Python Standard Library

: Python

10/06/07 20:10:08

"We'd like to pretend that 'Fredrik' is a role, but even hundreds of volunteers couldn't possibly keep up. No, 'Fredrik' is the result of crossing an http server with a spam filter with an emacs whatsit and some other stuff besides."
-Gordon McMillan, June 1998

Python 2.0 200 . . 360 .

0.1.

"Those people who have nothing better to do than post on the Internet all day long are rarely the ones who have the most insights."
- Jakob Nielsen, December 1998

Python, Python , comp.lang.python . , . . . , .

3,000 , , .

, . , , Python . .

, , bug , fredrik@pythonware.com . , .

http://www.pythonware.com/people/fredrik/librarybook.htm

Tkinter?

, ( )Tkinter ui(user-interface :)
. , , ,
Tkinter .

,
http://www.pythonware.com/people/fredrik/tkinterbook.htm . (,
404)

DocBook SGML , , Secret Labs'
PythonWorks, Excosoft Documentor, James Clark's Jade DSSSL processor,
Norm Walsh's DocBook stylesheets, , Python .

: Tim Peters, Guido van Rossum, David Ascher, Mark Lutz,
Rael Dornfest, PythonWare : Matthew Ellis, H?kan Karlsson,
Rune Uhlin.

Lenny Muellner, SGML ,
Christien Shangraw, CD (
http://examples.oreilly.com/pythonsl , 404,).

0.2.

:

. .

e.g. Python

, , , , , .

.

0.3.

, Python 1.5.2 Python 2.0 . Python 2.4/2.5 ..... .

, Windows, Solaris, Linux .

. , , (?) .

, "-example- " " ". , - *(the eff-bot guide to) The Standard Python Library* .

CD (http://examples.oreilly.com/pythonsl). http://www.pythonware.com/people/fredrik/librarybook.htm . (ft, 404. ~)

0.4.

Python QQ : 43680167

Feather ( ) QQ: 85660100

1.

"Since the functions in the C runtime library are not part of the Win32 API, we believe the number of applications that will be affected by this bug to be very limited."
- Microsoft, January 1999

1.1.

Python , Python , .

. Python .

1.1.1.

: (len,
int, range ...) __builtin__ , exceptions
.

Python , .

1.1.2.

Python POSIX API C .
.

: os ;
(, ,) os.path ;
time/datetime .

[!Feather : datetime Py2.3 ,]

, . Python
/ .

1.1.3.

. string
. math (pi, e), cmath
math .

1.1.4.

re Python .
.

1.1.5.

sys , , .
operator . copy
, Python 2.0 gc .

1.2. _ _builtin_ _

Python . ;
Python .

1.2.1.

Python .
apply . Example 1-1 .

1.2.1.1. Example 1-1. apply

```
File: builtin-apply-example-1.py

def function(a, b):
    print a, b

apply(function, ("whither", "canada?"))
apply(function, (1, 2 + 3))
```

```
whither canada?
1 5
```

, apply 3
, Example 1-2 .

1.2.1.2. Example 1-2. apply

```
File: builtin-apply-example-2.py

def function(a, b):
    print a, b

apply(function, ("crunchy", "frog"))
apply(function, ("crunchy",), {"b": "frog"})
apply(function, (), {"a": "crunchy", "b": "frog"})
```

```
crunchy frog
crunchy frog
crunchy frog
```

apply ,
. Example 1-3 .

1.2.1.3. Example 1-3. apply

```
File: builtin-apply-example-3.py

class Rectangle:
    def _ _init_ _(self, color="white", width=10, height=10):
        print "create a", color, self, "sized", width, "x", height

class RoundedRectangle(Rectangle):
    def _ _init_ _(self, **kw):
        apply(Rectangle._ _init_ _, (self,), kw)

rect = Rectangle(color="green", height=

       100, width=100) rect= RoundedRectangle(color=

            "blue",

height = 20)< /FONT> <
FONT  face=     >
```

```
create a green <Rectangle instance at 8c8260> sized 100 x 100
create a blue <RoundedRectangle instance at 8c84c0> sized 10 x 20
```

Python 2.0 .
, * , ** .

:

```
result = function(*args, **kwargs)
result =
apply(function, args, kwargs) < /FONT
>
```

1.2.2.

Python , import
(from-import).
import _ _import_ _ .

. ( ) ,
. Example 1-4 , "-plugin "
.

1.2.2.1. Example 1-4. _ _import_ _

```
File: builtin-import-example-1.py

import glob, os

modules = []

for module_file in glob.glob("*-plugin.py"):
    try:
        module_name, ext =
os.path.splitext(os.path.basename(module_file))
        module = _ _import_ _(module_name)
        modules.append(module)
    except ImportError:
        pass # ignore broken modules

# say hello to all modules
for module in modules:
    module.hello()
```

```
example-plugin says hello
```

plug-in "-" (hyphens).
import , Python "-" .

Example 1-5 Example 1-4 plug-in .

1.2.2.2. Example 1-5. Plug-in

File: example-plugin.py

```
def hello():
    print "example-plugin says hello"
```

Example 1-6 .

1.2.2.3. Example 1-6. _ _import_ _

File: builtin-import-example-2.py

```
def getfunctionbyname(module_name, function_name):
    module = _ _import_ _(module_name)
    return getattr(module, function_name)

print repr(getfunctionbyname("dumbdbm", "open"))
```

```
<function open at 794fa0>
```

(lazy module loading).
Example 1-7 string .

1.2.2.4. Example 1-7. _ _import_ _

File: builtin-import-example-3.py

```
class LazyImport:
    def _ _init_ _(self, module_name):
        self.module_name = module_name
        self.module = None
    def _ _getattr_ _(self, name):
        if self.module is None:
            self.module = _ _import_ _(self.module_name)
        return getattr(self.module, name)

string = LazyImport("string")

print string.lowercase

abcdefghijklmnopqrstuvwxyz
```

Python . [Example 1-8 #eg-1-8
3 *hello.py* .

1.2.2.5. Example 1-8. reload

```
File: builtin-reload-example-1.py

import hello
reload(hello)
reload(hello)
```

```
hello again, and welcome to the show
hello again, and welcome to the show
hello again, and welcome to the show
```

reload .

[!Feather : ^ , .]

, ,
. , ( ).

, from-import .

1.2.3.

dir , , , .
Python , . Example 1-9
dir .

1.2.3.1. Example 1-9. dir

```
File: builtin-dir-example-1.py

def dump(value):
    print value, "=>", dir(value)

import sys

dump(0)
dump(1.0)
dump(0.0j) # complex number
dump([]) # list
dump({}) # dictionary
dump("string")
dump(len) # function
dump(sys) # module
```

```
0 => []
1.0 => []
0j => ['conjugate', 'imag', 'real']
[] => ['append', 'count', 'extend', 'index', 'insert',
    'pop', 'remove', 'reverse', 'sort']
{} => ['clear', 'copy',
'get', 'has_key',
'items', 'keys', 'update', 'values']
string =
    > [] <built-in function
len> = > ['_ _doc_ _', '_
    _name_ _', '_
_self_ _']
<module 'sys' (built-in)> =
  >  ['_ _doc_ _', '_ _name_ _',
    '_ _stderr_ _', '_ _stdin_ _', '_ _stdout_ _', 'argv',
    'builtin_module_names', 'copyright', 'dllhandle',
    'exc_info', 'exc_type', 'exec_prefix', 'executable',
...
```

Example 1-10 getmember

.

1.2.3.2. Example 1-10. dir

```
File: builtin-dir-example-2.py

class A:
    def a(self):
        pass
    def b(self):
        pass

class B(A):
    def c(self):
        pass
    def d(self):
        pass

def getmembers(klass, members=None):
    # get a list of all class members, ordered by class
    if members is None:
        members = []
    for k in klass._ _bases_ _:
        getmembers(k, members)
    for m in dir(klass):
        if m not in members:
            members.append(m)
    return members

print getmembers(A)
print getmembers(B)
print getmembers(IOError)

<
```

```
 FONT face=
```

```
        >
```

```
['_ _doc_ _', '_ _module_ _', 'a', 'b']
['_ _doc_ _', '_ _module_ _', 'a', 'b', 'c', 'd']
['_ _doc_ _', '_ _getitem_ _', '_ _init_ _', '_ _module_ _', '_ _str_
_']
```

getmembers . ,
. , .

[!Feather : , ,
]

vars , .
vars , (locals()).
Example 1-11 .

1.2.3.3. Example 1-11. vars

```
File: builtin-vars-example-1.py

book = "library2"
pages = 250
scripts = 350


print "the %(book)s book contains more than %(scripts)s scripts" % vars()
```

```
the library book contains more than 350 scripts
```

1.2.4.

Python ,
. , , ,
:

```
def function(value):
    print value
function(1)
function(1.0)
function("one")
```

type (Example 1-12) . *type descriptor (* *)* , Python .

1.2.4.1. Example 1-12. type

```
File: builtin-type-example-1.py

def dump(value):
    print type(value), value

dump(1)
dump(1.0)
dump("one")
```

```
<type 'int'> 1
<type 'float'> 1.0
<type 'string'> one
```

, is (?) . (Example 1-13).

1.2.4.2. Example 1-13. type

```
File: builtin-type-example-2.py

def load(file):
    if isinstance(file, type("")):
        file = open(file, "rb")
    return file.read()

print len(load("samples/sample.jpg")), "bytes"
print len(load(open("samples/sample.jpg", "rb"))), "bytes"
```

```
4672 bytes
4672 bytes
```

callable , Example 1-14 ,
(apply). , , lambda , ,
_ _call_ _ , True.

1.2.4.3. Example 1-14. callable

```
File: builtin-callable-example-1.py

def dump(function):
    if callable(function):
        print function, "is callable"
    else:
        print function, "is *not* callable"

class A:
    def method(self, value):
        return value

class B(A):
    def _ _call_ _(self, value):
        return value

a = A()
b = B()

dump(0) # simple objects
dump("string")
dump(callable)
dump(dump) # function

dump(A) # classes
dump(B)
dump(B.method)

dump(a) # instances
dump(b)
dump(b.method)
```

<

FONT

face =

>

```
0 is *not* callable
string is *not* callable
<built-in function callable> is callable
<function dump at 8ca320> is callable
A is callable
B is callable
<unbound method A.method> is callable
<A instance at 8caa10> is *not* callable
<B instance at 8cab00> is callable
<method A.method of B instance at 8cab00> is callable
```

(A B) ; , ( ). A , __call__ .

operator (, ,) . , ( ), .

. Python ; , (special class type), (special instance type).

type ; ! , isinstance , ( ) . Example 1-15 isinstance .

1.2.4.4. Example 1-15. isinstance

File: builtin-isinstance-example-1.py

```
class A:
    pass

class B:
    pass

class C(A):
    pass

class D(A, B):
    pass

def dump(object):
    print object, "=>",
    if isinstance(object, A):
        print "A",
    if isinstance(object, B):
        print "B",
    if isinstance(object, C):
        print "C",
    if isinstance(object, D):
        print "D",
    print

a = A()
b =
```

```
     B() c=
    C()

d =
  D()
dump(a)  dump(b)
dump(c)
```

```
  dump(d)
    dump(0)

dump("string")< /FONT
    > < FONT
```

```
face =


>
```

```
<A instance at 8ca6d0> => A
<B instance at 8ca750> => B
<C instance at 8ca780> =
  > A  C
<D instance at 8ca7b0> =  >
A B D 0 =   >
string =
>
```

issubclass , ,
. Example 1-16 .

, isinstance , issubclass
TypeError .

1.2.4.5. Example 1-16. issubclass

```
File: builtin-issubclass-example-1.py

class A:
    pass

class B:
    pass

class C(A):
    pass

class D(A, B):
    pass

def dump(object):
    print object, "=>",
    if issubclass(object, A):
        print "A",
    if issubclass(object, B):
        print "B",
    if issubclass(object, C):
        print "C",
    if issubclass(object, D):
        print "D",
    print

dump(A)
dump(B)
dump(C)
dump(D)
dump(O)
dump("string")
```

```
A => A
B => B
C => A  C
D = >
A  B D
O =
>
Traceback (innermost last):
  File "builtin-issubclass-example-1.py", line 29, in ?
  File "builtin-issubclass-example-1.py", line 15, in dump
TypeError: arguments must be classes
```

1.2.5.   Python

Python . eval
Python . , ,
Python . Example 1-17 .

1.2.5.1. Example 1-17. eval

```
File: builtin-eval-example-1.py

def dump(expression):
    result = eval(expression)
    print expression, "=>", result, type(result)

dump("1")
dump("1.0")
dump("'string'")
dump("1.0 + 2.0")
dump("'*' * 10")
dump("len('world')")
```

<

FONT
face =
>

```
1 => 1 <type 'int'>
1.0 => 1.0 <type 'float'>
'string' =  > string
<type  'string'> 1.0 +
2.0 =   >
3.0 <type 'float'>  '*' * 10=
 > **********  <type 'string'> len('world')=
>  5 <type 'int'>
```

, eval .
, __import__ os ,
(Example 1-18).

1.2.5.2. Example 1-18. eval

```
File: builtin-eval-example-2.py

print eval("_ _import_ _('os').getcwd()")
print eval("_ _import_ _('os').remove('file')")
```

```
/home/fredrik/librarybook Traceback (innermost
 last): File "builtin-eval-example-2", line 2, in
 ? File"<string>", line 0, in ?
os.error: (2, 'No such file or directory')
```

os.error , *Python* !

, . eval 2 , . , :

```
>>> print eval("_ _import_
_('os').remove('file')", {}) Traceback
  (innermost last):File "<stdin>", line 1, in
  ? File"<string>", line 0, in ?
os.error: (2, 'No such file or directory')
```

.... *os.error* .

Python , _ _builtins_ _ ( ), :

```
>>> namespace = {}  >>>
 print eval("_ _import_ _('os').remove('file')", namespace)
Traceback (innermost last):
  File "<stdin>",line 1, in ? File
  "<string>", line0, in ? os.error: (2,
'No such file or directory') >>> namespace.keys()
 ['_
_builtins_ _'] <
/FONT >
```

namespace , .

[!Feather : RP , _ _builtins_ _ _
builtins _]

, Python ,
, _ _builtins_ _ .
Example 1-19 .

1.2.5.3. Example 1-19. eval

```
File: builtin-eval-example-3.py

print eval("_ _import_ _('os').getcwd()", {})
print eval("_ _import_ _('os').remove('file')", {"_ _builtins_ _": {}})
```

```
/home/fredrik/librarybook Traceback (innermost
  last): File "builtin-eval-example-3.py", line 2, in
  ? File"<string>", line 0, in ?
NameError: _ _import_ _
```

, CPU . (,
eval("'*'*1000000*2*2*2*2*2*2*2*2*2")
).

1.2.6.

eval . , compile
exec (Example 1-20).

1.2.6.1. Example 1-20. compile

```
File: builtin-compile-example-1.py

NAME = "script.py"

BODY =

"""  prnt
'owl-stretching time' """
try:

compile(BODY,
    NAME, "exec") except
SyntaxError, v: print
    "syntax error:", v, "in", NAME #

syntax error: invalid syntax in script.py

< /FONT

>
```

, compile , exec
, Example 1-21 .

1.2.6.2. Example 1-21.

```
File: builtin-compile-example-2.py
```

```
BODY = """
print 'the ant, an introduction'
"""

code =  compile(BODY,"<script>", "exec")

print code

exec code

< FONT

face =




>
```

```
<code object ? at 8c6be0, file"<script>", line 0>
the ant, an introduction
```

Example 1-22 . write , indent dedent . .

1.2.6.3. Example 1-22.

```
File: builtin-compile-example-3.py

import sys, string

class CodeGeneratorBackend:
    "Simple code generator for Python"

    def begin(self, tab="\t"):
        self.code = []
        self.tab = tab
        self.level = 0
    def end(self):
        self.code.append("") # make sure there's a newline at the end
        return compile(string.join(self.code, "\n"), "<code>", "exec")

    def write(self, string):
        self.code.append(self.tab * self.level + string)

    def indent(self):
        self.level = self.level +

1 # in

2.0 and
    later, this can be written

    as: self.level +=
        1  def
        dedent(self):  if
        self.level =



              =

      0:
        raise SyntaxError, "internal error in

    code generator"
        self.level =
                      self.level- 1

    # or:
        self.level - =

        1






      #
        #  try
        it  out!
        c =




            CodeGeneratorBackend()

    c.begin() c.write("for
        i  in range(5):") c.indent()
        c.write("print 'code generation made easy!'") c.dedent() exec
c.end()< /FONT > < FONT


         face =









>
```

```
code generation made easy!
code generation made easy!
code generation made easy!
code generation made easy!
code generation made easy!
```

Python execfile , , ,
. Example 1-23 .

1.2.6.4. Example 1-23. execfile

File: builtin-execfile-example-1.py

```
execfile("hello.py")

def EXECFILE(filename, locals=None, globals=None):
    exec compile(open(filename).read(), filename, "exec") in locals,
globals

EXECFILE("hello.py")
```

<

FONT face= >

```
hello again, and welcome to the show
hello again, and welcome to the show
```

Example 1-24 Example 1-23 hello.py .

1.2.6.5. Example 1-24. hello.py

```
File: hello.py

print "hello again, and welcome to the show"
```

1.2.7. _ _builtin_ _

Python ,
_ _builtin_ _ . Example 1-25
open . open ,
.

1.2.7.1. Example 1-25. _ _builtin_ _

```
File: builtin-open-example-1.py
```

```
def open(filename, mode="rb"):
    import _ _builtin_ _
    file = _ _builtin_ _.open(filename, mode)
    if file.read(5) not in("GIF87", "GIF89"): raise

IOError, "not aGIF
    file" file.seek(0) return file
    fp =

             open("samples/sample.gif")
    print
    len(fp.read()), "bytes"

fp =
 open("samples/sample.jpg") print

len(fp.read()),  "bytes"
< /FONT> <

FONT

  face=

>
```

```
3565 bytes
Traceback (innermost last):
  File "builtin-open-example-1.py", line 12, in ?
  File "builtin-open-example-1.py", line 5, in open
IOError: not a GIF file
```

[!Feather : open() ? GIF

,

.

file() open() ,]

1.3. exceptions

exceptions . Python , _ _builtin_ _ . , .

1.5.2 , 2.0 .

:

- *Exception* . ( ) .
- *SystemExit(Exception)* sys.exit . try-except , .
- *StandardError(Exception)* (*SystemExit*).
- *KeyboardInterrupt(StandardError)* Control-C( ) . " " try-except .
- *ImportError(StandardError)* Python .
- *EnvironmentError* . (, bug).
- *IOError(EnvironmentError)* I/O .
- *OSError(EnvironmentError)* os .
- *WindowsError(OSError)* os Windows .
- *NameError(StandardError)* Python .
- *UnboundLocalError(NameError)* , , . 2.0 ; *NameError* .
- *AttributeError(StandardError)* , Python ( ) , , , .
- *SyntaxError(StandardError)* , , .
- (2.0) *IndentationError(SyntaxError)* . 2.0 , *SyntaxError* .

- (2.0) *TabError(IndentationError)* , -tt . 2.0 , *SyntaxError* .
- *TypeError(StandardError)* , .
- *AssertionError(StandardError)* assert (false).
- *LookupError(StandardError)* .
- *IndexError(LookupError)* , ( ) .
- *KeyError(LookupError)* ( ) .
- *ArithmeticError(StandardError)* .
- *OverflowError(ArithmeticError)* (,).
- *ZeroDivisionError(ArithmeticError)* , 0 .
- *FloatingPointError(ArithmeticError)* , .
- *ValueError(StandardError)* , .
- (2.0) *UnicodeError(ValueError)* , Unicode . 2.0 .
- *RuntimeError(StandardError)* , , , .
- *NotImplementedError(RuntimeError)* , , .
- *SystemError(StandardError)* , . (, "eval_code2: NULL globals") . 5 . (raise SystemError).
- *MemoryError(StandardError)* , . ; , .

. *Exception* ( ) , __str__ . Example 1-26 exceptions .

1.3.0.1. Example 1-26. exceptions

```
File: exceptions-example-1.py

# python imports this module by itself, so the following
# line isn't really needed
# python                    ,
# import exceptions

class HTTPError(Exception):
    # indicates an HTTP protocol error
    def _ _init_ _(self, url, errcode, errmsg):
        self.url = url
        self.errcode = errcode
        self.errmsg =
```

```
     errmsg def _ _str_ _(self):
    return ( "<HTTPError for %s: %s %s>"
        %  (self.url,
        self.errcode,  self.errmsg)
        )  try:
    raise HTTPError("http://www.python.org/foo", 200, "Not
        Found") except
            HTTPError, error: print "url", "=
```

```
 >", error.url
    print "errcode", "=>", error.errcode
    print "errmsg", "=
      >",
```

```
        error.errmsg raise #
        reraise  exception
        < /FONT
        >
```

```
      <
     FONT face=
          >
```

```
url => http://www.python.org/foo
errcode =>  200
errmsg =
>  Not Found
Traceback (innermost last):
  File "exceptions-example-1", line 16, in ?
HTTPError: <HTTPError for http://www.python.org/foo: 200 Not Found>
```

1.4. os

, posix nt. os

.

1.4.1.

open / file , , Example 1-27 .

os .

1.4.1.1. Example 1-27. os

```
File: os-example-3.py

import os
import string

def replace(file, search_for, replace_with):
    # replace strings in a text file

    back = os.path.splitext(file)[0] + ".bak"
    temp = os.path.splitext(file)[0] + ".tmp"

    try:
        # remove old temp file, if any
        os.remove(temp)
    except os.error:
        pass

    fi = open(file)
    fo = open(temp, "w")

    for s in

fi.readlines(): fo.write(string.replace(s,
search_for, replace_with))

fi.close() fo.close() try: #
    remove old backup file, if any os.remove(back)

    except  os.error: pass #
    rename  original to backup...

    os.rename(file,
        back) # ...and temporary to original os.rename(temp,
        file)
    # #
        try

    it  out!
    file =
    "samples/sample.txt"



        replace(file,
    "hello", "tjena")
        replace(file,

    "tjena",  "hello")
    < /FONT
```

```
>
```

1.4.2.

os .

listdir ( ) , Example 1-28 . Unix Windows (. ..) .

1.4.2.1. Example 1-28. os

```
File: os-example-5.py

import os

for file in os.listdir("samples"):
    print file
```

```
sample.au
sample.jpg
sample.wav
...
```

getcwd chdir . Example 1-29 .

1.4.2.2. Example 1-29. os

```
File: os-example-4.py

import os

# where are we?
cwd = os.getcwd()
print "1", cwd

# go down
```

```
os.chdir("samples")
print "2", os.getcwd()

# go back up
os.chdir(os.pardir)
print "3", os.getcwd()
```

```
1 /ematter/librarybook
2 /ematter/librarybook/samples
3 /ematter/librarybook
```

makedirs removedirs Example 1-30 .

1.4.2.3. Example 1-30. os /

```
File: os-example-6.py

import os

os.makedirs("test/multiple/levels")

fp =

open("test/multiple/levels/file", "w") fp.write("inspector
praline") fp.close()
#

remove the file os.remove("test/multiple/levels/file")
#
```

and all empty directories above it os.removedirs("test/multiple/levels")
<
/FONT

>

removedirs . mkdir
rmdir . Example 1-31 .

1.4.2.4. Example 1-31. os /

```
File: os-example-7.py

import os

os.mkdir("test")
os.rmdir("test")

os.rmdir("samples") # this will fail
```

```
Traceback (innermost last):
  File "os-example-7", line 6, in ?
OSError: [Errno 41] Directory not empty: 'samples'
```

, shutil rmtree .

1.4.3.

stat , Example 1-32 . (stat_result , 10), st_mode ( ), st_ino (inode number), st_dev (device), st_nlink (number of hard links), st_uid (ID), st_gid (ID), st_size (,), st_atime ( ), st_mtime ( ), st_ctime (; Unix /metadata , Windows) - .

[!Feather : 9 . , , struct.]

1.4.3.1. Example 1-32. os

File: os-example-1.py

```
import os
import time

file = "samples/sample.jpg"

def dump(st):
    mode, ino, dev, nlink, uid, gid, size, atime, mtime, ctime = st
    print "- size:", size, "bytes"
    print "- owner:", uid, gid
    print "- created:", time.ctime(ctime)
    print "- last accessed:", time.ctime(atime)
    print "- last modified:", time.ctime(mtime)
    print "- mode:", oct(mode)
    print "- inode/dev:", ino, dev

#
# get stats for a filename

st =


 os.stat(file)
```

```
print  "stat",

file dump(st)
    print # # get stats for an open file fp =

     open(file)

st =

      os.fstat(fp.fileno())

print  "fstat",

file

dump(st) <
/FONT >

< FON
T

              face =
```

>

```
stat samples/sample.jpg
- size: 4762 bytes
- owner: 0 0
- created: Tue Sep 07 22:45:58 1999
- last accessed: Sun Sep 19 00:00:00 1999
- last modified: Sun May 19 01:42:16 1996
- mode: 0100666
- inode/dev: 0 2

fstat samples/sample.jpg
- size: 4762 bytes
- owner: 0 0
- created: Tue Sep 07 22:45:58 1999
- last accessed: Sun Sep 19 00:00:00 1999
- last modified: Sun May 19 01:42:16 1996
- mode: 0100666
- inode/dev: 0 0
```

Unix , (st_inode , st_dev)
Unix ,
.

stat .
.

chmod utime Example 1-33 .

1.4.3.2. Example 1-33. os

File: os-example-2.py

```
import os
import stat, time

infile = "samples/sample.jpg"
outfile = "out.jpg"

# copy contents
fi = open(infile, "rb")
fo = open(outfile, "wb")

while 1:
    s = fi.read(10000)
    if not s:
        break fo.write(s)

fi.close() fo.close()
# copy mode

and  timestamp
st =


os.stat(infile)  os.chmod(outfile, stat.S_IMODE(st[stat.ST_MODE]))
os.utime(outfile,  (st[stat.ST_ATIME], st[stat.ST_MTIME]))

print "original",
    " =
    >" print "mode",
        oct(stat.S_IMODE(st[stat.ST_MODE]))
    print

"atime",
time.ctime(st[stat.ST_ATIME])

print "mtime", time.ctime(st[stat.ST_MTIME]) print "copy",
" =
 >" st=

  os.stat(outfile)

print  "mode",
oct(stat.S_IMODE(st[stat.ST_MODE]))

  print
"atime",  time.ctime(st[stat.ST_ATIME]) print
"mtime",  time.ctime(st[stat.ST_MTIME]) <

/FONT >
    < FONT

face =

>
```

```
original =>
mode 0666
atime Thu Oct 14 15:15:50 1999
```

```
mtime Mon Nov 13 15:42:36 1995
copy =



>
mode 0666
atime Thu Oct 14 15:15:50 1999
mtime Mon Nov 13 15:42:36 1995
```

1.4.4.

system , , Example 1-34 .

1.4.4.1. Example 1-34. os

```
File: os-example-8.py

import os

if os.name == "nt":
    command =



 "dir"  else:
    command = "ls
-l"
    os.system(command)  < /FONT

><

FONT



 face =


    >
```

```
-rwxrw-r--   1 effbot  effbot        76 Oct  9 14:17 README
-rwxrw-r--   1 effbot  effbot      1727 Oct  7 19:00
SimpleAsyncHTTP.py
-rwxrw-r--   1 effbot  effbot       314 Oct  7 20:29 aifc-example-1.py
-rwxrw-r--   1 effbot  effbot       259 Oct  7 20:38
anydbm-example-1.py
...
```

shell , shell . Windows 95/98 , shell command.com , 0.

11os.system11 shell , (os.system("viewer %s" % file) , file "sample.jpg; rm -rf $HOME" ....). , exec spawn ( ).

exec (" "). Example 1-35 , "goodbye" .

1.4.4.2. Example 1-35. os

```
File: os-exec-example-1.py

import os
import sys

program = "python"
arguments = ["hello.py"]

print os.execvp(program, (program,) +  tuple(arguments))
print "goodbye"

hello again, and welcome to the show
```

Python exec . Example 1-35 execvp , , ( ) , . *Python Library Reference* .

Unix , exec , fork wait
, Example 1-36 . fork , wait
.

1.4.4.3. Example 1-36. os (Unix)

```
File: os-exec-example-2.py

import os
import sys

def run(program, *args):
    pid = os.fork()
    if not pid:
        os.execvp(program, (program,) +  args)
    return os.wait()[0]

run("python", "hello.py")

print "goodbye"
```

```
hello again, and welcome to the show
goodbye
```

fork 0 (fork),
0 (PID). ,
"not pid " .

fork wait Windows , spawn ,
Example 1-37 . , spawn ,
.

1.4.4.4. Example 1-37. os (Windows)

```
File: os-spawn-example-1.py
```

```
import os
import string

def run(program, *args):
    # find executable
    for path in string.split(os.environ["PATH"], os.pathsep):
        file = os.path.join(path, program) + ".exe"
        try:
            return os.spawnv(os.P_WAIT, file, (file,) + args)
        except os.error:
            pass
    raise os.error, "cannot find executable"

run("python", "hello.py")

print "goodbye"
```

```
hello again, and welcome to the show
goodbye
```

spawn . Example 1-38 run mode ; os.P_NOWAIT , , os.P_WAIT spawn .

os.P_OVERLAY , spawn exec , os.P_DETACH , , .

1.4.4.5. Example 1-38. os (Windows)

File: os-spawn-example-2.py

```
import os
```

```
import os
import string

if os.name in ("nt", "dos"):
    exefile = ".exe"
else:
    exefile = ""

def spawn(program, *args):
    try:
        # possible 2.0 shortcut!
        return os.spawnvp(program, (program,) + args)
    except AttributeError:
        pass
    try:
        spawnv =


os.spawnv except

AttributeError: # assume it's unix
    pid =

    os.fork()  if

not pid: os.execvp(program,
    (program,)
        + args) return os.wait()[0]
        else: # got spawnv but
    no spawnp:
        go
    look
        for  an
    executable for

        path in string.split(os.environ["PATH"], os.pathsep):
        file =






          os.path.join(path, program) +
            exefile  try: return spawnv(os.P_WAIT, file,
            (file,)
                + args) except os.error: pass raise
            IOError, "cannot
                find
        executable" # # try

it out!
spawn("python", "hello.py")

print "goodbye" < /FONT >
    < FONT

    face =
```

```
            >
```

```
hello again, and welcome to the show
goodbye
```

Example 1-39 spawnvp . (/
), spawnv
. , exec fork .

1.4.5. (Daemon Processes)

Unix , fork ("
/daemon"). , (fork off) ,
, Example 1-40 .

1.4.5.1. Example 1-40. os (Unix)

```
File: os-example-14.py

import os
import time

pid =




os.fork()  if
pid: os._exit(0)
    # kill original print

"daemon started" time.sleep(10)
print
"daemon terminated" <
/FONT




>
```

, setpgrp
" /process group leader". , ( )
:

```
os.setpgrp()
```

mode flags(
?), user mode mask:

```
os.umask(0)
```

, *stdout/stderr* , (
stdout stderr ,
).

```
class NullDevice:
    def write(self, s):
        pass
sys.stdin.close()
sys.stdout =


NullDevice()
sys.stderr =
NullDevice()
```

< /FONT
>

, Python print C printf/fprintf (device)
, sys.stdout.write()
IOError , ...

, _exit . sys.exit ,
(caller) *SystemExit* , . Example
1-41 .

1.4.5.2. Example 1-41. os

```
File: os-example-9.py

import os
import sys

try:
    sys.exit(1)
except SystemExit, value:
    print "caught exit(%s)" % value

try:
    os._exit(2)
except SystemExit, value:
    print "caught exit(%s)" % value

print "bye!"
```

```
caught exit(1)
```

1.5. os.path

os.path ( ) . (import) os , os.path .

1.5.1.

os.path . , , . Example 1-42 .

1.5.1.1. Example 1-42. os.path

```
File: os-path-example-1.py

import os

filename = "my/little/pony"

print "using", os.name, "..."
print "split", "=>", os.path.split(filename)
print "splitext", "=>", os.path.splitext(filename)
print "dirname", "=>", os.path.dirname(filename)
print "basename", "=>", os.path.basename(filename)

print "join",

" =



>", os.path.join(os.path.dirname(filename),
os.path.basename(filename))<
  /FONT>
  <
 FONT face=
      >
```

```
using nt ...
split => ('my/little', 'pony')
splitext => ('my/little/pony', '')
dirname = > my/little
basename =
>  pony
join =
>  my/little\pony
```

split ( ).

os.path
Example 1-43

1.5.1.2. Example 1-43. os.path

```
File: os-path-example-2.py

import os

FILES = (
    os.curdir,
    "/",
    "file",
    "/file",
    "samples",
    "samples/sample.jpg",
    "directory/file",
    "../directory/file",
    "/directory/file"
    )

for file in FILES:
    print file, "=>",
    if os.path.exists(file):
        print "EXISTS",
    if os.path.isabs(file):
        print "ISABS",
    if os.path.isdir(file):
        print "ISDIR",
    if os.path.isfile(file):
        print "ISFILE",
    if os.path.islink(file):
        print "ISLINK",
    if os.path.ismount(file):
        print "ISMOUNT",
    print
```

<

FONT

face =

>

```
. => EXISTS ISDIR
/ => EXISTS ISABS ISDIR ISMOUNT
file =>
/file => ISABS
samples => EXISTS ISDIR
samples/sample.jpg => EXISTS ISFILE
directory/file =>
../directory/file =>
/directory/file => ISABS
```

expanduser Unix shell (~, Windows), Example 1-44 .

1.5.1.3. Example 1-44. os.path

```
File: os-path-expanduser-example-1.py

import os

print os.path.expanduser("~/.pythonrc")

# /home/effbot/.pythonrc
```

expandvars , Example 1-45 .

