Python & SA

Guixing<khsing.cn@gmail.com>

do what?

install software or hardware

backup

repair

monitor

analysis

adjust

even more

find

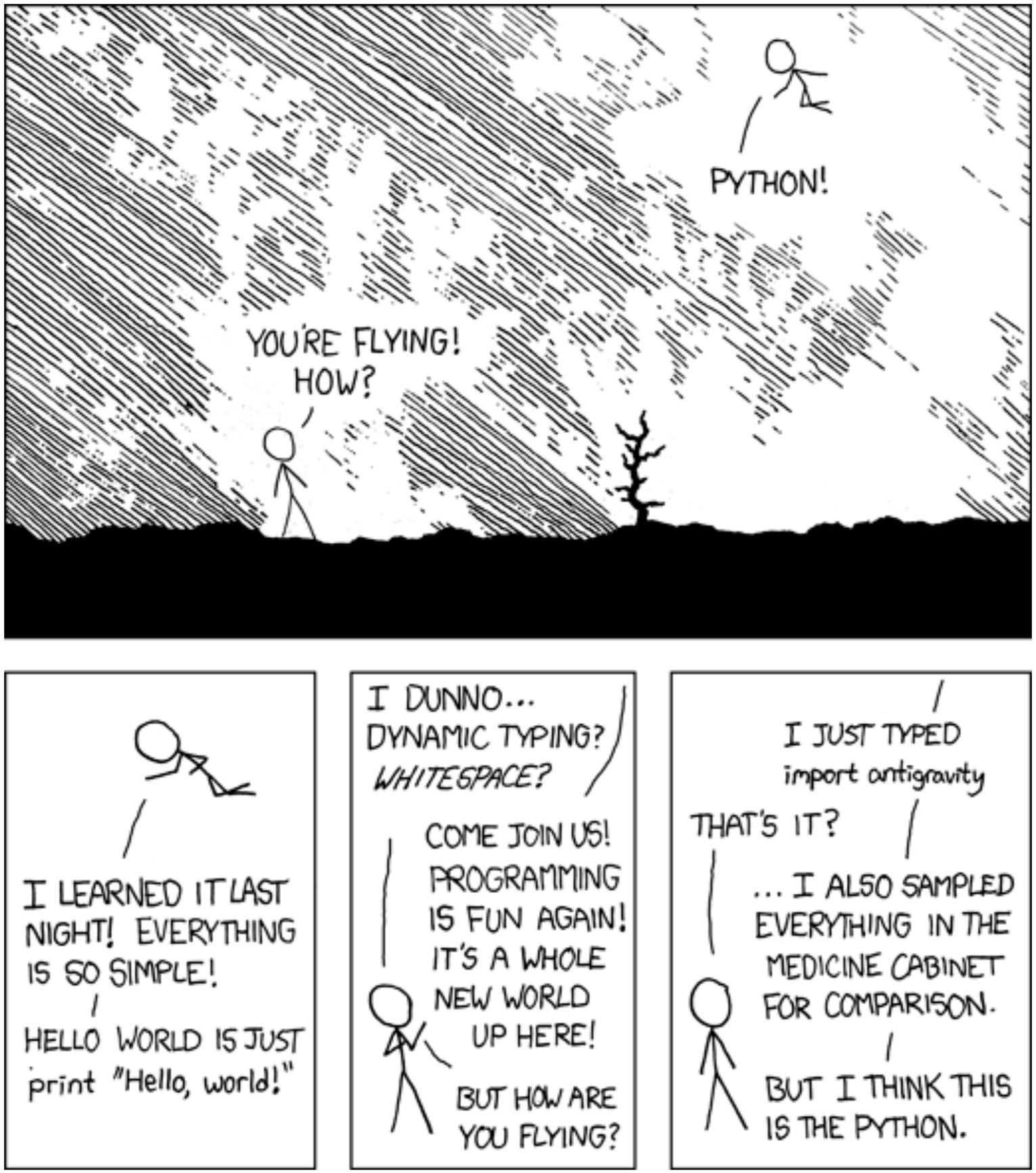
program bugs

fix it

meeting

HOPE FOR

Automatic


PYTHON!
YOU'RE FLYING! HOW?
I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST print "Hello, world!"
I DUNNO... DYNAMIC TYPING? WHITESPACE?
COME JOIN US! PROGRAMMING IS FUN AGAIN! IT'S A WHOLE NEW WORLD UP HERE!
BUT HOW ARE YOU FLYING?
I JUST TYPED import antigravity
THAT'S IT?
... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.
BUT I THINK THIS IS THE PYTHON.

