```
 4  5  20 5400 e37c   0 0000  39  01 9864 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 14
Request timeout for icmp_seq 15
Request timeout for icmp_seq 16
Request timeout for icmp_seq 17
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 6dba   0 0000  39  01 0e27 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 18
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 0b2c   0 0000  39  01 70b5 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 19
Request timeout for icmp_seq 20
Request timeout for icmp_seq 21
Request timeout for icmp_seq 22
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 6e95   0 0000  39  01 0d4c 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 23
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 601a   0 0000  39  01 1bc7 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 24
Request timeout for icmp_seq 25
Request timeout for icmp_seq 26
Request timeout for icmp_seq 27
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 4a1f   0 0000  39  01 31c2 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 28
Request timeout for icmp_seq 29
Request timeout for icmp_seq 30
Request timeout for icmp_seq 31
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 0275   0 0000  39  01 796c 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 32
Request timeout for icmp_seq 33
Request timeout for icmp_seq 34
Request timeout for icmp_seq 35
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 91b2   0 0000  39  01 ea2e 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 36
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 a685   0 0000  39  01 d55b 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 37
Request timeout for icmp_seq 38
Request timeout for icmp_seq 39
Request timeout for icmp_seq 40
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 3d8c   0 0000  39  01 3e55 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 41
36 bytes from te-7-1-acr03.statecollege.pa.pitt.comcast.net (162.151.68.250): Destination Host Unreachable
Vr HL TOS  Len   ID Flg  off TTL Pro  cks      Src      Dst
 4  5  20 5400 9f36   0 0000  39  01 dcaa 104.38.16.163  174.60.222.162

Request timeout for icmp_seq 42
Request timeout for icmp_seq 43
Request timeout for icmp_seq 44
```

~ — tmux

```
Restored session: Mon Jan 29 08:45:55 EST 2018
Flashpoint:~ eflubacher$ cd cmpsci122/Labs/
.DS_Store  lab1/      lab2/      lab3/
Flashpoint:~ eflubacher$ cd cmpsci122/Labs/lab3/
Flashpoint:lab3 eflubacher$ ;s
-bash: syntax error near unexpected token `;'
Flashpoint:lab3 eflubacher$ ls
LAB3.py         Lab3.pdf
Flashpoint:lab3 eflubacher$ vim LAB3.py


Press ENTER or type command to continue
Flashpoint:lab3 eflubacher$
```