1.5.1.4. Example 1-45. os.path

```
File: os-path-expandvars-example-1.py

import os

os.environ["USER"] = "user"

print os.path.expandvars("/home/$USER/config")
print os.path.expandvars("$USER/folders")
```

```
/home/user/config
user/folders
```

1.5.2.

walk (Example 1-46).

, , .

1.5.2.1. Example 1-46. os.path

```
File: os-path-walk-example-1.py

import os

def callback(arg, directory, files):
    for file in files:
        print os.path.join(directory, file), repr(arg)

os.path.walk(".", callback, "secret message")
```

```
./aifc-example-1.py 'secret message'
./anydbm-example-1.py 'secret message'
./array-example-1.py 'secret message'
...
./samples 'secret message'
./samples/sample.jpg 'secret message'
./samples/sample.txt 'secret message'
./samples/sample.zip 'secret message'
./samples/articles 'secret message'
./samples/articles/article-1.txt 'secret message'
./samples/articles/article-2.txt 'secret message'
...
```

walk (,
). Example 1-47 index ,
for-in .

1.5.2.2. Example 1-47. os.listdir

```
File: os-path-walk-example-2.py

import os

def index(directory):
    # like os.listdir, but traverses directory trees
    stack = [directory]
    files = []
    while stack:
        directory = stack.pop()
        for file in os.listdir(directory):
            fullname = os.path.join(directory, file)
            files.append(fullname)
            if os.path.isdir(fullname) and not
os.path.islink(fullname):
                stack.append(fullname)
    return files

for file in index("."):
    print file

.\aifc-example-1.py
.\anydbm-example-1.py
.\array-example-1.py
...
```

( ) , Example 1-48
. *DirectoryWalker* ,
. (generator?)

1.5.2.3. Example 1-48. DirectoryWalker

```
File: os-path-walk-example-3.py

import os

class DirectoryWalker:
    # a forward iterator that traverses a directory tree
    def __init__(self, directory):
        self.stack = [directory]
        self.files = []
        self.index = 0

    def __getitem__(self, index):
        while 1:
            try:
                file = self.files[self.index]
                self.index = self.index + 1
            except IndexError:
                # pop next directory from stack
                self.directory = self.stack.pop()
                self.files = os.listdir(self.directory)
                self.index = 0
            else:
                # got a filename
                fullname = os.path.join(self.directory, file)
                if os.path.isdir(fullname) and not
os.path.islink(fullname):
                    self.stack.append(fullname)
                return fullname

for file in DirectoryWalker("."):
    print file
```

```
.\aifc-example-1.py
.\anydbm-example-1.py
.\array-example-1.py
...
```

DirectoryWalker __getitem__ . ( ) , .

, , Example 1-49 , os.stat ( ). stat (os.path.isdir os.path.islink stat), .

1.5.2.4. Example 1-49. DirectoryStatWalker

File: os-path-walk-example-4.py

```
import os, stat
```

```
class DirectoryStatWalker:
    # a forward iterator that traverses a directory tree, and
    # returns the filename and additional file information

    def _ _init_ _(self, directory):
        self.stack = [directory]
        self.files = []
        self.index = 0

    def _ _getitem_ _(self, index):
        while 1:
            try:
                file = self.files[self.index]
                self.index = self.index + 1
            except IndexError:
                # pop next directory from stack
                self.directory = self.stack.pop()
                self.files = os.listdir(self.directory)
                self.index = 0
            else:
                # got a filename
                fullname = os.path.join(self.directory, file)
                st = os.stat(fullname)
                mode = st[stat.ST_MODE]
                if stat.S_ISDIR(mode) and not stat.S_ISLNK(mode):
                    self.stack.append(fullname)
                return fullname, st

for file, st in DirectoryStatWalker("."):
    print file, st[stat.ST_SIZE]
```

```
.\aifc-example-1.py 336
.\anydbm-example-1.py 244
.\array-example-1.py 526
```

1.6. stat

Example 1-50 stat , os.stat .

1.6.0.1. Example 1-50. Using the stat Module

File: stat-example-1.py

```
import stat
import os, time

st = os.stat("samples/sample.txt")

print "mode", "=>", oct(stat.S_IMODE(st[stat.ST_MODE]))

print "type", "=>",
if stat.S_ISDIR(st[stat.ST_MODE]):
    print "DIRECTORY",
if stat.S_ISREG(st[stat.ST_MODE]):
    print "REGULAR",
if stat.S_ISLNK(st[stat.ST_MODE]):
    print "LINK",
print

print "size", "=>", st[stat.ST_SIZE]

print "last accessed", "=>", time.ctime(st[stat.ST_ATIME])
print "last modified", "=>", time.ctime(st[stat.ST_MTIME])
print "inode changed", "=>", time.ctime(st[stat.ST_CTIME])
```

```
mode => 0664
type => REGULAR
size => 305
last accessed => Sun Oct 10 22:12:30 1999
last modified => Sun Oct 10 18:39:37 1999
inode changed => Sun Oct 10 15:26:38 1999
```

1.7. string

string , Example 1-51 .

1.7.0.1. Example 1-51. string

```
File: string-example-1.py

import string

text = "Monty Python's Flying Circus"

print "upper", "=>", string.upper(text)
print "lower", "=>", string.lower(text)
print "split", "=>", string.split(text)
print "join", "=>", string.join(string.split(text), "+")
print "replace", "=>", string.replace(text, "Python", "Java")
print "find", "=>", string.find(text, "Python"), string.find(text,
"Java")
print "count", "=>", string.count(text, "n")
```

```
upper => MONTY PYTHON'S FLYING CIRCUS
lower => monty python's flying circus
split => ['Monty', "Python's", 'Flying', 'Circus']
join => Monty+Python's+Flying+Circus
replace => Monty Java's Flying Circus
find => 6 -1
count => 3
```

Python 1.5.2 , string strop
.

Python1.6 ,
Example 1-52 , string
.

1.7.0.2. Example 1-52. string

```
File: string-example-2.py

text = "Monty Python's Flying Circus"

print "upper", "=>", text.upper()
print "lower", "=>", text.lower()
print "split", "=>", text.split()
print "join", "=>", "+".join(text.split())
print "replace", "=>", text.replace("Python", "Perl")
print "find", "=>", text.find("Python"), text.find("Perl")
print "count", "=>", text.count("n")
```

```
upper => MONTY PYTHON'S FLYING CIRCUS
lower => monty python's flying circus
split => ['Monty', "Python's", 'Flying', 'Circus']
join => Monty+Python's+Flying+Circus
replace => Monty Perl's Flying Circus
find => 6 -1
count => 3
```

, , string
, (Example 1-53).

1.7.0.3. Example 1-53. string

```
File: string-example-3.py

import string

print int("4711"),
print string.atoi("4711"),
print string.atoi("11147", 8), # octal
print string.atoi("1267", 16), # hexadecimal
print string.atoi("3mv", 36) # whatever...

print string.atoi("4711", 0),
print string.atoi("04711", 0),
print string.atoi("0x4711", 0)

print float("4711"),
print string.atof("1"),
print string.atof("1.23e5")
```

```
4711 4711 4711 4711 4711
4711 2505 18193
4711.0 1.0 123000.0
```

(1.6) int
float string

atoi , (number base).
0, : "0x,"
16 ( ), "0," 8 ( ). 10
( ), .

1.6 , int atoi .
, int float Unicode .

1.8. re

"Some people, when confronted with a problem, think 'I know, I'll use regular expressions.' Now they have two problems."

- Jamie Zawinski, on comp.lang.emacs

re (regular expression) , (match), (search). ( ) .

match , Example 1-54 . (,) , . group .

1.8.0.1. Example 1-54. re

File: re-example-1.py

```
import re

text = "The Attila the Hun Show"

# a single character
m = re.match(".", text)
if m: print repr("."), "=>", repr(m.group(0))

# any string of characters
m = re.match(".*", text)
if m: print repr(".*"), "=>", repr(m.group(0))

# a string of letters (at least one)                    (        )
m = re.match("\w+", text)
if m: print repr("\w+"), "=>", repr(m.group(0))

# a string of digits
m = re.match("\d+", text)
if m: print repr("\d+"), "=>", repr(m.group(0))
```

```
 '.' => 'T'
'.*' => 'The Attila the Hun Show'
'\\w+' => 'The'
```

. , group Example 1-55 . group(1) , group(2) , ... group , .

1.8.0.2. Example 1-55. re

```
File: re-example-2.py

import re

text ="10/15/99"

m = re.match("(\d{2})/(\d{2})/(\d{2,4})", text)
if m:
    print m.group(1, 2, 3)

('10', '15', '99')
```

search , Example 1-56 .
, ,
. , *None* .

1.8.0.3. Example 1-56. re

```
File: re-example-3.py

import re

text = "Example 3: There is 1 date 10/25/95 in here!"

m = re.search("(\d{1,2})/(\d{1,2})/(\d{2,4})", text)

print m.group(1), m.group(2), m.group(3)

month, day, year = m.group(1, 2, 3)
print month, day, year

date = m.group(0)
print date

10 25 95
10 25 95
10/25/95
```

Example 1-57 sub , .

1.8.0.4. Example 1-57. re

```
File: re-example-4.py

import re
```

```
text = "you're no fun anymore..."

# literal replace (string.replace is faster)
#          (string.replace         )
print re.sub("fun", "entertaining", text)

# collapse all non-letter sequences to a single dash
#                            "-"(dansh,      )
print re.sub("[^\w]+", "-", text)

# convert all words to beeps
#                   BEEP
print re.sub("\S+", "-BEEP-", text)
```

```
you're no entertaining anymore...
you-re-no-fun-anymore-
-BEEP- -BEEP- -BEEP- -BEEP-
```

(callback) sub . Example 1-58 .

1.8.0.5. Example 1-58. re ( )

```
File: re-example-5.py

import re
import string

text = "a line of text\\012another line of text\\012etc..."

def octal(match):
    # replace octal code with corresponding ASCII character
    #           ASCII
    return chr(string.atoi(match.group(1), 8))

octal_pattern = re.compile(r"\\(\d\d\d)")

print text
print octal_pattern.sub(octal, text)
```

```
a line of text\012another line of text\012etc...
a line of text
another line of text
etc...
```

, re , ,
. Python1.5.2 , 20 ,
2.0 , 100 .

, Example 1-59 .
, .

1.8.0.6. Example 1-59. re

```
File: re-example-6.py

import re, string

def combined_pattern(patterns):
    p = re.compile(
        string.join(map(lambda x: "("+x+")", patterns), "|")
        )
    def fixup(v, m=p.match, r=range(0,len(patterns))):
        try:
            regs = m(v).regs
        except AttributeError:
            return None # no match, so m.regs will fail
        else:
            for i in r:
                if regs[i+1] != (-1, -1):
                    return i
    return fixup

#
# try it out!

patterns = [
    r"\d+",
    r"abc\d{2,4}",
    r"p\w+"
]

p = combined_pattern(patterns)

print p("129391")
print p("abc800")
print p("abc1600")
print p("python")
print p("perl")
print p("tcl")
```

```
0
1
1
2
2
None
```

1.9. math

math . C

, ,

, . Example 1-60 math .

1.9.0.1. Example 1-60. math

File: math-example-1.py

```
import math

print "e", "=>", math.e
print "pi", "=>", math.pi
print "hypot", "=>", math.hypot(3.0, 4.0)

# and many others...
```

```
e => 2.71828182846
pi => 3.14159265359
hypot => 5.0
```

Python Library Reference .

1.10. cmath

Example 1-61 cmath .

1.10.0.1. Example 1-61. cmath

File: cmath-example-1.py

```
import cmath

print "pi", "=>", cmath.pi
print "sqrt(-1)", "=>", cmath.sqrt(-1)

pi => 3.14159265359
sqrt(-1) => 1j
```

Python Library Reference .

1.11. operator

operator Python " " .
map filter , operator
lambda . .
Example 1-62 operator .

1.11.0.1. Example 1-62. operator

```
File: operator-example-1.py

import operator

sequence = 1, 2, 4

print "add", "=>", reduce(operator.add, sequence)
print "sub", "=>", reduce(operator.sub, sequence)
print "mul", "=>", reduce(operator.mul, sequence)
print "concat", "=>", operator.concat("spam", "egg")
print "repeat", "=>", operator.repeat("spam", 5)
print "getitem", "=>", operator.getitem(sequence, 2)
print "indexOf", "=>", operator.indexOf(sequence, 2)
print "sequenceIncludes", "=>", operator.sequenceIncludes(sequence, 3)

add => 7
sub => -5
mul => 8
concat => spamegg
repeat => spamspamspamspamspam

getitem => 4
indexOf => 1
```

```
sequenceIncludes => 0
```

Example 1-63 operator .

1.11.0.2. Example 1-63. operator

```
File: operator-example-2.py

import operator
import UserList

def dump(data):
    print type(data), "=>",
    if operator.isCallable(data):
        print "CALLABLE",
    if operator.isMappingType(data):
        print "MAPPING",
    if operator.isNumberType(data):
        print "NUMBER",
    if operator.isSequenceType(data):
        print "SEQUENCE",
    print

dump(0)
dump("string")
dump("string"[0])
dump([1, 2, 3])
dump((1, 2, 3))
dump({"a": 1})
dump(len) # function
dump(UserList) # module
dump(UserList.UserList) # class
dump(UserList.UserList()) # instance
```

```
<type 'int'> => NUMBER
<type 'string'> => SEQUENCE
<type 'string'> => SEQUENCE
<type 'list'> => SEQUENCE
<type 'tuple'> => SEQUENCE
<type 'dictionary'> => MAPPING
<type 'builtin_function_or_method'> => CALLABLE
<type 'module'> =>
<type 'class'> => CALLABLE
<type 'instance'> => MAPPING NUMBER SEQUENCE
```

operator .
isNumberType , isMappingType , isSequenceType ,
.

( ) .
isSequenceType , (
).

1.12. copy

copy , , Example 1-64 .

copy(object) => object " / (shallow)" (copy). " / (shallow)" , (container) , .

1.12.0.1. Example 1-64. copy

```
File: copy-example-1.py

import copy

a = [[1],[2],[3]]
b = copy.copy(a)

print "before", "=>"
print a
print b

# modify original
a[0][0] = 0
a[1] = None

print "after", "=>"
print a
print b
```

```
before =>
[[1], [2], [3]]
[[1], [2], [3]]
after =>
[[0], None, [3]]
```

```
[[0], [2], [3]]
```

[:] ( ) , copy .

, deepcopy(object) => object (deepcopy), Example 1-65 , ,

1.12.0.2. Example 1-65. copy (Collections)

```
File: copy-example-2.py

import copy

a = [[1],[2],[3]]
b = copy.deepcopy(a)

print "before", "=>"
print a
print b

# modify original
a[0][0] = 0
a[1] = None

print "after", "=>"
print a
print b
```

```
before =>
[[1], [2], [3]]
[[1], [2], [3]]
after =>
[[0], None, [3]]
[[1], [2], [3]]
```

1.13. sys

sys Python .

1.13.1.

, argv , Example 1-66
. .

1.13.1.1. Example 1-66. sys

```
File: sys-argv-example-1.py

import sys

print "script name is", sys.argv[0]

if len(sys.argv) > 1:
    print "there are", len(sys.argv)-1, "arguments:"
    for arg in sys.argv[1:]:
        print arg
else:
    print "there are no arguments!"
```

```
script name is sys-argv-example-1.py
there are no arguments!
```

("python < sys-argv-example-1.py "), . python (-c), "-c".

1.13.2.

path , Python (Python , ,). Python , , PYTHONPATH , (Windows) . , , Example 1-67 .

1.13.2.1. Example 1-67. sys

```
File: sys-path-example-1.py

import sys

print "path has", len(sys.path), "members"

# add the sample directory to the path
sys.path.insert(0, "samples")
import sample
```

```
# nuke the path
sys.path = []
import random # oops!
```

```
path has 7 members
this is the sample module!
Traceback (innermost last):
  File "sys-path-example-1.py", line 11, in ?
    import random # oops!
ImportError: No module named random
```

builtin_module_names Python ,
Example 1-68 .

1.13.2.2. Example 1-68. sys

```
File: sys-builtin-module-names-example-1.py

import sys

def dump(module):
    print module, "=>",
    if module in sys.builtin_module_names:
        print "<BUILTIN>"
    else:
        module = _ _import_ _(module)
        print module._ _file_ _

dump("os")
dump("sys")
dump("string")
dump("strop")
dump("zlib")
```

```
os => C:\python\lib\os.pyc
sys => <BUILTIN>
string => C:\python\lib\string.pyc
strop => <BUILTIN>
zlib => C:\python\zlib.pyd
```

modules . import
.

Example 1-69 , Python
.

1.13.2.3. Example 1-69. sys

```
File: sys-modules-example-1.py

import sys

print sys.modules.keys()
```

```
['os.path', 'os', 'exceptions', '__main__', 'ntpath', 'strop', 'nt',
'sys', '__builtin__', 'site', 'signal', 'UserDict', 'string', 'stat']
```

1.13.3.

getrefcount (Example 1-70) -
. Python , 0 ,
.

1.13.3.1. Example 1-70. sys

```
File: sys-getrefcount-example-1.py

import sys

variable = 1234

print sys.getrefcount(0)
print sys.getrefcount(variable)
print sys.getrefcount(None)
```

```
50
3
192
```

,
.

== ===

Example 1-71 platform , .

1.13.3.2. Example 1-71. sys

```
File: sys-platform-example-1.py

import sys

#
# emulate "import os.path" (sort of)...

if sys.platform == "win32":
    import ntpath
    pathmodule = ntpath
elif sys.platform == "mac":
    import macpath
    pathmodule = macpath
else:
    # assume it's a posix platform
    import posixpath
    pathmodule = posixpath

print pathmodule
```

Windows 9X/NT(win32), Macintosh(mac) .
Unix , platform "uname -r " ,
irix6 , linux2 , sunos5 (Solaris).

1.13.4.

setprofiler (profiling function).
( ),
. Example 1-72 .

1.13.4.1. Example 1-72. sys

```
File: sys-setprofiler-example-1.py

import sys

def test(n):
    j = 0
    for i in range(n):
        j = j + i
    return n

def profiler(frame, event, arg):
    print event, frame.f_code.co_name, frame.f_lineno, "->", arg
```

```
# profiler is activated on the next call, return, or exception
#                          ,     ,
sys.setprofile(profiler)

# profile this function call
#
test(1)

# disable profiler
#
sys.setprofile(None)

# don't profile this call
#
test(2)
```

```
call test 3 -> None
return test 7 -> 1
```

, profile .

Example 1-73 settrace , trace

.

1.13.4.2. Example 1-73. sys

```
File: sys-settrace-example-1.py

import sys

def test(n):
    j = 0
    for i in range(n):
        j = j + i
    return n

def tracer(frame, event, arg):
    print event, frame.f_code.co_name, frame.f_lineno, "->", arg
    return tracer

# tracer is activated on the next call, return, or exception
#                        ,     ,
sys.settrace(tracer)
```

```
# trace this function call
#
test(1)

# disable tracing
#
sys.settrace(None)

# don't trace this call
#
test(2)

call test 3 -> None
line test 3 -> None
line test 4 -> None
line test 5 -> None
line test 5 -> None
line test 6 -> None
line test 5 -> None
line test 7 -> None
return test 7 -> 1
```

, pdb (debug) .

1.13.5. /

stdin , stdout , stderr I/O . , print , . , (device), . Example 1-74 .

1.13.5.1. Example 1-74. sys

```
File: sys-stdout-example-1.py

import sys
import string

class Redirect:

    def _ _init_ _(self, stdout):
        self.stdout = stdout

    def write(self, s):
        self.stdout.write(string.lower(s))

# redirect standard output (including the print statement)
#                (    print    )
old_stdout = sys.stdout
sys.stdout = Redirect(sys.stdout)

print "HEJA SVERIGE",
print "FRISKT HUM\303\226R"

# restore standard output
#
sys.stdout = old_stdout

print "M\303\205\303\205\303\205\303\205L!"
```

```
heja sverige friskt hum\303\266r
M\303\205\303\205\303\205\303\205L!
```

, write .

(C Python softspace . , Python . Python , C , softspace .)

1.13.6.

, . , sys.exit , . Example 1-75 .

1.13.6.1. Example 1-75. sys

```
File: sys-exit-example-1.py

import sys

print "hello"

sys.exit(1)

print "there"
```

```
hello
```

sys.exit . *SystemExit* . sys.exit , Example 1-76 .

1.13.6.2. Example 1-76. sys.exit

```
File: sys-exit-example-2.py

import sys

print "hello"

try:
    sys.exit(1)
except SystemExit:
    pass

print "there"
```

```
hello
there
```

( ), " "(exit handler), . Example 1-77 .

1.13.6.3. Example 1-77. sys.exit

```
File: sys-exitfunc-example-1.py

import sys

def exitfunc():
    print "world"

sys.exitfunc = exitfunc

print "hello"
sys.exit(1)
print "there" # never printed #        print
```

```
hello
world
```

Python 2.0 , atexit .

1.14. atexit

(2.0) atexit ( ), .

register , , Example 1-78 . , exit .

1.14.0.1. Example 1-78. atexit

```
File: atexit-example-1.py

import atexit

def exit(*args):
    print "exit", args

# register two exit handler
atexit.register(exit)
atexit.register(exit, 1)
atexit.register(exit, "hello", "world")
```

```
exit ('hello', 'world')
exit (1,)
exit ()
```

sys.exitfunc (hook) .

1.15. time

time . C .

(, 1970.1.1 . Unix), struct ( ).

1.15.1.

Example 1-79 time .

1.15.1.1. Example 1-79. time

```
File: time-example-1.py

import time

now = time.time()

print now, "seconds since", time.gmtime(0)[:6]
print
print "or in other words:"
print "- local time:", time.localtime(now)
print "- utc:", time.gmtime(now)
```

```
937758359.77 seconds since (1970, 1, 1, 0, 0, 0)

or in other words:
- local time: (1999, 9, 19, 18, 25, 59, 6, 262, 1)
- utc: (1999, 9, 19, 16, 25, 59, 6, 262, 0)
```

localtime gmtime , , , , , , , , . (,), (0) , 1 1 .

1.15.2.

, time , Example 1-80 .

1.15.2.1. Example 1-80. time

```
File: time-example-2.py

import time

now = time.localtime(time.time())

print time.asctime(now)
print time.strftime("%y/%m/%d %H:%M", now)
print time.strftime("%a %b %d", now)
print time.strftime("%c", now)
print time.strftime("%I %p", now)
print time.strftime("%Y-%m-%d %H:%M:%S %Z", now)

# do it by hand...
year, month, day, hour, minute, second, weekday, yearday, daylight = now
print "%04d-%02d-%02d" % (year, month, day)
print "%02d:%02d:%02d" % (hour, minute, second)
print ("MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN")[weekday], yearday

Sun Oct 10 21:39:24 1999
99/10/10 21:39
Sun Oct 10
Sun Oct 10 21:39:24 1999
09 PM
1999-10-10 21:39:24 CEST
1999-10-10
21:39:24
SUN 283
```

1.15.3.

, time strptime , strftime . , , Example 1-81 .

1.15.3.1. Example 1-81. time.strptime

```
File: time-example-6.py

import time

# make sure we have a strptime function!
#          strptime
try:
    strptime = time.strptime
except AttributeError:
    from strptime import strptime

print strptime("31 Nov 00", "%d %b %y")
print strptime("1 Jan 70 1:30pm", "%d %b %y %I:%M%p")
```

C , time.strptime . , Example 1-82 .

1.15.3.2. Example 1-82. strptime

File: strptime.py

```
import re
import string

MONTHS = ["Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug",
          "Sep", "Oct", "Nov", "Dec"]

SPEC = {
    # map formatting code to a regular expression fragment
    "%a": "(?P<weekday>[a-z]+)",
    "%A": "(?P<weekday>[a-z]+)",
    "%b": "(?P<month>[a-z]+)",
    "%B": "(?P<month>[a-z]+)",
    "%C": "(?P<century>\d\d?)",
    "%d": "(?P<day>\d\d?)",
    "%D": "(?P<month>\d\d?)/(?P<day>\d\d?)/(?P<year>\d\d)",
    "%e": "(?P<day>\d\d?)",
    "%h": "(?P<month>[a-z]+)",
    "%H": "(?P<hour>\d\d?)",
    "%I": "(?P<hour12>\d\d?)",
    "%j": "(?P<yearday>\d\d?\d?)",
    "%m": "(?P<month>\d\d?)",
    "%M": "(?P<minute>\d\d?)",
    "%p": "(?P<ampm12>am|pm)",
    "%R": "(?P<hour>\d\d?):(?P<minute>\d\d?)",
    "%S": "(?P<second>\d\d?)",
    "%T": "(?P<hour>\d\d?):(?P<minute>\d\d?):(?P<second>\d\d?)",
    "%U": "(?P<week>\d\d)",
    "%w": "(?P<weekday>\d)",
    "%W": "(?P<weekday>\d\d)",
    "%y": "(?P<year>\d\d)",
    "%Y": "(?P<year>\d\d\d\d)",
    "%%": "%"
}

class TimeParser:
    def _ _init_ _(self, format):
        # convert strptime format string to regular expression
        format = string.join(re.split("(?:\s|%t|%n)+", format))
        pattern = []
        try:
            for spec in re.findall("%\w|%%|.", format):
                if spec[0] == "%":
                    spec = SPEC[spec]
                pattern.append(spec)
        except KeyError:
            raise ValueError, "unknown specificer: %s" % spec
        self.pattern = re.compile("(?i)" + string.join(pattern, ""))
    def match(self, daytime):
        # match time string
        match = self.pattern.match(daytime)
        if not match:
            raise ValueError, "format mismatch"
        get = match.groupdict().get
        tm = [0] * 9
        # extract date elements
        y = get("year")
        if y:
            y = int(y)
            if y < 68:
                y = 2000 + y
            elif y < 100:
                y = 1900 + y
            tm[0] = y
        m = get("month")
        if m:
            if m in MONTHS:
                m = MONTHS.index(m) + 1
            tm[1] = int(m)
        d = get("day")
        if d: tm[2] = int(d)
        # extract time elements
        h = get("hour")
        if h:
            tm[3] = int(h)
        else:
            h = get("hour12")
            if h:
                h = int(h)
                if string.lower(get("ampm12", "")) == "pm":
                    h = h + 12
                tm[3] = h
        m = get("minute")
        if m: tm[4] = int(m)
        s = get("second")
        if s: tm[5] = int(s)
        # ignore weekday/yearday for now
        return tuple(tm)

def strptime(string, format="%a %b %d %H:%M:%S %Y"):
    return TimeParser(format).match(string)

if _ _name_ _ == "_ _main_ _":
    # try it out
    import time
    print strptime("2000-12-20 01:02:03", "%Y-%m-%d %H:%M:%S")
    print strptime(time.ctime(time.time()))
```

```
(2000, 12, 20, 1, 2, 3, 0, 0, 0)
(2000, 11, 15, 12, 30, 45, 0, 0, 0)
```

1.15.4.

, (local time) . mktime , Example 1-83 .

1.15.4.1. Example 1-83. time ( )

```
File: time-example-3.py

import time

t0 = time.time()
tm = time.localtime(t0)

print tm

print t0
print time.mktime(tm)
```

```
(1999, 9, 9, 0, 11, 8, 3, 252, 1)
936828668.16
936828668.0
```

, 1.5.2 UTC (Universal Time, Coordinated:) (Python C). Example 1-84 Python , timegm .

1.15.4.2. Example 1-84. UTC ( )

```
File: time-example-4.py

import time

def _d(y, m, d, days=(0,31,59,90,120,151,181,212,243,273,304,334,365)):
    # map a date to the number of days from a reference point
    return (((y - 1901)*1461)/4 + days[m-1] + d +
        ((m > 2 and not y % 4 and (y % 100 or not y % 400)) and 1))

def timegm(tm, epoch=_d(1970,1,1)):
    year, month, day, h, m, s = tm[:6]
    assert year >= 1970
    assert 1 <= month <= 12
    return (_d(year, month, day) - epoch)*86400 + h*3600 + m*60 + s

t0 = time.time()
tm = time.gmtime(t0)

print tm

print t0
print timegm(tm)
```

```
(1999, 9, 8, 22, 12, 12, 2, 251, 0)
936828732.48
936828732
```

1.6 , calendar calendar.timegm .

1.15.5. Timing

time Python , Example 1-85 .
"wall time" (real world time), " " (CPU).

1.15.5.1. Example 1-85. time

```
File: time-example-5.py

import time

def procedure():
    time.sleep(2.5)

# measure process time
```

```
t0 = time.clock()
procedure()
print time.clock() - t0, "seconds process time"

# measure wall time
t0 = time.time()
procedure()
print time.time() - t0, "seconds wall time"
```

```
0.0 seconds process time
2.50903499126 seconds wall time
```

. (Windows), clock wall time.

. , 30 . (: On many systems, it wraps around after just over 30 minutes.)

timing (Windows , ~), wall time.

1.16. types

types , Example 1-86 . . is .

1.16.0.1. Example 1-86. types

```
File: types-example-1.py

import types

def check(object):
    print object,

    if type(object) is types.IntType:
        print "INTEGER",
    if type(object) is types.FloatType:
        print "FLOAT",
    if type(object) is types.StringType:
        print "STRING",
    if type(object) is types.ClassType:
        print "CLASS",
    if type(object) is types.InstanceType:
        print "INSTANCE",
    print

check(0)
check(0.0)
check("0")

class A:
    pass

class B:
    pass

check(A)
check(B)

a = A()
b = B()

check(a)
check(b)
```

```
0 INTEGER
0.0 FLOAT
0 STRING
A CLASS
B CLASS
<A instance at 796960> INSTANCE
<B instance at 796990> INSTANCE
```

, .
, issubclass isinstance .

types . ,
( ) .

1.17. gc

(, 2.0) gc .

Python ;
, .

2.0 , Python , .
, . Example 1-87
.

gc.collect .
.

1.17.0.1. Example 1-87. gc

```
File: gc-example-1.py

import gc

# create a simple object that links to itself
class Node:

    def _ _init_ _(self, name):
        self.name = name
        self.parent = None
        self.children = []

    def addchild(self, node):
        node.parent = self
        self.children.append(node)

    def _ _repr_ _(self):
        return "<Node %s at %x>" % (repr(self.name), id(self))

# set up a self-referencing structure
root = Node("monty")

root.addchild(Node("eric"))
root.addchild(Node("john"))
root.addchild(Node("michael"))

# remove our only reference
del root

print gc.collect(), "unreachable objects"
print gc.collect(), "unreachable objects"
```

```
12 unreachable objects
```

```
0 unreachable objects
```

, gc.disable , , Python 1.5.2 .

2.

"Now, imagine that your friend kept complaining that she didn't want to visit you since she found it too hard to climb up the drain pipe, and you kept telling her to use the friggin' stairs like everyone else..."
- eff-bot, June 1998

2.1.

Python . , Python , .

2.1.1.

fileinput . , for-in .

StringIO (cStringIO ,) . StringIO .

2.1.2.

UserDict , UserList , UserString . , . .

2.1.3.

random . whrandom , .

[!Feather : whrandom 2.1 . random .]

2.1.4.

md5 sha (cryptographically strong message signatures , "message digests",).

crypt DES . Unix .

rotor . 2.4 .

[!Feather : 2.3 , .]

2.2. fileinput

fileinput , Example 2-1 .

2.2.0.1. Example 2-1. fileinput

```
File: fileinput-example-1.py

import fileinput
import sys

for line in fileinput.input("samples/sample.txt"):
    sys.stdout.write("-> ")
    sys.stdout.write(line)

-> We will perhaps eventually be writing only small
-> modules which are identified by name as they are
-> used to build larger ones, so that devices like
-> indentation, rather than delimiters, might become
-> feasible for expressing local structure in the
-> source language.
->      -- Donald E. Knuth, December 1974
```

fileinput (meta information). isfirstline , filename , lineno , Example 2-2 .

2.2.0.2. Example 2-2. fileinput

File: fileinput-example-2.py

```
File: shutil-example-1.py

import shutil
import os

for file in os.listdir("."):
    if os.path.splitext(file)[1] == ".py":
        print file
        shutil.copy(file, os.path.join("backup", file))
```

```
aifc-example-1.py
anydbm-example-1.py
array-example-1.py
...
```

copytree (cp -r), rmtree (rm -r). Example 2-5 .

2.3.0.2. Example 2-5. shutil /

```
File: shutil-example-2.py

import shutil
import os

SOURCE = "samples"
BACKUP = "samples-bak"

# create a backup directory
shutil.copytree(SOURCE, BACKUP)

print os.listdir(BACKUP)

# remove it
shutil.rmtree(BACKUP)

print os.listdir(BACKUP)
```

```
['sample.wav', 'sample.jpg', 'sample.au', 'sample.msg', 'sample.tgz',
...
Traceback (most recent call last):
 File "shutil-example-2.py", line 17, in ?
   print os.listdir(BACKUP)
os.error: No such file or directory
```

2.4. tempfile

Example 2-6 tempfile
.

2.4.0.1. Example 2-6. tempfile

```
File: tempfile-example-1.py

import tempfile
import os

tempfile = tempfile.mktemp()

print "tempfile", "=>", tempfile

file = open(tempfile, "w+b")
file.write("*" * 1000)
file.seek(0)
print len(file.read()), "bytes"
file.close()

try:
    # must remove file when done
    os.remove(tempfile)
except OSError:
    pass

tempfile => C:\TEMP\~160-1
1000 bytes
```

TemporaryFile , , Example 2-7
. . (Unix ,
, . ,
.)

2.4.0.2. Example 2-7. tempfile

```
File: tempfile-example-2.py

import tempfile

file = tempfile.TemporaryFile()
```

```
for i in range(100):
    file.write("*" * 100)

file.close() # removes the file!
```

2.5. StringIO

Example 2-8 StringIO .
( ). .

