Pystone(1.1) time for 50000 passes = 0.280017
```

PYPY 的编译姿势尼玛是有多蛋疼

—— 碧海潮生

二进制

代码加密

嵌入式

跑得更快

## Tutorial: Writing an Interpreter with PyPy

## 库

Lib

py2exe

Cython

## 前情回顾

```
cdef public int main(int argc, char **argv):
    print 'hello world!'
```

## libsum.pyx

```
cdef public int sum(int a, int b):
    c = a + b
    return c
```

## test.c

```
#include <Python.h>
#include "libsum.h"

int main(int argc, char **argv)
{
    Py_Initialize();
    PySys_SetArgv(argc, argv);
    initlibsum();
    int c = sum(11, 7);
    printf("%d\n", c);
    Py_Finalize();
    return 0;
}
```

```
$ cython libsum.pyx
$ ls
libsum.c  libsum.h  libsum.pyx  test.c
$ gcc -fPIC -I/usr/include/python2.7 -shared libsum.c -o libsum.so
$ ls
libsum.c  libsum.h  libsum.pyx  libsum.so  test.c
$ gcc -I/usr/include/python2.7 -L. -lpython2.6 -lsum test.c -o test
$ export LD_LIBRARY_PATH=$(pwd)
$ ./test
18
```

## Embedding？

# Python 对象跨语言传递方式之一

```
class Foo:
    pass

cdef public int new_foo():
    foo = Foo()
    id1 = id(foo)
    objects[id1] = foo
    return id1

cdef public int delete_foo(int id1):
    del objects[id1]

cdef public void print_foo(int id1):
    print objects[id1]

objects = {}
```

pypy

## Standalone

```
def target(driver,
args):
    return
entry_point, [...]
```

## Shared

```
def target(driver, 
args):
    return [
        (entry_point1, 
[...]),
        (entry_point2, 
[...]),
        (entry_point3, 
[...]),
                           ...
    ]
```

## 增加对多个 entry_point 的支持

## /pypy/translator/c/genc.py

```
class CExtModuleBuilder:
    def getentrypointptr(self):
        return [new_wrapper(ep, self.translator)  for ep, _ in spec]

    def compile(self):
        export_symbols = [self.db.get(ep)  for ep in \
            self.getentrypointptr()] + ['RPython_StartupCode']
        if self.config.translation.countmallocs:
            export_symbols.append('malloc_counters')
        extsymeci = ExternalCompilationInfo(export_symbols=export_symbols)
        self.eci = self.eci.merge(extsymeci)
        files = [self.c_source_filename] + self.extrafiles
        self.translator.platform.compile(files, self.eci,
standalone=False)
        self._compiled = True

    def get_entry_point(self, isolated=False):
        return drv.extmod_name
```

## /pypy/translator/driver.py

```
class TranslationDriver:
    def task_annotate(self):
        translator = self.translator
        policy = self.policy
        annmodel.DEBUG = self.config.translation.debug
        annotator = translator.buildannotator(policy=policy)
        for func, inputtypes in self.secondary_entrypoints:
            if inputtypes == Ellipsis:
                continue
            rettype = annotator.build_types(func, inputtypes, False)
        annotator.complete()
        self.sanity_check_annotation()
        annotator.simplify()

    task_annotate = taskdef \
        (task_annotate, [], "Annotating&simplifying")
```

## libsum.py

```
def sum(a, b):
    c = a + b
    return 0

def target(drv, args):
    return [(sum, [int, int])]
```

## sum.h

```
#define sum(a, b) \
    pypy_g_sum((a), (b));

int pypy_g_sum(int, int);
```

## test.c

```
#include <stdio.h>
#include "sum.h"

int main()
{
    int c = sum(11, 7);
    printf("%d\n", c);
    return 0;
}
```

## 共享库生成位置

**shared_library_name** = driver.c_entryp + '.' + t.platform.so_ext
**dest** = driver.cbuilder.targetdir.join(**shared_library_name**).strpath

```
$ python translate.py --cflags="-I/usr/include/python2.7" libsum.py
$ ls
libsum.py  libsum.so  sum.h test.c
$ gcc -I/usr/include/python2.7 -L. -lpython2.7 -lsum test.c -o test
$ export LD_LIBRARY_PATH=$(pwd)
$ ./test
18
```

## libsum.py