~ — tmux

```
    common = intersect(class_sig, dict_keys)
    if 'typemap' in class_sig:
        class_sig.remove('typemap')
    if tuple(class_sig) != tuple(dict_keys):
        print "Conformability error"
#       print "Class signature: " + `class_sig`
#       print "Dictionary keys: " + `dict_keys`
        print "Not matched in class signature: " + `setdiff(class_sig, common)`
        print "Not matched in dictionary keys: " + `setdiff(dict_keys, common)`
        sys.exit(1)
    else:
        for x in dict_keys:
            setattr(toclass, x, fromdict[x])
Listing 3

# The tricky part -- initializing objects from the configuration global
# `Configuration' is the top level of the object tree we're going
# to mung

Configuration = Controls()
copy_instance(Configuration, configuration)
Configuration.servers = [];
for server in configuration[`servers']:
    Newsite = Server()
    copy_instance(Newsite, server)
    Configuration.servers.append(Newsite)
    Newsite.users = [];
    for user in server['users']:
        Newuser = User()
        copy_instance(Newuser, user)
        Newsite.users.append(Newuser)
Listing 4

That doesn't look too bad for deep black magic, does it? Thirty-two lines, counting comments. Just from knowing what I
've said about the class structure, the calling code is even readable. But the size of this code isn't the real shocke
r. Brace yourself: this code only took me about ninety minutes to write -- and it worked correctly the first time I ra
n it.

To say I was astonished would have been positively wallowing in understatement. It's remarkable enough when implementa
tions of simple techniques work exactly as expected the first time; but my first metaclass hack in a new language, six
 days from a cold standing start? Even if we stipulate that I am a fairly talented hacker, this is an amazing testamen
t to Python's clarity and elegance of design.

There was simply no way I could have pulled off a coup like this in Perl, even with my vastly greater experience level
 in that language. It was at this point I realized I was probably leaving Perl behind.

Conclusion
This was my most dramatic Python moment. But, when all is said and done, it was just a clever hack. The long-term usef
ulness of a language comes not in its ability to support clever hacks, but from how well and how unobtrusively it supp
orts the day-to-day work of programming. The day-to-day work of programming consists not of writing new programs, but
mostly reading and modifying existing ones.

So the real punchline of the story is this: weeks and months after writing fetchmailconf, I could still read the fetch
mailconf code and grok what it was doing without serious mental effort. And the true reason I no longer write Perl for
 anything but tiny projects is that was never true when I was writing large masses of Perl code. I fear the prospect o
f ever having to modify keeper or anthologize again -- but fetchmailconf gives me no qualms at all.

Perl still has its uses. For tiny projects (100 lines or fewer) that involve a lot of text pattern matching, I am stil
l more likely to tinker up a Perl-regexp-based solution than to reach for Python. For good recent examples of such thi
ngs, see the timeseries and growthplot scripts in the fetchmail distribution. Actually, these are much like the things
 Perl did in its original role as a sort of combination awk/sed/grep/sh, before it had functions and direct access to
the operating system API. For anything larger or more complex, I have come to prefer the subtle virtues of Python -- a
nd I think you will, too.

All listings referred to in this article are available by anonymous download in the file ftp://ftp.linuxjournal.com/pu
b/lj/listings/issue73/3882.tgz

About the Author
Eric Raymond is a Linux advocate and the author of The Cathedral & The Bazaar. He can be reached via e-mail at esr@thy
rsus.com.
(END)
```

[3] 0:less*  "Flashpoint.local" 21:51 29-Jan-18

~ — tmux

```
Restored session: Mon Jan 29 08:45:55 EST 2018
Flashpoint:~ eflubacher$ cd cmpsci122/Labs/
.DS_Store  lab1/      lab2/      lab3/
Flashpoint:~ eflubacher$ cd cmpsci122/Labs/lab3/
Flashpoint:lab3 eflubacher$ ;s
-bash: syntax error near unexpected token `;'
Flashpoint:lab3 eflubacher$ ls
LAB3.py         Lab3.pdf
Flashpoint:lab3 eflubacher$ vim LAB3.py


Press ENTER or type command to continue
Flashpoint:lab3 eflubacher$
```

~ — tmux

```
Flashpoint:ulab2 eflubacher$ vim myline.txt
Flashpoint:ulab2 eflubacher$ cat article.txt myline.txt > article2.txt
Flashpoint:ulab2 eflubacher$ less article2.txt
Flashpoint:ulab2 eflubacher$ tail -f article2.txt
Perl still has its uses. For tiny projects (100 lines or fewer) that involve a lot of text pattern matching, I am stil
l more likely to tinker up a Perl-regexp-based solution than to reach for Python. For good recent examples of such thi
ngs, see the timeseries and growthplot scripts in the fetchmail distribution. Actually, these are much like the things
 Perl did in its original role as a sort of combination awk/sed/grep/sh, before it had functions and direct access to
the operating system API. For anything larger or more complex, I have come to prefer the subtle virtues of Python -- a
nd I think you will, too.

All listings referred to in this article are available by anonymous download in the file ftp://ftp.linuxjournal.com/pu
b/lj/listings/issue73/3882.tgz

About the Author
Eric Raymond is a Linux advocate and the author of The Cathedral & The Bazaar. He can be reached via e-mail at esr@thy
rsus.com.



This is my line
```

[3] 0:tail*  "Flashpoint.local" 21:54 29-Jan-18