2.5.0.1. Example 2-8. StringIO

```
File: stringio-example-1.py

import StringIO

MESSAGE = "That man is depriving a village somewhere of a computer
scientist."

file = StringIO.StringIO(MESSAGE)

print file.read()
```

```
That man is depriving a village somewhere of a computer scientist.
```

StringIO , getvalue
. Example 2-9 .

2.5.0.2. Example 2-9. StringIO

```
File: stringio-example-2.py

import StringIO

file = StringIO.StringIO()
file.write("This man is no ordinary man. ")
file.write("This is Mr. F. G. Superman.")

print file.getvalue()
```

```
This man is no ordinary man. This is Mr. F. G. Superman.
```

StringIO Python , Example 2-10 .

2.5.0.3. Example 2-10. StringIO

```
File: stringio-example-3.py

import StringIO
import string, sys

stdout = sys.stdout

sys.stdout = file = StringIO.StringIO()

print """
According to Gbaya folktales, trickery and guile
are the best ways to defeat the python, king of
snakes, which was hatched from a dragon at the
world's start. -- National Geographic, May 1997
"""

sys.stdout = stdout

print string.upper(file.getvalue())
```

```
ACCORDING TO GBAYA FOLKTALES, TRICKERY AND GUILE
ARE THE BEST WAYS TO DEFEAT THE PYTHON, KING OF
SNAKES, WHICH WAS HATCHED FROM A DRAGON AT THE
WORLD'S START. -- NATIONAL GEOGRAPHIC, MAY 1997
```

2.6. cStringIO

cStringIO , StringIO .
StringIO , . Example 2-11 cStringIO
, .

2.6.0.1. Example 2-11. cStringIO

```
File: cstringio-example-1.py

import cStringIO

MESSAGE = "That man is depriving a village somewhere of a computer
scientist."

file = cStringIO.StringIO(MESSAGE)
```

```
print file.read()

That man is depriving a village somewhere of a computer scientist.
```

, Python ,
cStringIO StringIO , Example 2-12 .

2.6.0.2. Example 2-12. StringIO

```
File: cstringio-example-2.py

try:
    import cStringIO
    StringIO = cStringIO
except ImportError:
    import StringIO

print StringIO

<module  'StringIO' (built-in)>
```

2.7. mmap

(2.0) mmap , Example 2-13
. , .

2.7.0.1. Example 2-13. mmap

```
File: mmap-example-1.py

import mmap
import os

filename = "samples/sample.txt"

file = open(filename, "r+")
size = os.path.getsize(filename)

data = mmap.mmap(file.fileno(), size)

# basics
print data
print len(data), size

# use slicing to read from the file
#
print repr(data[:10]), repr(data[:10])

# or use the standard file interface
#
print repr(data.read(10)), repr(data.read(10))
```

```
<mmap object at 008A2A10>
302 302
'We will pe' 'We will pe'
'We will pe' 'rhaps even'
```

Windows , (`r+` , `w+` , `a+`), mmap .

[!Feather : , a+ , r+ w+]

Example 2-14 , , .

2.7.0.2. Example 2-14.

File: mmap-example-2.py

```
import mmap
import os, string, re

def mapfile(filename):
    file = open(filename, "r+")
    size = os.path.getsize(filename)
    return mmap.mmap(file.fileno(), size)

data = mapfile("samples/sample.txt")

# search
index = data.find("small")
print index, repr(data[index-5:index+15])

# regular expressions work too!
m = re.search("small", data)
print m.start(), m.group()
```

```
43 'only small\015\012modules '
43 small
```

2.8. UserDict

UserDict (Python).

Example 2-15 , " /+" .

2.8.0.1. Example 2-15. UserDict

```
File: userdict-example-1.py

import UserDict

class FancyDict(UserDict.UserDict):

    def _ _init_ _(self, data = {}, **kw):
        UserDict.UserDict._ _init_ _(self)
        self.update(data)
        self.update(kw)

    def _ _add_ _(self, other):
        dict = FancyDict(self.data)
        dict.update(b)
        return dict

a = FancyDict(a = 1)
b = FancyDict(b = 2)

print a + b
```

```
{'b': 2, 'a': 1}
```

2.9. UserList

UserList (Python).

Example 2-16 , *AutoList* ,
.

2.9.0.1. Example 2-16. UserList

```
File: userlist-example-1.py

import UserList

class AutoList(UserList.UserList):

    def _ _setitem_ _(self, i, item):
        if i == len(self.data):
            self.data.append(item)
        else:
            self.data[i] = item

list = AutoList()

for i in range(10):
    list[i] = i

print list
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

2.10. UserString

(2.0) UserString , *UserString* *MutableString* .
, ,
( ).

MutableString ,
. ,
array . Example 2-17 UserString .

2.10.0.1. Example 2-17. UserString

```
File: userstring-example-1.py

import UserString

class MyString(UserString.MutableString):
    def append(self, s):
        self.data = self.data + s

    def insert(self, index, s):
        self.data = self.data[index:] + s + self.data[index:]

    def remove(self, s):
        self.data = self.data.replace(s, "")

file = open("samples/book.txt")
text = file.read()
file.close()

book = MyString(text)

for bird in ["gannet", "robin", "nuthatch"]:
    book.remove(bird)

print book
```

```
...
C: The one without the !
P: The one without the -!!! They've ALL got the !! It's a
Standard British Bird, the , it's in all the books!!!
...
```

2.11. traceback

Example 2-18 traceback
(Traceback) , . Example 2-18 .

2.11.0.1. Example 2-18. traceback

```
File: traceback-example-1.py

# note! importing the traceback module messes up the
# exception state, so you better do that here and not
# in the exception handler
#     !     traceback                    ,
#
import traceback
```

```
try:
    raise SyntaxError, "example"
except:
    traceback.print_exc()
```

```
Traceback (innermost last):
  File "traceback-example-1.py", line 7, in ?
SyntaxError: example
```

Example 2-19 StringIO .

2.11.0.2. Example 2-19. traceback

```
File: traceback-example-2.py

import traceback
import StringIO

try:
    raise IOError, "an i/o error occurred"
except:
    fp = StringIO.StringIO()
    traceback.print_exc(file=fp)
    message = fp.getvalue()

    print "failure! the error was:", repr(message)
```

```
failure! the error was: 'Traceback (innermost last):\012  File
"traceback-example-2.py", line 5, in ?\012IOError: an i/o error
occurred\012'
```

extract_tb , , Example 2-20 .

2.11.0.3. Example 2-20. traceback Module Traceback

```
File: traceback-example-3.py

import traceback
import sys

def function():
    raise IOError, "an i/o error occurred"

try:
    function()
except:
    info = sys.exc_info()
    for file, lineno, function, text in traceback.extract_tb(info[2]):
        print file, "line", lineno, "in", function
        print "=>", repr(text)
    print "** %s: %s" % info[:2]
```

```
traceback-example-3.py line 8 in ?
=> 'function()'
traceback-example-3.py line 5 in function
=> 'raise IOError, "an i/o error occurred"'
** exceptions.IOError: an i/o error occurred
```

2.12. errno

errno , ENOENT (" ")
EPERM (" "). .
Example 2-21 errno .

, *IOError* ,
. ,
.

2.12.0.1. Example 2-21. errno

```
File: errno-example-1.py

import errno

try:
    fp = open("no.such.file")
except IOError, (error, message):
    if error == errno.ENOENT:
        print "no such file"
    elif error == errno.EPERM:
        print "permission denied"
    else:
        print message
```

```
no such file
```

Example 2-22 , errorcode
(symbolic name).

2.12.0.2. Example 2-22. errorcode

```
File: errno-example-2.py

import errno

try:
    fp = open("no.such.file")
except IOError, (error, message):
    print error, repr(message)
    print errno.errorcode[error]

# 2 'No such file or directory'
# ENOENT
```

2.13. getopt

getopt ,
. Example 2-23 .

2 . (:)
.

2.13.0.1. Example 2-23. getopt

```
File: getopt-example-1.py

import getopt
import sys

# simulate command-line invocation
#
sys.argv = ["myscript.py", "-l", "-d", "directory", "filename"]

# process options
#
opts, args = getopt.getopt(sys.argv[1:], "ld:")

long = 0
directory = None
```

```
for o, v in opts:
    if o == "-l":
        long = 1
    elif o == "-d":
        directory = v

print "long", "=", long
print "directory", "=", directory
print "arguments", "=", args
```

```
long = 1
directory = directory
arguments = ['filename']
```

getopt , Example 2-24 , 3 . (=) , .

2.13.0.2. Example 2-24. getopt

```
File: getopt-example-2.py

import getopt
import sys

# simulate command-line invocation
#
sys.argv = ["myscript.py", "--echo", "--printer", "lp01", "message"]

opts, args = getopt.getopt(sys.argv[1:], "ep:", ["echo", "printer="])

# process options
#
echo = 0
printer = None

for o, v in opts:
    if o in ("-e", "--echo"):
        echo = 1
    elif o in ("-p", "--printer"):
        printer = v

print "echo", "=", echo
print "printer", "=", printer
print "arguments", "=", args
```

```
echo = 1
printer = lp01
arguments = ['message']
```

[!Feather : , :
myscript.py -e -p lp01 message
myscript.py --echo --printer=lp01 message
]

2.14. getpass

getpass . Example 2-25 .

getpass(prompt) , , . , "Password: ".

getuser() , .

2.14.0.1. Example 2-25. getpass

```
File: getpass-example-1.py

import getpass

usr = getpass.getuser()

pwd = getpass.getpass("enter password for user %s: " % usr)

print usr, pwd
```

```
enter password for user mulder:
mulder trustno1
```

2.15. glob

glob , Unix shell .

, . (*) ,
(?) . , [0-9]
. .

glob(pattern) . Example 2-26
.

2.15.0.1. Example 2-26. glob

File: glob-example-1.py

```
import glob

for file in glob.glob("samples/*.jpg"):
    print file
```

```
samples/sample.jpg
```

glob , os.listdir . glob
fnmatch .

2.16. fnmatch

fnmatch . Example 2-27 .

Unix shell . (*) ,
(?) . , [0-9]
. .

2.16.0.1. Example 2-27. fnmatch

File: fnmatch-example-1.py

```
import fnmatch
import os

for file in os.listdir("samples"):
    if fnmatch.fnmatch(file, "*.jpg"):
        print file
```

```
sample.jpg
```

Example 2-28 translate
.

2.16.0.2. Example 2-28. fnmatch

File: fnmatch-example-2.py

```
import fnmatch
import os, re

pattern = fnmatch.translate("*.jpg")

for file in os.listdir("samples"):
    if re.match(pattern, file):
        print file

print "(pattern was %s)" % pattern
```

```
sample.jpg
(pattern was .*\.jpg$)
```

glob find fnmatch .

2.17. random

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."
- John von Neumann, 1951

random .

(Wichmann Hill , 1982)
, Example 2-29 .

2.17.0.1. Example 2-29. random

File: random-example-1.py

```
import random

for i in range(5):
    # random float: 0.0 <= number < 1.0
    print random.random(),

    # random float: 10 <= number < 20
    print random.uniform(10, 20),

    # random integer: 100 <= number <= 1000
    print random.randint(100, 1000),

    # random integer: even numbers in 100 <= number < 1000
    print random.randrange(100, 1000, 2)
```

```
0.946842713956 19.5910069381 709 172
0.573613195398 16.2758417025 407 120
0.363241598013 16.8079747714 916 580
0.602115173978 18.386796935 531 774
0.526767588533 18.0783794596 223 344
```

randint , .
.

Example 2-30 choice , .
, , (,).

2.17.0.2. Example 2-30. random

```
File: random-example-2.py

import random

# random choice from a list
for i in range(5):
    print random.choice([1, 2, 3, 5, 9])
```

```
2
3
1
9
1
```

2.0 , shuffle (
). Example 2-31 .

2.17.0.3. Example 2-31. random

```
File: random-example-4.py

import random

try:
    # available in 2.0 and later
    shuffle = random.shuffle
except AttributeError:
    def shuffle(x):
        for i in xrange(len(x)-1, 0, -1):
            # pick an element in x[:i+1] with which to exchange x[i]
            j = int(random.random() * (i+1))
            x[i], x[j] = x[j], x[i]

cards = range(52)

shuffle(cards)

myhand = cards[:5]

print myhand
```

```
[4, 8, 40, 12, 30]
```

random . Example 2-32
gauss ( ) .

2.17.0.4. Example 2-32. random

```
File: random-example-3.py

import random

histogram = [0] * 20

# calculate histogram for gaussian
# noise, using average=5, stddev=1
for i in range(1000):
    i = int(random.gauss(5, 1) * 2)
    histogram[i] = histogram[i] + 1

# print the histogram
m = max(histogram)
for v in histogram:
    print "*" * (v * 50 / m)
```

```
****
**********
*************************
***********************************
************************************************
**************************************************
*************************************
***************************
*************
***
*
```

Python Library Reference
.

.

, , , .

.

2.18. whrandom

2.1 , . random .
- Feather

Example 2-33 whrandom , . (
Wichmann Hill, 1982).
( ), random .

2.18.0.1. Example 2-33. whrandom

File: whrandom-example-1.py

```
import whrandom

# same as random
print whrandom.random()
print whrandom.choice([1, 2, 3, 5, 9])
print whrandom.uniform(10, 20)
print whrandom.randint(100, 1000)
```

```
0.113412062346
```

```
1
16.8778954689
799
```

Example 2-34 *whrandom* .

2.18.0.2. Example 2-34. whrandom

```
File: whrandom-example-2.py

import whrandom

# initialize all generators with the same seed
rand1 = whrandom.whrandom(4,7,11)
rand2 = whrandom.whrandom(4,7,11)
rand3 = whrandom.whrandom(4,7,11)

for i in range(5):
    print rand1.random(), rand2.random(), rand3.random()

0.123993532536 0.123993532536 0.123993532536
0.180951499518 0.180951499518 0.180951499518
0.291924111809 0.291924111809 0.291924111809
0.952048889363 0.952048889363 0.952048889363
0.969794283643 0.969794283643 0.969794283643
```

2.19. md5

md5 (Message-Digest Algorithm 5) ( ).

md5 128 . , md5 . , md5 , . Example 2-35 md5 .

2.19.0.1. Example 2-35. md5

```
File: md5-example-1.py

import md5

hash = md5.new()
hash.update("spam, spam, and eggs")
```

```
print repr(hash.digest())
```

```
 'L\005J\243\266\355\243u`\305r\203\267\020F\303'
```

. Example 2-36 base64 .

2.19.0.2. Example 2-36. md5 base64 md5

```
File: md5-example-2.py

import md5
import string
import base64

hash = md5.new()
hash.update("spam, spam, and eggs")

value = hash.digest()
print hash.hexdigest()
# before 2.0, the above can be written as
#    2.0    ,               :
# print string.join(map(lambda v: "%02x" % ord(v), value), "")

print base64.encodestring(value)
```

```
4c054aa3b6eda37560c57283b71046c3
TAVKo7bto3VgxXKDtxBGww==
```

Example 2-37 md5 ( ).

2.19.0.3. Example 2-37. md5

```
File: md5-example-3.py

import md5
import string, random

def getchallenge():
    # generate a 16-byte long random string.  (note that the built-
    # in pseudo-random generator uses a 24-bit seed, so this is not
    # as good as it may seem...)
    #          16                    .
    #          24          (seed),                     ..
    challenge = map(lambda i: chr(random.randint(0, 255)), range(16))
    return string.join(challenge, "")

def getresponse(password, challenge):
    # calculate combined digest for password and challenge
    #                (challenge)
    m = md5.new()
    m.update(password)
    m.update(challenge)
    return m.digest()

#
# server/client communication
#       /

# 1. client connects.  server issues challenge.
# 1.            ,               (challenge)

print "client:", "connect"

challenge = getchallenge()

print "server:", repr(challenge)

# 2. client combines password and challenge, and calculates
# the response.
# 2.                      (challenge)

client_response = getresponse("trustno1", challenge)

print "client:", repr(client_response)

# 3. server does the same, and compares the result with the
# client response.  the result is a safe login in which the
# password is never sent across the communication channel.
# 3.                  ,                           ,
#                      .                                .

server_response = getresponse("trustno1", challenge)

if server_response == client_response:
    print "server:", "login ok"
```

```
client: connect
server: '\334\352\227Z#\272\273\212KG\330\265\032>\311o'
client: "I'\305\240-x\245\237\035\225A\254\233\337\225\001"
server: login ok
```

Example 2-38 md5 , ( ).

2.19.0.4. Example 2-38. md5

File: md5-example-4.py

```
import md5
import array

class HMAC_MD5:
    # keyed md5 message authentication

    def _ _init_ _(self, key):
        if len(key) > 64:
            key = md5.new(key).digest()
        ipad = array.array("B", [0x36] * 64)
        opad = array.array("B", [0x5C] * 64)
        for i in range(len(key)):
            ipad[i] = ipad[i] ^ ord(key[i])
            opad[i] = opad[i] ^ ord(key[i])
        self.ipad = md5.md5(ipad.tostring())
        self.opad = md5.md5(opad.tostring())

    def digest(self, data):
        ipad = self.ipad.copy()
        opad = self.opad.copy()
        ipad.update(data)
        opad.update(ipad.digest())
        return opad.digest()

#
# simulate server end
#

key = "this should be a well-kept secret"
message = open("samples/sample.txt").read()

signature = HMAC_MD5(key).digest(message)
```

```
# (send message and signature across a public network)
# (                         )

#
# simulate client end
#

key = "this should be a well-kept secret"

client_signature = HMAC_MD5(key).digest(message)

if client_signature == signature:
    print "this is the original message:"
    print
    print message
else:
    print "someone has modified the message!!!"
```

copy (snapshot). (Example 2-38 padded key).

HMAC-MD5:Keyed-MD5 for Message Authentication (http://www.research.ibm.com/security/draft-ietf-ipsec-hmac-md5-00.txt) by Krawczyk, .

2.20. sha

sha ( ) , Example 2-39 . md5 , 160 .

2.20.0.1. Example 2-39. sha

```
File: sha-example-1.py

import sha

hash = sha.new()
hash.update("spam, spam, and eggs")

print repr(hash.digest())
print hash.hexdigest()

'\321\333\003\026I\331\272-j\303\247\240\345\343Tvq\364\346\311'
d1db031649d9ba2d6ac3a7a0e5e3547671f4e6c9
```

sha , md5 .

2.21. crypt

(, Unix) crypt DES , Unix , .

Example 2-40 crypt.crypt , *salt* , *salt* . .

2.21.0.1. Example 2-40. crypt

```
File: crypt-example-1.py

import crypt

import random, string

def getsalt(chars = string.letters + string.digits):
    # generate a random 2-character 'salt'
    #           2      'salt'
    return random.choice(chars) + random.choice(chars)

print crypt.crypt("bananas", getsalt())

'py8UGrijma1j6'
```

, , *salt* . , . Example 2-41 pwd .

2.21.0.2. Example 2-41. crypt

```
File: crypt-example-2.py

import pwd, crypt
```

```
def login(user, password):
    "Check if user would be able to log in using password"
    try:
        pw1 = pwd.getpwnam(user)[1]
        pw2 = crypt.crypt(password, pw1[:2])
        return pw1 == pw2
    except KeyError:
        return 0 # no such user

user = raw_input("username:")
password = raw_input("password:")

if login(user, password):
    print "welcome", user
else:
    print "login failed"
```

md5 .

2.22. rotor

2.3 , 2.4 . . - Feather

( ) rotor . Example 2-42 . WWII Enigma engine.

2.22.0.1. Example 2-42. rotor

```
File: rotor-example-1.py

import rotor

SECRET_KEY = "spam"
MESSAGE = "the holy grail"

r = rotor.newrotor(SECRET_KEY)

encoded_message = r.encrypt(MESSAGE)
decoded_message = r.decrypt(encoded_message)
```

```
print "original:", repr(MESSAGE)
print "encoded message:", repr(encoded_message)
print "decoded message:", repr(decoded_message)

original: 'the holy grail'
encoded message: '\227\271\244\015\305sw\3340\337\252\237\340U'
decoded message: 'the holy grail'
```

2.23. zlib

( ) zlib "zlib" . ("deflate".)

Example 2-43 compress decompress
.

2.23.0.1. Example 2-43. zlib

```
File: zlib-example-1.py

import zlib

MESSAGE = "life of brian"

compressed_message = zlib.compress(MESSAGE)
decompressed_message = zlib.decompress(compressed_message)

print "original:", repr(MESSAGE)
print "compressed message:", repr(compressed_message)
print "decompressed message:", repr(decompressed_message)

original: 'life of brian'
compressed message:
'x\234\313\311LKU\310OSH*\312L\314\003\000!\010\004\302'
decompressed message: 'life of brian'
```

, Example 2-44 .

2.23.0.2. Example 2-44. zlib

```
File: zlib-example-2.py

import zlib
import glob
```

```
for file in glob.glob("samples/*"):

    indata = open(file, "rb").read()
    outdata = zlib.compress(indata, zlib.Z_BEST_COMPRESSION)

    print file, len(indata), "=>", len(outdata),
    print "%d%%" % (len(outdata) * 100 / len(indata))

samples\sample.au 1676 => 1109 66%
samples\sample.gz 42 => 51 121%
samples\sample.htm 186 => 135 72%
samples\sample.ini 246 => 190 77%
samples\sample.jpg 4762 => 4632 97%
samples\sample.msg 450 => 275 61%
samples\sample.sgm 430 => 321 74%
samples\sample.tar 10240 => 125 1%
samples\sample.tgz 155 => 159 102%
samples\sample.txt 302 => 220 72%
samples\sample.wav 13260 => 10992 82%
```

, Example 2-45 .

2.23.0.3. Example 2-45. zlib

```
File: zlib-example-3.py

import zlib

encoder = zlib.compressobj()

data = encoder.compress("life")
data = data + encoder.compress(" of ")
data = data + encoder.compress("brian")
data = data + encoder.flush()

print repr(data)
print repr(zlib.decompress(data))

'x\234\313\311LKU\310OSH*\312L\314\003\000!\010\004\302'
'life of brian'
```

Example 2-46 , , .

2.23.0.4. Example 2-46.

File: zlib-example-4.py

```
import zlib
import string, StringIO

class ZipInputStream:

    def _ _init_ _(self, file):
        self.file = file
        self._ _rewind()

    def _ _rewind(self):
        self.zip = zlib.decompressobj()
        self.pos = 0 # position in zipped stream
        self.offset = 0 # position in unzipped stream
        self.data = ""

    def _ _fill(self, bytes):
        if self.zip:
            # read until we have enough bytes in the buffer
            while not bytes or len(self.data) < bytes:
                self.file.seek(self.pos)
                data = self.file.read(16384)
                if not data:
                    self.data = self.data + self.zip.flush()
                    self.zip = None # no more data
                    break
                self.pos = self.pos + len(data)
                self.data = self.data + self.zip.decompress(data)

    def seek(self, offset, whence=0):
        if whence == 0:
            position = offset
        elif whence == 1:
            position = self.offset + offset
        else:
            raise IOError, "Illegal argument"
        if position < self.offset:
            raise IOError, "Cannot seek backwards"

        # skip forward, in 16k blocks
        while position > self.offset:
            if not self.read(min(position - self.offset, 16384)):
                break

    def tell(self):
        return self.offset

    def read(self, bytes = 0):
        self._ _fill(bytes)
        if bytes:
            data = self.data[:bytes]
            self.data = self.data[bytes:]
        else:
            data = self.data
            self.data = ""
        self.offset = self.offset + len(data)
        return data

    def readline(self):
        # make sure we have an entire line
        while self.zip and "\n" not in self.data:
            self._ _fill(len(self.data) + 512)
        i = string.find(self.data, "\n") + 1
        if i <= 0:
            return self.read()
        return self.read(i)

    def readlines(self):
        lines = []
        while 1:
            s = self.readline()
            if not s:
                break
            lines.append(s)
        return lines

#
# try it out

data = open("samples/sample.txt").read()
data = zlib.compress(data)

file = ZipInputStream(StringIO.StringIO(data))
for line in file.readlines():
    print line[:-1]
```

We will perhaps eventually be writing only small
modules which are identified by name as they are
used to build larger ones, so that devices like
indentation, rather than delimiters, might become
feasible for expressing local structure in the
source language.
-- Donald E. Knuth, December 1974

2.24. code

code .

compile_command compile ,
Python .

Example 2-47 , ,
(code object). :

```
a = (
  1,
  2,
  3
)
print a
```

2 , .

2.24.0.1. Example 2-47. code

File: code-example-1.py

```
import code
import string

#
SCRIPT = [
    "a = (",
    "  1,",
    "  2,",
    "  3 ",
    ")",
    "print a"
]

script = ""

for line in SCRIPT:
    script = script + line + "\n"
    co = code.compile_command(script, "<stdin>", "exec")
    if co:
        # got a complete statement.  execute it!
        print "-"*40
        print script,
        print "-"*40
        exec co
        script = ""

----------------------------------------
a = (
  1,
  2,
  3
)
----------------------------------------
----------------------------------------
print a
----------------------------------------
(1, 2, 3)
```

InteractiveConsole , Python
.

( ) (
push). raw_input .
, . Example 2-48 code
.

2.24.0.2. Example 2-48. code

```
File: code-example-2.py

import code

console = code.InteractiveConsole()
console.interact()

Python 1.5.2
```

```
Copyright 1991-1995 Stichting Mathematisch Centrum, Amsterdam
(InteractiveConsole)
>>> a = (
...     1,
...     2,
...     3
... )
>>> print a
(1, 2, 3)
```

Example 2-49 keyboard .
.

2.24.0.3. Example 2-49. code Debugging

File: code-example-3.py

```
def keyboard(banner=None):
    import code, sys

    # use exception trick to pick up the current frame
    try:
        raise None
    except:
        frame = sys.exc_info()[2].tb_frame.f_back

    # evaluate commands in current namespace
    namespace = frame.f_globals.copy()
    namespace.update(frame.f_locals)

    code.interact(banner=banner, local=namespace)

def func():
    print "START"
    a = 10
    keyboard()
    print "END"

func()
```

```
START
Python 1.5.2
Copyright 1991-1995 Stichting Mathematisch Centrum, Amsterdam
(InteractiveConsole)
>>> print a
```

```
10
>>> print keyboard
<function keyboard at 9032c8>
^Z
END
```

3.

"Well, since you last asked us to stop, this thread has moved from discussing languages suitable for professional programmers via accidental users to computer-phobic users. A few more iterations can make this thread really interesting..."
- eff-bot, June 1996

3.1.

Python .
, Python .

Unix Windows .

3.1.1.

Python , .
, .

, . .
, .

, Python *global interpreter lock (* *)* .
Python ; Python
, (socket
,).

.
. :

```
def getitem(key):
    item = cache.get(key)
    if item is None:
        # not in cache; create a new one
        item = create_new_item(key)
        cache[key] = item
    return item
```

key getitem , create_new_item . , .

lock objects . *lock object* , *getitem* .

3.1.2.

, *(process)* . shell / . Python .

os . 1.4.4 .

3.2. threading

( ) threading , Example 3-1 . Java . thread , .

Thread , run , . , start . run .

3.2.0.1. Example 3-1. threading

```
File: threading-example-1.py

import threading
import time, random

class Counter:
    def _ _init_ _(self):
        self.lock = threading.Lock()
        self.value = 0

    def increment(self):
        self.lock.acquire() # critical section
        self.value = value = self.value + 1
        self.lock.release()
        return value

counter = Counter()

class Worker(threading.Thread):

    def run(self):
        for i in range(10):
            # pretend we're doing something that takes 10?00 ms
            value = counter.increment() # increment global counter
            time.sleep(random.randint(10, 100) / 1000.0)
            print self.getName(), "-- task", i, "finished", value

#
# try it

for i in range(10):
    Worker().start() # start a worker
```

```
Thread-1 -- task 0 finished 1
Thread-3 -- task 0 finished 3
Thread-7 -- task 0 finished 8
Thread-1 -- task 1 finished 7
Thread-4 -- task 0 Thread-5 -- task 0 finished 4
finished 5
Thread-8 -- task 0 Thread-6 -- task 0 finished 9
finished 6
...
Thread-6 -- task 9 finished 98
Thread-4 -- task 9 finished 99
Thread-9 -- task 9 finished 100
```

Example 3-1 *Lock* *Counter* (critical section). acquire release , Counter 100.

3.3. Queue

Queue (queue) , Example 3-2 .
.

3.3.0.1. Example 3-2. Queue

File: queue-example-1.py

```
import threading
import Queue
import time, random

WORKERS = 2

class Worker(threading.Thread):

    def _ _init_ _(self, queue):
        self._ _queue = queue
        threading.Thread._ _init_ _(self)

    def run(self):
        while 1:
            item = self._ _queue.get()
            if item is None:
                break # reached end of queue

            # pretend we're doing something that takes 10?00 ms
            time.sleep(random.randint(10, 100) / 1000.0)

            print "task", item, "finished"

#
# try it

queue = Queue.Queue(0)

for i in range(WORKERS):
    Worker(queue).start() # start a worker

for i in range(10):
    queue.put(i)

for i in range(WORKERS):
    queue.put(None) # add end-of-queue markers
```

```
task 1 finished
task 0 finished
task 3 finished
task 2 finished
task 4 finished
task 5 finished
task 7 finished
task 6 finished
task 9 finished
task 8 finished
```

Example 3-3 . ,
(producer threads) , (pop off).

3.3.0.2. Example 3-3. Queue

File: queue-example-2.py

```
import threading
import Queue

import time, random

WORKERS = 2

class Worker(threading.Thread):

    def _ _init_ _(self, queue):
        self._ _queue = queue
        threading.Thread._ _init_ _(self)

    def run(self):
        while 1:
            item = self._ _queue.get()
            if item is None:
                break # reached end of queue

            # pretend we're doing something that takes 10?00 ms
            time.sleep(random.randint(10, 100) / 1000.0)

            print "task", item, "finished"

#
```

```
# run with limited queue

queue = Queue.Queue(3)

for i in range(WORKERS):
    Worker(queue).start() # start a worker

for item in range(10):
    print "push", item
    queue.put(item)

for i in range(WORKERS):
    queue.put(None) # add end-of-queue markers
```

```
push 0
push 1
push 2
push 3
push 4
push 5
task 0 finished
push 6
task 1 finished
push 7
task 2 finished
push 8
task 3 finished
push 9
task 4 finished
task 6 finished
task 5 finished
task 7 finished
task 9 finished
task 8 finished
```

Queue . Example 3-4
. ,
( ).

3.3.0.3. Example 3-4. Queue

File: queue-example-3.py

```
import Queue
import bisect
```

```
Empty = Queue.Empty

class PriorityQueue(Queue.Queue):
    "Thread-safe priority queue"

    def _put(self, item):
        # insert in order
        bisect.insort(self.queue, item)

#
# try it

queue = PriorityQueue(0)

# add items out of order
queue.put((20, "second"))
queue.put((10, "first"))
queue.put((30, "third"))

# print queue contents
try:
    while 1:
        print queue.get_nowait()
except Empty:
    pass
```

```
third
second
first
```

Example 3-5 (stack) (, ,
,).

3.3.0.4. Example 3-5. Queue

```
File: queue-example-4.py

import Queue

Empty = Queue.Empty

class Stack(Queue.Queue):
    "Thread-safe stack"
    def _put(self, item):
        # insert at the beginning of queue, not at the end
        self.queue.insert(0, item)

    # method aliases
    push = Queue.Queue.put
    pop = Queue.Queue.get
    pop_nowait = Queue.Queue.get_nowait

#
# try it

stack = Stack(0)

# push items on stack
stack.push("first")
stack.push("second")
stack.push("third")

# print stack contents
try:
    while 1:
        print stack.pop_nowait()
except Empty:
    pass
```

```
third
second
first
```

3.4. thread

( ) thread (low_level) , Example 3-6 . . , threading .

3.4.0.1. Example 3-6. thread

```
File: thread-example-1.py

import thread
import time, random
```

```
def worker():
    for i in range(50):
        # pretend we're doing something that takes 10?00 ms
        time.sleep(random.randint(10, 100) / 1000.0)
        print thread.get_ident(), "-- task", i, "finished"

#
# try it out!

for i in range(2):
    thread.start_new_thread(worker, ())

time.sleep(1)

print "goodbye!"