```
def sum(a, b):
    c = a + b
    return 0

def target(drv, args):
    return [(sum, [int, int])]
```

## libsum.pyx

```
cdef sum(int a, int b):
    cdef c = a + b
    return 0
```

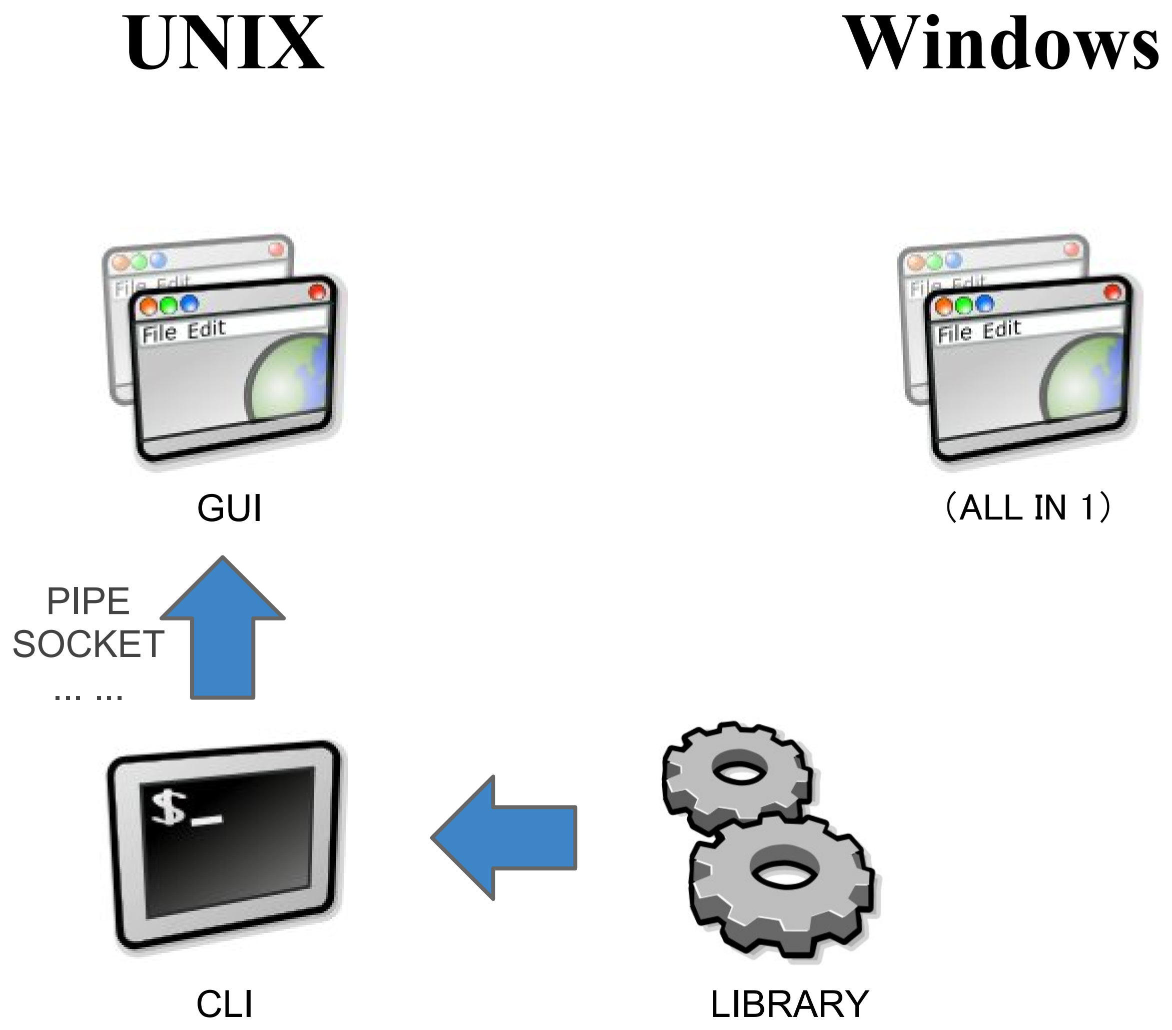
二进制　　代码加密　　嵌入式　　跑得更快

......

## 跨语言库

## 界面

User Interface



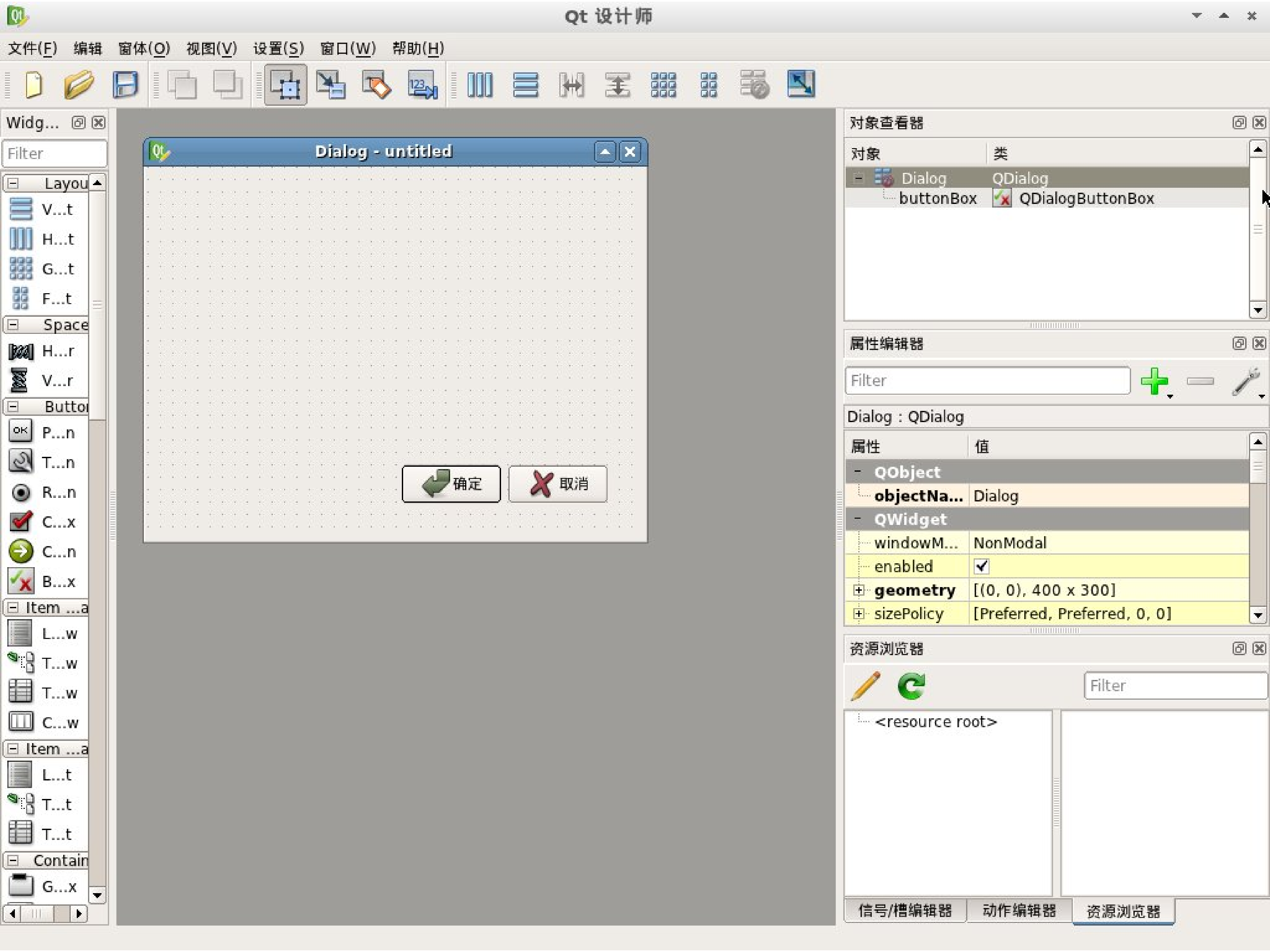
注:开发者取向

Tk

PySide
Python for Qt

wxPython

pygame



| 对象 | 类 |
|---|---|
| Dialog | QDialog |
| buttonBox | QDialogButtonBox |

Dialog : QDialog

| 属性 | 值 |
|---|---|
| QObject | |
| objectNa... | Dialog |
