

We interrupt this program to annoy you  
and make things generally irritating –  
Python

Peter Meerwald

2nd May 2007

# Plan

April 26 – Introduction (Concepts, data types, . . . )

May 3 – CGI

May 10 – Strings, pattern matching & regular expressions

May 24 – Data structures, classes, exceptions

May 31 – Misc.

June 14 – Interface a database

June 31 – Web framework (Django)

# About...

- <http://www.python.org>
- Created circa 1990 by Guido van Rossum, Benevolent Dictator for Life
- Simple, intuitive, dynamic language with OO and functional elements
- Open source, current version 2.5, implementations in C, Java (Jython), .net (IronPython) and Python (PyPy)

# What to do with Python?

Scripts, CGI, glue code, embedded language, . . .

Zope (Application server)

Plone (Content Management System)

Django, TurboGears (Framework)

Mercurial (Source repository)

SCons (Build control)

MailMan (Mail list administration)

NumPy & SciPy (Scientific computation)

youtube.com, google.com

# hello, world

```
#!/usr/bin/env python
def say_hello():
    """Just prints 'hello, world'.
    Know what Python refers to, BTW?"""
    print 'hello, world'

say_hello()
```

# Overview

- Basics (data types, typing, flow control, function, modules)
- Data structures (list, tuple, dictionary)
- Strings
- Classes
- Standard Library
- Exceptions

## Data types

```
s = 'this is "quoted"'
```

```
t = 'new\nline'
```

```
i = 10
```

```
b = True
```

```
j = 3.14
```

```
tuple = (1,2)
```

```
list = ['a', 'b', 3, ('d', 5)]
```

```
dictionary = {'To': 'guido', 'length': 128}
```

# Flow Control

```
i = 0
while True:
    if i == 0 or i == 1:
        print 'zero or one'
    elif i < 10: print 'small'
    elif i == 15: break
    else:
        print 'big'
    i += 1
print 'i = %d' % (i)
```

# Functions

```
def f(i):  
    pass  
def g(x=[]):  
    return len(x)  
  
print g('blabla')  
print g()  
print g(['b', 'l', 'a'])
```

# Modules

```
import sys  
sys.exit(0)
```

---

```
from sys import exit  
exit(0)
```

---

```
from sys import *  
exit(0)
```

---

## (Duck) Typing

“If it walks like a duck and quacks like a duck, it must be a duck”

- Emphasize interface over type, allow polymorphism
- Avoid type checks, catch an exception if it fails

```
>>> def times10(x):
```

```
    return x * 10
```

```
>>> times10(3)
```

```
30
```

```
>>> times10('-')
```

```
-----
```

## Lists

```
>>> l = ['spam', 'eggs', 3.14, 42]
>>> l[0]
'spam'
>>> l[-2]
3.14
>>> l[1:]
['eggs', 3.14, 42]
>>> l[:2] + ['bacon', 'pepper']
['spam', 'eggs', 'bacon', 'pepper']
>>> len(l)
4
>>> l.append(100)
['spam', 'eggs', 3.14, 42, 100]
```

## List Comprehension / Looping

```
>>> v = [1,2,3,4]
```

```
>>> len(v)
```

```
4
```

```
>>> [x*x for x in v]
```

```
[1, 4, 9, 16]
```

```
>>> [x*x for x in v if x > 2]
```

```
[9,16]
```

```
>>>for i in xrange(1,10,2): print i
```

```
1,3,5,7,9
```

## Dictionaries

```
>>> weight = {'galahad': 78, 'lancelot': 80, 'arthur': 101}
>>> weight['arthur']
101
>>> weight.has_key('bjelli')
False
>>> weight['god'] = 0
>>> weight.keys()
['god', 'arthur', 'galahad', 'lancelot']
```

# Extending a Web Server with Python

- CGI - create new process handling one request
- Fast CGI - process(es) handling many requests
- `mod_python` - interpreter running embedded in the web server
- Web Server in Python (`BaseHTTPServer`, `Twisted`, etc.)

# Fast CGI

Instead of creating a new process for every request, persistent process(es) handle many requests, communication over sockets.

By Open Market, Inc., circa 1996: <http://www.fastcgi.com>

Wide adoption, but no standard.

Need language binding, eg. <http://jonpy.sourceforge.net/fcgi.html>

---

```
import jon.cgi as cgi
import jon.fcgi as fcgi
class Handler(cgi.Handler):
    def process(self, req):
        req.set_header("Content-Type", "text/plain")
        req.write("Hello, world!\n")
fcgi.Server({fcgi.FCGI_RESPONDER: Handler}).run()
```

# mod\_python

```
LoadModule python_module libexec/mod_python.so
<Directory /some/directory/htdocs/test>
    AddHandler mod_python .py
    PythonHandler mptest
    PythonDebug On
</Directory>
```

---

```
from mod_python import apache
def handler(req):
    req.write("Hello World!")
    return apache.OK
```

# Server in Python

```
import BaseHTTPServer
import CGIHTTPServer
httpd = BaseHTTPServer.HTTPServer(\
    ('localhost', 8123), \
    CGIHTTPServer.CGIHTTPRequestHandler)
httpd.serve_forever()
```

# Common Gateway Interface (CGI)

Input:

- Environment variables (QUERY\_STRING, CONTENT\_LENGTH, REQUEST\_METHOD, REMOTE\_ADDR, request header lines as HTTP\_XXX)
- Command-line arguments (not used)
- stdin (for HTTP POST)

Output:

- Response to stdout

```
print 'Content-type: text/html\n'
```

```
print '<html>...'
```
- Logging to stderr (may appear in web server's log file)

## cgibtb and cgi Module

```
import cgitb
cgitb.enable() # special exception handler, formatted report
import cgi
form = cgi.FieldStorage()
name = form.getvalue('name') # attention! can return a list...
addr = form.getfirst('addr', 'N/A')
print cgi.escape(addr) # converts &, <, > to HTML-safe sequences
```

## FieldStorage Details

```
import cgi
form = cgi.FieldStorage() # only instantiate once
form.name
form.type # eg. application/x-www-form-urlencoded
form.headers # dictionary of all headers
form['name'].value
form['name'].file # file handle
```

# Environment Variables

```
import os
os.environ['HTTP_ACCEPT']
# text/xml,application/xml,application/xhtml+xml,...
# text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
os.environ['REMOTE_ADDR']
# 127.0.0.1
os.environ['REQUEST_METHOD']
# GET
os.environ['HTTP_USER_AGENT']
# Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.7.10) ...
```

# File Upload

```
<html><body>  
<form enctype="multipart/form-data" action="fileupload.py" method="post">  
<p>File: <input type="file" name="file"></p>  
<p><input type="submit" value="Upload"></p>  
</form></body></html>
```

---

```
form = cgi.FieldStorage()  
fileitem = form['file']  
# strip leading path from file name to avoid directory traversal attacks  
fn = os.path.basename(fileitem.filename)  
open('files/' + fn, 'wb').write(fileitem.file.read())
```

# CGI Scripts in Python

<http://www.imladris.com/Scripts/PythonForWindows.html>