311 -- task 0 finished
265 -- task 0 finished
265 -- task 1 finished
311 -- task 1 finished
...
265 -- task 17 finished
311 -- task 13 finished
265 -- task 18 finished
goodbye!
```

, . threading
. ( )

3.5. commands

(Unix) commands . Example 3-7
.

3.5.0.1. Example 3-7. commands

```
File: commands-example-1.py

import commands

stat, output = commands.getstatusoutput("ls -lR")
```

```
print "status", "=>", stat
print "output", "=>", len(output), "bytes"
```

```
status => 0
output => 171046 bytes
```

3.6. pipes

(Unix) pipes " (conversion pipelines)" . . Example 3-8 .

3.6.0.1. Example 3-8. pipes

```
File: pipes-example-1.py

import pipes

t = pipes.Template()

# create a pipeline
#    " - "
t.append("sort", "--")
t.append("uniq", "--")

# filter some text
#
t.copy("samples/sample.txt", "")
```

```
Alan Jones (sensible party)
Kevin Phillips-Bong (slightly silly)
Tarquin
Fin-tim-lin-bin-whin-bim-lin-bus-stop-F'tang-F'tang-Olé-Biscuitbarrel
```

3.7. popen2

popen2 , stdin stdout (stderr).

python 1.5.2 , Unix . 2.0 , Windows . Example 3-9 .

3.7.0.1. Example 3-9. popen2 Module to Sort Strings

File: popen2-example-1.py

```
import popen2, string

fin, fout = popen2.popen2("sort")

fout.write("foo\n")
fout.write("bar\n")
fout.close()

print fin.readline(),
print fin.readline(),
fin.close()
```

```
bar
foo
```

Example 3-10 .

3.7.0.2. Example 3-10. popen2 gnuchess

File: popen2-example-2.py

```
import popen2
import string

class Chess:
    "Interface class for chesstool-compatible programs"

    def _ _init_ _(self, engine = "gnuchessc"):
        self.fin, self.fout = popen2.popen2(engine)
        s = self.fin.readline()
        if s != "Chess\n":
            raise IOError, "incompatible chess program"

    def move(self, move):
        self.fout.write(move + "\n")
        self.fout.flush()
        my = self.fin.readline()
        if my == "Illegal move":
            raise ValueError, "illegal move"
        his = self.fin.readline()
        return string.split(his)[2]

    def quit(self):
        self.fout.write("quit\n")
        self.fout.flush()

#
# play a few moves

g = Chess()

print g.move("a2a4")
print g.move("b2b3")

g.quit()
```

```
b8c6
e7e5
```

3.8. signal

signal (signal handler), Example 3-11 . , .

3.8.0.1. Example 3-11. signal

```
File: signal-example-1.py

import signal
import time

def handler(signo, frame):
    print "got signal", signo

signal.signal(signal.SIGALRM, handler)

# wake me up in two seconds
signal.alarm(2)
```

```
now = time.time()

time.sleep(200)

print "slept for", time.time() - now, "seconds"

got signal 14
slept for 1.99262607098 seconds
```

4.

"PALO ALTO, Calif. - Intel says its Pentium Pro and new Pentium II chips have a flaw that can cause computers to sometimes make mistakes but said the problems could be fixed easily with rewritten software."
- Reuters telegram

4.1.

Python . / Python .

4.1.1.

Python / . struct (C struct) Python . array (C arrays) Python .

4.1.2.

marshal pickle Python / .

marshal (Self-Describing Formats), , code . Python (.pyc).

pickle , , . pickle Python , , cPickle , C , marshal .

4.1.3.

, repr %
.

pprint Python ( ).

repr .
: 30 ,
, .

4.1.4.

Python , base64 , binhex (Macintosh
) , quoted printable , uu .

4.2. array

array . ,
. .

Examples 4-1 4-5 . Example 4-1 *array*
, tostring (internal buffer) .

4.2.0.1. Example 4-1. array

```
File: array-example-1.py

import array

a = array.array("B", range(16)) # unsigned char
b = array.array("h", range(16)) # signed short

print a
print repr(a.tostring())

print b
print repr(b.tostring())
```

```
array('B', [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])
```

```
'\000\001\002\003\004\005\006\007\010\011\012\013\014\015\016\017'

array('h', [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15])
'\000\000\001\000\002\000\003\000\004\000\005\000\006\000\007\000
\010\000\011\000\012\000\013\000\014\000\015\000\016\000\017\000'
```

array , Example 4-2 . ,
.

4.2.0.2. Example 4-2.

File: array-example-2.py

```
import array

a = array.array("B", [1, 2, 3])

a.append(4)

a = a + a

a = a[2:-2]

print a
print repr(a.tostring())
for i in a:
    print i,

array('B', [3, 4, 1, 2])
'\003\004\001\002'
3 4 1 2
```

(,
) , Example 4-3 .

4.2.0.3. Example 4-3.

File: array-example-3.py

```
import array

a = array.array("i", "fish license") # signed integer

print a
print repr(a.tostring())
print a.tolist()
```

```
array('i', [1752394086, 1667853344, 1702063717])
'fish license'
[1752394086, 1667853344, 1702063717]
```

, Example 4-4
(endianess) .

4.2.0.4. Example 4-4. array

```
File: array-example-4.py

import array

def little_endian():
    return ord(array.array("i",[1]).tostring()[0])

if little_endian():
    print "little-endian platform (intel, alpha)"
else:
    print "big-endian platform (motorola, sparc)"
```

```
big-endian platform (motorola, sparc)
```

Python 2.0 sys.byteorder ,
("little " "big "), Example 4-5 .

4.2.0.5. Example 4-5. sys.byteorder (Python 2.0)

```
File: sys-byteorder-example-1.py

import sys

# 2.0 and later
if sys.byteorder == "little":
    print "little-endian platform (intel, alpha)"
else:
    print "big-endian platform (motorola, sparc)"
```

```
big-endian platform (motorola, sparc)
```

4.3. struct

struct Python . pack
, . upack
, . Example 4-6 .

4.3.0.1. Example 4-6. struct

```
File: struct-example-1.py

import struct

# native byteorder
buffer = struct.pack("ihb", 1, 2, 3)
print repr(buffer)
print struct.unpack("ihb", buffer)

# data from a sequence, network byteorder
data = [1, 2, 3]
buffer = apply(struct.pack, ("!ihb",) + tuple(data))
print repr(buffer)
print struct.unpack("!ihb", buffer)

# in 2.0, the apply statement can also be written as:
# buffer = struct.pack("!ihb", *data)
```

```
'\001\000\000\000\002\000\003'
(1, 2, 3)
'\000\000\000\001\000\002\003'
(1, 2, 3)
```

4.4. xdrlib

xdrlib Python Sun external data
representation (XDR) , Example 4-7 .

4.4.0.1. Example 4-7. xdrlib

```
File: xdrlib-example-1.py

import xdrlib
```

```
#
# create a packer and add some data to it

p = xdrlib.Packer()
p.pack_uint(1)
p.pack_string("spam")

data = p.get_buffer()

print "packed:", repr(data)

#
# create an unpacker and use it to decode the data

u = xdrlib.Unpacker(data)

print "unpacked:", u.unpack_uint(), repr(u.unpack_string())

u.done()
```

```
packed: '\000\000\000\001\000\000\000\004spam'
unpacked: 1 'spam'
```

Sun remote procedure call (RPC) XDR . Example 4-8 , RPC .

4.4.0.2. Example 4-8. xdrlib RPC

File: xdrlib-example-2.py

```
import xdrlib

# some constants (see the RPC specs for details)
RPC_CALL = 1
RPC_VERSION = 2

MY_PROGRAM_ID = 1234 # assigned by Sun
MY_VERSION_ID = 1000
MY_TIME_PROCEDURE_ID = 9999

AUTH_NULL = 0

transaction = 1

p = xdrlib.Packer()
```

```
# send a Sun RPC call package
p.pack_uint(transaction)
p.pack_enum(RPC_CALL)
p.pack_uint(RPC_VERSION)
p.pack_uint(MY_PROGRAM_ID)
p.pack_uint(MY_VERSION_ID)
p.pack_uint(MY_TIME_PROCEDURE_ID)
p.pack_enum(AUTH_NULL)
p.pack_uint(0)
p.pack_enum(AUTH_NULL)
p.pack_uint(0)

print repr(p.get_buffer())
```

```
'\000\000\000\001\000\000\000\001\000\000\000\002\000\000\004\322
\000\000\003\350\000\000\'\017\000\000\000\000\000\000\000\000\000
\000\000\000\000\000\000\000'
```

4.5. marshal

marshal - ,
. Example 4-9 .

marshal . ,
, .
(little-endian order) , (
), .

4.5.0.1. Example 4-9. marshal

```
File: marshal-example-1.py

import marshal

value = (
    "this is a string",
    [1, 2, 3, 4],
    ("more tuples", 1.0, 2.3, 4.5),
    "this is yet another string"
    )

data = marshal.dumps(value)
```

```
# intermediate format
print type(data), len(data)

print "-"*50
print repr(data)
print "-"*50

print marshal.loads(data)
```

```
<type 'string'> 118
--------------------------------------------------
'(\004\000\000\000s\020\000\000\000this is a string
[\004\000\000\000i\001\000\000\000i\002\000\000\000
i\003\000\000\000i\004\000\000\000(\004\000\000\000
s\013\000\000\000more tuplesf\0031.0f\0032.3f\0034.
5s\032\000\000\000this is yet another string'
--------------------------------------------------
('this is a string', [1, 2, 3, 4], ('more tuples',
1.0, 2.3, 4.5), 'this is yet another string')
```

marshal code (Python).
Example 4-10 .

4.5.0.2. Example 4-10. marshal

```
File: marshal-example-2.py

import marshal

script = """
print 'hello'
"""

code = compile(script, "<script>", "exec")

data = marshal.dumps(code)

# intermediate format
print type(data), len(data)

print "-"*50
print repr(data)
print "-"*50
```

```
exec marshal.loads(data)

<type 'string'> 81
--------------------------------------------------
'c\000\000\000\000\001\000\000\000s\017\000\000\00
0\177\000\000\177\002\000d\000\000GHd\001\000S(\00
2\000\000\000s\005\000\000\000helloN(\000\000\000\
000(\000\000\000\000s\010\000\000\000<script>s\001
\000\000\000?\002\000s\000\000\000\000'
--------------------------------------------------
hello
```

4.6. pickle

pickle marshal , , .
marshal , , , .

4.6.0.1. Example 4-11. pickle

```
File: pickle-example-1.py

import pickle

value = (
    "this is a string",
    [1, 2, 3, 4],
    ("more tuples", 1.0, 2.3, 4.5),
    "this is yet another string"
    )

data = pickle.dumps(value)

# intermediate format
print type(data), len(data)

print "-"*50
print data
print "-"*50

print pickle.loads(data)

<type 'string'> 121
--------------------------------------------------
```

```
(S'this is a string'
p0
(lp1
I1
aI2
aI3
aI4
a(S'more tuples'
p2
F1.0
F2.3
F4.5
tp3
S'this is yet another string'
p4
tp5
.
--------------------------------------------------
('this is a string', [1, 2, 3, 4], ('more tuples',
1.0, 2.3, 4.5), 'this is yet another string')
```

, pickle code (copy_reg
).

, pickle . ,
, . Example
4-12 .

4.6.0.2. Example 4-12. pickle

File: pickle-example-2.py

```
import pickle
import math

value = (
    "this is a long string" * 100,
    [1.2345678, 2.3456789, 3.4567890] * 100
    )

# text mode
data = pickle.dumps(value)
print type(data), len(data), pickle.loads(data) == value

# binary mode
```

```
data = pickle.dumps(value, 1)
print type(data), len(data), pickle.loads(data) == value
```

4.7. cPickle

(,) cPickle pickle . Example 4-13 .

4.7.0.1. Example 4-13. cPickle

```
File: cpickle-example-1.py

try:
    import cPickle
    pickle = cPickle
except ImportError:
    import pickle
```

4.8. copy_reg

copy_reg . pickle copy .

, pickle Python code , :

```
File: copy-reg-example-1.py

import pickle

CODE = """
print 'good evening'
"""

code = compile(CODE, "<string>", "exec")

exec code
exec pickle.loads(pickle.dumps(code))

good evening
Traceback (innermost last):
...
```

```
pickle.PicklingError: can't pickle 'code' objects
```

code . :
pickler , code ,
unpickler , , . Example 4-14 .

4.8.0.1. Example 4-14. copy_reg code pickle

```
File: copy-reg-example-2.py

import copy_reg
import pickle, marshal, types

#
# register a pickle handler for code objects

def code_unpickler(data):
    return marshal.loads(data)

def code_pickler(code):
    return code_unpickler, (marshal.dumps(code),)

copy_reg.pickle(types.CodeType, code_pickler, code_unpickler)

#
# try it out

CODE = """
print "suppose he's got a pointed stick"
"""

code = compile(CODE, "<string>", "exec")

exec code
exec pickle.loads(pickle.dumps(code))
```

```
suppose he's got a pointed stick
suppose he's got a pointed stick
```

pickle , unpickler
.

Example 4-15 pickle .

4.8.0.2. Example 4-15. copy_reg pickle

```
File: copy-reg-example-3.py

import copy_reg
import pickle, types
import StringIO

#
# register a pickle handler for file objects

def file_unpickler(position, data):
    file = StringIO.StringIO(data)
    file.seek(position)
    return file

def file_pickler(code):
    position = file.tell()
    file.seek(0)
    data = file.read()
    file.seek(position)
    return file_unpickler, (position, data)

copy_reg.pickle(types.FileType, file_pickler, file_unpickler)

#
# try it out

file = open("samples/sample.txt", "rb")

print file.read(120),
print "<here>",
print pickle.loads(pickle.dumps(file)).read()
```

We will perhaps eventually be writing only small modules, which are identified by name as they are used to build larger <here> ones, so that devices like indentation, rather than delimiters, might become feasible for expressing local structure in the source language.

-- Donald E. Knuth, December 1974

4.9. pprint

pprint (pretty printer) Python .
(,).

4.9.0.1. Example 4-16. pprint

```
File: pprint-example-1.py

import pprint

data = (
    "this is a string", [1, 2, 3, 4], ("more tuples",
    1.0, 2.3, 4.5), "this is yet another string"
    )

pprint.pprint(data)
```

```
('this is a string',
 [1, 2, 3, 4],
 ('more tuples', 1.0, 2.3, 4.5),
 'this is yet another string')
```

4.10. repr

repr repr . (,
). Example 4-17 .

4.10.0.1. Example 4-17. repr

```
File: repr-example-1.py

# note: this overrides the built-in 'repr' function
from repr import repr

# an annoyingly recursive data structure
data = (
    "X" * 100000,
    )
data = [data]
data.append(data)

print repr(data)
```

```
[('XXXXXXXXXXXX...XXXXXXXXXXXXX',), [('XXXXXXXXXXXX...XXXXXXXXXX
XXX',), [('XXXXXXXXXXXX...XXXXXXXXXXXXX',), [('XXXXXXXXXXXX...XX
XXXXXXXXXXX',), [('XXXXXXXXXXXX...XXXXXXXXXXXXX',), [(...), [...
]]]]]]]
```

4.11. base64

base64 . 3
4 ,
:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
0123456789+/
```

, = .

Example 4-18 encode decode .

4.11.0.1. Example 4-18. base64

```
File: base64-example-1.py

import base64

MESSAGE = "life of brian"

file = open("out.txt", "w")
file.write(MESSAGE)
file.close()

base64.encode(open("out.txt"), open("out.b64", "w"))
base64.decode(open("out.b64"), open("out.txt", "w"))

print "original:", repr(MESSAGE)
print "encoded message:", repr(open("out.b64").read())
print "decoded message:", repr(open("out.txt").read())

original:  'life of brian'
encoded message: 'bGlmZSBvZiBicmlhbg==\012'
decoded message: 'life of brian'
```

Example 4-19 encodestring decodestring
. encode decode . StringIO
.

4.11.0.2. Example 4-19. base64

```
File: base64-example-2.py

import base64

MESSAGE = "life of brian"

data = base64.encodestring(MESSAGE)

original_data = base64.decodestring(data)

print "original:", repr(MESSAGE)
print "encoded data:", repr(data)
print "decoded data:", repr(original_data)
```

```
original: 'life of brian'
encoded data: 'bGlmZSBvZiBicmlhbg==\012'
decoded data: 'life of brian'
```

Example 4-20 HTTP .

4.11.0.3. Example 4-20. base64

```
File: base64-example-3.py

import base64

def getbasic(user, password):
    # basic authentication (according to HTTP)
    return base64.encodestring(user + ":" + password)

print getbasic("Aladdin", "open sesame")
```

```
'QWxhZGRpbjpvcGVuIHNlc2FtZQ=='
```

, Example 4-21 , GIF
Python , Tkinter .

4.11.0.4. Example 4-21. base64 Tkinter GIF

```
File: base64-example-4.py

import base64, sys

if not sys.argv[1:]:
    print "Usage: gif2tk.py giffile >pyfile"
    sys.exit(1)

data = open(sys.argv[1], "rb").read()

if data[:4] != "GIF8":
    print sys.argv[1], "is not a GIF file"
    sys.exit(1)

print '# generated from', sys.argv[1], 'by gif2tk.py'
print
print 'from Tkinter import PhotoImage'
print
print 'image = PhotoImage(data="""'
print base64.encodestring(data),
print '""")'
```

```
# generated from samples/sample.gif by gif2tk.py

from Tkinter import PhotoImage

image = PhotoImage(data="""
R0lGODlhoAB4APcAAAAAAIAAAACAAICAAAAAgIAAgACAgICAgAQEBIwEBIyMBJRUlISE
/LRUBAQE
...
AjmQBFmQBnmQCJmQCrmQDNmQDvmQEBmREnkRAQEAOw==
""")
```

4.12. binhex

binhex Macintosh BinHex . Example 4-22 .

4.12.0.1. Example 4-22. binhex

```
File: binhex-example-1.py

import binhex
import sys

infile = "samples/sample.jpg"

binhex.binhex(infile, sys.stdout)
```

```
(This file must be converted with BinHex 4.0)

:#R0KEA"XC5jUF'F!2j!)!*!%%TS!N!4RdrrBrq!!%%T'58B!!3%!!!%!!3!!rpX
!3`!)"JB("J8)"`F(#3N)#J`8$3`,#``C%K-2&"dD(aiG'K`F)#3Z*b!L,#-F(#J
h+5``-63d0"mR16di-M`Z-c3brpX!3`%*#3N-#``B$3dB-L%F)6+3-[r!!"%)!)!
!J!-")J!#%3%$%3(ra!!I!!!""3'3"J#3#!%#!`3&"JF)#3S,rm3!Y4!!!J%$!`)
%!`8&"!3!!&p!3)$!!34"4)K-8%'%e&K"b*a&$+"ND%))d+a`495dI!N-f*bJJN
```

binhex hexbin .

4.13. quopri

quopri MIME (quoted printable encoding).

U.S. ASCII , , , U.S. ASCII . mail , . Example 4-23 .

4.13.0.1. Example 4-23. quopri

File: quopri-example-1.py

```
import quopri
import StringIO

# helpers (the quopri module only supports file-to-file conversion)

def encodestring(instring, tabs=0):
    outfile = StringIO.StringIO()
    quopri.encode(StringIO.StringIO(instring), outfile, tabs)
    return outfile.getvalue()

def decodestring(instring):
    outfile = StringIO.StringIO()
    quopri.decode(StringIO.StringIO(instring), outfile)
    return outfile.getvalue()

#
# try it out

MESSAGE = "? i ?a ? e ?!"

encoded_message = encodestring(MESSAGE)
decoded_message = decodestring(encoded_message)

print "original:", MESSAGE
print "encoded message:", repr(encoded_message)
print "decoded message:", decoded_message
```

```
original: ? i ?a ? e ?!
encoded message: '=E5 i =E5a =E4 e =F6!\012'
decoded message: ? i ?a ? e ?!
```

Example 4-23 , U.S. (=) . ("=3D ") , ("=20 "). . , .

(Europeans generally hate this encoding and strongly believe that certain U.S. programmers deserve to be slapped in the head with a huge great fish to the jolly music of Edward German....)

4.14. uu

uu . , base64 .

uu 3 (24) 4 (6), chr(32) ( ) chr(95) . uu 40% .

, (Unix) , end :

```
begin 666 sample.jpg
M_]C_X  02D9)1@ ! 0   0 !  #_VP!#  @&!@<&!O@'!P<)'O@*#!O-# L+
...more lines like this...
```

```
end
```

uu : encode decode .

encode(infile, outfile, filename) , . Example 4-24 . infile outfile . filename .

4.14.0.1. Example 4-24. uu

```
File: uu-example-1.py

import uu
import os, sys

infile = "samples/sample.jpg"

uu.encode(infile, sys.stdout, os.path.basename(infile))

begin 666 sample.jpg
M_]C_X  02D9)1@ ! O   O !  #_VP!#  @&!@<&!O@'!P<)"O@*#!O-# L+
M#!D2$P\4'1H?'AT:'!P@)"XG("(L(QP<*#<I+# Q-#OT'R<Y/3@R/"XS-#+_
MVP!# OD)"OP+#!@-#1@R(1PA,C(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R,C(R
M,C(R,C(R,C(R,C(R,C(R,C(R,C+_P  1" "  ( # 2(  A$! Q$!_\O
M'P   O4! O$! O$           $" PO%!@<("OH+_\O M1   @$# P($ P4%
```

decode(infile, outfile) uu . , . Example 4-25 .

4.14.0.2. Example 4-25. uu uu

```
File: uu-example-2.py

import uu
import StringIO

infile = "samples/sample.uue"
outfile = "samples/sample.jpg"

#
# decode

fi = open(infile)
fo = StringIO.StringIO()
```

```
uu.decode(fi, fo)

#
# compare with original data file

data = open(outfile, "rb").read()

if fo.getvalue() == data:
    print len(data), "bytes ok"
```

4.15. binascii

binascii , base64 , binhex , uu . Example 4-26 .

2.0 , .

4.15.0.1. Example 4-26. binascii

```
File: binascii-example-1.py

import binascii

text = "hello, mrs teal"

data = binascii.b2a_base64(text)
text = binascii.a2b_base64(data)
print text, "<=>", repr(data)

data = binascii.b2a_uu(text)
text = binascii.a2b_uu(data)
print text, "<=>", repr(data)

data = binascii.b2a_hqx(text)
text = binascii.a2b_hqx(data)[0]
print text, "<=>", repr(data)

# 2.0 and newer
data = binascii.b2a_hex(text)
text = binascii.a2b_hex(data)
print text, "<=>", repr(data)
```

```
hello, mrs teal <=> 'aGVsbG8sIG1ycyB0ZWFs\012'
hello, mrs teal <=> '/:&5L;&\\L(&UR<R!T96%L\012'
hello, mrs teal <=> 'D\'9XE\'mX)\'ebFb"dC@&X'
hello, mrs teal <=> '68656c6c6f2c206d7273207465616c'
```

5.

5.1.

.

5.1.1. Markup

Python (Extensible Markup Language , XML) (Hypertext Markup Language , HTML) . Python (Standard Generalized Markup Language , SGML) .

, HTML XML SGML . (start tags), (end tags), ( ), (entity references) :

```
<document name="sample.xml">
    <header>This is a header</header>
    <body>This is the body text.  The text can contain
    plain text (&quot;character data&quot;), tags, and
    entities.
    </body>
</document>
```

, <document> , <header> , <body> . , "/ " . , name .

(element) . document header body .

&quot; (character entity).
, & . , "
< (&lt;) " " > (&gt;) " .

XML , HTML , SGML , . XML
, , (well-formed).
XML , <document> <Document>
.

HTML , HTML ; ,
<P> , ,
</P> . HTML . , XML
, HTML HTML .

SGML , (declaration)
, DTD (document type description ,)
. , HTML XML SGML ,
SGML , HTML DTD .

Python makeup . SGML , Python
sgmllib . DTD ,
.

Python HTML SGML . htmllib
formatter . formatter .

Python XML . sgmllib xmllib ,
expat ( ). xmllib ,
xml .

5.1.2.

ConfigParser , Windows INI .

netrc .netrc , shlex shell
.

5.1.3.

Python GZIP ZIP (2.0) .
zlib , gzip zipfile .

5.2. xmllib

xmllib .

xmlib XML , XML
, Example 5-1 . ,
, .

XML xmllib ( ).
, , , .

, start_tag end_tag
, tag . start
( ).

5.2.0.1. Example 5-1. xmllib

File: xmllib-example-1.py

```
import xmllib

class Parser(xmllib.XMLParser):
    # get quotation number

    def _ _init_ _(self, file=None):
        xmllib.XMLParser._ _init_ _(self)
        if file:
            self.load(file)

    def load(self, file):
        while 1:
            s = file.read(512)
            if not s:
                break
            self.feed(s)
        self.close()

    def start_quotation(self, attrs):
        print "id =>", attrs.get("id")
        raise EOFError

try:
    c = Parser()
    c.load(open("samples/sample.xml"))
```

```
except EOFError:
    pass
```

id => 031

Example 5-2 ( ) (rendering engine). (_ _tags), . style , , .

5.2.0.2. Example 5-2. xmllib

File: xmllib-example-2.py

```
import xmllib
import string, sys

STYLESHEET = {
    # each element can contribute one or more style elements
    "quotation": {"style": "italic"},
    "lang": {"weight": "bold"},
    "name": {"weight": "medium"},
}

class Parser(xmllib.XMLParser):
    # a simple styling engine

    def _ _init_ _(self, renderer):
        xmllib.XMLParser._ _init_ _(self)
        self._ _data = []
        self._ _tags = []
        self._ _renderer = renderer

    def load(self, file):
        while 1:
            s = file.read(8192)
            if not s:
                break
            self.feed(s)
        self.close()

    def handle_data(self, data):
        self._ _data.append(data)

    def unknown_starttag(self, tag, attrs):
        if self._ _data:
            text = string.join(self._ _data, "")
            self._ _renderer.text(self._ _tags, text)
        self._ _tags.append(tag)
        self._ _data = []

    def unknown_endtag(self, tag):
        self._ _tags.pop()
        if self._ _data:
            text = string.join(self._ _data, "")
            self._ _renderer.text(self._ _tags, text)
        self._ _data = []

class DumbRenderer:

    def _ _init_ _(self):
        self.cache = {}

    def text(self, tags, text):
        # render text in the style given by the tag stack
        tags = tuple(tags)
        style = self.cache.get(tags)
        if style is None:
            # figure out a combined style
            style = {}
            for tag in tags:
                s = STYLESHEET.get(tag)
                if s:
                    style.update(s)
            self.cache[tags] = style # update cache
        # write to standard output
        sys.stdout.write("%s =>\n" % style)
        sys.stdout.write("  " + repr(text) + "\n")

#
# try it out

r = DumbRenderer()
c = Parser(r)
c.load(open("samples/sample.xml"))
```

```
{'style': 'italic'} =>
  'I\'ve had a lot of developers come up to me and\012say,
  "I haven\'t had this much fun in a long time. It sure
  beats\012writing '
{'style': 'italic', 'weight': 'bold'} =>
  'Cobol'
{'style': 'italic'} =>
  '" -- '
{'style': 'italic', 'weight': 'medium'} =>
  'James Gosling'
{'style': 'italic'} =>
  ', on\012'
{'weight': 'bold'} =>
  'Java'
{'style': 'italic'} =>
  '.'
```

5.3. xml.parsers.expat

( ) xml.parsers.expat James Clark's Expat XML parser . Example 5-3 .

5.3.0.1. Example 5-3. xml.parsers.expat

```
File: xml-parsers-expat-example-1.py

from xml.parsers import expat

class Parser:

    def _ _init_ _(self):
        self._parser = expat.ParserCreate()
        self._parser.StartElementHandler = self.start
        self._parser.EndElementHandler = self.end
        self._parser.CharacterDataHandler = self.data

    def feed(self, data):
        self._parser.Parse(data, 0)

    def close(self):
        self._parser.Parse("", 1) # end of data
        del self._parser # get rid of circular references

    def start(self, tag, attrs):
        print "START", repr(tag), attrs
    def end(self, tag):
        print "END", repr(tag)

    def data(self, data):
        print "DATA", repr(data)

p = Parser()
p.feed("<tag>data</tag>")
p.close()
```

```
START u'tag' {}
DATA u'data'
END u'tag'
```

, Unicode .
, UTF-8 . ,
XML *encoding* . Example 5-4 .

5.3.0.2. Example 5-4. xml.parsers.expat ISO Latin-1

```
File: xml-parsers-expat-example-2.py

from xml.parsers import expat

class Parser:

    def _ _init_ _(self):
        self._parser = expat.ParserCreate()
        self._parser.StartElementHandler = self.start
        self._parser.EndElementHandler = self.end
        self._parser.CharacterDataHandler = self.data

    def feed(self, data):
        self._parser.Parse(data, 0)

    def close(self):
        self._parser.Parse("", 1) # end of data
        del self._parser # get rid of circular references

    def start(self, tag, attrs):
        print "START", repr(tag), attrs

    def end(self, tag):
        print "END", repr(tag)
    def data(self, data):
        print "DATA", repr(data)

p = Parser()
p.feed("""\
<?xml version='1.0' encoding='iso-8859-1'?>
<author>
<name>fredrik lundh</name>
<city>link?ping</city>
</author>
"""
)
p.close()
```

```
START u'author' {}
DATA u'\012'
START u'name' {}
DATA u'fredrik lundh'
END u'name'
DATA u'\012'
START u'city' {}
DATA u'link\366ping'
END u'city'
DATA u'\012'
END u'author'
```

5.4. sgmllib

sgmllib , SGML . xmllib
, ( ). Example 5-5 .

xmllib , , ,
. ,
.

5.4.0.1. Example 5-5. sgmllib Title

File: sgmllib-example-1.py

```
import sgmllib
import string
```

```
class FoundTitle(Exception):
    pass

class ExtractTitle(sgmllib.SGMLParser):

    def __init__(self, verbose=0):
        sgmllib.SGMLParser.__init__(self, verbose)
        self.title = self.data = None

    def handle_data(self, data):
        if self.data is not None:
            self.data.append(data)

    def start_title(self, attrs):
        self.data = []

    def end_title(self):
        self.title = string.join(self.data, "")
        raise FoundTitle # abort parsing!

def extract(file):
    # extract title from an HTML/SGML stream
    p = ExtractTitle()
    try:
        while 1:
            # read small chunks
            s = file.read(512)
            if not s:
                break
            p.feed(s)
        p.close()
    except FoundTitle:
        return p.title
    return None

#
# try it out

print "html", "=>", extract(open("samples/sample.htm"))
print "sgml", "=>", extract(open("samples/sample.sgm"))
```

```
html => A Title.
sgml => Quotations
```

unknown_starttag unknown_endtag .
Example 5-6 .

5.4.0.2. Example 5-6. sgmllib SGML

File: sgmllib-example-2.py

```
import sgmllib
import cgi, sys

class PrettyPrinter(sgmllib.SGMLParser):
    # A simple SGML pretty printer

    def _ _init_ _(self):
        # initialize base class
        sgmllib.SGMLParser._ _init_ _(self)
        self.flag = 0

    def newline(self):
        # force newline, if necessary
        if self.flag:
            sys.stdout.write("\n")
        self.flag = 0

    def unknown_starttag(self, tag, attrs):
        # called for each start tag

        # the attrs argument is a list of (attr, value)
        # tuples. convert it to a string.
        text = ""
        for attr, value in attrs:
            text = text + " %s='%s'" % (attr, cgi.escape(value))

        self.newline()
        sys.stdout.write("<%s%s>\n" % (tag, text))

    def handle_data(self, text):
        # called for each text section
        sys.stdout.write(text)
        self.flag = (text[-1:] != "\n")

    def handle_entityref(self, text):
        # called for each entity
        sys.stdout.write("&%s;" % text)
    def unknown_endtag(self, tag):
        # called for each end tag
        self.newline()
        sys.stdout.write("<%s>" % tag)

#
# try it out

file = open("samples/sample.sgm")

p = PrettyPrinter()
p.feed(file.read())
p.close()
```

```
<chapter>
<title>
Quotations
<title>
<epigraph>
<attribution>
eff-bot, June 1997
<attribution>
<para>
<quote>
Nobody expects the Spanish Inquisition! Amongst
our weaponry are such diverse elements as fear, surprise,
ruthless efficiency, and an almost fanatical devotion to
Guido, and nice red uniforms &mdash; oh, damn!
<quote>
<para>
<epigraph>
<chapter>
```

Example 5-7 SGML XML " ",
, .

,
. .

5.4.0.3. Example 5-7. sgmllib

File: sgmllib-example-3.py

```
import sgmllib
```

```
class WellFormednessChecker(sgmllib.SGMLParser):
    # check that an SGML document is 'well-formed'
    # (in the XML sense).

    def __init__(self, file=None):
        sgmllib.SGMLParser.__init__(self)
        self.tags = []
        if file:
            self.load(file)

    def load(self, file):
        while 1:
            s = file.read(8192)
            if not s:
                break
            self.feed(s)
        self.close()

    def close(self):
        sgmllib.SGMLParser.close(self)
        if self.tags:
            raise SyntaxError, "start tag %s not closed" % self.tags[-1]

    def unknown_starttag(self, start, attrs):
        self.tags.append(start)

    def unknown_endtag(self, end):
        start = self.tags.pop()
        if end != start:
            raise SyntaxError, "end tag %s does't match start tag %s" %\
                  (end, start)

try:
    c = WellFormednessChecker()
    c.load(open("samples/sample.htm"))
except SyntaxError:
    raise # report error
else:
    print "document is well-formed"
```

```
Traceback (innermost last):
...
SyntaxError: end tag head does't match start tag meta
```

, Example 5-8 HTML SGML . ,
start end .

5.4.0.4. Example 5-8. sgmllib SGML

File: sgmllib-example-4.py

```
import sgmllib
import cgi, string, sys

class SGMLFilter(sgmllib.SGMLParser):
    # sgml filter.  override start/end to manipulate
    # document elements

    def _ _init_ _(self, outfile=None, infile=None):
        sgmllib.SGMLParser._ _init_ _(self)
        if not outfile:
            outfile = sys.stdout
        self.write = outfile.write
        if infile:
            self.load(infile)

    def load(self, file):
        while 1:
            s = file.read(8192)
            if not s:
                break
            self.feed(s)
        self.close()

    def handle_entityref(self, name):
        self.write("&%s;" % name)

    def handle_data(self, data):
        self.write(cgi.escape(data))

    def unknown_starttag(self, tag, attrs):
        tag, attrs = self.start(tag, attrs)
        if tag:
            if not attrs:
                self.write("<%s>" % tag)
            else:
                self.write("<%s" % tag)
                for k, v in attrs:
                    self.write(" %s=%s" % (k, repr(v)))
                self.write(">")

    def unknown_endtag(self, tag):
        tag = self.end(tag)
        if tag:
            self.write("</%s>" % tag)

    def start(self, tag, attrs):
        return tag, attrs # override

    def end(self, tag):
        return tag # override

class Filter(SGMLFilter):

    def fixtag(self, tag):
        if tag == "em":
            tag = "i"
        if tag == "string":
            tag = "b"
        return string.upper(tag)

    def start(self, tag, attrs):
        return self.fixtag(tag), attrs

    def end(self, tag):
        return self.fixtag(tag)

c = Filter()
c.load(open("samples/sample.htm"))
```

5.5. htmllib

htmlib (tag-driven) HTML , . Example 5-9 . HTML formatter .