| QWidget | |
| windowM... | NonModal |
| enabled | ✓ |
| geometry | [(0, 0), 400 x 300] |
| sizePolicy | [Preferred, Preferred, 0, 0] |

Tk

PySide
Python for Qt

wxPython

pygame

# Beginning Game Development with Python and Pygame – From Novice to Professional

用 Python 和 PyGame 写游 戏 – 从入门到精通

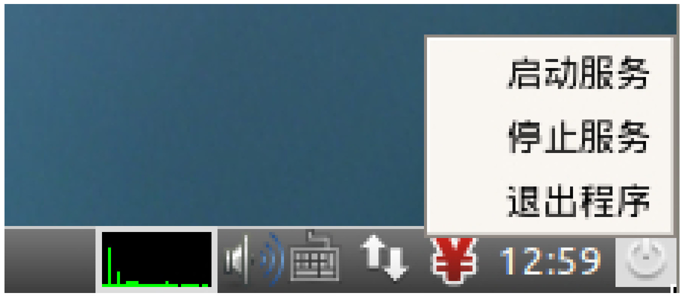
## 服务

Service

## pywin32

win32serviceutil.
ServiceFramework

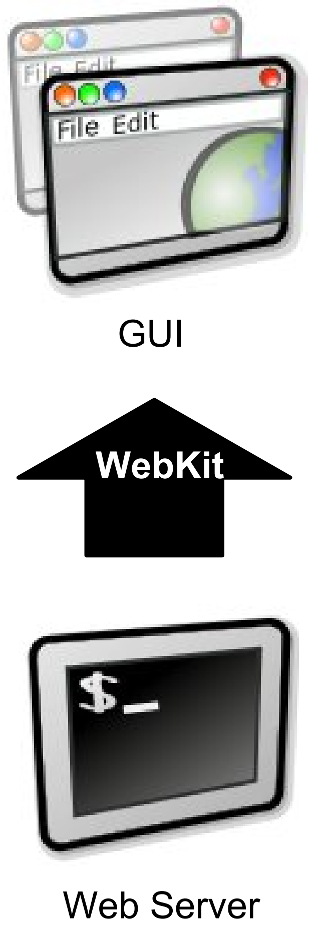
## Systray ?

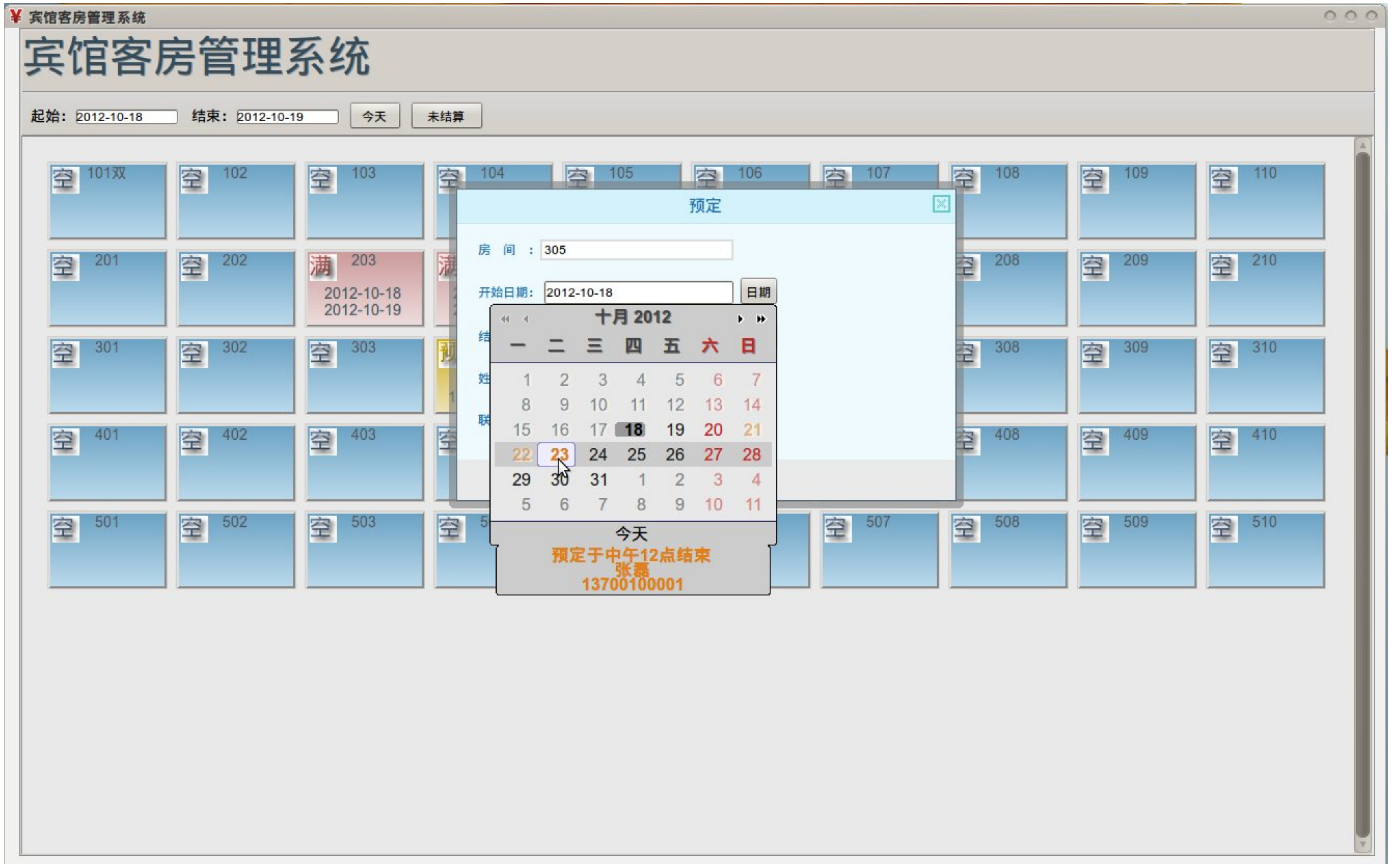

启动服务
停止服务
退出程序
12:59