try

before try

No God

Not Perfect

More light

Continuable

introduce Generator

iteration

for

[list] & (tuple)

```
>>>for i in [1,2,3,4]
...     print i
1
2
3
4
>>>for i in (1,2,3,4):
...
```

{dict}

```
>>>d = {'a':1,'b':2}
>>>for i in d:
...     print i,d[i]
```

```
a 1
b 2
```

“string”

```
>>>for i in “python”:
...     print i
p
y
t
h
o
n
```

also file

file

```
>>>for i in open('example.txt'):
...     print i
1st line
2nd line
```

iterable

.__iter__(self)

.next(self)

```
class countdown(object):
    def __init__(self,start):
        self.count = start
    def __iter__(self):
        return self
    def next(self):
        if self.count <= 0:
            raise StopIteration
        r = self.count
        self.count -= 1
        return self
```

```
>>>a = countdown(5)
>>>for i in a:
...     print i
...
5 4 3 2 1
>>>
```

yield

```
def d1(t):
    newt=[]
    for i in t:
        newt.append(i+1)
    return newt
```

```
def d2(t):
    for i in t:
        yield i+1
```

yield

```
def countdown(n):
    while n > 0:
        yield n
        n -= 1

>>>a = countdown(5)
>>>for i in a: print i
5 4 3 2 1
```

this is

a

Generator

Generator Expressions

generator & yield

```
(expr for i in s if condition)

#same as

for i in s:
    if condition:
        yield expr
```

1st Problem

analyze access log

```
127.0.0.1 - - [31/Mar/2009:21:53:25 +0800]
"GET /apache_pb.gif HTTP/1.1" 200 2326

host referrer user [datetime] "request"
status bytes

re.compile(r'(\S+) (\S+) (\S+) \[(.*?)\]
"(\S+) (\S+) (\S+)" (\S+) (\S+)')

match().groups()

line = ('127.0.0.1','-','-','31/Mar/
2009:21:53:25 +0800','GET','/
apache_pb.gif','HTTP/1.1','200','2326')
```

```
cols=('host','referrer','user','datetime',
'method','request','proto','status','bytes
')

dict(zip(cols,line))

{'bytes': '2326',
 'datetime': '31/Mar/2009:21:53:25 +0800',
'host': '127.0.0.1',
'method': '"GET',
'proto': 'HTTP/1.1',
'referrer': '-',
'request': '/apache_pb.gif',
'status': '200',
'user': '-'}
```

RealOne

```
lines=(dict(zip(cols,logpats.match(i).grou
ps())) for i in open('./access_log'))

for i in lines: print i

{'status': '200',
'proto': 'HTTP/1.1',
'referrer': '-',
'request': '/',
'datetime': '31/Mar/2009:21:53:25 +0800',
'bytes': '1456',
'host': '::1',
'user': '-',
'method': 'GET'
}
```

```
def fmap(dictseq,col,func):
    for d in dictseq:
        d[col] = func(d[col])
        yield d

log = fmap(lines,'status',int)
log = fmap(log,'bytes',lambda s:int(s) if
s != '-' else 0)
```

```
{'status': 200,
...
'bytes': 1456,
...
}
```

Show Time

```
>>>print sum(r['bytes'] for r in log)
2456

>>>print "%d %s" %
max(r['bytes'],r['request'] for r in log)
233 /aaa.jpg

>>>print set(r['host'] for r in log)

>>>print set(r['request'] for r in log if
r['status'] == 404)
```

2nd Problem

find

specific files

os.walk

Is a Generator

```
for p,dl,fl in os.walk(t)

#t: a top directory

#p: current path
#dl: directory list in current
#fl: file list in current
```

pyfind

```
import os
import fnmatch

def pyfind(top,pats):
    for p,dl,fl in os.walk(top):
        for f in fnmatch.filter(fl,pats):
            yield os.path.join(p,f)
```

pyfind big file

```
if os.path.getsize(ffp) > s*1024*1024:
        yield ffp
```

os.path

get stat

```
os.path.getctime
os.path.getmtime
os.path.getatime
os.path.getsize

os.stat
```

do a test

```
os.path.exists
os.path.isabs
os.path.isdir
os.path.isfile
os.path.islink
os.path.ismount
```

deal with path

```
os.path.abspath
os.path.join
os.path.basename
os.path.dirname
os.path.commonprefix
os.path.split
os.path.splitext
```

next

process

thread

queue

Reference

- Generator Tricks for Systems Programmers
  http://www.dabeaz.com/generators-uk/index.html
- Python Module of the Week
  http://www.doughellmann.com/projects/PyMOTW/
- Python documentation
  http://docs.python.org/
- Book: Python for Unix and Linux System Administration
  http://www.amazon.com/Python-Unix-Linux-System-Administration/dp/0596515820

Q&A

Thanks