5.5.0.1. Example 5-9. htmllib

File: htmllib-example-1.py

```
import htmllib
import formatter
import string

class Parser(htmllib.HTMLParser):
    # return a dictionary mapping anchor texts to lists
    # of associated hyperlinks

    def _ _init_ _(self, verbose=0):
        self.anchors = {}
        f = formatter.NullFormatter()
        htmllib.HTMLParser._ _init_ _(self, f, verbose)

    def anchor_bgn(self, href, name, type):
        self.save_bgn()
        self.anchor = href

    def anchor_end(self):
        text = string.strip(self.save_end())
        if self.anchor and text:
            self.anchors[text] = self.anchors.get(text, []) +
[self.anchor]

file = open("samples/sample.htm")
html = file.read()
file.close()

p = Parser()
p.feed(html)
p.close()

for k, v in p.anchors.items():
    print k, "=>", v

print
```

```
link => ['http://www.python.org']
```

HTML , , sgmllib .

5.6. htmlentitydefs

htmlentitydefs HTML ISO Latin-1 .
Example 5-10 .

5.6.0.1. Example 5-10. htmlentitydefs

```
File: htmlentitydefs-example-1.py

import htmlentitydefs

entities = htmlentitydefs.entitydefs

for entity in "amp", "quot", "copy", "yen":
    print entity, "=", entities[entity]
```

```
amp = &
quot = "
copy = \302\251
yen = \302\245
```

Example 5-11
(cgi.escape).

5.6.0.2. Example 5-11. htmlentitydefs

```
File: htmlentitydefs-example-2.py

import htmlentitydefs
import re
import cgi

pattern = re.compile("&(\w+?);")

def descape_entity(m, defs=htmlentitydefs.entitydefs):
    # callback: translate one entity to its ISO Latin value
    try:
        return defs[m.group(1)]
    except KeyError:
        return m.group(0) # use as is

def descape(string):
    return pattern.sub(descape_entity, string)

print descape("&lt;spam&amp;eggs&gt;")
print descape(cgi.escape("<spam&eggs>"))
```

```
<spam&eggs>
<spam&eggs>
```

, Example 5-12 XML ISO Latin-1
XML . cgi.escape , ASCII .

5.6.0.3. Example 5-12. ISO Latin-1

```
File: htmlentitydefs-example-3.py

import htmlentitydefs
import re, string

# this pattern matches substrings of reserved and non-ASCII characters
pattern = re.compile(r"[&<>\"\x80-\xff]+")

# create character map
entity_map = {}

for i in range(256):
    entity_map[chr(i)] = "&%d;" % i

for entity, char in htmlentitydefs.entitydefs.items():
    if entity_map.has_key(char):
        entity_map[char] = "&%s;" % entity

def escape_entity(m, get=entity_map.get):
    return string.join(map(get, m.group()), "")

def escape(string):
    return pattern.sub(escape_entity, string)

print escape("<spam&eggs>")
print escape("\303\245 i \303\245a \303\244 e \303\266")
```

```
&lt;spam&amp;eggs&gt;
&aring; i &aring;a &auml; e &ouml;
```

5.7. formatter

formatter htmllib (formatter classes).

, *formatter* *writer* . formatter HTML (event stream), writer . Example 5-13 .

, *AbstractFormatter* . writer . *AbstractWriter* .

5.7.0.1. Example 5-13. formatter HTML

```
File: formatter-example-1.py

import formatter
import htmllib

w = formatter.AbstractWriter()
f = formatter.AbstractFormatter(w)

file = open("samples/sample.htm")

p = htmllib.HTMLParser(f)
p.feed(file.read())
p.close()

file.close()
```

```
send_paragraph(1)
new_font(('h1', 0, 1, 0))
send_flowing_data('A Chapter.')
send_line_break()
send_paragraph(1)
new_font(None)
send_flowing_data('Some text. Some more text. Some')
send_flowing_data(' ')
new_font((None, 1, None, None))
send_flowing_data('emphasized')
new_font(None)
send_flowing_data(' text. A')
send_flowing_data(' link')
send_flowing_data('[1]')
send_flowing_data('.')
```

formatter *NullWriter* , ; *DumbWriter* , . Example 5-14 .

5.7.0.2. Example 5-14. formatter HTML

File: formatter-example-2.py

```
import formatter
import htmllib

w = formatter.DumbWriter() # plain text
f = formatter.AbstractFormatter(w)

file = open("samples/sample.htm")

# print html body as plain text
p = htmllib.HTMLParser(f)
p.feed(file.read())
p.close()

file.close()

# print links
print
print
i = 1
for link in p.anchorlist:
    print i, "=>", link
    i = i + 1
```

```
A Chapter.

Some text. Some more text. Some emphasized text. A link[1].

1 => http://www.python.org
```

Example 5-15 Writer , *DumbWriter* , .

5.7.0.3. Example 5-15. formatter Writer

File: formatter-example-3.py

```
import formatter
import htmllib, string

class Writer(formatter.DumbWriter):
    def _ _init_ _(self):
        formatter.DumbWriter._ _init_ _(self)
        self.tag = ""
        self.bold = self.italic = 0
        self.fonts = []

    def new_font(self, font):
        if font is None:
            font = self.fonts.pop()
            self.tag, self.bold, self.italic = font
        else:
            self.fonts.append((self.tag, self.bold, self.italic))
            tag, bold, italic, typewriter = font
            if tag is not None:
                self.tag = tag
            if bold is not None:
                self.bold = bold
            if italic is not None:
                self.italic = italic

    def send_flowing_data(self, data):
        if not data:
            return
        atbreak = self.atbreak or data[0] in string.whitespace
        for word in string.split(data):
            if atbreak:
                self.file.write(" ")
            if self.tag in ("h1", "h2", "h3"):
                word = string.upper(word)
            if self.bold:
                word = "*" + word + "*"
            if self.italic:
                word = "_" + word + "_"
            self.file.write(word)
            atbreak = 1
        self.atbreak = data[-1] in string.whitespace

w = Writer()
f = formatter.AbstractFormatter(w)

file = open("samples/sample.htm")

# print html body as plain text
p = htmllib.HTMLParser(f)
```

```
p.feed(file.read())
p.close()
```

```
_A_ _CHAPTER._

Some text. Some more text. Some *emphasized* text. A link[1].
```

5.8. ConfigParser

ConfigParser .

Windows INI ,
(section), .

, Example 5-16 :

```
[book]
title: The Python Standard Library
author: Fredrik Lundh
email: fredrik@pythonware.com
version: 2.0-001115

[ematter]
pages: 250

[hardcopy]
pages: 350
```

Example 5-16 ConfigParser .

5.8.0.1. Example 5-16. ConfigParser

```
File: configparser-example-1.py

import ConfigParser
import string

config = ConfigParser.ConfigParser()

config.read("samples/sample.ini")

# print summary
print
```

```
print string.upper(config.get("book", "title"))
print "by", config.get("book", "author"),
print  "(" + config.get("book", "email") + ")"
print
print config.get("ematter", "pages"), "pages"
print

# dump entire config file
for section in config.sections():
    print section
    for option in config.options(section):
        print " ", option, "=", config.get(section, option)
```

```
THE PYTHON STANDARD LIBRARY
by Fredrik Lundh (fredrik@pythonware.com)

250 pages

book
  title = The Python Standard Library
  email = fredrik@pythonware.com
  author = Fredrik Lundh
  version = 2.0-001115
  _ _name_ _ = book
ematter
  _ _name_ _ = ematter
  pages = 250
hardcopy
  _ _name_ _ = hardcopy
  pages = 350
```

Python 2.0 , ConfigParser , Example 5-17 .

5.8.0.2. Example 5-17. ConfigParser

File: configparser-example-2.py

```
import ConfigParser
import sys

config = ConfigParser.ConfigParser()

# set a number of parameters
config.add_section("book")
config.set("book", "title", "the python standard library")
config.set("book", "author", "fredrik lundh")

config.add_section("ematter")
config.set("ematter", "pages", 250)

# write to screen
config.write(sys.stdout)
```

```
[book]
title = the python standard library
author = fredrik lundh

[ematter]
pages = 250
```

5.9. netrc

netrc *.netrc* , Example 5-18 .
home FTP . (
: "chmod 0600 ~/.netrc,").

5.9.0.1. Example 5-18. netrc

```
File: netrc-example-1.py

import netrc


# default is $HOME/.netrc
info = netrc.netrc("samples/sample.netrc")

login, account, password = info.authenticators("secret.fbi")
print "login", "=>", repr(login)
print "account", "=>", repr(account)
print "password", "=>", repr(password)
```

```
login => 'mulder'
account => None
password => 'trustno1'
```

5.10. shlex

shlex Unix shell lexer (tokenizer). Example 5-19 .

5.10.0.1. Example 5-19. shlex

```
File: shlex-example-1.py

import shlex

lexer = shlex.shlex(open("samples/sample.netrc", "r"))
lexer.wordchars = lexer.wordchars + "._"

while 1:
    token = lexer.get_token()
    if not token:
        break
    print repr(token)

'machine'
'secret.fbi'
'login'
'mulder'
'password'
'trustno1'
'machine'
'non.secret.fbi'
'login'
'scully'
'password'
'noway'
```

5.11. zipfile

(2.0) zipfile ZIP .

5.11.1.

namelist infolist , , *ZipInfo* . Example 5-20 .

5.11.1.1. Example 5-20. zipfile ZIP

File: zipfile-example-1.py

```
import zipfile

file = zipfile.ZipFile("samples/sample.zip", "r")

# list filenames
for name in file.namelist():
    print name,
print

# list file information
for info in file.infolist():
    print info.filename, info.date_time, info.file_size
```

```
sample.txt sample.jpg
sample.txt (1999, 9, 11, 20, 11, 8) 302
sample.jpg (1999, 9, 18, 16, 9, 44) 4762
```

5.11.2. ZIP

read ZIP . ,
. Example 5-21 .

5.11.2.1. Example 5-21. zipfile ZIP

File: zipfile-example-2.py

```
import zipfile

file = zipfile.ZipFile("samples/sample.zip", "r")

for name in file.namelist():
    data = file.read(name)
    print name, len(data), repr(data[:10])
```

```
sample.txt 302 'We will pe'
sample.jpg 4762 '\377\330\377\340\000\020JFIF'
```

5.11.3. ZIP

, , ZIP write
.

Example 5-22 samples ZIP .

5.11.3.1. Example 5-22. zipfile ZIP

File: zipfile-example-3.py

```
import zipfile
import glob, os

# open the zip file for writing, and write stuff to it

file = zipfile.ZipFile("test.zip", "w")

for name in glob.glob("samples/*"):
    file.write(name, os.path.basename(name), zipfile.ZIP_DEFLATED)

file.close()

# open the file again, to see what's in it

file = zipfile.ZipFile("test.zip", "r")
for info in file.infolist():
    print info.filename, info.date_time, info.file_size,
info.compress_size
```

```
sample.wav (1999, 8, 15, 21, 26, 46) 13260 10985
sample.jpg (1999, 9, 18, 16, 9, 44) 4762 4626
sample.au (1999, 7, 18, 20, 57, 34) 1676 1103
...
```

write .
zipfile.ZIP_STORED , .
zlib , zipfile.ZIP_DEFLATED .

zipfile . , ,
ZipInfo , . Example 5-23
.

5.11.3.2. Example 5-23. zipfile ZIP

File: zipfile-example-4.py

```
import zipfile
import glob, os, time

file = zipfile.ZipFile("test.zip", "w")

now = time.localtime(time.time())[:6]

for name in ("life", "of", "brian"):
    info = zipfile.ZipInfo(name)
    info.date_time = now
    info.compress_type = zipfile.ZIP_DEFLATED
    file.writestr(info, name*1000)

file.close()

# open the file again, to see what's in it

file = zipfile.ZipFile("test.zip", "r")

for info in file.infolist():
    print info.filename, info.date_time, info.file_size,
info.compress_size
```

```
life (2000, 12, 1, 0, 12, 1) 4000 26
of (2000, 12, 1, 0, 12, 1) 2000 18
brian (2000, 12, 1, 0, 12, 1) 5000 31
```

5.12. gzip

gzip gzip , Example 5-24 .

5.12.0.1. Example 5-24. gzip

File: gzip-example-1.py

```
import gzip

file = gzip.GzipFile("samples/sample.gz")

print file.read()
```

```
Well it certainly looks as though we're in for
a splendid afternoon's sport in this the 127th
```

Upperclass Twit of the Year Show.

seek tell . Example 5-25
.

5.12.0.2. Example 5-25. gzip seek/tell

```
File: gzip-example-2.py

import gzip

class gzipFile(gzip.GzipFile):
    # adds seek/tell support to GzipFile

    offset = 0

    def read(self, size=None):
        data = gzip.GzipFile.read(self, size)
        self.offset = self.offset + len(data)
        return data

    def seek(self, offset, whence=0):
        # figure out new position (we can only seek forwards)
        if whence == 0:
            position = offset
        elif whence == 1:
            position = self.offset + offset
        else:
            raise IOError, "Illegal argument"
        if position < self.offset:
            raise IOError, "Cannot seek backwards"

        # skip forward, in 16k blocks
        while position > self.offset:
            if not self.read(min(position - self.offset, 16384)):
                break

    def tell(self):
        return self.offset

#
# try it

file = gzipFile("samples/sample.gz")
file.seek(80)
```

```
print file.read()
```

```
this the 127th
Upperclass Twit of the Year Show.
```

6.

"To be removed from our list of future commercial postings by [SOME] PUBLISHING COMPANY an Annual Charge of Ninety Five dollars is required. Just send $95.00 with your Name, Address and Name of the Newsgroup to be removed from our list."
- Newsgroup spammer, July 1996

" ' ' , 95 .
95 , , , ,
."
- , 1996 7

6.1.

Python ,
.

6.2. rfc822

rfc822 (RFC 822
, HTTP).

, RFC 822 , ,
.

For example, here's a short mail message. The first five lines make up the message header, and the actual message (a single line, in this case) follows after an empty line:

. , .

```
Message-Id: <20001114144603.00abb310@oreilly.com>
```

```
Date: Tue, 14 Nov 2000 14:55:07 -0500
To: "Fredrik Lundh" <fredrik@effbot.org>
From: Frank
Subject: Re: python library book!

Where is it?
```

,

Example 6-1 .

6.2.0.1. Example 6-1. rfc822

```
File: rfc822-example-1.py

import rfc822

file = open("samples/sample.eml")

message = rfc822.Message(file)

for k, v in message.items():
    print k, "=", v

print len(file.read()), "bytes in body"
```

```
subject = Re: python library book!
from = "Frank" <your@editor>
message-id = <20001114144603.00abb310@oreilly.com>
to = "Fredrik Lundh" <fredrik@effbot.org>
date = Tue, 14 Nov 2000 14:55:07 -0500
25 bytes in body
```

(message object) ,
Example 6-2 .

6.2.0.2. Example 6-2. rfc822

```
File: rfc822-example-2.py

import rfc822

file = open("samples/sample.eml")

message = rfc822.Message(file)
```

```
print message.getdate("date")
print message.getaddr("from")
print message.getaddrlist("to")

(2000, 11, 14, 14, 55, 7, 0, 0, 0)
('Frank', 'your@editor')
[('Fredrik Lundh', 'fredrik@effbot.org')]
```

(,) . 9 , time .

6.3. mimetools

(Multipurpose Internet Mail Extensions, MIME) RFC 822 ASCII , .

mimetools MIME . rfc822 *Message* , MIME . Example 6-3 .

6.3.0.1. Example 6-3. mimetools

```
File: mimetools-example-1.py

import mimetools

file = open("samples/sample.msg")

msg = mimetools.Message(file)

print "type", "=>", msg.gettype()
print "encoding", "=>", msg.getencoding()
print "plist", "=>", msg.getplist()

print "header", "=>"
for k, v in msg.items():
    print "  ", k, "=", v

type => text/plain
encoding => 7bit
plist => ['charset="iso-8859-1"']
header =>
```

```
   mime-version = 1.0
   content-type = text/plain;
 charset="iso-8859-1"
   to = effbot@spam.egg
   date = Fri, 15 Oct 1999 03:21:15 -0400
   content-transfer-encoding = 7bit
   from = "Fredrik Lundh" <fredrik@pythonware.com>
   subject = By the way...
...
```

6.4. MimeWriter

MimeWriter MIME " " , Example 6-4 .

6.4.0.1. Example 6-4. MimeWriter

```
File: mimewriter-example-1.py

import MimeWriter

# data encoders
#
import quopri
import base64
import StringIO

import sys

TEXT = """
here comes the image you asked for.  hope
it's what you expected.

</F>"""

FILE = "samples/sample.jpg"

file = sys.stdout

#
# create a mime multipart writer instance

mime = MimeWriter.MimeWriter(file)
```

```
mime.addheader("Mime-Version", "1.0")

mime.startmultipartbody("mixed")

# add a text message
#

part = mime.nextpart()
part.addheader("Content-Transfer-Encoding", "quoted-printable")
part.startbody("text/plain")

quopri.encode(StringIO.StringIO(TEXT), file, 0)

# add an image
#

part = mime.nextpart()
part.addheader("Content-Transfer-Encoding", "base64")
part.startbody("image/jpeg")

base64.encode(open(FILE, "rb"), file)

mime.lastpart()
```

:

```
Content-Type: multipart/mixed;
    boundary='host.1.-852461.936831373.130.24813'

--host.1.-852461.936831373.130.24813
Content-Type: text/plain
Context-Transfer-Encoding: quoted-printable

here comes the image you asked for.  hope
it's what you expected.

</F>

--host.1.-852461.936831373.130.24813
Content-Type: image/jpeg
Context-Transfer-Encoding: base64

/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UH
Rof
```

```
HBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhM
jIy
...
1e5vLrSYbJnEVpEgjCLx5mPUOqsVKOUaxjdNIS+1U6pfzTR8IzEhj2HrVG6m8m18xc8cI
KSC
tCuFyC746j/Cq2pTia4WztfmKjGBXTCmo6IUpt==

--host.1.-852461.936831373.130.24813--
```

[Example 6-5 #eg-6-5]   .

6.4.0.2. Example 6-5. MimeWriter

File: mimewriter-example-2.py

```
import MimeWriter
import string, StringIO, sys
import re, quopri, base64

# check if string contains non-ascii characters
must_quote = re.compile("[\177-\377]").search


#
# encoders

def encode_quoted_printable(infile, outfile):
    quopri.encode(infile, outfile, 0)

class Writer:

    def _ _init_ _(self, file=None, blurb=None):
        if file is None:
            file = sys.stdout
        self.file = file
        self.mime = MimeWriter.MimeWriter(file)
        self.mime.addheader("Mime-Version", "1.0")

        file = self.mime.startmultipartbody("mixed")
        if blurb:
            file.write(blurb)

    def close(self):
        "End of message"
        self.mime.lastpart()
        self.mime = self.file = None

    def write(self, data, mimetype="text/plain"):
        "Write data from string or file to message"

        # data is either an opened file or a string
        if type(data) is type(""):
            file = StringIO.StringIO(data)
        else:
            file = data
            data = None

        part = self.mime.nextpart()

        typ, subtyp = string.split(mimetype, "/", 1)

        if typ == "text":

            # text data
            encoding = "quoted-printable"
            encoder = lambda i, o: quopri.encode(i, o, 0)

            if data and not must_quote(data):
                # copy, don't encode
                encoding = "7bit"
                encoder = None

        else:

            # binary data (image, audio, application, ...)
            encoding = "base64"
            encoder = base64.encode

        #
        # write part headers

        if encoding:
            part.addheader("Content-Transfer-Encoding", encoding)

        part.startbody(mimetype)

        #
        # write part body
        if encoder:
            encoder(file, self.file)
        elif data:
            self.file.write(data)
        else:
            while 1:
                data = infile.read(16384)
                if not data:
                    break
                outfile.write(data)

#
# try it out

BLURB = "if you can read this, your mailer is not MIME-aware\n"

mime = Writer(sys.stdout, BLURB)

# add a text message
mime.write("""\
here comes the image you asked for.  hope
it's what you expected.
""", "text/plain")

# add an image
mime.write(open("samples/sample.jpg", "rb"), "image/jpeg")

mime.close()
```

6.5. mailbox

mailbox , Example 6-6 . RFC 822 , .

6.5.0.1. Example 6-6. mailbox

```
File: mailbox-example-1.py

import mailbox

mb = mailbox.UnixMailbox(open("/var/spool/mail/effbot"))

while 1:
    msg = mb.next()
    if not msg:
        break
    for k, v in msg.items():
        print k, "=", v
    body = msg.fp.read()
    print len(body), "bytes in body"
```

```
subject = for he's a ...
message-id = <199910150027.CAA03202@spam.egg>
received = (from fredrik@pythonware.com)
 by spam.egg (8.8.7/8.8.5) id CAA03202
 for effbot; Fri, 15 Oct 1999 02:27:36 +0200
from = Fredrik Lundh <fredrik@pythonware.com>
date = Fri, 15 Oct 1999 12:35:36 +0200
to = effbot@spam.egg
1295 bytes in body
```

6.6. mailcap

mailcap *mailcap* ,
(Unix). Example 6-7 .

6.6.0.1. Example 6-7. mailcap Capability

```
File: mailcap-example-1.py

import mailcap

caps = mailcap.getcaps()

for k, v in caps.items():
    print k, "=", v
```

```
image/* = [{'view': 'pilview'}]
application/postscript = [{'view': 'ghostview'}]
```

Example 6-7 , pilview (view) ,
ghostscript viewer PostScript . Example 6-8
mailcap .

6.6.0.2. Example 6-8. mailcap

```
File: mailcap-example-2.py

import mailcap

caps = mailcap.getcaps()

command, info = mailcap.findmatch(
    caps, "image/jpeg", "view", "samples/sample.jpg"
    )

print command

pilview samples/sample.jpg
```

6.7. mimetypes

mimetypes url (uniform resource locator ,) MIME . , Apache Netscape . Example 6-9 .

6.7.0.1. Example 6-9. mimetypes

```
File: mimetypes-example-1.py

import mimetypes
import glob, urllib

for file in glob.glob("samples/*"):
    url = urllib.pathname2url(file)
    print file, mimetypes.guess_type(url)

samples\sample.au ('audio/basic', None)
samples\sample.ini (None, None)
samples\sample.jpg ('image/jpeg', None)
samples\sample.msg (None, None)
samples\sample.tar ('application/x-tar', None)
samples\sample.tgz ('application/x-tar', 'gzip')
samples\sample.txt ('text/plain', None)
samples\sample.wav ('audio/x-wav', None)
samples\sample.zip ('application/zip', None)
```

6.8. packmail

( ) packmail Unix shell .
, . Example 6-10
, Example 6-11 .

6.8.0.1. Example 6-10. packmail

File: packmail-example-1.py

```
import packmail
import sys

packmail.pack(sys.stdout, "samples/sample.txt", "sample.txt")
```

```
echo sample.txt
sed "s/^X//" >sample.txt <<"!"
XWe will perhaps eventually be writing only small
Xmodules, which are identified by name as they are
Xused to build larger ones, so that devices like
Xindentation, rather than delimiters, might become
Xfeasible for expressing local structure in the
Xsource language.
X    -- Donald E. Knuth, December 1974
!
```

====Example 6-11. packmail ===[eg-6-11]

File: packmail-example-2.py

```
import packmail
import sys

packmail.packtree(sys.stdout, "samples")
```

, , .

6.9. mimify

mimify MIME (ISO Latin 1
) . , :

```
$ mimify.py -e raw-message mime-message
$ mimify.py -d mime-message raw-message
```

, Example 6-12 .

6.9.0.1. Example 6-12. mimify

```
File: mimify-example-1.py

import mimify
import sys

mimify.unmimify("samples/sample.msg", sys.stdout, 1)
```

MIME , ,
base64 . unmimify base64
:

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary='boundary'

this is a multipart sample file.  the two
parts both contain ISO Latin 1 text, with
different encoding techniques.

--boundary
Content-Type: text/plain
Content-Transfer-Encoding: quoted-printable

sillmj=F6lke! blindstyre! medisterkorv!

--boundary
Content-Type: text/plain
Content-Transfer-Encoding: base64

a29tIG5lciBiYXJhLCBvbSBkdSB09nJzIQ==

--boundary--
```

( ):

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary= 'boundary'

this is a multipart sample file.  the two
```

```
parts both contain ISO Latin 1 text, with
different encoding techniques.

--boundary
Content-Type: text/plain

sillmj?lke! blindstyre! medisterkorv!

--boundary
Content-Type: text/plain

kom ner bara, om du t?rs!
```

Example 6-13 .

6.9.0.2. Example 6-13. mimify

File: mimify-example-2.py

```
import mimify
import StringIO, sys

#
# decode message into a string buffer

file = StringIO.StringIO()

mimify.unmimify("samples/sample.msg", file, 1)

#
# encode message from string buffer

file.seek(0) # rewind

mimify.mimify(file, sys.stdout)
```

6.10. multifile

multifile MIME
. Example 6-14 .

6.10.0.1. Example 6-14. multifile

File: multifile-example-1.py

```
import multifile
import cgi, rfc822

infile = open("samples/sample.msg")

message = rfc822.Message(infile)

# print parsed header
for k, v in message.items():
    print k, "=", v

# use cgi support function to parse content-type header
type, params = cgi.parse_header(message["content-type"])

if type[:10] == "multipart/":

    # multipart message
    boundary = params["boundary"]

    file = multifile.MultiFile(infile)

    file.push(boundary)

    while file.next():

        submessage = rfc822.Message(file)

        # print submessage
        print "-" * 68
        for k, v in submessage.items():
            print k, "=", v
        print
        print file.read()

    file.pop()

else:

    # plain message
    print infile.read()
```

7.

"Increasingly, people seem to misinterpret complexity as sophistication, which is baffling - the incomprehensible should cause suspicion rather than admiration. Possibly this trend results from a mistaken belief that using a somewhat mysterious device confers an aura of power on the user."
- Niklaus Wirth

7.1.

Python socket socket . Internet , Internet .

, : Internet Time Protocol (Internet) Hypertext Transfer Protocol (, HTTP).

7.1.1. Internet

Internet (RFC 868, Postel Harrenstien, 1983) .

, Unix ( ) . . (4 , 1900 1 1).

, :

File: rfc868.txt

```
Network Working Group                                    J. Postel -
ISI
Request for Comments: 868                          K. Harrenstien - SRI
                                                                   May
1983

                               Time Protocol
```

This RFC specifies a standard for the ARPA Internet community. Hosts on the ARPA Internet that choose to implement a Time Protocol are expected to adopt and implement this standard.

RFC ARPA Internet community .
ARPA Internet .

This protocol provides a site-independent, machine readable date and time. The Time service sends back to the originating source the time in seconds since midnight on January first 1900.

, .
1900 1 1 .

One motivation arises from the fact that not all systems have a date/time clock, and all are subject to occasional human or machine error. The use of time-servers makes it possible to quickly confirm or correct a system's idea of the time, by making a brief poll of several independent sites on the network.

, ,
.
.

This protocol may be used either above the Transmission Control Protocol (TCP) or above the User Datagram Protocol (UDP).

TCP UDP .

When used via TCP the time service works as follows:

TCP :

- S: Listen on port 37 (45 octal).
- U: Connect to port 37.
- S: Send the time as a 32 bit binary number.
- U: Receive the time.
- U: Close the connection.
- S: Close the connection.

- S: 37 (45) .
- U: 37 .
- S: 32 .
- U: .
- U: .
- S: .

The server listens for a connection on port 37. When the connection is established, the server returns a 32-bit time value and closes the connection. If the server is unable to determine the time at its site, it should either refuse the connection or close it without sending anything.

37 . , 32

. ,

.

When used via UDP the time service works as follows:

TCP :

S: Listen on port 37 (45 octal).
U: Send an empty datagram to port 37.
S: Receive the empty datagram.
S: Send a datagram containing the time as a 32 bit binary number.
U: Receive the time datagram.

S: 37 (45) .
U: 37 .
S: .
S: (32) .
U: .

The server listens for a datagram on port 37. When a datagram arrives, the server returns a datagram containing the 32-bit time value. If the server is unable to determine the time at its site, it should discard the arriving datagram and make no reply.

37 . , 32
. , ,
.

The Time

The time is the number of seconds since 00:00 (midnight) 1 January 1900 GMT, such that the time 1 is 12:00:01 am on 1 January 1900 GMT; this base will serve until the year 2036.

1900 1 1 0 ,
2036 . .

For example:

```
   the time  2,208,988,800 corresponds to 00:00  1 Jan 1970 GMT,
             2,398,291,200 corresponds to 00:00  1 Jan 1976 GMT,
             2,524,521,600 corresponds to 00:00  1 Jan 1980 GMT,
             2,629,584,000 corresponds to 00:00  1 May 1983 GMT,
        and -1,297,728,000 corresponds to 00:00 17 Nov 1858 GMT.
```

:

```
             2,208,988,800      to 00:00  1 Jan 1970 GMT,
             2,398,291,200      to 00:00  1 Jan 1976 GMT,
             2,524,521,600      to 00:00  1 Jan 1980 GMT,
             2,629,584,000      to 00:00  1 May 1983 GMT,
            -1,297,728,000      to 00:00 17 Nov 1858 GMT.
```

RFC868.txt Translated By Andelf(gt: andelf@gmail.com)
, . Thx.

7.1.2. HTTP

(HTTP, RFC 2616) .
(Version 1.1) 100 .

, .
, :

```
GET /hello.txt HTTP/1.0
Host: hostname
User-Agent: name

[optional request body ,                ]
```

:

```
HTTP/1.0 200 OK
Content-Type: text/plain
```

```
Content-Length: 7

Hello
```

headers ( ) , header Host / .

header "\r\n " , header , ( ).

HTTP , , , , , Hypertext TransferProtocol - HTTP/1.1 (http://www.w3.org/Protocols).

7.2. socket

socket socket . socket .

, Example 7-1 , 4 , .

7.2.0.1. Example 7-1. socket

File: socket-example-1.py

```
import socket
import struct, time

# server
HOST = "www.python.org"
PORT = 37

# reference time (in seconds since 1900-01-01 00:00:00)
TIME1970 = 2208988800L # 1970-01-01 00:00:00

# connect to server
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))

# read 4 bytes, and convert to time value
t = s.recv(4)
```

```
t = struct.unpack("!I", t)[0]
t = int(t - TIME1970)

s.close()

# print results
print "server time is", time.ctime(t)
print "local clock is", int(time.time()) - t, "seconds off"
```

```
server time is Sat Oct 09 16:42:36 1999
local clock is 8 seconds off
```

socket (factory function) (Internet
stream socket , TCP socket) socket . connect
socket . , recv
.

socket , ,
socket bind ( ) , ,
.

Example 7-2 , 8037 (1024
, Unix root).

7.2.0.2. Example 7-2. socket

File: socket-example-2.py

```
import socket
import struct, time

# user-accessible port
PORT = 8037

# reference time
TIME1970 = 2208988800L

# establish server
service = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
service.bind(("", PORT))
service.listen(1)

print "listening on port", PORT

while 1:
    # serve forever
    channel, info = service.accept()
    print "connection from", info
    t = int(time.time()) + TIME1970
    t = struct.pack("!I", t)
    channel.send(t) # send timestamp
    channel.close() # disconnect
```

```
listening on port 8037
connection from ('127.0.0.1', 1469)
connection from ('127.0.0.1', 1470)
...
```

listen socket . (
) . accept
.

accept socket ,
. socket ;
socket .

Example 7-3 , (Example 7-1)
, .

7.2.0.3. Example 7-3.

```
File: timeclient.py

import socket
import struct, sys, time

# default server
host = "localhost"
port = 8037

# reference time (in seconds since 1900-01-01 00:00:00)
TIME1970 = 2208988800L # 1970-01-01 00:00:00

def gettime(host, port):
    # fetch time buffer from stream server
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect((host, port))
    t = s.recv(4)
    s.close()
    t = struct.unpack("!I", t)[0]
    return int(t - TIME1970)

if _ _name_ _ == "_ _main_ _":
    # command-line utility
    if sys.argv[1:]:
        host = sys.argv[1]
        if sys.argv[2:]:
            port = int(sys.argv[2])
        else:
            port = 37 # default for public servers

    t = gettime(host, port)
    print "server time is", time.ctime(t)
    print "local clock is", int(time.time()) - t, "seconds off"
```

```
server time is Sat Oct 09 16:58:50 1999
local clock is 0 seconds off
```

Example 7-3 ; timeclient
, gettime .

, (TCP) socket . UDP sockets
( ). socket ;
, . ,
. , .

Example 7-4 , socket .
sendto , .
recvfrom .

7.2.0.4. Example 7-4. socket

File: socket-example-4.py

```
import socket
import struct, time

# server
HOST = "localhost"
PORT = 8037

# reference time (in seconds since 1900-01-01 00:00:00)
TIME1970 = 2208988800L # 1970-01-01 00:00:00

# connect to server
```

```
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# send empty packet
s.sendto("", (HOST, PORT))

# read 4 bytes from server, and convert to time value
t, server = s.recvfrom(4)
t = struct.unpack("!I", t)[0]
t = int(t - TIME1970)

s.close()

print "server time is", time.ctime(t)
print "local clock is", int(time.time()) - t, "seconds off"
```

```
server time is Sat Oct 09 16:42:36 1999
local clock is 8 seconds off
```

recvfrom : . .

Example 7-5 .

Example 7-5. socket

File: socket-example-5.py

```
import socket
import struct, time

# user-accessible port
PORT = 8037

# reference time
TIME1970 = 2208988800L

# establish server
service = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
service.bind(("", PORT))

print "listening on port", PORT

while 1:
    # serve forever
    data, client = service.recvfrom(0)
    print "connection from", client
    t = int(time.time()) + TIME1970
    t = struct.pack("!I", t)
    service.sendto(t, client) # send timestamp
```

```
listening on port 8037
connection from ('127.0.0.1', 1469)
connection from ('127.0.0.1', 1470)
...
```

bind socket ,
recvfrom .

7.3. select

select socket , ,
, Example 7-6 .

socket select , (
, ,):

- listen , , socket (accept). socket (, recv).
- connect socket , socket .
- connect , socket .