```
from PySide import QtCore, QtGui
class Dialog(QtGui.QDialog):
    def __init__(self):
        QtGui.QDialog.__init__(self)
        menu = QtGui.QMenu(self)
        menu.addAction(QtGui.QAction(
            u'退出服务’, self,
            triggered=QtGui.qApp.quit))
        tray = QtGui.QSystemTrayIcon(self)
        tray.setIcon(QtGui.QIcon('￥.png’))
        tray.setContextMenu(menu)
        tray.show()
```

```
def serve_forever():
    def app(environ, start_response):
        start_response('200 OK',
            [('Content-Type', 'text/plain')])
        return ['hello world!']
    from wsgiref import simple_server
    server = simple_server.make_server('', 8000, app)
    server.serve_forever()

if '__main__' == __name__:
    import sys, thread
    app = QtGui.QApplication(sys.argv)
    dialog = Dialog()
    thread.start_new_thread(serve_forever, ())
    sys.exit(app.exec_())
```

## Web Style !


GUI
WebKit
$_
Web Server



|一|二|三|四|五|六|日|
|---|---|---|---|---|---|---|
|1|2|3|4|5|6|7|
|8|9|10|11|12|13|14|
|15|16|17|18|19|20|21|
|22|23|24|25|26|27|28|
|29|30|31|1|2|3|4|
|5|6|7|8|9|10|11|

起始: 2012-10-18 结束: 2012-10-19

房 间 : 305

开始日期: 2012-10-18

## Duck-Typing

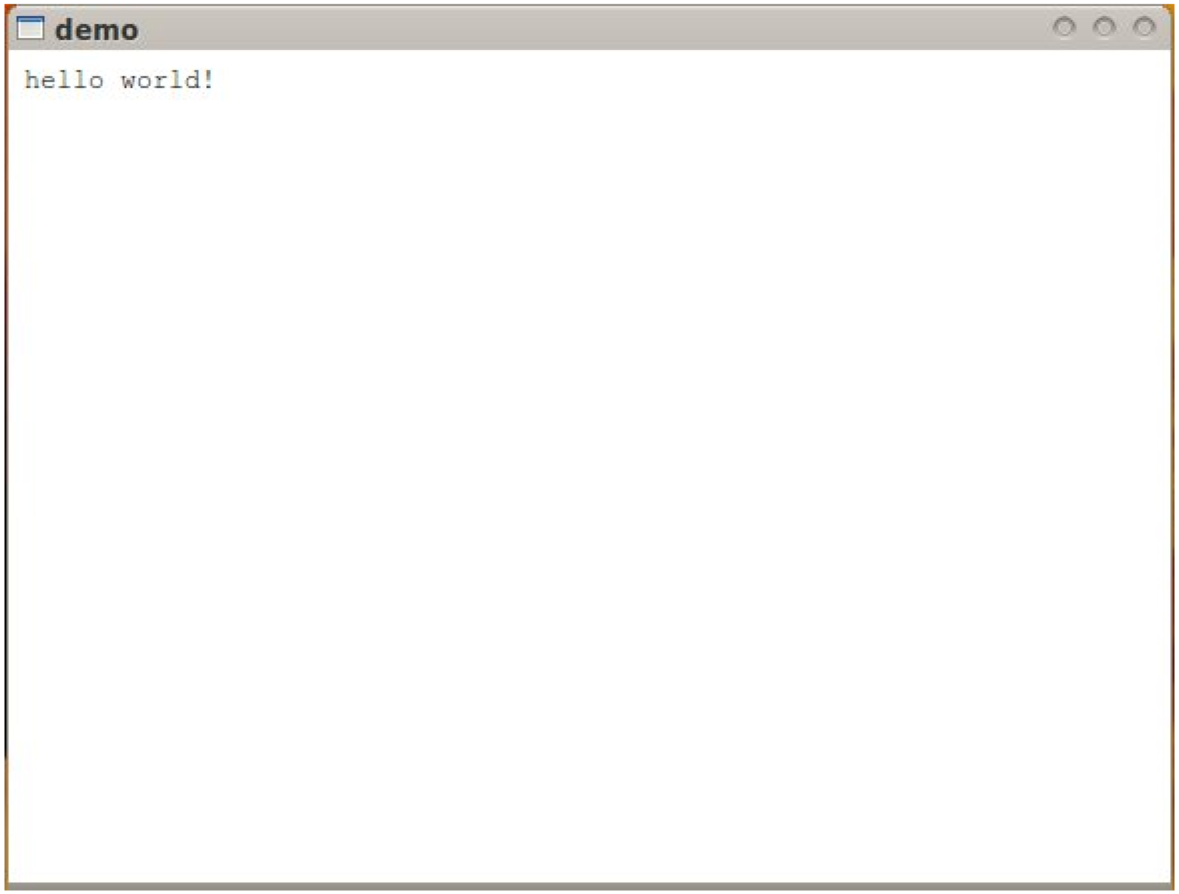
# 前情回顾

```
from PySide import QtCore, QtGui
class Dialog(QtGui.QDialog):
    def __init__(self):
        QtGui.QDialog.__init__(self)
        menu = QtGui.QMenu(self)
        menu.addAction(QtGui.QAction(
            u'退出服务’, self,
            triggered=QtGui.qApp.quit))
        tray = QtGui.QSystemTrayIcon(self)
        tray.setIcon(QtGui.QIcon('￥.png’))
        tray.setContextMenu(menu)
        tray.show()
```