7.3.0.1. Example 7-6. select socket

File: select-example-1.py

```
import select
import socket
import time

PORT = 8037

TIME1970 = 2208988800L

service = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
service.bind(("", PORT))
service.listen(1)

print "listening on port", PORT

while 1:
    is_readable = [service]
    is_writable = []
    is_error = []
    r, w, e = select.select(is_readable, is_writable, is_error, 1.0)
    if r:
        channel, info = service.accept()
        print "connection from", info
        t = int(time.time()) + TIME1970
        t = chr(t>>24&255) + chr(t>>16&255) + chr(t>>8&255) + chr(t&255)
        channel.send(t) # send timestamp
        channel.close() # disconnect
    else:
        print "still waiting"
```

```
listening on port 8037
still waiting
still waiting
connection from ('127.0.0.1', 1469)
still waiting
connection from ('127.0.0.1', 1470)
...
```

Example 7-6 , socket ,
. channel socket ,
4 . ,
is_writable , select
.

socket (setblocking),
select socket . asyncore ( )
, .
, .

7.4. asyncore

asyncore " (reactive)" socket .
, socket ,

. *dispatcher* , (
) socket .

- handle_connect : .
- handle_expt : .
- handle_accept : socket .
  (callback) accept socket .
- handle_read : socket .
  recv .
- handle_write : socket . send
  .
- handle_close : socket .
- handle_error(type, value, traceback)
  Python .
  sys.stdout .

Example 7-7 , socket .

7.4.0.1. Example 7-7. asyncore

File: asyncore-example-1.py

```
import asyncore
import socket, time

# reference time (in seconds since 1900-01-01 00:00:00)
TIME1970 = 2208988800L # 1970-01-01 00:00:00

class TimeRequest(asyncore.dispatcher):
    # time requestor (as defined in RFC 868)

    def _ _init_ _(self, host, port=37):
        asyncore.dispatcher._ _init_ _(self)
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.connect((host, port))

    def writable(self):
        return 0 # don't have anything to write

    def handle_connect(self):
        pass # connection succeeded

    def handle_expt(self):
        self.close() # connection failed, shutdown
    def handle_read(self):
        # get local time
        here = int(time.time()) + TIME1970

        # get and unpack server time
        s = self.recv(4)
        there = ord(s[3]) + (ord(s[2])<<8) + (ord(s[1])<<16) +
(ord(s[0])<<24L)

        self.adjust_time(int(here - there))

        self.handle_close() # we don't expect more data

    def handle_close(self):
        self.close()

    def adjust_time(self, delta):
        # override this method!
        print "time difference is", delta

#
# try it out

request = TimeRequest("www.python.org")

asyncore.loop()
```

```
log: adding channel <TimeRequest  at 8cbe90>
time difference is 28
log: closing channel 192:<TimeRequest connected at 8cbe90>
```

, *dispatcher* log .

Example 7-8 . *dispatcher* , socket , .

7.4.0.2. Example 7-8. asyncore

```
File: asyncore-example-2.py

import asyncore
import socket, time

# reference time
```

```
TIME1970 = 2208988800L

class TimeChannel(asyncore.dispatcher):

    def handle_write(self):
        t = int(time.time()) + TIME1970
        t = chr(t>>24&255) + chr(t>>16&255) + chr(t>>8&255) + chr(t&255)
        self.send(t)
        self.close()

class TimeServer(asyncore.dispatcher):

    def _ _init_ _(self, port=37):
        self.port = port
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.bind(("", port))
        self.listen(5)
        print "listening on port", self.port

    def handle_accept(self):
        channel, addr = self.accept()
        TimeChannel(channel)

server = TimeServer(8037)
asyncore.loop()
```

```
log: adding channel <TimeServer  at 8cb940>
listening on port 8037
log: adding channel <TimeChannel  at 8b2fd0>
log: closing channel 52:<TimeChannel connected at 8b2fd0>
```

dispatcher , *dispatcher_with_send* .
.

Example 7-9 *dispatcher_with_send* *AsyncHTTP* . , HTTP GET "consumer"

7.4.0.3. Example 7-9. asyncore HTTP

```
File: SimpleAsyncHTTP.py

import asyncore
import string, socket
import StringIO
import mimetools, urlparse

class AsyncHTTP(asyncore.dispatcher_with_send):
    # HTTP requester

    def _ _init_ _(self, uri, consumer):
        asyncore.dispatcher_with_send._ _init_ _(self)

        self.uri = uri
        self.consumer = consumer

        # turn the uri into a valid request
        scheme, host, path, params, query, fragment =
urlparse.urlparse(uri)
        assert scheme == "http", "only supports HTTP requests"
        try:
            host, port = string.split(host, ":", 1)
            port = int(port)
        except (TypeError, ValueError):
            port = 80 # default port
        if not path:
            path = "/"
        if params:
            path = path + ";" + params
        if query:
            path = path + "?" + query

        self.request = "GET %s HTTP/1.0\r\nHost: %s\r\n\r\n" % (path,
host)

        self.host = host
        self.port = port

        self.status = None
        self.header = None

        self.data = ""

        # get things going!
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.connect((host, port))

    def handle_connect(self):
        # connection succeeded
        self.send(self.request)

    def handle_expt(self):
        # connection failed; notify consumer (status is None)
        self.close()
        try:
            http_header = self.consumer.http_header
        except AttributeError:
            pass
        else:
            http_header(self)

    def handle_read(self):
        data = self.recv(2048)
        if not self.header:
            self.data = self.data + data
            try:
                i = string.index(self.data, "\r\n\r\n")
            except ValueError:
                return # continue
            else:
                # parse header
                fp = StringIO.StringIO(self.data[:i+4])
                # status line is "HTTP/version status message"
                status = fp.readline()
                self.status = string.split(status, " ", 2)
                # followed by a rfc822-style message header
                self.header = mimetools.Message(fp)
                # followed by a newline, and the payload (if any)
                data = self.data[i+4:]
                self.data = ""
                # notify consumer (status is non-zero)
                try:
                    http_header = self.consumer.http_header
                except AttributeError:
                    pass
                else:
                    http_header(self)
                if not self.connected:
                    return # channel was closed by consumer

        self.consumer.feed(data)

    def handle_close(self):
        self.consumer.close()
        self.close()
```

Example 7-10 .

7.4.0.4. Example 7-10. SimpleAsyncHTTP

```
File: asyncore-example-3.py

import SimpleAsyncHTTP
import asyncore

class DummyConsumer:
    size = 0

    def http_header(self, request):
        # handle header
        if request.status is None:
            print "connection failed"
        else:
            print "status", "=>", request.status
            for key, value in request.header.items():
                print key, "=", value

    def feed(self, data):
        # handle incoming data
        self.size = self.size + len(data)

    def close(self):
        # end of data
        print self.size, "bytes in body"

#
# try it out

consumer = DummyConsumer()

request = SimpleAsyncHTTP.AsyncHTTP(
    "http://www.pythonware.com",
    consumer
    )

asyncore.loop()
```

```
log: adding channel <AsyncHTTP  at 8e2850>
```

```
status => ['HTTP/1.1', '200', 'OK\015\012']
server = Apache/Unix (Unix)
content-type = text/html
content-length = 3730
...
3730 bytes in body
log: closing channel 156:<AsyncHTTP connected at 8e2850>
```

consumer htmllib xmllib , HTML XML . http_header ; , .

Example 7-10 . Example 7-11 consumer , .

7.4.0.5. Example 7-11. SimpleAsyncHTTP

File: asyncore-example-4.py

```
import SimpleAsyncHTTP
import asyncore

class DummyConsumer:
    size = 0

    def http_header(self, request):
        # handle header
        if request.status is None:
            print "connection failed"
        else:
            print "status", "=>", request.status
            for key, value in request.header.items():
                print key, "=", value

    def feed(self, data):
        # handle incoming data
        self.size = self.size + len(data)

    def close(self):
        # end of data
        print self.size, "bytes in body"

class RedirectingConsumer:

    def _ _init_ _(self, consumer):
        self.consumer = consumer

    def http_header(self, request):
        # handle header
        if request.status is None or\
           request.status[1] not in ("301", "302"):
            try:
                http_header = self.consumer.http_header
            except AttributeError:
                pass
            else:
                return http_header(request)
        else:
            # redirect!
            uri = request.header["location"]
            print "redirecting to", uri, "..."
            request.close()
            SimpleAsyncHTTP.AsyncHTTP(uri, self)

    def feed(self, data):
        self.consumer.feed(data)

    def close(self):
        self.consumer.close()

#
# try it out

consumer = RedirectingConsumer(DummyConsumer())

request = SimpleAsyncHTTP.AsyncHTTP(
    "http://www.pythonware.com/library",
    consumer
    )

asyncore.loop()
```

```
log: adding channel <AsyncHTTP  at 8e64b0>
redirecting to http://www.pythonware.com/library/ ...
log: closing channel 48:<AsyncHTTP connected at 8e64b0>
log: adding channel <AsyncHTTP  at 8ea790>
status => ['HTTP/1.1', '200', 'OK\015\012']
server = Apache/Unix (Unix)
content-type = text/html
```

```
content-length = 387
...
387 bytes in body
log: closing channel 236:<AsyncHTTP connected at 8ea790>
```

301 ( ) 302 ( ), consumer . consumer consumer .

7.5. asynchat

asynchat asyncore . (line-oriented) . (push "producer" .

Example 7-12 HTTP . HTTP HTML ( ).

7.5.0.1. Example 7-12. asynchat HTTP

File: asynchat-example-1.py

```
import asyncore, asynchat
import os, socket, string

PORT = 8000

class HTTPChannel(asynchat.async_chat):

    def _ _init_ _(self, server, sock, addr):
        asynchat.async_chat._ _init_ _(self, sock)
        self.set_terminator("\r\n")
        self.request = None
        self.data = ""
        self.shutdown = 0

    def collect_incoming_data(self, data):
        self.data = self.data + data

    def found_terminator(self):
        if not self.request:
            # got the request line
            self.request = string.split(self.data, None, 2)
            if len(self.request) != 3:
                self.shutdown = 1
            else:
                self.push("HTTP/1.0 200 OK\r\n")
                self.push("Content-type: text/html\r\n")
                self.push("\r\n")
            self.data = self.data + "\r\n"
            self.set_terminator("\r\n\r\n") # look for end of headers
        else:
            # return payload.
            self.push("<html><body><pre>\r\n")
            self.push(self.data)
            self.push("</pre></body></html>\r\n")
            self.close_when_done()

class HTTPServer(asyncore.dispatcher):

    def _ _init_ _(self, port):
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.bind(("", port))
        self.listen(5)

    def handle_accept(self):
        conn, addr = self.accept()
        HTTPChannel(self, conn, addr)

#
# try it out

s = HTTPServer(PORT)
print "serving at port", PORT, "..."
asyncore.loop()
```

```
GET / HTTP/1.1
Accept: */*
Accept-Language: en, sv
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; Bruce/1.0)
Host: localhost:8000
Connection: Keep-Alive
```

producer ("push") .
asyncore producer more . ,
.

Example 7-13 HTTP ,
FileProducer , kb .

7.5.0.2. Example 7-13. asynchat HTTP

File: asynchat-example-2.py

```
import asyncore, asynchat
import os, socket, string, sys
import StringIO, mimetools

ROOT = "."

PORT = 8000

class HTTPChannel(asynchat.async_chat):

    def _ _init_ _(self, server, sock, addr):
        asynchat.async_chat._ _init_ _(self, sock)
        self.server = server
        self.set_terminator("\r\n\r\n")
        self.header = None
        self.data = ""
        self.shutdown = 0

    def collect_incoming_data(self, data):
        self.data = self.data + data
        if len(self.data) > 16384:
            # limit the header size to prevent attacks
            self.shutdown = 1

    def found_terminator(self):
        if not self.header:
            # parse http header
            fp = StringIO.StringIO(self.data)
            request = string.split(fp.readline(), None, 2)
            if len(request) != 3:
                # badly formed request; just shut down
                self.shutdown = 1
            else:
                # parse message header
                self.header = mimetools.Message(fp)
                self.set_terminator("\r\n")
                self.server.handle_request(
                    self, request[0], request[1], self.header
                    )
                self.close_when_done()
            self.data = ""
        else:
            pass # ignore body data, for now

    def pushstatus(self, status, explanation="OK"):
        self.push("HTTP/1.0 %d %s\r\n" % (status, explanation))


class FileProducer:
    # a producer that reads data from a file object

    def _ _init_ _(self, file):
        self.file = file

    def more(self):
        if self.file:
            data = self.file.read(2048)
            if data:
                return data
            self.file = None
        return ""


class HTTPServer(asyncore.dispatcher):

    def _ _init_ _(self, port=None, request=None):
        if not port:
            port = 80
        self.port = port
        if request:
            self.handle_request = request # external request handler
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.bind(("", port))
        self.listen(5)

    def handle_accept(self):
        conn, addr = self.accept()
        HTTPChannel(self, conn, addr)
    def handle_request(self, channel, method, path, header):
        try:
            # this is not safe!
            while path[:1] == "/":
                path = path[1:]
            filename = os.path.join(ROOT, path)
            print path, "=>", filename
            file = open(filename, "r")
        except IOError:
            channel.pushstatus(404, "Not found")
            channel.push("Content-type: text/html\r\n")
            channel.push("\r\n")
            channel.push("<html><body>File not
found.</body></html>\r\n")
        else:
            channel.pushstatus(200, "OK")
            channel.push("Content-type: text/html\r\n")
            channel.push("\r\n")
            channel.push_with_producer(FileProducer(file))

#
# try it out

s = HTTPServer(PORT)
print "serving at port", PORT
asyncore.loop()
```

```
serving at port 8000
log: adding channel <HTTPServer  at 8e54d0>
log: adding channel <HTTPChannel  at 8e64a0>
samples/sample.htm => .\samples/sample.htm
log: closing channel 96:<HTTPChannel connected at 8e64a0>
```

7.6. urllib

urllib HTTP , FTP , gopher .
URL .

URL . urlopen ,
, Example 7-14 .

7.6.0.1. Example 7-14. urllib

```
File: urllib-example-1.py

import urllib

fp = urllib.urlopen("http://www.python.org")

op = open("out.html", "wb")

n = 0

while 1:
    s = fp.read(8192)
    if not s:
        break
    op.write(s)
    n = n + len(s)

fp.close()
op.close()

for k, v in fp.headers.items():
    print k, "=", v

print "copied", n, "bytes from", fp.url
```

```
server = Apache/1.3.6 (Unix)
content-type = text/html
accept-ranges = bytes
date = Mon, 11 Oct 1999 20:11:40 GMT
connection = close
etag = "741e9-7870-37f356bf"
content-length = 30832
last-modified = Thu, 30 Sep 1999 12:25:35 GMT
copied 30832 bytes from http://www.python.org
```

. headers *Message* (
mimetools), url URL .
.

urlopen , *FancyURLopener*
open . .
Example 7-15 .

7.6.0.2. Example 7-15. urllib

```
File: urllib-example-3.py

import urllib

class myURLOpener(urllib.FancyURLopener):
    # read an URL, with automatic HTTP authentication

    def setpasswd(self, user, passwd):
        self._ _user = user
        self._ _passwd = passwd

    def prompt_user_passwd(self, host, realm):
        return self._ _user, self._ _passwd

urlopener = myURLOpener()
urlopener.setpasswd("mulder", "trustno1")

fp = urlopener.open("http://www.secretlabs.com")
print fp.read()
```

7.7. urlparse

urlparse URL , URL . Example 7-16 .

7.7.0.1. Example 7-16. urlparse

```
File: urlparse-example-1.py

import urlparse

print urlparse.urlparse("http://host/path;params?query#fragment")

('http', 'host', '/path', 'params', 'query', 'fragment')
```

HTTP URL (HTTP), Example 7-17 .

7.7.0.2. Example 7-17. urlparse HTTP (HTTP Locators)

```
File: urlparse-example-2.py
```

```
import urlparse

scheme, host, path, params, query, fragment =\
        urlparse.urlparse("http://host/path;params?query#fragment")

if scheme == "http":
    print "host", "=>", host
    if params:
        path = path + ";" + params
    if query:
        path = path + "?" + query
    print "path", "=>", path
```

```
host => host
path => /path;params?query
```

Example 7-18 urlunparse
URL .

7.7.0.3. Example 7-18. urlparse HTTP (HTTP Locators)

File: urlparse-example-3.py

```
import urlparse

scheme, host, path, params, query, fragment =\
        urlparse.urlparse("http://host/path;params?query#fragment")

if scheme == "http":
    print "host", "=>", host
    print "path", "=>", urlparse.urlunparse(
    (None, None, path, params, query, None)
    )
```

```
host => host
path => /path;params?query
```

Example 7-19 urljoin .

7.7.0.4. Example 7-19. urlparse

File: urlparse-example-4.py

```
import urlparse
```

```
base = "http://spam.egg/my/little/pony"

for path in "/index", "goldfish", "../black/cat":
    print path, "=>", urlparse.urljoin(base, path)
```

```
/index => http://spam.egg/index
goldfish => http://spam.egg/my/little/goldfish
../black/cat => http://spam.egg/my/black/cat
```

7.8. cookie

(2.0) HTTP cookie . Example 7-20 .

7.8.0.1. Example 7-20. cookie

```
File: cookie-example-1.py

import Cookie
import os, time

cookie = Cookie.SimpleCookie()
cookie["user"] = "Mimi"
cookie["timestamp"] = time.time()

print cookie

# simulate CGI roundtrip
os.environ["HTTP_COOKIE"] = str(cookie)

print

cookie = Cookie.SmartCookie()
cookie.load(os.environ["HTTP_COOKIE"])

for key, item in cookie.items():
    # dictionary items are "Morsel" instances
    # use value attribute to get actual value
    print key, repr(item.value)
```

```
Set-Cookie: timestamp=736513200;
```

```
Set-Cookie: user=Mimi;

user 'Mimi'
timestamp '736513200'
```

7.9. robotparser

(2.0) robotparser robots.txt , Robot Exclusion Protocol (? http://info.webcrawler.com/mak/projects/robots/robots.html).

HTTP ( ), . Example 7-21 .

7.9.0.1. Example 7-21. robotparser

```
File: robotparser-example-1.py

import robotparser

r = robotparser.RobotFileParser()
r.set_url("http://www.python.org/robots.txt")
r.read()

if r.can_fetch("*", "/index.html"):
    print "may fetch the home page"

if r.can_fetch("*", "/tim_one/index.html"):
    print "may fetch the tim peters archive"

may fetch the home page
```

7.10. ftplib

ftplib File Transfer Protocol (FTP ,) .

Example 7-22 . ( ) (, Unix ls Windows/DOS dir).

7.10.0.1. Example 7-22. ftplib

```
File: ftplib-example-1.py

import ftplib

ftp = ftplib.FTP("www.python.org")
ftp.login("anonymous", "ftplib-example-1")

print ftp.dir()

ftp.quit()

total 34
drwxrwxr-x  11 root     4127         512 Sep 14 14:18 .
drwxrwxr-x  11 root     4127         512 Sep 14 14:18 ..
drwxrwxr-x   2 root     4127         512 Sep 13 15:18 RCS
lrwxrwxrwx   1 root     bin           11 Jun 29 14:34 README ->
welcome.msg
drwxr-xr-x   3 root     wheel        512 May 19  1998 bin
drwxr-sr-x   3 root     1400         512 Jun  9  1997 dev
drwxrwxr--   2 root     4127         512 Feb  8  1998 dup
drwxr-xr-x   3 root     wheel        512 May 19  1998 etc
...
```

; retr . ,
. Example 7-23 lambda
.

7.10.0.2. Example 7-23. ftplib

```
File: ftplib-example-2.py

import ftplib
import sys

def gettext(ftp, filename, outfile=None):
    # fetch a text file
    if outfile is None:
        outfile = sys.stdout
    # use a lambda to add newlines to the lines read from the server
    ftp.retrlines("RETR " + filename, lambda s, w=outfile.write:
w(s+"\n"))

def getbinary(ftp, filename, outfile=None):
    # fetch a binary file
    if outfile is None:
        outfile = sys.stdout
    ftp.retrbinary("RETR " + filename, outfile.write)

ftp = ftplib.FTP("www.python.org")
ftp.login("anonymous", "ftplib-example-2")

gettext(ftp, "README")
getbinary(ftp, "welcome.msg")
```

```
WELCOME to python.org, the Python programming language home site.

You are number %N of %M allowed users.  Ni!

Python Web site: http://www.python.org/

CONFUSED FTP CLIENT?  Try begining your login password with '-' dash.
This turns off continuation messages that may be confusing your client.
...
```

, Example 7-24 FTP .
.

7.10.0.3. Example 7-24. ftplib

```
File: ftplib-example-3.py

import ftplib
import os

def upload(ftp, file):
    ext = os.path.splitext(file)[1]
    if ext in (".txt", ".htm", ".html"):
        ftp.storlines("STOR " + file, open(file))
    else:
        ftp.storbinary("STOR " + file, open(file, "rb"), 1024)

ftp = ftplib.FTP("ftp.fbi.gov")
ftp.login("mulder", "trustno1")

upload(ftp, "trixie.zip")
upload(ftp, "file.txt")
upload(ftp, "sightings.jpg")
```

7.11. gopherlib

gopherlib gopher , Example 7-25 .

7.11.0.1. Example 7-25. gopherlib

```
File: gopherlib-example-1.py

import gopherlib

host = "gopher.spam.egg"

f = gopherlib.send_selector("1/", host)
for item in gopherlib.get_directory(f):
    print item

['0', "About Spam.Egg's Gopher Server", "0/About's Spam.Egg's
Gopher Server", 'gopher.spam.egg', '70', '+']
['1', 'About Spam.Egg', '1/Spam.Egg', 'gopher.spam.egg', '70', '+']
['1', 'Misc', '1/Misc', 'gopher.spam.egg', '70', '+']
...
```

7.12. httplib

httplib HTTP , Example 7-26 .

7.12.0.1. Example 7-26. httplib

```
File: httplib-example-1.py

import httplib

USER_AGENT = "httplib-example-1.py"

class Error:
    # indicates an HTTP error
    def _ _init_ _(self, url, errcode, errmsg, headers):
        self.url = url
        self.errcode = errcode
        self.errmsg = errmsg
        self.headers = headers
    def _ _repr_ _(self):
        return (
            "<Error for %s: %s %s>" %
            (self.url, self.errcode, self.errmsg)
            )

class Server:

    def _ _init_ _(self, host):
        self.host = host

    def fetch(self, path):
        http = httplib.HTTP(self.host)

        # write header
        http.putrequest("GET", path)
        http.putheader("User-Agent", USER_AGENT)
        http.putheader("Host", self.host)
        http.putheader("Accept", "*/*")
        http.endheaders()

        # get response
        errcode, errmsg, headers = http.getreply()

        if errcode != 200:
            raise Error(errcode, errmsg, headers)

        file = http.getfile()
        return file.read()

if _ _name_ _ == "_ _main_ _":

    server = Server("www.pythonware.com")
    print server.fetch("/index.htm")
```

httplib HTTP . asyncore .

7.12.1.

httplib HTTP , POST , Example 7-27 .

7.12.1.1. Example 7-27. httplib

File: httplib-example-2.py

```
import httplib

USER_AGENT = "httplib-example-2.py"

def post(host, path, data, type=None):

    http = httplib.HTTP(host)

    # write header
    http.putrequest("PUT", path)
    http.putheader("User-Agent", USER_AGENT)
    http.putheader("Host", host)
    if type:
        http.putheader("Content-Type", type)
    http.putheader("Content-Length", str(len(size)))
    http.endheaders()

    # write body
    http.send(data)

    # get response
    errcode, errmsg, headers = http.getreply()

    if errcode != 200:
        raise Error(errcode, errmsg, headers)

    file = http.getfile()
    return file.read()

if _ _name_ _ == "_ _main_ _":

    post("www.spam.egg", "/bacon.htm", "a piece of data", "text/plain")
```

7.13. poplib

poplib (Example 7-28) Post Office Protocol (POP3) . "pop" ( ) .

7.13.0.1. Example 7-28. poplib

File: poplib-example-1.py

```
import poplib
import string, random
import StringIO, rfc822

SERVER = "pop.spam.egg"

USER  = "mulder"
PASSWORD = "trustno1"

# connect to server
server = poplib.POP3(SERVER)

# login
server.user(USER)
server.pass_(PASSWORD)

# list items on server
resp, items, octets = server.list()

# download a random message
id, size = string.split(random.choice(items))
resp, text, octets = server.retr(id)

text = string.join(text, "\n")
file = StringIO.StringIO(text)

message = rfc822.Message(file)

for k, v in message.items():
    print k, "=", v

print message.fp.read()
```

```
subject = ANN: (the eff-bot guide to) The Standard Python Library
message-id = <199910120808.KAA09206@spam.egg>
received = (from fredrik@spam.egg)
 by spam.egg (8.8.7/8.8.5) id KAA09206
 for mulder; Tue, 12 Oct 1999 10:08:47 +0200
from = Fredrik Lundh <fredrik@spam.egg>
date = Tue, 12 Oct 1999 10:08:47 +0200
to = mulder@spam.egg

...
```

7.14. imaplib

imaplib Internet Message Access Protocol (IMAP, Internet
) . ,
. Example 7-29 .

7.14.0.1. Example 7-29. imaplib

```
File: imaplib-example-1.py

import imaplib
import string, random
import StringIO, rfc822

SERVER = "imap.spam.egg"

USER  = "mulder"
PASSWORD = "trustno1"

# connect to server
server = imaplib.IMAP4(SERVER)

# login
server.login(USER, PASSWORD)
server.select()

# list items on server
resp, items = server.search(None, "ALL")
items = string.split(items[0])

# fetch a random item
id = random.choice(items)
resp, data = server.fetch(id, "(RFC822)")
text = data[0][1]

file = StringIO.StringIO(text)

message = rfc822.Message(file)

for k, v in message.items():
    print k, "=", v

print message.fp.read()
```

```
server.logout()
```

```
subject = ANN: (the eff-bot guide to) The Standard Python Library
message-id = <199910120816.KAA12177@larch.spam.egg>
to = mulder@spam.egg
date = Tue, 12 Oct 1999 10:16:19 +0200 (MET DST)
from = <effbot@spam.egg>
received = (effbot@spam.egg) by imap.algonet.se (8.8.8+Sun/8.6.12)
id KAA12177 for effbot@spam.egg; Tue, 12 Oct 1999 10:16:19 +0200
(MET DST)

body text for test 5
```

7.15. smtplib

smtplib Simple Mail Transfer Protocol (SMTP ,) . Unix , Example 7-30 .

poplib imaplib .

7.15.0.1. Example 7-30. smtplib

```
File: smtplib-example-1.py

import smtplib
import string, sys

HOST = "localhost"

FROM = "effbot@spam.egg"
TO = "fredrik@spam.egg"

SUBJECT = "for your information!"

BODY = "next week: how to fling an otter"

body = string.join((
    "From: %s" % FROM,
    "To: %s" % TO,
    "Subject: %s" % SUBJECT,
    "",
    BODY), "\r\n")
```

```
print body

server = smtplib.SMTP(HOST)
server.sendmail(FROM, [TO], body)
server.quit()
```

```
From: effbot@spam.egg
To: fredrik@spam.egg
Subject: for your information!

next week: how to fling an otter
```

7.16. telnetlib

telnetlib telnet .

Example 7-31 Unix , , .

7.16.0.1. Example 7-31. telnetlib

File: telnetlib-example-1.py

```
import telnetlib
import sys

HOST = "spam.egg"

USER = "mulder"
PASSWORD = "trustno1"

telnet = telnetlib.Telnet(HOST)

telnet.read_until("login: ")
telnet.write(USER + "\n")

telnet.read_until("Password: ")
telnet.write(PASSWORD + "\n")

telnet.write("ls librarybook\n")
telnet.write("exit\n")

print telnet.read_all()
```

```
[spam.egg mulder]$ ls
README                                  os-path-isabs-example-1.py
SimpleAsyncHTTP.py                      os-path-isdir-example-1.py
aifc-example-1.py                       os-path-isfile-example-1.py
anydbm-example-1.py                     os-path-islink-example-1.py
array-example-1.py                      os-path-ismount-example-1.py
...
```

7.17. nntplib

nntplib (Network News Transfer Protocol, NNTP) .

7.17.1.

, .
Example 7-32 ,
.

7.17.1.1. Example 7-32. nntplib

```
File: nntplib-example-1.py

import nntplib
import string

SERVER = "news.spam.egg"
GROUP  = "comp.lang.python"
AUTHOR = "fredrik@pythonware.com" # eff-bots human alias

# connect to server
server = nntplib.NNTP(SERVER)

# choose a newsgroup
resp, count, first, last, name = server.group(GROUP)
print "count", "=>", count
print "range", "=>", first, last

# list all items on the server
resp, items = server.xover(first, last)
```

```
# extract some statistics
authors = {}
subjects = {}
for id, subject, author, date, message_id, references, size, lines in
items:
    authors[author] = None
    if subject[:4] == "Re: ":
        subject = subject[4:]
    subjects[subject] = None
    if string.find(author, AUTHOR) >= 0:
        print id, subject

print "authors", "=>", len(authors)
print "subjects", "=>", len(subjects)
```

```
count => 607
range => 57179 57971
57474 Three decades of Python!
...
57477 More Python books coming...
authors => 257
subjects => 200
```

7.17.2.

, article , Example 7-33 .

7.17.2.1. Example 7-33. nntplib

```
File: nntplib-example-2.py

import nntplib
import string

SERVER = "news.spam.egg"
GROUP  = "comp.lang.python"
KEYWORD = "tkinter"

# connect to server
server = nntplib.NNTP(SERVER)

resp, count, first, last, name = server.group(GROUP)
resp, items = server.xover(first, last)
```

```
for id, subject, author, date, message_id, references, size, lines in
items:
    if string.find(string.lower(subject), KEYWORD) >= 0:
        resp, id, message_id, text = server.article(id)
        print author
        print subject
        print len(text), "lines in article"
```

```
"Fredrik Lundh" <fredrik@pythonware.com>
Re: Programming Tkinter (In Python)
110 lines in article
...
```

Example 7-34 ,
Message (rfc822).

7.17.2.2. Example 7-34. nntplib rfc822

```
File: nntplib-example-3.py

import nntplib
import string, random
import StringIO, rfc822

SERVER = "news.spam.egg"
GROUP  = "comp.lang.python"

# connect to server
server = nntplib.NNTP(SERVER)

resp, count, first, last, name = server.group(GROUP)
for i in range(10):
    try:
        id = random.randint(int(first), int(last))
        resp, id, message_id, text = server.article(str(id))
    except (nntplib.error_temp, nntplib.error_perm):
        pass # no such message (maybe it was deleted?)
    else:
        break # found a message!
else:
    raise SystemExit

text = string.join(text, "\n")
file = StringIO.StringIO(text)
```

```
message = rfc822.Message(file)

for k, v in message.items():
    print k, "=", v

print message.fp.read()
```

```
mime-version = 1.0
content-type = text/plain; charset="iso-8859-1"
message-id = <008501bf1417$1cf90b70$f29b12c2@sausage.spam.egg>
lines = 22
...
from = "Fredrik Lundh" <fredrik@pythonware.com>
nntp-posting-host = parrot.python.org
subject = ANN: (the eff-bot guide to) The Standard Python Library
...
</F>
```

, htmllib , uu , base64 .

7.18. SocketServer

SocketServer socket .
, .

Example 7-35 Internet .
timeclient .

7.18.0.1. Example 7-35. SocketServer

```
File: socketserver-example-1.py

import SocketServer
import time

# user-accessible port
PORT = 8037

# reference time
TIME1970 = 2208988800L

class TimeRequestHandler(SocketServer.StreamRequestHandler):
    def handle(self):
        print "connection from", self.client_address
        t = int(time.time()) + TIME1970
        b = chr(t>>24&255) + chr(t>>16&255) + chr(t>>8&255) + chr(t&255)
        self.wfile.write(b)

server = SocketServer.TCPServer(("", PORT), TimeRequestHandler)
print "listening on port", PORT
server.serve_forever()
```

```
connection from ('127.0.0.1', 1488)
connection from ('127.0.0.1', 1489)
...
```

7.19. BaseHTTPServer

SocketServer , HTTP .

Example 7-36 . path URL , URL (path).

7.19.0.1. Example 7-36. BaseHTTPServer

```
File: basehttpserver-example-1.py

import BaseHTTPServer
import cgi, random, sys

MESSAGES = [
    "That's as maybe, it's still a frog.",
    "Albatross! Albatross! Albatross!",
    "It's Wolfgang Amadeus Mozart.",
    "A pink form from Reading.",
    "Hello people, and welcome to 'It's a Tree.'"
    "I simply stare at the brick and it goes to sleep.",
]

class Handler(BaseHTTPServer.BaseHTTPRequestHandler):

    def do_GET(self):
        if self.path != "/":
            self.send_error(404, "File not found")
            return
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()
        try:
            # redirect stdout to client
            stdout = sys.stdout
            sys.stdout = self.wfile
            self.makepage()
        finally:
            sys.stdout = stdout # restore

    def makepage(self):
        # generate a random message
        tagline = random.choice(MESSAGES)
        print "<html>"
        print "<body>"
        print "<p>Today's quote: "
        print "<i>%s</i>" % cgi.escape(tagline)
        print "</body>"
        print "</html>"

PORT = 8000

httpd = BaseHTTPServer.HTTPServer(("", PORT), Handler)
print "serving at port", PORT
httpd.serve_forever()
```

HTTP SimpleHTTPServer CGIHTTPServer .

7.20. SimpleHTTPServer

SimpleHTTPServer HTTP , GET HEAD . ( ). Example 7-37 .