```
def serve_forever():
    def app(environ, start_response):
        start_response('200 OK',
            [('Content-Type', 'text/plain')])
        return ['hello world!']
    from wsgiref import simple_server
    server = simple_server.make_server('', 8000, app)
    server.serve_forever()

if '__main__' == __name__:
    import sys, thread
    app = QtGui.QApplication(sys.argv)
    dialog = Dialog()
    thread.start_new_thread(serve_forever, ())
    sys.exit(app.exec_())
```

```
from PySide import QtCore, QtGui
from PySide import QtWebKit
class Browser(QtWebKit.QWebView):
    def __init__(self):
        QtWebKit.QWebView.__init__(self)

        menu = QtGui.QMenu(self)
        menu.addAction(QtGui.QAction(
            u'退出程序’, self,
            triggered=QtGui.qApp.quit))
        tray = QtGui.QSystemTrayIcon(self)
        tray.setIcon(QtGui.QIcon('￥.png’))
        tray.setContextMenu(menu)
        tray.show()

        self.load(QtCore.QUrl(
            u'http://127.0.0.1:8000/'))
        self.show()
```

```
def serve_forever():
    def app(environ, start_response):
        start_response('200 OK',
            [('Content-Type', 'text/plain')])
        return ['hello world!']
    from wsgiref import simple_server
    server = simple_server.make_server('', 8000, app)
    server.serve_forever()

if '__main__' == __name__:
    import sys, thread
    app = QtGui.QApplication(sys.argv)
    browser = Browser()
    thread.start_new_thread(serve_forever, ())
    sys.exit(app.exec_())
```



```
hello world!
```

# OnBeforeNavigate

直接与 UI 端通信

`<a href="`**`my://do some stuff`**`">hello world!</a>`

# AJAX

将任务发送给本地服务器

## JAVASCRIPT ↔ PYTHON

## XUL、QML vs HTML

```python
# libsum.py

def sum(a, b):
    return a + b
```

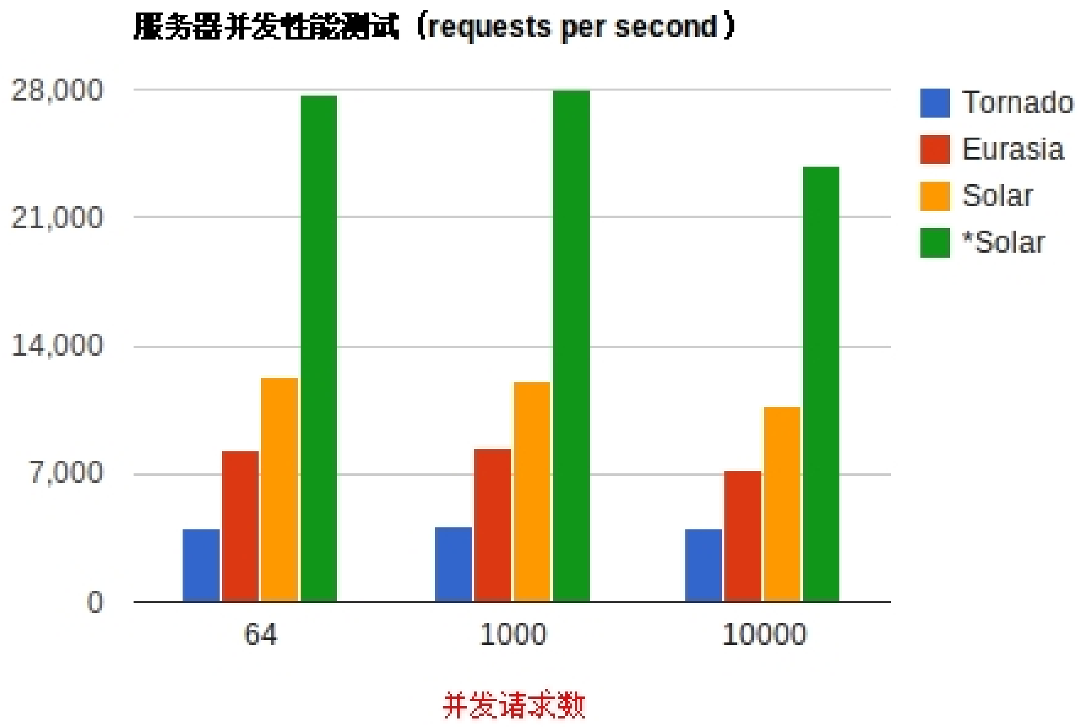
```html
<html>
<head>
<title>test</title>
</head>
<body>
<script>
var c = libsum.sum(11, 7);
alert(c);  // 18!
</script>
</body>
</html>
```

## Liubao

www.liubao.com
code.google.com/p/liubao

## Eurasia

code.google.com/p/eurasia



AMD Athlon(tm) II X2 250 Processor

## Eurasia 3.2

File IO、Pipe、PostgreSQL、Durus

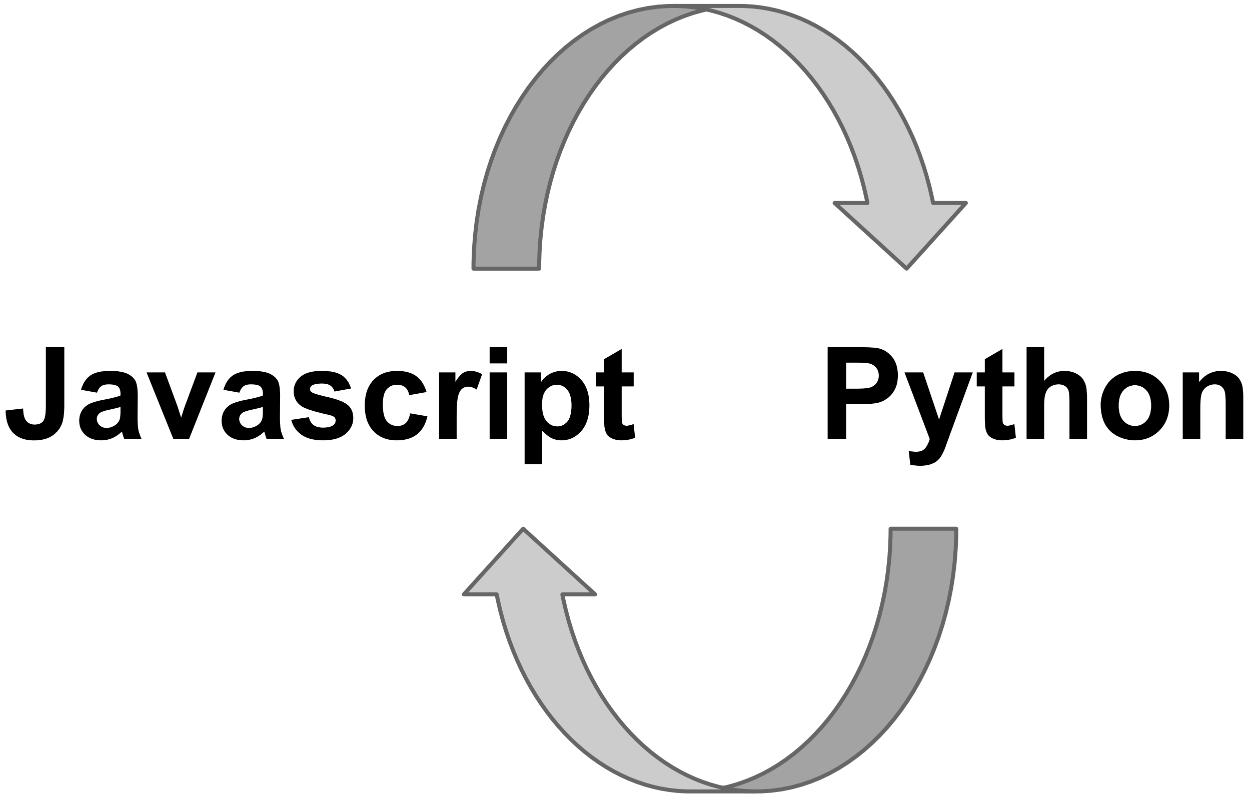
……

## Liubao 1.0