7.20.0.1. Example 7-37. SimpleHTTPServer

File: simplehttpserver-example-1.py

```
import SimpleHTTPServer
import SocketServer
```

```
# minimal web server.  serves files relative to the
# current directory.

PORT = 8000

Handler = SimpleHTTPServer.SimpleHTTPRequestHandler

httpd = SocketServer.TCPServer(("", PORT), Handler)

print "serving at port", PORT
httpd.serve_forever()
```

```
serving at port 8000
localhost - - [11/Oct/1999 15:07:44] code 403, message Directory listing
not sup
ported
localhost - - [11/Oct/1999 15:07:44] "GET / HTTP/1.1" 403 -
localhost - - [11/Oct/1999 15:07:56] "GET /samples/sample.htm HTTP/1.1"
200 -
```

(`..`).
, .

Example 7-38 web . HTTP
URI . urllib .

7.20.0.2. Example 7-38. SimpleHTTPServer

```
File: simplehttpserver-example-2.py

# a truly minimal HTTP proxy

import SocketServer
import SimpleHTTPServer
import urllib

PORT = 1234

class Proxy(SimpleHTTPServer.SimpleHTTPRequestHandler):
    def do_GET(self):
        self.copyfile(urllib.urlopen(self.path), self.wfile)

httpd = SocketServer.ForkingTCPServer(('', PORT), Proxy)
```

```
print "serving at port", PORT
httpd.serve_forever()
```

7.21. CGIHTTPServer

CGIHTTPServer (common gateway interface , CGI) HTTP . Example 7-39 .

7.21.0.1. Example 7-39. CGIHTTPServer

```
File: cgihttpserver-example-1.py

import CGIHTTPServer
import BaseHTTPServer

class Handler(CGIHTTPServer.CGIHTTPRequestHandler):
    cgi_directories = ["/cgi"]

PORT = 8000

httpd = BaseHTTPServer.HTTPServer(("", PORT), Handler)
print "serving at port", PORT
httpd.serve_forever()
```

7.22. cgi

cgi CGI . CGI .

Example 7-40 CGI ,
( )

7.22.0.1. Example 7-40. cgi

```
File: cgi-example-1.py

import cgi
import os, urllib

ROOT = "samples"

# header
```

```
print "text/html"
print

query = os.environ.get("QUERY_STRING")
if not query:
    query = "."

script = os.environ.get("SCRIPT_NAME", "")
if not script:
    script = "cgi-example-1.py"

print "<html>"
print "<head>"
print "<title>file listing</title>"
print "</head>"
print "</html>"

print "<body>"

try:
    files = os.listdir(os.path.join(ROOT, query))
except os.error:
    files = []

for file in files:
    link = cgi.escape(file)
    if os.path.isdir(os.path.join(ROOT, query, file)):
        href = script + "?" + os.path.join(query, file)
        print "<p><a href= '%s'>%s</a>" % (href, cgi.escape(link))
    else:
        print "<p>%s" % link

print "</body>"
print "</html>"
```

```
text/html

<html>
<head>
<title>file listing</title>
</head>
</html>
<body>
<p>sample.gif
```

```
<p>sample.gz
<p>sample.netrc
...
<p>sample.txt
<p>sample.xml
<p>sample~
<p><a href='cgi-example-1.py?web'>web</a>
</body>
</html>
```

7.23. webbrowser

(2.0) webbrowser web . open , URL , . open , . Example 7-41 .

7.23.0.1. Example 7-41. webbrowser

```
File: webbrowser-example-1.py

import webbrowser
import time

webbrowser.open("http://www.pythonware.com")

# wait a while, and then go to another page
time.sleep(5)
webbrowser.open(
    "http://www.pythonware.com/people/fredrik/librarybook.htm"
    )
```

Unix , lynx , Netscape , Mosaic , Konquerer , Grail . Windows Macintosh , (Internet).

8.

8.1. locale

locale C (localization) , Example 8-1
. , locale , .
(int , float , string locale
.)

====Example 8-1. locale =====[eg-8-1]

```
File: locale-example-1.py

import locale

print "locale", "=>", locale.setlocale(locale.LC_ALL, "")

# integer formatting
value = 4711
print locale.format("%d", value, 1), "==",
print locale.atoi(locale.format("%d", value, 1))

# floating point
value = 47.11
print locale.format("%f", value, 1), "==",
print locale.atof(locale.format("%f", value, 1))

info = locale.localeconv()
print info["int_curr_symbol"]
```

```
locale => Swedish_Sweden.1252
4,711 == 4711
47,110000 == 47.11
SEK
```

Example 8-2 locale locale .

8.1.0.1. Example 8-2. locale locale

```
File: locale-example-2.py

import locale

language, encoding = locale.getdefaultlocale()

print "language", language
print "encoding", encoding
```

```
language sv_SE
encoding cp1252
```

8.2. unicodedata

(2.0) unicodedata Unicode , , , . Example 8-3 .

8.2.0.1. Example 8-3. unicodedata

```
File: unicodedata-example-1.py

import unicodedata

for char in [u"A", u"-", u"1", u"\N{LATIN CAPITAL LETTER O WITH
DIAERESIS}"]:
    print repr(char),
    print unicodedata.category(char),
    print repr(unicodedata.decomposition(char)),

    print unicodedata.decimal(char, None),
    print unicodedata.numeric(char, None)
```

```
u'A' Lu '' None None
u'-' Pd '' None None
u'1' Nd '' 1 1.0
u'\303\226' Lu '004F 0308' None None
```

Python 2.0 CJK . 0x3400-0x4DB5 , 0x4E00-0x9FA5 , 0xAC00-D7A3 , , :

```
def remap(char):
    # fix for broken unicode property database in Python 2.0
    c = ord(char)
    if 0x3400 <= c <= 0x4DB5:
        return unichr(0x3400)
    if 0x4E00 <= c <= 0x9FA5:
        return unichr(0x4E00)
    if 0xAC00 <= c <= 0xD7A3:
        return unichr(0xAC00)
    return char
```

Python 2.1 bug .

8.3. ucnhash

(2.0) ucnhash Unicode . \N{} Unicode . Example 8-4 .

8.3.0.1. Example 8-4. ucnhash

```
File: ucnhash-example-1.py

# Python imports this module automatically, when it sees
# the first \N{} escape
# import ucnhash

print repr(u"\N{FROWN}")
print repr(u"\N{SMILE}")
print repr(u"\N{SKULL AND CROSSBONES}")
```

```
u'\u2322'
u'\u2323'
u'\u2620'
```

9.

"Wot? No quote?"
- Guido van Rossum

9.1.

Python .

Pythonware Image Library (PIL , http://www.pythonware.com/products/pil/), *PythonWare Sound Toolkit* (PST , http://www.pythonware.com/products/pst/).

: PST , , pymedia .

9.2. imghdr

imghdr . bmp , gif , jpeg , pbm , pgm , png , ppm , rast (Sun raster), rgb (SGI), tiff , xbm . Example 9-1 .

9.2.0.1. Example 9-1. imghdr

```
File: imghdr-example-1.py

import imghdr

result = imghdr.what("samples/sample.jpg")

if result:
    print "file format:", result
else:
    print "cannot identify file"
```

```
file format: jpeg
```

```
#    PIL
import Image

im = Image.open("samples/sample.jpg")
print im.format, im.mode, im.size
```

9.3. sndhdr

sndhdr , , . Example 9-2 .

, what , , , . help(sndhdr) .

9.3.0.1. Example 9-2. sndhdr

```
File: sndhdr-example-1.py
```

```
import sndhdr

result = sndhdr.what("samples/sample.wav")

if result:
    print "file format:", result
else:
    print "cannot identify file"

file format: ('wav', 44100, 1, -1, 16)
```

9.4. whatsound

( ) whatsound sndhdr . Example 9-3 .

9.4.0.1. Example 9-3. whatsound

File: whatsound-example-1.py

```
import whatsound # same as sndhdr

result = whatsound.what("samples/sample.wav")

if result:
    print "file format:", result
else:
    print "cannot identify file"

file format: ('wav', 44100, 1, -1, 16)
```

9.5. aifc

aifc AIFF AIFC (SGI Macintosh). Example 9-4 .

9.5.0.1. Example 9-4. aifc

File: SimpleAsyncHTTP.py

```
import asyncore
import string, socket
import StringIO
import mimetools, urlparse

class AsyncHTTP(asyncore.dispatcher_with_send):
    # HTTP requestor

    def __init__(self, uri, consumer):
        asyncore.dispatcher_with_send.__init__(self)

        self.uri = uri
        self.consumer = consumer

        # turn the uri into a valid request
        scheme, host, path, params, query, fragment =
urlparse.urlparse(uri)
        assert scheme == "http", "only supports HTTP requests"
        try:
            host, port = string.split(host, ":", 1)
            port = int(port)
        except (TypeError, ValueError):
            port = 80 # default port
        if not path:
            path = "/"
        if params:
            path = path + ";" + params
        if query:
            path = path + "?" + query

        self.request = "GET %s HTTP/1.0\r\nHost: %s\r\n\r\n" % (path,
host)

        self.host = host
        self.port = port

        self.status = None
        self.header = None

        self.data = ""

        # get things going!
        self.create_socket(socket.AF_INET, socket.SOCK_STREAM)
        self.connect((host, port))

    def handle_connect(self):
```

```
    # connection succeeded
    self.send(self.request)

def handle_expt(self):
    # connection failed; notify consumer (status is None)
    self.close()
    try:
        http_header = self.consumer.http_header
    except AttributeError:
        pass
    else:
        http_header(self)

def handle_read(self):
    data = self.recv(2048)
    if not self.header:
        self.data = self.data + data
        try:
            i = string.index(self.data, "\r\n\r\n")
        except ValueError:
            return # continue
        else:
            # parse header
            fp = StringIO.StringIO(self.data[:i+4])
            # status line is "HTTP/version status message"
            status = fp.readline()
            self.status = string.split(status, " ", 2)
            # followed by a rfc822-style message header
            self.header = mimetools.Message(fp)
            # followed by a newline, and the payload (if any)
            data = self.data[i+4:]
            self.data = ""
            # notify consumer (status is non-zero)
            try:
                http_header = self.consumer.http_header
            except AttributeError:
                pass
            else:
                http_header(self)
            if not self.connected:
                return # channel was closed by consumer

    self.consumer.feed(data)
def handle_close(self):
    self.consumer.close()
    self.close()
```

9.6. sunau

sunau Sun AU . Example 9-5 .

9.6.0.1. Example 9-5. sunau

```
File: sunau-example-1.py

import sunau

w = sunau.open("samples/sample.au", "r")

if w.getnchannels() == 1:
    print "mono,",
else:
    print "stereo,",

print w.getsampwidth()*8, "bits,",
print w.getframerate(), "Hz sampling rate"
```

```
mono, 16 bits, 8012 Hz sampling rate
```

9.7. sunaudio

sunaudio Sun AU , . sunau Sun AU . Example 9-6

9.7.0.1. Example 9-6. sunaudio

```
File: sunaudio-example-1.py

import sunaudio

file = "samples/sample.au"

print sunaudio.gethdr(open(file, "rb"))
```

```
(6761, 1, 8012, 1, 'sample.au')
```

9.8. wave

wave Microsoft WAV , Example 9-7 .

9.8.0.1. Example 9-7. wave

```
File: wave-example-1.py

import wave

w = wave.open("samples/sample.wav", "r")

if w.getnchannels() == 1:
    print "mono,",
else:
    print "stereo,",

print w.getsampwidth()*8, "bits,",
print w.getframerate(), "Hz sampling rate"
```

```
mono, 16 bits, 44100 Hz sampling rate
```

9.9. audiodev

(Unix) audiodev Sun SGI . Example 9-8 .

9.9.0.1. Example 9-8. audiodev

```
File: audiodev-example-1.py

import audiodev
import aifc

sound = aifc.open("samples/sample.aiff", "r")

player = audiodev.AudioDev()

player.setoutrate(sound.getframerate())
player.setsampwidth(sound.getsampwidth())
player.setnchannels(sound.getnchannels())

bytes_per_frame = sound.getsampwidth() * sound.getnchannels()
bytes_per_second = sound.getframerate() * bytes_per_frame

while 1:
    data = sound.readframes(bytes_per_second)
    if not data:
        break
    player.writeframes(data)

player.wait()
```

9.10. winsound

(Windows) winsound Winodws Wave . Example 9-9 .

9.10.0.1. Example 9-9. winsound

```
File: winsound-example-1.py

import winsound

file = "samples/sample.wav"

winsound.PlaySound(
    file,
    winsound.SND_FILENAME|winsound.SND_NOWAIT,
    )
```

flag :

- SND_FILENAME - sound wav
- SND_ALIAS - sound
- SND_LOOP - PlaySound ; SND_ASYNC
- SND_MEMORY - sound wav
- SND_PURGE - sound
- SND_ASYNC - ,
- SND_NODEFAULT - sound beep
- SND_NOSTOP - sound
- SND_NOWAIT - sound

10.

"Unlike mainstream component programming, scripts usually do not introduce new components but simply 'wire' existing ones. Scripts can be seen as introducing behavior but no new state ... Of course, there is nothing to stop a 'scripting' language from introducing persistent state — it then simply turns into a normal programming language."
- Clemens Szyperski, in Component Software

10.1.

Python (database manager) ,
Unix dbm . ,
. (shelve)

10.2. anydbm

anydbm .

, anydbm ,
dbhash , gdbm , dbm , dumbdbm . ,
ImportError .

open ( ),
Example 10-1 .

10.2.0.1. Example 10-1. anydbm

```
File: anydbm-example-1.py

import anydbm

db = anydbm.open("database", "c")
db["1"] = "one"
db["2"] = "two"
db["3"] = "three"
db.close()
```

```
db = anydbm.open("database", "r")
for key in db.keys():
    print repr(key), repr(db[key])

'2' 'two'
'3' 'three'
'1' 'one'
```

10.3. whichdb

whichdb , Example 10-2 .

10.3.0.1. Example 10-2. whichdb

```
File: whichdb-example-1.py

import whichdb

filename = "database"

result = whichdb.whichdb(filename)

if result:
    print "file created by", result
    handler = _ _import_ _(result)
    db = handler.open(filename, "r")
    print db.keys()
else:
    # cannot identify data base
    if result is None:
        print "cannot read database file", filename
    else:
        print "cannot identify database file", filename
    db = None
```

_ _import_ _ (
?).

10.4. shelve

shelve . shelve , , pickle . Example 10-3 .

10.4.0.1. Example 10-3. shelve

```
File: shelve-example-1.py

import shelve

db = shelve.open("database", "c")
db["one"] = 1
db["two"] = 2
db["three"] = 3
db.close()

db = shelve.open("database", "r")
for key in db.keys():
    print repr(key), repr(db[key])
```

```
'one' 1
'three' 3
'two' 2
```

Example 10-4 shelve .

10.4.0.2. Example 10-4. shelve

```
File: shelve-example-3.py

import shelve
import gdbm

def gdbm_shelve(filename, flag="c"):
    return shelve.Shelf(gdbm.open(filename, flag))

db = gdbm_shelve("dbfile")
```

10.5. dbhash

( ) dbhash bsddb dbm . Example 10-5 .

10.5.0.1. Example 10-5. dbhash

```
File: dbhash-example-1.py

import dbhash

db = dbhash.open("dbhash", "c")
db["one"] = "the foot"
db["two"] = "the shoulder"
db["three"] = "the other foot"
db["four"] = "the bridge of the nose"
db["five"] = "the naughty bits"
db["six"] = "just above the elbow"
db["seven"] = "two inches to the right of a very naughty bit indeed"
db["eight"] = "the kneecap"
db.close()

db = dbhash.open("dbhash", "r")
for key in db.keys():
    print repr(key), repr(db[key])
```

10.6. dbm

( ) dbm dbm (Unix). Example 10-6 .

10.6.0.1. Example 10-6. dbm

```
File: dbm-example-1.py

import dbm

db = dbm.open("dbm", "c")
db["first"] = "bruce"
db["second"] = "bruce"
db["third"] = "bruce"
db["fourth"] = "bruce"
db["fifth"] = "michael"
db["fifth"] = "bruce" # overwrite
db.close()

db = dbm.open("dbm", "r")
for key in db.keys():
    print repr(key), repr(db[key])

'first' 'bruce'
'second' 'bruce'
'fourth' 'bruce'
'third' 'bruce'
'fifth' 'bruce'
```

10.7. dumbdbm

dumbdbm , dbm , Python . : (*.dat*) , (*.dir*) .

10.7.0.1. Example 10-7. dumbdbm

```
File: dumbdbm-example-1.py

import dumbdbm

db = dumbdbm.open("dumbdbm", "c")
db["first"] = "fear"
db["second"] = "surprise"
db["third"] = "ruthless efficiency"
db["fourth"] = "an almost fanatical devotion to the Pope"
db["fifth"] = "nice red uniforms"
db.close()

db = dumbdbm.open("dumbdbm", "r")
for key in db.keys():
    print repr(key), repr(db[key])

'first' 'fear'
'third' 'ruthless efficiency'
'fifth' 'nice red uniforms'
'second' 'surprise'
'fourth' 'an almost fanatical devotion to the Pope'
```

10.8. gdbm

( ) gdbm GNU dbm , Example 10-8 .

10.8.0.1. Example 10-8. gdbm

```
File: gdbm-example-1.py

import gdbm

db = gdbm.open("gdbm", "c")
db["1"] = "call"
db["2"] = "the"
db["3"] = "next"
db["4"] = "defendant"
db.close()


db = gdbm.open("gdbm", "r")

keys = db.keys()
keys.sort()
for key in keys:
    print db[key],

call the next defendant
```

11.

.

11.1. dis

dis Python .
.

.
:

```
$ dis.py hello.py

          0 SET_LINENO          0

          3 SET_LINENO          1
          6 LOAD_CONST              0 ('hello again, and welcome to the
show')
          9 PRINT_ITEM
         10 PRINT_NEWLINE
         11 LOAD_CONST              1 (None)
         14 RETURN_VALUE
```

dis . dis , , , code . Example 11-1 .

11.1.0.1. Example 11-1. dis

```
File: dis-example-1.py

import dis

def procedure():
    print 'hello'

dis.dis(procedure)

          0 SET_LINENO          3

          3 SET_LINENO          4
          6 LOAD_CONST          1 ('hello')
          9 PRINT_ITEM
         10 PRINT_NEWLINE
         11 LOAD_CONST          0 (None)
         14 RETURN_VALUE
```

11.2. pdb

pdb Python (debugger). bdb .

(n , help):

```
$ pdb.py hello.py
> hello.py(0)?()
(Pdb) n
> hello.py()
(Pdb) n
hello again, and welcome to the show
```

```
--Return--
> hello.py(1)?()->None
(Pdb)
```

Example 11-2 .

11.2.0.1. Example 11-2. pdb

```
File: pdb-example-1.py

import pdb

def test(n):
    j = 0
    for i in range(n):
        j = j + i
    return n

db = pdb.Pdb()
db.runcall(test, 1)

> pdb-example-1.py(3)test()
-> def test(n):
(Pdb) s
> pdb-example-1.py(4)test()
-> j = 0
(Pdb) s
> pdb-example-1.py(5)test()
-> for i in range(n):
...
```

11.3. bdb

bdb . ,
Example 11-3 .

Bdb , user (
). set .

11.3.0.1. Example 11-3. bdb

```
File: bdb-example-1.py
```

```
import bdb
import time

def spam(n):
    j = 0
    for i in range(n):
        j = j + i
    return n

def egg(n):
    spam(n)
    spam(n)
    spam(n)
    spam(n)

def test(n):
    egg(n)

class myDebugger(bdb.Bdb):

    run = 0

    def user_call(self, frame, args):
        name = frame.f_code.co_name or "<unknown>"
        print "call", name, args
        self.set_continue() # continue

    def user_line(self, frame):
        if self.run:
            self.run = 0
            self.set_trace() # start tracing
        else:
            # arrived at breakpoint
            name = frame.f_code.co_name or "<unknown>"
            filename = self.canonic(frame.f_code.co_filename)
            print "break at", filename, frame.f_lineno, "in", name
        print "continue..."
        self.set_continue() # continue to next breakpoint

    def user_return(self, frame, value):
        name = frame.f_code.co_name or "<unknown>"
        print "return from", name, value
        print "continue..."
        self.set_continue() # continue
    def user_exception(self, frame, exception):
        name = frame.f_code.co_name or "<unknown>"
        print "exception in", name, exception
        print "continue..."
        self.set_continue() # continue

db = myDebugger()
db.run = 1
db.set_break("bdb-example-1.py", 7)
db.runcall(test, 1)
```

```
continue...
call egg None
call spam None
break at C:\ematter\librarybook\bdb-example-1.py 7 in spam
continue...
call spam None
break at C:\ematter\librarybook\bdb-example-1.py 7 in spam
continue...
call spam None
break at C:\ematter\librarybook\bdb-example-1.py 7 in spam
continue...
call spam None
break at C:\ematter\librarybook\bdb-example-1.py 7 in spam
continue...
```

11.4. profile

profile Python .

, , :

```
$ profile.py hello.py

hello again, and welcome to the show

         3 function calls in 0.785 CPU seconds

   Ordered by: standard name

   ncalls  tottime  percall  cumtime  percall
filename:lineno(function)
```

```
        1  0.001    0.001    0.002    0.002 <string>:1(?)
        1  0.001    0.001    0.001    0.001 hello.py:1(?)
        1  0.783    0.783    0.785    0.785
profile:0(execfile('hello.py'))
        0  0.000             0.000          profile:0(profiler)
```

Example 11-4 , profile .

11.4.0.1. Example 11-4. U profile

```
File: profile-example-1.py

import profile

def func1():
    for i in range(1000):
        pass

def func2():
    for i in range(1000):
        func1()

profile.run("func2()")

        1003 function calls in 2.380 CPU seconds

  Ordered by: standard name

  ncalls  tottime  percall  cumtime  percall
filename:lineno(function)
        1  0.000    0.000    2.040    2.040 <string>:1(?)
     1000  1.950    0.002    1.950    0.002
profile-example-1.py:3(func1)
        1  0.090    0.090    2.040    2.040
profile-example-1.py:7(func2)
        1  0.340    0.340    2.380    2.380 profile:0(func2())
        0  0.000             0.000          profile:0(profiler)
```

pstats .

11.5. pstats

pstats Python . Example 11-5 .

11.5.0.1. Example 11-5. pstats

```
File: pstats-example-1.py

import pstats
import profile

def func1():
    for i in range(1000):
        pass

def func2():
    for i in range(1000):
        func1()

p = profile.Profile()
p.run("func2()")

s = pstats.Stats(p)
s.sort_stats("time", "name").print_stats()

         1003 function calls in 1.574 CPU seconds

   Ordered by: internal time, function name

   ncalls  tottime  percall  cumtime  percall
filename:lineno(function)
     1000    1.522    0.002    1.522    0.002
pstats-example-1.py:4(func1)
        1    0.051    0.051    1.573    1.573
pstats-example-1.py:8(func2)
        1    0.001    0.001    1.574    1.574 profile:0(func2())
        1    0.000    0.000    1.573    1.573 <string>:1(?)
        0    0.000             0.000          profile:0(profiler)
```

11.6. tabnanny

(2.0) tabnanny Python . tab , nanny ( ) .

badtabs.py , if 4 1 tab . .

```
$ tabnanny.py -v samples/badtabs.py
';samples/badtabs.py': *** Line 3: trouble in tab city! ***
offending line:         print "world"

indent not equal e.g. at tab sizes 1, 2, 3, 5, 6, 7, 9
```

Python tab 8 , . (tab 8) . , nanny .

Example 11-6 tabnanny .

11.6.0.1. Example 11-6. tabnanny

```
File: tabnanny-example-1.py

import tabnanny

FILE = "samples/badtabs.py"

file = open(FILE)
for line in file.readlines():
    print repr(line)

# let tabnanny look at it
tabnanny.check(FILE)
```

```
'if 1:\012'
'    \011print "hello"\012'
'        print "world"\012'
samples/badtabs.py 3 '        print "world"'\012'
```

sys.stdout StringIO .

12.

12.1.

. .
(Unix , Windows)

12.2. fcntl

(Unix) fcntl Unix ioctl fcntl . I/O "out of band" , , . . (out of band management: . , . : http://en.wikipedia.org/wiki/Out-of-band_management)

, Unix man .

Unix . Example 12-1 flock , *advisory lock* .

3 . ( ):

python fcntl-example-1.py& python fcntl-example-1.py& python fcntl-example-1.py&

flock , counter .

12.2.0.1. Example 12-1. Using the fcntl Module

```
File: fcntl-example-1.py

import fcntl, FCNTL
import os, time

FILE = "counter.txt"

if not os.path.exists(FILE):
    # create the counter file if it doesn't exist
        #     counter
    file = open(FILE, "w")
    file.write("0")
    file.close()

for i in range(20):
    # increment the counter
    file = open(FILE, "r+")
    fcntl.flock(file.fileno(), FCNTL.LOCK_EX)
    counter = int(file.readline()) + 1
    file.seek(0)
    file.write(str(counter))
    file.close() # unlocks the file
    print os.getpid(), "=>", counter
    time.sleep(0.1)
```

```
30940 => 1
30942 => 2
30941 => 3
30940 => 4
30941 => 5
30942 => 6
```

12.3. pwd

(Unix) pwd Unix /password " "(/etc/passwd) . ( ) . Example 12-2 .

12.3.0.1. Example 12-2. pwd

```
File: pwd-example-1.py

import pwd
import os

print pwd.getpwuid(os.getgid())
print pwd.getpwnam("root")

('effbot', 'dsWjk8', 4711, 4711, 'eff-bot', '/home/effbot', '/bin/bosh')
('root', 'hs2giiw', 0, 0, 'root', '/root', '/bin/bash')
```

getpwall . .

, getpwall , Example 12-3 .

12.3.0.2. Example 12-3. pwd

```
File: pwd-example-2.py
```

```
import pwd
import os

# preload password dictionary
_pwd = {}
for info in pwd.getpwall():
    _pwd[info[0]] = _pwd[info[2]] = info

def userinfo(uid):
    # name or uid integer
    return _pwd[uid]

print userinfo(os.getuid())
print userinfo("root")

('effbot', 'dsWjk8', 4711, 4711, 'eff-bot', '/home/effbot', '/bin/bosh')
('root', 'hs2giiw', 0, 0, 'root', '/root', '/bin/bash')
```

12.4. grp

(Unix) grp Unix /group (/etc/group) . getgrgid id (Example 12-4), getgrnam .

12.4.0.1. Example 12-4. grp

File: grp-example-1.py

```
import grp
import os

print grp.getgrgid(os.getgid())
print grp.getgrnam("wheel")

('effbot', '', 4711, ['effbot'])
('wheel', '', 10, ['root', 'effbot', 'gorbot', 'timbot'])
```

getgrall .

, getgrall , . Example 12-5 groupinfo id (int) (str) .

12.4.0.2. Example 12-5.   grp

```
File: grp-example-2.py

import grp
import os

# preload password dictionary
_grp = {}
for info in grp.getgrall():
    _grp[info[0]] = _grp[info[2]] = info

def groupinfo(gid):
    # name or gid integer
    return _grp[gid]

print groupinfo(os.getgid())
print groupinfo("wheel")

('effbot', '', 4711, ['effbot'])
('wheel', '', 10, ['root', 'effbot', 'gorbot', 'timbot'])
```

12.5. nis

(????? Unix , ???) nis ??????? NIS (Network Information Services , ??????????? ,???) ???????, ?? Example 12-6 ???. ??????????? NIS ?????л?????.

12.5.0.1. Example 12-6. ??? nis ???

```
File: nis-example-1.py

import nis
import string

print nis.cat("ypservers")
print string.split(nis.match("bacon", "hosts.byname"))

{'bacon.spam.egg': 'bacon.spam.egg'}
['194.18.155.250', 'bacon.spam.egg', 'bacon', 'spam-010']
```

12.6. curses

(????? Unix ???)
curses ????????????????????????, ?????????????????????. ?? Example 12-7 ???.

12.6.0.1. Example 12-7. ??? curses ???

File: curses-example-1.py

```
import curses

text = [
    "a very simple curses demo",
    "",
    "(press any key to exit)"
]

# connect to the screen
# ???????
screen = curses.initscr()

# setup keyboard
# ???ü???
curses.noecho() # no keyboard echo
curses.cbreak() # don't wait for newline

# screen size
# ??????
rows, columns = screen.getmaxyx()

# draw a border around the screen
# ???????
screen.border()

# display centered text
# ???????
y = (rows - len(text)) / 2

for line in text:
    screen.addstr(y, (columns-len(line))/2, line)
    y = y + 1

screen.getch()
```

```
curses.endwin()
```

12.7. termios

(Unix ,) termios Unix .
.

Example 12-8 , (ECHO).

12.7.0.1. Example 12-8. termios

```
File: termios-example-1.py

import termios, TERMIOS
import sys

fileno = sys.stdin.fileno()

attr = termios.tcgetattr(fileno)
orig = attr[:]

print "attr =>", attr[:4] # flags

# disable echo flag
attr[3] = attr[3] & ~TERMIOS.ECHO

try:
    termios.tcsetattr(fileno, TERMIOS.TCSADRAIN, attr)
    message = raw_input("enter secret message: ")
    print
finally:
    # restore terminal settings
    termios.tcsetattr(fileno, TERMIOS.TCSADRAIN, orig)

print "secret =>", repr(message)
```

```
attr => [1280, 5, 189, 35387]
enter secret message:
secret => 'and now for something completely different'
```

12.8. tty

(Unix) tty tty . Example 12-9 "raw" .

12.8.0.1. Example 12-9. tty

```
File: tty-example-1.py

import tty
import os, sys

fileno = sys.stdin.fileno()

tty.setraw(fileno)
print raw_input("raw input: ")

tty.setcbreak(fileno)
print raw_input("cbreak input: ")

os.system("stty sane") # ...

raw input: this is raw input
cbreak input: this is cbreak input
```

12.9. resource

(Unix ,) resource . Example 12-10 , Example 12-11 .

12.9.0.1. Example 12-10. resource

```
File: resource-example-1.py

import resource

print "usage stats", "=>", resource.getrusage(resource.RUSAGE_SELF)
print "max cpu", "=>", resource.getrlimit(resource.RLIMIT_CPU)
print "max data", "=>", resource.getrlimit(resource.RLIMIT_DATA)
print "max processes", "=>", resource.getrlimit(resource.RLIMIT_NPROC)
print "page size", "=>", resource.getpagesize()
```

```
usage stats => (0.03, 0.02, 0, 0, 0, 0, 75, 168, 0, 0, 0, 0, 0, 0, 0,
0)
max cpu => (2147483647, 2147483647)
max data => (2147483647, 2147483647)
max processes => (256, 256)
page size => 4096
```

12.9.0.2. Example 12-11. resource

```
File: resource-example-2.py

import resource

resource.setrlimit(resource.RLIMIT_CPU, (0, 1))

# pretend we're busy
for i in range(1000):
    for j in range(1000):
        for k in range(1000):
            pass
```

```
CPU time limit exceeded
```

12.10. syslog

(Unix) syslog (syslogd). , log , *ar/log/messages* , */var/adm/syslog* , . (,). Example 12-12 .

12.10.0.1. Example 12-12. syslog

```
File: syslog-example-1.py

import syslog
import sys

syslog.openlog(sys.argv[0])

syslog.syslog(syslog.LOG_NOTICE, "a log notice")
syslog.syslog(syslog.LOG_NOTICE, "another log notice: %s" % "watch
out!")
```

```
syslog.closelog()
```

12.11. msvcrt

(Windows/DOS) msvcrt Microsoft Visual C/C++ Runtime Library (MSVCRT) .

Example 12-13 getch , .

12.11.0.1. Example 12-13. msvcrt

```
File: msvcrt-example-1.py

import msvcrt

print "press 'escape' to quit..."

while 1:
    char = msvcrt.getch()
    if char == chr(27):
        break
    print char,
    if char == chr(13):
        print
```

```
press 'escape' to quit...
h e l l o
```

kbhit (getch), Example 12-14 .

12.11.0.2. Example 12-14. msvcrt

```
File: msvcrt-example-2.py

import msvcrt
import time

print "press SPACE to enter the serial number"

while not msvcrt.kbhit() or msvcrt.getch() != " ":
    # do something else
    print ".",
    time.sleep(0.1)

print

# clear the keyboard buffer
#
while msvcrt.kbhit():
    msvcrt.getch()

serial = raw_input("enter your serial number: ")

print "serial number is", serial
```

```
press SPACE to enter the serial number
. . . . . . . . . . . . . . . . . . . . . . . . .
enter your serial number: 10
serial number is 10
```

: : cmd . IDLE ,
. . IDLE python ,
socket , .

locking Windows , Example 12-15 .

12.11.0.3. Example 12-15. msvcrt

```
File: msvcrt-example-3.py

import msvcrt
import os

LK_UNLCK = 0 # unlock the file region
LK_LOCK = 1 # lock the file region
LK_NBLCK = 2 # non-blocking lock
LK_RLCK = 3 # lock for writing
LK_NBRLCK = 4 # non-blocking lock for writing


FILE = "counter.txt"

if not os.path.exists(FILE):
    file = open(FILE, "w")
    file.write("0")
    file.close()
```

```
for i in range(20):
    file = open(FILE, "r+")
    # look from current position (0) to end of file
    msvcrt.locking(file.fileno(), LK_LOCK, os.path.getsize(FILE))
    counter = int(file.readline()) + 1
    file.seek(0)
    file.write(str(counter))
    file.close() # unlocks the file
    print os.getpid(), "=>", counter
    time.sleep(0.1)
```

```
208 => 21
208 => 22
208 => 23
208 => 24
208 => 25
208 => 26
```

12.12. nt

(, Windows) nt os Windows . , os . Example 12-16 .

12.12.0.1. Example 12-16. nt

```
File: nt-example-1.py

import nt

# in real life, use os.listdir and os.stat instead!
for file in nt.listdir("."):
    print file, nt.stat(file)[6]
```

```
aifc-example-1.py 314
anydbm-example-1.py 259
array-example-1.py 48
```

12.13. _winreg

(Windows , 2.0) _winreg Windows . Example 12-17 .