```python
# /www/demo/libsum.py

def sum(self, a, b):
    return a + b
```

```html
<html>
<head>
<title>file:/www/index.html</title>
<script src="__api__.js"></script>
</head>
<body>
<script>

var mysite = connect('http://mysite/');
var c = mysite.demo.libsum.sum(11, 7);
alert(c);  // 18!

</script>
</body>
</html>
```

$$\begin{tikzcd}\textbf{Javascript} \arrow[r, bend left] & \textbf{Python}\end{tikzcd}$$

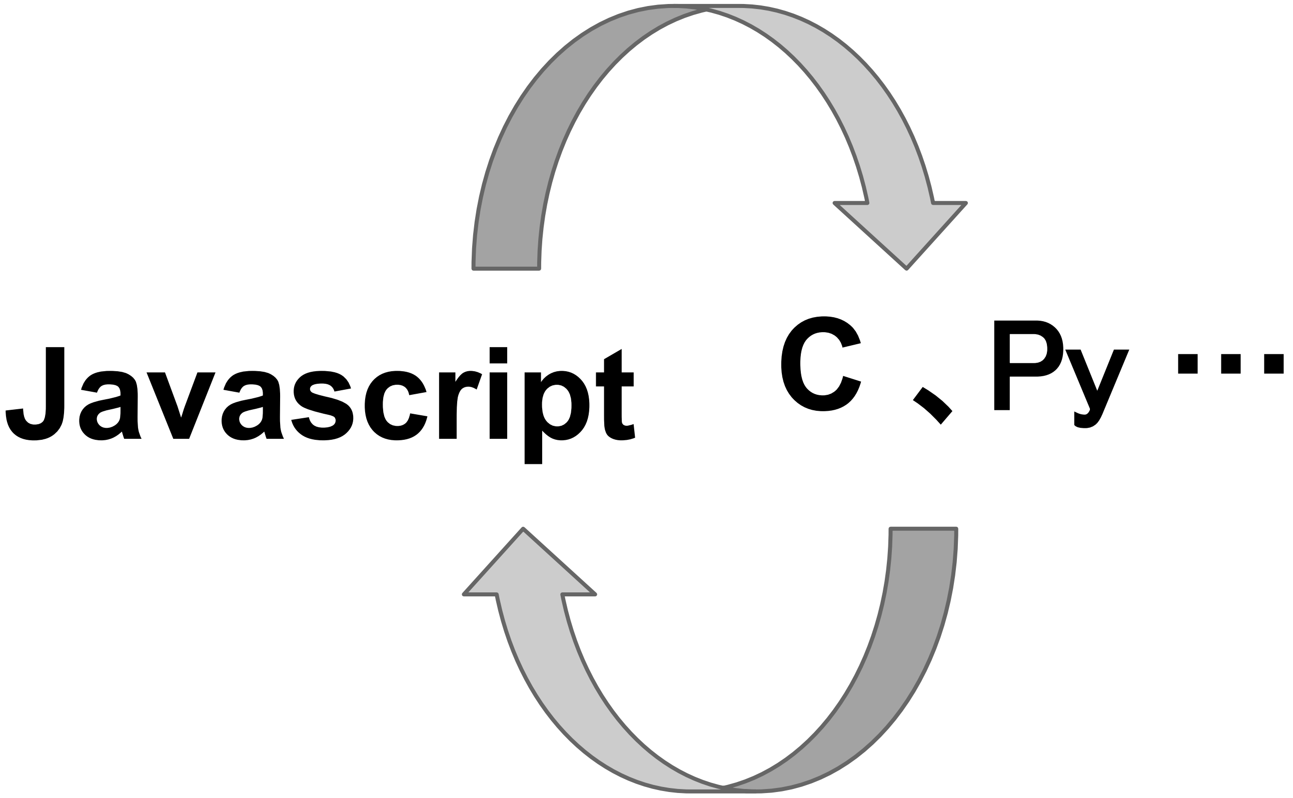

Javascript
Python

```python
# /www/demo/libsum.py

def server_sum(self, a, b):
    c = self.client_sum(a + b)
    return c
```

```html
<html>
<head>
<title>file:/www/index.html</title>
<script src="__api__.js"></script>
</head>
<body>
<script>

var client_sum = function(a, b) {
    return a + b;
};

var mysite = connect('http://mysite/');
var c = mysite.demo.libsum.server_sum(11, 7);
alert(c);  // 18!

</script>
</body>
</html>
```

```
# /www/demo/libsum.so

def server_sum(self, a, b):
    c = self.client_sum(a + b)
    return c
```

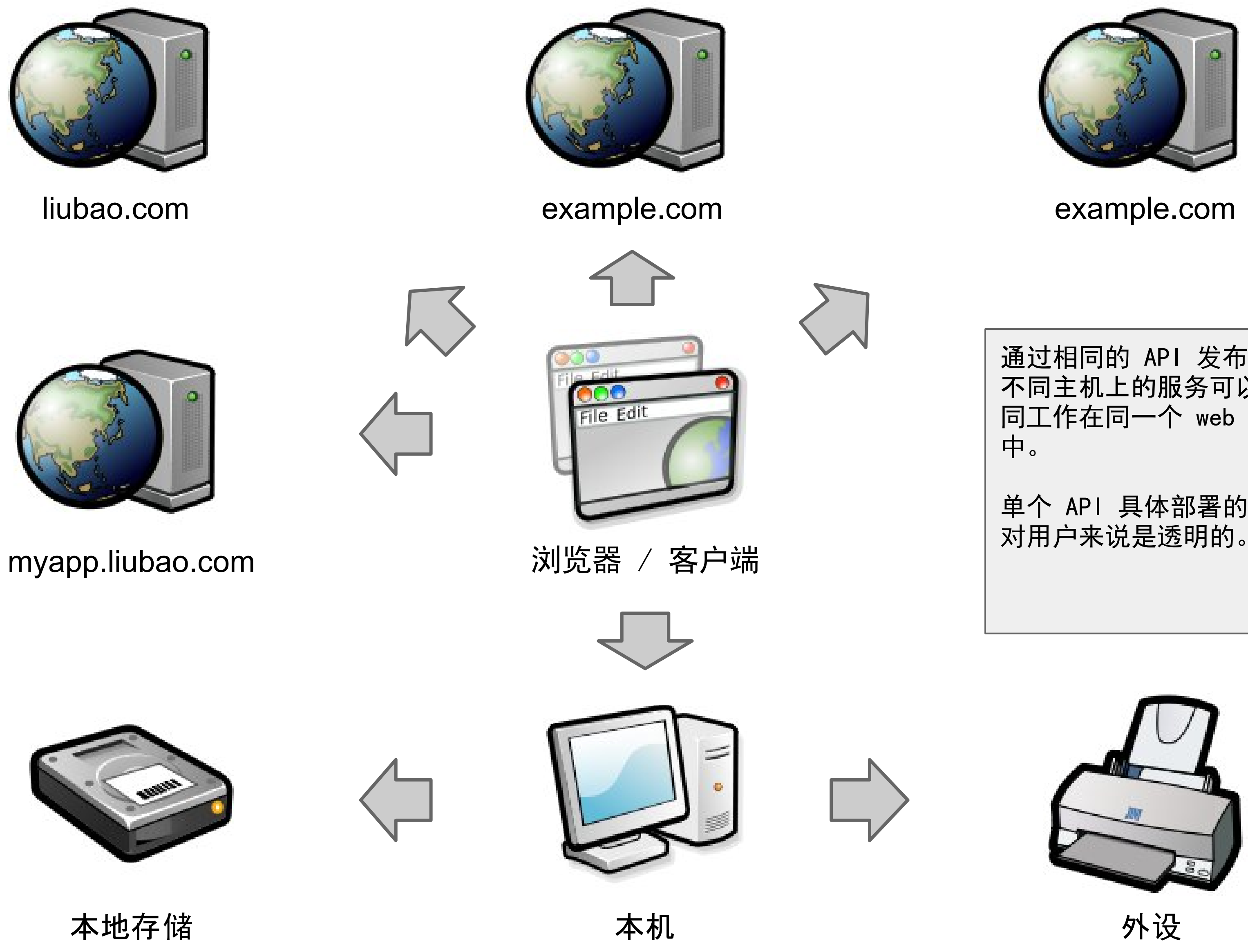
```
<html>
<head>
<title>file:/www/index.html</title>
<script src="__api__.js"></script>
</head>
<body>
<script>

var client_sum = function(a, b) {
    return a + b;
};

var mysite = connect('http://mysite/');
var c = mysite.demo.libsum.server_sum(11, 7);
alert(c);  // 18!

</script>
</body>
</html>
```


Javascript
C 、Py …

## 双向跨域

可直接嵌入当前网站

```
<html>
<head>
<title>file:/www/index.html</title>
</head>
<body>
<script>

var pycon  = connect('http://www.pycon.org/'),
    liubao = connect('http://www.liubao.com/'),
    c =  pycon.libsum.sum(11, 7),
    d = liubao.libpow.pow(c, 2);

alert(d); // 324!

</script>
</body>
</html>
```



通过相同的 API 发布协议不同主机上的服务可以协同工作在同一个 web 页面中。

单个 API 具体部署的位置对用户来说是透明的。

## 依赖

```
HTTP/1.0
JavaScript
```

## 语言瓶颈

## JS 缺少 __getattr__ 、*args 、**kwargs

```
var api = connect('http://www.pycon.org/');
var c = api.libsum.sum(11, 7);
```

```
var call = connect('http://www.pycon.org/').
submit;
var c = call('/libsum/sum', [11, 7], {});
```

## JS 有 async 无 coroutine

```
var api = connect('http://www.pycon.org/');
var c = api.libsum.sum(11, 7);
alert(c);  // 18!
```

⬇