12.13.0.1. Example 12-17. _winreg

```
File: winreg-example-1.py

import _winreg

explorer = _winreg.OpenKey(
    _winreg.HKEY_CURRENT_USER,
    "Software\\Microsoft\\Windows\CurrentVersion\\Explorer"
    )

# list values owned by this registry key
#
try:
    i = 0
    while 1:
      name, value, type= _winreg.EnumValue(explorer, i)
      print repr(name),
      i += 1
except WindowsError:
    print

value, type = _winreg.QueryValueEx(explorer, "Logon User Name")

print
print "user is", repr(value)
```

```
'Logon User Name' 'CleanShutdown' 'ShellState' 'Shutdown Setting'
'Reason Setting' 'FaultCount' 'FaultTime' 'IconUnderline'...

user is u'Effbot'
```

12.14. posix

(, Unix/POSIX) posix os Unix POSIX . os . Example 12-18 .

12.14.0.1. Example 12-18. posix

```
File: posix-example-1.py

import posix

for file in posix.listdir("."):
    print file, posix.stat(file)[6]
```

```
aifc-example-1.py 314
anydbm-example-1.py 259
array-example-1.py 48
```

13.

.

13.1. dospath

dospath (Example 13-1) DOS os.path . DOS .

13.1.0.1. Example 13-1. dospath

```
File: dospath-example-1.py

import dospath

file = "/my/little/pony"

print "isabs", "=>", dospath.isabs(file)
print "dirname", "=>", dospath.dirname(file)
print "basename", "=>", dospath.basename(file)
print "normpath", "=>", dospath.normpath(file)
print "split", "=>", dospath.split(file)
print "join", "=>", dospath.join(file, "zorba")
```

```
isabs => 1
dirname => /my/little
basename => pony
normpath => \my\little\pony
split => ('/my/little', 'pony')
```

```
join => /my/little/pony\zorba
```

Python DOS .

13.2. macpath

macpath (Example 13-2) Macintosh os.path . Macintosh .

13.2.0.1. Example 13-2. macpath

```
File: macpath-example-1.py

import macpath

file = "my:little:pony"

print "isabs", "=>", macpath.isabs(file)
print "dirname", "=>", macpath.dirname(file)
print "basename", "=>", macpath.basename(file)
print "normpath", "=>", macpath.normpath(file)
print "split", "=>", macpath.split(file)
print "join", "=>", macpath.join(file, "zorba")
```

```
isabs => 1
dirname => my:little
basename => pony
normpath => my:little:pony
split => ('my:little', 'pony')
join => my:little:pony:zorba
```

13.3. ntpath

ntpath (Example 13-3) Windows os.path . Windows .

13.3.0.1. Example 13-3. ntpath

```
File: ntpath-example-1.py
```

```
import ntpath

file = "/my/little/pony"

print "isabs", "=>", ntpath.isabs(file)
print "dirname", "=>", ntpath.dirname(file)
print "basename", "=>", ntpath.basename(file)
print "normpath", "=>", ntpath.normpath(file)
print "split", "=>", ntpath.split(file)
print "join", "=>", ntpath.join(file, "zorba")
```

```
isabs => 1
dirname => /my/little
basename => pony

normpath => \my\little\pony
split => ('/my/little', 'pony')
join => /my/little/pony\zorba
```

.

13.4. posixpath

posixpath (Example 13-4) Unix POSIX
os.path . POSIX . ,
URL .

13.4.0.1. Example 13-4. posixpath

File: posixpath-example-1.py

```
import posixpath

file = "/my/little/pony"

print "isabs", "=>", posixpath.isabs(file)
print "dirname", "=>", posixpath.dirname(file)
print "basename", "=>", posixpath.basename(file)
print "normpath", "=>", posixpath.normpath(file)
print "split", "=>", posixpath.split(file)
print "join", "=>", posixpath.join(file, "zorba")
```

```
isabs => 1
dirname => /my/little
basename => pony
normpath => /my/little/pony
split => ('/my/little', 'pony')
join => /my/little/pony/zorba
```

13.5. strop

( ) strop string C .string
, .

Python . Example
13-5 .

13.5.0.1. Example 13-5. strop

```
File: strop-example-1.py

import strop
import sys

# assuming we have an executable named ".../executable", add a
# directory named ".../executable-extra" to the path

if strop.lower(sys.executable)[-4:] == ".exe":
    extra = sys.executable[:-4] # windows
else:
    extra = sys.executable

sys.path.insert(0, extra + "-extra")

import mymodule
```

Python 2.0 , strop ,
. "sys.executable.lower() "
"strop.lower(sys.executable) " .

13.6. imp

imp import . Example 13-6 import , .

13.6.0.1. Example 13-6. imp

```
File: imp-example-1.py

import imp
import sys

def my_import(name, globals=None, locals=None, fromlist=None):
    try:
        module = sys.modules[name] # already imported?
    except KeyError:
        file, pathname, description = imp.find_module(name)
        print "import", name, "from", pathname, description
        module = imp.load_module(name, file, pathname, description)
    return module

import _ _builtin_ _
_ _builtin_ _._ _import_ _ = my_import

import xmllib

import xmllib from /python/lib/xmllib.py ('.py', 'r', 1)
import re from /python/lib/re.py ('.py', 'r', 1)

import sre from /python/lib/sre.py ('.py', 'r', 1)
import sre_compile from /python/lib/sre_compile.py ('.py', 'r', 1)
import _sre from /python/_sre.pyd ('.pyd', 'rb', 3)
```

. knee .

13.7. new

new , , , , Python . Example 13-7 .

1.5.2 , Python , . 2.0 , .

13.7.0.1. Example 13-7.   new

```
File: new-example-1.py

import new

class Sample:

    a = "default"

    def _ _init_ _(self):
        self.a = "initialised"

    def _ _repr_ _(self):
        return self.a

#
# create instances

a = Sample()
print "normal", "=>", a

b = new.instance(Sample, {})
print "new.instance", "=>", b

b._ _init_ _()
print "after _ _init_ _", "=>", b

c = new.instance(Sample, {"a": "assigned"})
print "new.instance w. dictionary", "=>", c
```

```
normal => initialised
new.instance => default
after _ _init_ _ => initialised
new.instance w. dictionary => assigned
```

13.8. pre

(   ) pre   1.5.2   re   .
      . Example 13-8   .

13.8.0.1. Example 13-8.   pre

```
File: pre-example-1.py

import pre

p = pre.compile("[Python]+")

print p.findall("Python is not that bad")

['Python', 'not', 'th', 't']
```

13.9. sre

(,) sre re .
, . Example 13-9 .

13.9.0.1. Example 13-9. sre

```
File: sre-example-1.py

import sre

text = "The Bookshop Sketch"

# a single character
m = sre.match(".", text)
if m: print repr("."), "=>", repr(m.group(0))

# and so on, for all 're' examples...

'.' => 'T'
```

13.10. py_compile

py_compile Python . Python
import , .
Example 13-10 .

13.10.0.1. Example 13-10. py_compile

```
File: py-compile-example-1.py
```

```
import py_compile

# explicitly compile this module
py_compile.compile("py-compile-example-1.py")
```

compileall Python .

13.11. compileall

compileall (Python path) Python . (Unix , Python). Example 13-11 .

13.11.0.1. Example 13-11. compileall

```
File: compileall-example-1.py

import compileall

print "This may take a while!"

compileall.compile_dir(".", force=1)
```

```
This may take a while!
Listing . ...
Compiling .\SimpleAsyncHTTP.py ...
Compiling .\aifc-example-1.py ...
Compiling .\anydbm-example-1.py ...
...
```

13.12. ihooks

ihooks . . Example 13-12 .

13.12.0.1. Example 13-12. ihooks

```
File: ihooks-example-1.py

import ihooks, imp, os
```

```
def import_from(filename):
    "Import module from a named file"

    loader = ihooks.BasicModuleLoader()
    path, file = os.path.split(filename)
    name, ext  = os.path.splitext(file)
    m = loader.find_module_in_dir(name, path)
    if not m:
        raise ImportError, name
    m = loader.load_module(name, m)
    return m

colorsys = import_from("/python/lib/colorsys.py")

print colorsys
```

```
<module 'colorsys' from '/python/lib/colorsys.py'>
```

13.13. linecache

linecache . ( ). Example 13-13 .

13.13.0.1. Example 13-13. linecache

```
File: linecache-example-1.py

import linecache

print linecache.getline("linecache-example-1.py", 5)
```

```
print linecache.getline("linecache-example-1.py", 5)
```

traceback .

13.14. macurl2path

( ) macurl2path URL Macintosh .
, urllib . Example 13-14 .

13.14.0.1. Example 13-14. macurl2path

```
File: macurl2path-example-1.py

import macurl2path

file = ":my:little:pony"

print macurl2path.pathname2url(file)
print macurl2path.url2pathname(macurl2path.pathname2url(file))
```

```
my/little/pony
:my:little:pony
```

13.15. nturl2path

( ) nturl2path URL Windows .
Example 13-15 .

13.15.0.1. Example 13-15. nturl2path

```
File: nturl2path-example-1.py

import nturl2path

file = r"c:\my\little\pony"

print nturl2path.pathname2url(file)
print nturl2path.url2pathname(nturl2path.pathname2url(file))
```

```
///C|/my/little/pony
C:\my\little\pony
```

, urllib , Example 13-16 .

13.15.0.2. Example 13-16. urllib nturl2path

```
File: nturl2path-example-2.py
```

```
import urllib

file = r"c:\my\little\pony"

print urllib.pathname2url(file)
print urllib.url2pathname(urllib.pathname2url(file))
```

```
///C|/my/little/pony
C:\my\little\pony
```

13.16. tokenize

tokenize Python token . .

Example 13-17 , token .

13.16.0.1. Example 13-17. tokenize

```
File: tokenize-example-1.py

import tokenize

file = open("tokenize-example-1.py")

def handle_token(type, token, (srow, scol), (erow, ecol), line):
    print "%d,%d-%d,%d:\t%s\t%s" % \
        (srow, scol, erow, ecol, tokenize.tok_name[type], repr(token))

tokenize.tokenize(
    file.readline,
    handle_token
    )
```

```
1,0-1,6:     NAME    'import'
1,7-1,15:    NAME    'tokenize'
1,15-1,16:   NEWLINE '\012'
2,0-2,1:     NL      '\012'
3,0-3,4:     NAME    'file'
3,5-3,6:     OP      '='
3,7-3,11:    NAME    'open'
3,11-3,12:   OP      '('
3,12-3,35:   STRING  '"tokenize-example-1.py"'
3,35-3,36:   OP      ')'
3,36-3,37:   NEWLINE '\012'
...
```

tokenize : , token .

13.17. keyword

keyword (Example 13-18) Python . , key , value , Python .

13.17.0.1. Example 13-18. keyword

```
File: keyword-example-1.py

import keyword

name = raw_input("Enter module name: ")

if keyword.iskeyword(name):
    print name, "is a reserved word."
    print "here's a complete list of reserved words:"
    print keyword.kwlist

Enter module name: assert
assert is a reserved word.
here's a complete list of reserved words:
['and', 'assert', 'break', 'class', 'continue', 'def', 'del',
'elif', 'else', 'except', 'exec', 'finally', 'for', 'from',
'global', 'if', 'import', 'in', 'is', 'lambda', 'not', 'or',
'pass', 'print', 'raise', 'return', 'try', 'while']
```

13.18. parser

( ) parser Python .

Example 13-19 (abstract syntax tree , AST), AST , (

token) , 1 ,

. .

13.18.0.1. Example 13-19. parser

```
File: parser-example-1.py

import parser
import symbol, token

def dump_and_modify(node):
    name = symbol.sym_name.get(node[0])
    if name is None:
        name = token.tok_name.get(node[0])
    print name,
    for i in range(1, len(node)):
        item = node[i]
        if type(item) is type([]):
            dump_and_modify(item)
        else:
            print repr(item)
            if name == "NUMBER":
                # increment all numbers!
                node[i] = repr(int(item)+1)

ast = parser.expr("1 + 3")

list = ast.tolist()

dump_and_modify(list)

ast = parser.sequence2ast(list)

print eval(parser.compileast(ast))
```

```
eval_input testlist test and_test not_test comparison
expr xor_expr and_expr shift_expr arith_expr term factor
power atom NUMBER '1'
PLUS '+'
term factor power atom NUMBER '3'
NEWLINE ''
ENDMARKER ''
6
```

13.19. symbol

symbol Python . parser . Example 13-20 .

13.19.0.1. Example 13-20. symbol

```
File: symbol-example-1.py

import symbol

print "print", symbol.print_stmt
print "return", symbol.return_stmt
```

```
print 268
return 274
```

13.20. token

token Python tokenizer token . Example 13-21 .

13.20.0.1. Example 13-21. token

```
File: token-example-1.py

import token

print "NUMBER", token.NUMBER
print "PLUS", token.STAR
print "STRING", token.STRING
```

```
NUMBER 2
PLUS 16
STRING 3
```

14.

14.1.

. ,

.

14.2. pyclbr

pyclbr Python , Example 14-1 .

1.5.2 , readmodule , ,

.

14.2.0.1. Example 14-1. pyclbr

```
File: pyclbr-example-1.py

import pyclbr

mod = pyclbr.readmodule("cgi")

for k, v in mod.items():
    print k, v

MiniFieldStorage <pyclbr.Class instance at 7873b0>
InterpFormContentDict <pyclbr.Class instance at 79bd00>
FieldStorage <pyclbr.Class instance at 790e20>
SvFormContentDict <pyclbr.Class instance at 79b5e0>
StringIO <pyclbr.Class instance at 77dd90>
FormContent <pyclbr.Class instance at 79bd60>
FormContentDict <pyclbr.Class instance at 79a9c0>
```

2.0 , readmodule_ex , .
Example 14-2 .

14.2.0.2. Example 14-2. pyclbr

```
File: pyclbr-example-3.py

import pyclbr

# 2.0 and later
```

```
mod = pyclbr.readmodule_ex("cgi")

for k, v in mod.items():
    print k, v
```

```
MiniFieldStorage <pyclbr.Class instance at 00905D2C>
parse_header <pyclbr.Function instance at 00905BD4>
test <pyclbr.Function instance at 00906FBC>
print_environ_usage <pyclbr.Function instance at 00907C94>
parse_multipart <pyclbr.Function instance at 00905294>
FormContentDict <pyclbr.Class instance at 008D3494>
initlog <pyclbr.Function instance at 00904AAC>
parse <pyclbr.Function instance at 00904EFC>
StringIO <pyclbr.Class instance at 00903EAC>
SvFormContentDict <pyclbr.Class instance at 00906824>
...
```

, Example 14-3 .

14.2.0.3. Example 14-3. pyclbr

```
File: pyclbr-example-2.py

import pyclbr
import string

mod = pyclbr.readmodule("cgi")

def dump(c):
    # print class header
    s = "class " + c.name
    if c.super:
        s = s +  "(" + string.join(map(lambda v: v.name, c.super), ", ")
+ ")"
    print s + ":"
    # print method names, sorted by line number
    methods = c.methods.items()
    methods.sort(lambda a, b: cmp(a[1], b[1]))
    for method, lineno in methods:
        print "  def " + method
    print

for k, v in mod.items():
    dump(v)
```

```
class MiniFieldStorage:
  def __init__
  def __repr__

class InterpFormContentDict(SvFormContentDict):
  def __getitem__
  def values
  def items

...
```

14.3. filecmp

(2.0) filecmp , Example 14-4 .

14.3.0.1. Example 14-4. filecmp

```
File: filecmp-example-1.py

import filecmp

if filecmp.cmp("samples/sample.au", "samples/sample.wav"):
    print "files are identical"
else:
    print "files differ!"

# files differ!
```

1.5.2 , cmp dircmp .

14.4. cmd

cmd (command-line interfaces , CLI) . pdb , , Example 14-5 .

Cmd , do help . .

14.4.0.1. Example 14-5. cmd

File: cmd-example-1.py

```
import cmd
import string, sys

class CLI(cmd.Cmd):

    def _ _init_ _(self):
        cmd.Cmd._ _init_ _(self)
        self.prompt = '> '

    def do_hello(self, arg):
        print "hello again", arg, "!"

    def help_hello(self):
        print "syntax: hello [message]",
        print "-- prints a hello message"

    def do_quit(self, arg):
        sys.exit(1)

    def help_quit(self):
        print "syntax: quit",
        print "-- terminates the application"

    # shortcuts
    do_q = do_quit

#
# try it out

cli = CLI()
cli.cmdloop()
```

```
> help

Documented commands (type help <topic>):
========================================
hello           quit

Undocumented commands:
======================
help            q
```

```
> hello world
hello again world !
> q
```

14.5. rexec

Feather : 2.3 , :
http://www.amk.ca/python/howto/rexec/
http://mail.python.org/pipermail/python-dev/2002-December/031160.html

:
http://mail.python.org/pipermail/python-list/2003-November/234581.html

rexec exec , eval , import , Example 14-6 . , .

14.5.0.1. Example 14-6. rexec

```
File: rexec-example-1.py

import rexec

r = rexec.RExec()
print r.r_eval("1+2+3")
print r.r_eval("_ _import_ _('os').remove('file')")

6
Traceback (innermost last):
  File "rexec-example-1.py", line 5, in ?
    print r.r_eval("_ _import_ _('os').remove('file')")
  File "/usr/local/lib/python1.5/rexec.py", line 257, in r_eval
    return eval(code, m._ _dict_ _)
  File "<string>", line 0, in ?
AttributeError: remove
```

14.6. Bastion

Feather : 2.3 , :
http://www.amk.ca/python/howto/rexec/
http://mail.python.org/pipermail/python-dev/2003-January/031848.html

Bastion , , Example 14-7 .
.

, , .

14.6.0.1. Example 14-7. Bastion

```
File: bastion-example-1.py

import Bastion

class Sample:
    value = 0

    def _set(self, value):
        self.value = value

    def setvalue(self, value):
        if 10 < value <= 20:
            self._set(value)
        else:
            raise ValueError, "illegal value"

    def getvalue(self):
        return self.value

#
# try it

s = Sample()
s._set(100) # cheat
print s.getvalue()

s = Bastion.Bastion(Sample())
s._set(100) # attempt to cheat
print s.getvalue()

100
Traceback (innermost last):
...
AttributeError: _set
```

. Example 14- , ,
getvalue .

14.6.0.2. Example 14-8. Bastion

```
File: bastion-example-2.py

import Bastion

class Sample:
    value = 0

    def _set(self, value):
        self.value = value

    def setvalue(self, value):
        if 10 < value <= 20:
            self._set(value)
        else:
            raise ValueError, "illegal value"

    def getvalue(self):
        return self.value

#
# try it

def is_public(name):
    return name[:3] != "get"

s = Bastion.Bastion(Sample(), is_public)
s._set(100) # this works
print s.getvalue() # but not this
```

```
100
Traceback (innermost last):
...
AttributeError: getvalue
```

14.7. readline

( ) readline GNU readline ( ) Unix
. Example 14-9 .

, . input
raw_input .

14.7.0.1. Example 14-9. readline

File: readline-example-1.py

```
import readline # activate readline editing
```

14.8. rlcompleter

(, Unix) rlcompleter readline
.

. Esc
. Esc .
:

```
import readline
readline.parse_and_bind("tab: complete")
```

Example 14-10 .

14.8.0.1. Example 14-10. rlcompleter

File: rlcompleter-example-1.py

```
import rlcompleter
import sys

completer = rlcompleter.Completer()

for phrase in "co", "sys.p", "is":
    print phrase, "=>",
    # emulate readline completion handler
    try:
        for index in xrange(sys.maxint):
            term = completer.complete(phrase, index)
            if term is None:
                break
            print term,
    except:
        pass
    print
```

```
co => continue compile complex coerce completer
sys.p => sys.path sys.platform sys.prefix
is => is isinstance issubclass
```

14.9. statvfs

statvfs os.statvfs ( ) , . Example 14-11 .

14.9.0.1. Example 14-11. statvfs

```
File: statvfs-example-1.py

import statvfs
import os

st = os.statvfs(".")

print "preferred block size", "=>", st[statvfs.F_BSIZE]
print "fundamental block size", "=>", st[statvfs.F_FRSIZE]
print "total blocks", "=>", st[statvfs.F_BLOCKS]
print "total free blocks", "=>", st[statvfs.F_BFREE]
print "available blocks", "=>", st[statvfs.F_BAVAIL]
print "total file nodes", "=>", st[statvfs.F_FILES]
print "total free nodes", "=>", st[statvfs.F_FFREE]
print "available nodes", "=>", st[statvfs.F_FAVAIL]
print "max file name length", "=>", st[statvfs.F_NAMEMAX]
```

```
preferred block size => 8192
fundamental block size => 1024
total blocks => 749443
total free blocks => 110442
available blocks => 35497
total file nodes => 92158
total free nodes => 68164
available nodes => 68164
max file name length => 255
```

14.10. calendar

calendar Unix *cal* Python . /
.

prmonth(year, month) , Example 14-12 .

14.10.0.1. Example 14-12. calendar

```
File: calendar-example-1.py

import calendar
calendar.prmonth(1999, 12)

    December 1999
Mo Tu We Th Fr Sa Su
       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
```

prcal(year) , Example 14-13 .

14.10.0.2. Example 14-13. calendar

```
File: calendar-example-2.py

import calendar
calendar.prcal(2000)

                                  2000

       January                       February                       March
Mo Tu We Th Fr Sa Su          Mo Tu We Th Fr Sa Su          Mo Tu We Th Fr Sa
Su
                1  2              1  2  3  4  5  6                 1  2  3
4  5
 3  4  5  6  7  8  9           7  8  9 10 11 12 13           6  7  8  9 10 11
12
10 11 12 13 14 15 16          14 15 16 17 18 19 20          13 14 15 16 17 18
19
17 18 19 20 21 22 23          21 22 23 24 25 26 27          20 21 22 23 24 25
26
24 25 26 27 28 29 30          28 29                         27 28 29 30 31
31
```

```
        April                      May                      June
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa
Su
                1  2       1  2  3  4  5  6  7                1  2
3  4
 3  4  5  6  7  8  9       8  9 10 11 12 13 14       5  6  7  8  9 10
11
10 11 12 13 14 15 16      15 16 17 18 19 20 21      12 13 14 15 16 17
18
17 18 19 20 21 22 23      22 23 24 25 26 27 28      19 20 21 22 23 24
25
24 25 26 27 28 29 30      29 30 31                  26 27 28 29 30

        July                     August                   September
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa
Su
                1  2          1  2  3  4  5  6                   1
2  3
 3  4  5  6  7  8  9       7  8  9 10 11 12 13       4  5  6  7  8  9
10
10 11 12 13 14 15 16      14 15 16 17 18 19 20      11 12 13 14 15 16
17
17 18 19 20 21 22 23      21 22 23 24 25 26 27      18 19 20 21 22 23
24
24 25 26 27 28 29 30      28 29 30 31               25 26 27 28 29 30
31

       October                  November                  December
Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa Su      Mo Tu We Th Fr Sa
Su
                   1             1  2  3  4  5                   1
2  3
 2  3  4  5  6  7  8       6  7  8  9 10 11 12       4  5  6  7  8  9
10
 9 10 11 12 13 14 15      13 14 15 16 17 18 19      11 12 13 14 15 16
17
16 17 18 19 20 21 22      20 21 22 23 24 25 26      18 19 20 21 22 23
24
23 24 25 26 27 28 29      27 28 29 30               25 26 27 28 29 30
31
30 31

                                  ,                                 ,
                                . (          ,         )
```

.

14.11. sched

sched . Example 14-14
.

14.11.0.1. Example 14-14. sched

```
File: sched-example-1.py

import sched
import time, sys

scheduler = sched.scheduler(time.time, time.sleep)

# add a few operations to the queue
scheduler.enter(0.5, 100, sys.stdout.write, ("one\n",))
scheduler.enter(1.0, 300, sys.stdout.write, ("three\n",))
scheduler.enter(1.0, 200, sys.stdout.write, ("two\n",))

scheduler.run()
```

```
one
two
three
```

14.12. statcache

statcache . os.stat
, . Example 14-15 .

2.2 , os.stat() , ,
, .

14.12.0.1. Example 14-15. statcache

```
File: statcache-example-1.py

import statcache
import os, stat, time

now = time.time()
for i in range(1000):
    st = os.stat("samples/sample.txt")
print "os.stat", "=>", time.time() - now

now = time.time()
for i in range(1000):
    st = statcache.stat("samples/sample.txt")
print "statcache.stat", "=>", time.time() - now

print "mode", "=>", oct(stat.S_IMODE(st[stat.ST_MODE]))
print "size", "=>", st[stat.ST_SIZE]
print "last modified", "=>", time.ctime(st[stat.ST_MTIME])
```

```
os.stat => 0.371000051498
statcache.stat => 0.0199999809265
mode => 0666
size => 305
last modified => Sun Oct 10 18:39:37 1999
```

14.13. grep

grep , Example 14-16 .

2.1 , , .

14.13.0.1. Example 14-16. grep

```
File: grep-example-1.py

import grep
import glob

grep.grep("\<rather\>", glob.glob("samples/*.txt"))

# 4: indentation, rather than delimiters, might become
```

14.14. dircache

( ) statcache , os.listdir ,
, ~ MUHAHAHAHAHA~~~~ .
os.listdir . Example 14-17 .

14.14.0.1. Example 14-17. dircache

```
File: dircache-example-1.py

import dircache

import os, time

#
# test cached version

t0 = time.clock()

for i in range(100):
    dircache.listdir(os.sep)

print "cached", time.clock() - t0

#
# test standard version

t0 = time.clock()

for i in range(100):
    os.listdir(os.sep)

print "standard", time.clock() - t0
```

```
cached 0.0664509964968
standard 0.5560845807
```

14.15. dircmp

(, 1.5.2) dircmp , Example 14-18 .

14.15.0.1. Example 14-18. dircmp

```
File: dircmp-example-1.py
```

```
import dircmp

d = dircmp.dircmp()
d.new("samples", "oldsamples")
d.run()
d.report()
```

```
diff samples oldsamples
Only in samples : ['sample.aiff', 'sample.au', 'sample.wav']
Identical files : ['sample.gif', 'sample.gz', 'sample.jpg', ...]
```

Python 2.0 , filecmp .

14.16. cmp

(, 1.5.2) cmp , Example 14-19 .

14.16.0.1. Example 14-19. cmp

File: cmp-example-1.py

```
import cmp

if cmp.cmp("samples/sample.au", "samples/sample.wav"):
    print "files are identical"
else:
    print "files differ!"
```

```
files differ!
```

Python 2.0 , filecmp .

14.17. cmpcache

(, 1.5.2) cmpcache . cmp , . Example 14-20 .

14.17.0.1. Example 14-20. cmpcache

```
File: cmpcache-example-1.py

import cmpcache

if cmpcache.cmp("samples/sample.au", "samples/sample.wav"):
    print "files are identical"
else:
    print "files differ!"
```

```
files differ!
```

Python 2.0 , filecmp .

filecmp .

14.18. util

(, 1.5.2) util .
Examples 14-21 14-23 .

Example 14-21 remove(sequence, item) .

14.18.0.1. Example 14-21. util remove

```
File: util-example-1.py

def remove(sequence, item):
    if item in sequence:
        sequence.remove(item)
```

Example 14-22 readfile(filename) => string .

14.18.0.2. Example 14-22. util readfile

```
File: util-example-2.py

def readfile(filename):
    file = open(filename, "r")
    return file.read()
```

Example 14-23 `readopenfile(file) => string .

14.18.0.3. Example 14-23. util readopenfile

```
File: util-example-3.py

def readopenfile(file):
    return file.read()
```

14.19. soundex

(, 1.5.2) soundex hash , 6 .

2.0 , .

get_soundex(word) soundex . sound_similar(word1, word2) soundex . soundex . Example 14-24 .

14.19.0.1. Example 14-24. soundex

```
File: soundex-example-1.py

import soundex

a = "fredrik"
b = "friedrich"

print soundex.get_soundex(a), soundex.get_soundex(b)

print soundex.sound_similar(a, b)
```

```
F63620 F63620
1
```

14.20. timing

(, Unix) timing Python . Example 14-25 .

14.20.0.1. Example 14-25. timing

```
File: timing-example-1.py

import timing
import time

def procedure():
    time.sleep(1.234)

timing.start()
procedure()
timing.finish()

print "seconds:", timing.seconds()
print "milliseconds:", timing.milli()
print "microseconds:", timing.micro()

seconds: 1
milliseconds: 1239
microseconds: 1239999
```

Example 14-26 time timing .

14.20.0.2. Example 14-26. timing

```
File: timing-example-2.py

import time

t0 = t1 = 0

def start():
    global t0
    t0 = time.time()


def finish():
    global t1
    t1 = time.time()

def seconds():
    return int(t1 - t0)

def milli():
    return int((t1 - t0) * 1000)
```

```
def micro():
    return int((t1 - t0) * 1000000)
```

time.clock() time.time() CPU .

14.21. posixfile

(, Unix) posixfile (file-like object), . Example 14-27 . fcntl .

14.21.0.1. Example 14-27. posixfile

```
File: posixfile-example-1.py

import posixfile
import string

filename = "counter.txt"

try:
    # open for update
    file = posixfile.open(filename, "r+")
    counter = int(file.read(6)) + 1
except IOError:
    # create it
    file = posixfile.open(filename, "w")
    counter = 0

file.lock("w|", 6)

file.seek(0) # rewind
file.write("%06d" % counter)

file.close() # releases lock
```

14.22. bisect

bisect .

insort(sequence, item) , . _ _getitem_ _ insert . Example 14-28 .

14.22.0.1. Example 14-28. bisect

```
File: bisect-example-1.py

import bisect

list = [10, 20, 30]

bisect.insort(list, 25)
bisect.insort(list, 15)

print list
```

```
[10, 15, 20, 25, 30]
```

bisect(sequence, item) => index , . Example 14-29 .

14.22.0.2. Example 14-29. bisect

```
File: bisect-example-2.py

import bisect

list = [10, 20, 30]

print list
print bisect.bisect(list, 25)
print bisect.bisect(list, 15)
```

```
[10, 20, 30]
2
1
```

14.23. knee

knee Python 1.5 (package import) . Python , , , .

Python-X.tgz\Python-2.4.4\Demo\imputil\knee.py

, , Example 14-30 .

14.23.0.1. Example 14-30. knee

```
File: knee-example-1.py

import knee

# that's all, folks!
```

14.24. tzparse

( ) tzparse (time zone specification). TZ . Example 14-31 .

14.24.0.1. Example 14-31. tzparse

```
File: tzparse-example-1.py

import os
if not os.environ.has_key("TZ"):
    # set it to something...
    os.environ["TZ"] = "EST+5EDT;100/2,300/2"

# importing this module will parse the TZ variable
import tzparse

print "tzparams", "=>", tzparse.tzparams
print "timezone", "=>", tzparse.timezone
print "altzone", "=>", tzparse.altzone
print "daylight", "=>", tzparse.daylight
print "tzname", "=>", tzparse.tzname

tzparams => ('EST', 5, 'EDT', 100, 2, 300, 2)
timezone => 18000
altzone => 14400
daylight => 1
tzname => ('EST', 'EDT')
```

, .

14.25. regex

( ) regex (1.5) , Example 14-32 . re .

Python 1.5.2 regex re . re .

14.25.0.1. Example 14-32. regex

```
File: regex-example-1.py

import regex

text = "Man's crisis of identity in the latter half of the 20th century"

p = regex.compile("latter") # literal
print p.match(text)
print p.search(text), repr(p.group(0))

p = regex.compile("[0-9]+") # number
print p.search(text), repr(p.group(0))

p = regex.compile("\<\w\w\>") # two-letter word
print p.search(text), repr(p.group(0))

p = regex.compile("\w+$") # word at the end
print p.search(text), repr(p.group(0))
```

```
-1
32 'latter'
51 '20'
13 'of'
56 'century'
```

14.26. regsub

( ) regsub . Example 14-33 . re replace .

14.26.0.1. Example 14-33. regsub

```
File: regsub-example-1.py

import regsub

text = "Well, there's spam, egg, sausage, and spam."

print regsub.sub("spam", "ham", text) # just the first
print regsub.gsub("spam", "bacon", text) # all of them
```

```
Well, there's ham, egg, sausage, and spam.
Well, there's bacon, egg, sausage, and bacon.
```

14.27. reconvert

( ) reconvert (regex) (re) . Example 14-34 . .

14.27.0.1. Example 14-34. reconvert

```
File: reconvert-example-1.py

import reconvert

for pattern in "abcd", "a\(b*c\)d", "\<\w+\>":
    print pattern, "=>", reconvert.convert(pattern)
```

```
abcd => abcd
a\(b*c\)d => a(b*c)d
\<\w+\> => \b\w+\b
```

14.28. regex_syntax

( ) regex_syntax , Example 14-35 .

14.28.0.1. Example 14-35. regex_syntax

```
File: regex-syntax-example-1.py
```

```
import regex_syntax
import regex

def compile(pattern, syntax):
    syntax = regex.set_syntax(syntax)
    try:
        pattern = regex.compile(pattern)
    finally:
        # restore original syntax
        regex.set_syntax(syntax)
    return pattern

def compile_awk(pattern):
    return compile(pattern, regex_syntax.RE_SYNTAX_AWK)

def compile_grep(pattern):
    return compile(pattern, regex_syntax.RE_SYNTAX_GREP)

def compile_emacs(pattern):
    return compile(pattern, regex_syntax.RE_SYNTAX_EMACS)
```

14.29. find

(, 1.5.2) find
, Example 14-36 .

fnmatch .

14.29.0.1. Example 14-36. find

```
File: find-example-1.py

import find

# find all JPEG files in or beneath the current directory
for file in find.find("*.jpg", "."):
    print file
```

```
.\samples\sample.jpg
```

15. Py 2.0

, .

16.