```
var api = connect('http://www.pycon.org/').submit;
api('/libsum/sum', [11, 7], {}, function(c) {
    alert(c);  // 18!
});
```

## 解决方案

PyJs 、NodeJs …

## 综合应用

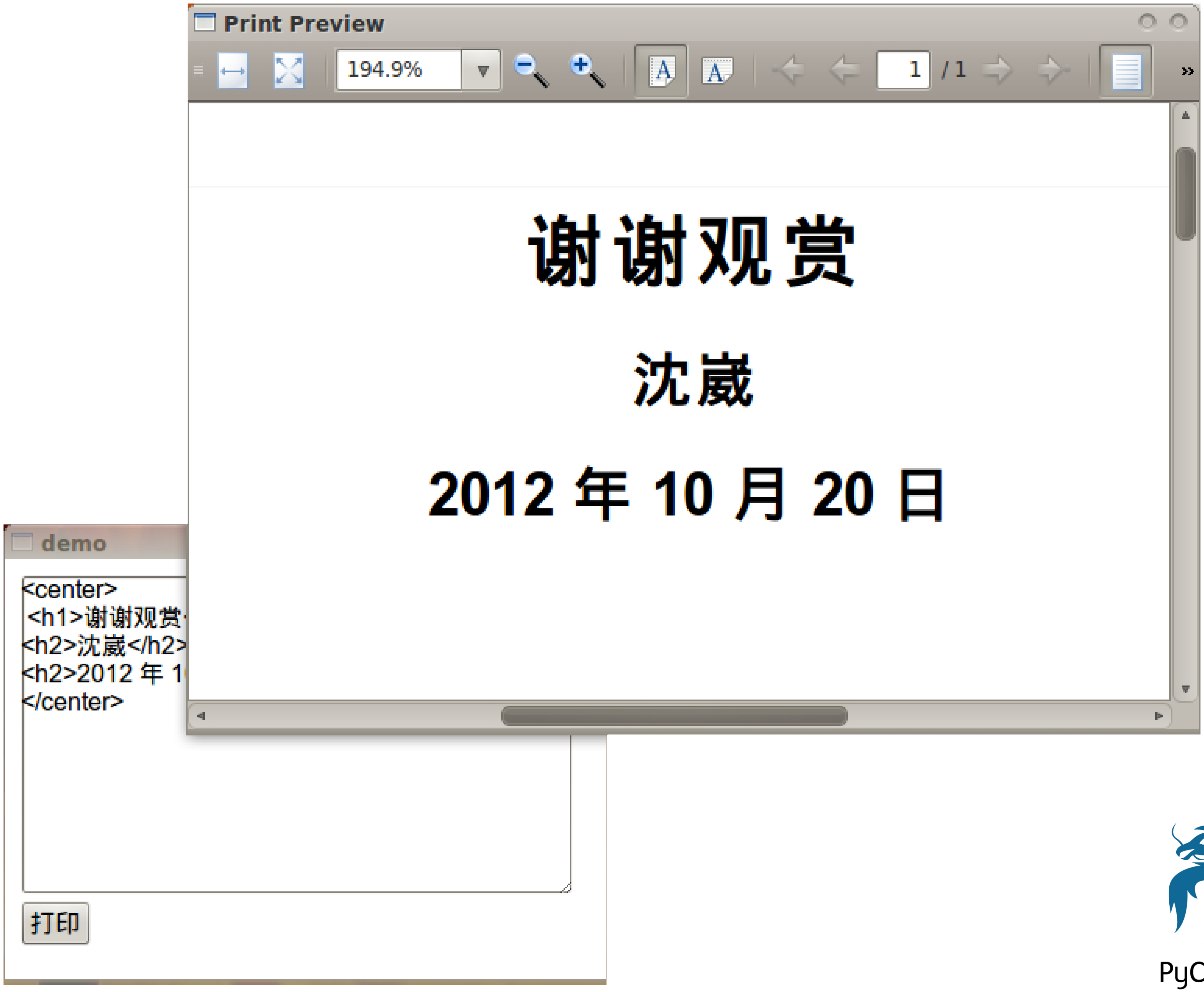
# 主程序

```
from PySide import QtCore, QtGui, QtWebKit

class Window(QtWebKit.QWebView):
    def __init__(self):
        QtWebKit.QWebView.__init__(self)
        menu = QtGui.QMenu(self)
        menu.addAction(QtGui.QAction(
            u'退出程序', self,
            triggered=QtGui.qApp.quit))
        tray = QtGui.QSystemTrayIcon(self)
        tray.setIcon(QtGui.QIcon('￥.png'))
        tray.setContextMenu(menu)
        tray.show()
        self.print_signal.connect(self.
_print_)
        self.load(QtCore.QUrl(
            u'http://127.0.0.1:8080/'))
        self.show()

    def print_(self, s):
        self.print_signal.emit(s)

    def _print_(self, s):
        doc = QtWebKit.QWebView()
        doc.setHtml(s)
        printer = QtGui.QPrinter()
        dialog = QtGui.QPrintPreviewDialog
(printer)
        dialog.paintRequested.connect(doc.
print_)
        dialog.exec_()

    print_signal = QtCore.Signal(object)
```

```
def main():
    import sys, thread
    app = QtGui.QApplication(sys.argv)
    sys.modules['config'] = {'window':
Window()}

    from liubao import httpd
    servers, serveforever = initialize
('httpd.conf')
    thread.start_new_thread
(serveforever, ())

    sys.exit(app.exec_())

if '__main__' == __name__:
  main()
```

```html
<html>
<head>
<meta charset="utf-8">
<title>file: /www/index.html</title>
<script src="__api__.js"></script>
</head>
<body>
<textarea id="text">
</textarea><br />
<button onclick="print">打印</button>
<script>
var api = connect('http://127.0.0.1:8080/').
submit;
var print = function() {
    var doc = document.getElementById('text').
value;
    api('/print_/print_', [doc]);
};
</script>
</body>
</html>
```

```python
# file: /www/print_.py

import config
def print_(self, doc):
    config['window'].print_(doc)
    self.alert('打印完毕！’)
```



# 谢谢观赏

## 沈崴

## 2012 年 10 月 20 日

```
<center>
 <h1>谢谢观赏
<h2>沈崴</h2>
<h2>2012 年 1
</center>
```

PyCon 2012 CHINA