



Will iPython replace Bash? s/bash/ipython/g

EuroPython 2013, 3th July - Firenze



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



What? Who? Why?

How ipython can speed up your daily bash work, improve reusability and your python confidence...without entirely replacing bash.

Roberto Polli - Community Manager @ Babel.it. Loves writing in C, Java and Python. Red Hat Certified Engineer and Virtualization Administrator.

Babel - Proud sponsor of this talk ;) Delivers large mail infrastructures based on Open Source software for Italian ISP and PA. Contributes to various FLOSS.



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Agenda - 1

Bash Rocks

Simple bash patterns

Python vs Bash: strenght and limits

Can python save the day?

SysAdmins can gently learn python!



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Agenda - 2

Real life scenarios

- using fields
- parsing /proc/ files
- reusing stuff
- flaskize
- test

You can manage it faster ;)



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Python vs Bash

- Bash is "The" shell
- Python is the widespread system language
- A reunion?

```
# python
for f in files:
    name, ext = f.splitext()
    shutil.move(f,name.lower()+ "."+ext)
```

```
# bash
for f in ${files[@]}; do
    name=${f%.*}; ext=${f##*.}
    mv "$f" "${name,,}.${ext}"
done
```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Bash rocks - 1

Job Control

```
for x in $(seq 1 10); do
    sleep $x &
done
jobs -l;
wait %1
```

Simple pipelines and streams

```
# zipinfo -t just_files.zip | xargs rm --
```

Clean I/O redirection

```
# kinit user <<< 'secret' # kerberos login
# script.py 2>py.err < /etc/hosts # redirect stdin
```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Bash rocks - 2

Braces and Globbing

```
# du -ms .[a-z]*  
# touch {01..20}.out
```

...bash3 behaves differently from bash2

History and readline substitution

```
# mv foo.out foo-$(date -I).out  
# !!:gs/foo/bar # same as above with "bar"
```

Long commands

```
# fc;
```



Wishes - 1

Safe system scripts

```
# touch -- -rf; rm * #ouch!
```

Readability

```
# mv "$file" "${file,,}"
```

```
# [ x$1 != x ] || exit
```

```
# ${WTF[@]}
```

Reusability

```
# . /etc/init.d/functions
```

```
# are you sure
```

```
# bash scripts
```

```
# are short?
```

```
RM=$(which rm)
```

```
? ${RM:?Missing rm}
```

```
WTF=(world taekwondo fed)
```

```
for file in ${A[@]}; do
```

```
    $RM -- "$file";
```

```
done
```




Wishes - 2

Code consistency

- `bash3 != bash4`

One language

- `perl`
- `awk`, `gawk`, `mawk`
- `grep`, `egrep`, `fgrep`
- `sed`, `gsed`,
- `gnuplot`, `bc`

```
#bash4
if [[ $file =~ ^p ]]
then
    strace $file |& \
        grep inputrc
fi
# bash3
if echo "$file" | \
    egrep -q "^p"
then
    strace 2>&1 $file | \
        grep inputrc
fi
```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Interactive Python

- CPython delivers an interactive interpreter
- Limited Readline and History support
- Requires parenthesis and string quoting
- Batteries included (math, telnetlib, shutil, ...)

```
$ telnet localhost 80
telnet: command not found

$ python -m telnetlib 0 80
```

**Present by default
in almost
all Linux distros!**



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Interactive Python

Cpython shell can autocomplete!

```
import readline as rl
import rlcompleter
rl.parse_and_bind('tab: complete')
```

...but you can use iPython!



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



iPython: interactivity += 1

Easy to install

- pip install ipython

Various flavours!

- vanilla# ipython
- math# ipython --pylab
- shell# ipython --profile=[py]sh

Customizable

~/ipython/profile_XX/ipython_config.py

Customize:
prompt
spacing
functions
shortcuts
modules
...



iPython

/bin/sh passthru:

- `ls -ld ~/python; # run via sh`
- `ll /tmp; # alias pointing to ls -l`

Capture both stdout and stderr

- `ret = !cat /etc/hosts #run via sh`
- `ret = !ls /MISSING`

Get exit status

- `print(_exit_code, "resembles $?")`

ipython uses
`os.system`

```
# output is  
# in a flexible  
# list (SList)  
type(ret)
```

`_exit_code` may
not be exactly
as expected



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



iPython features

Use SList as:

- list or string
- w or w/o newlines

Various Magics

%autocall

%edit

%debug

**Automagic string-ify
and parenthesis**

```
#ret = !cat /etc/resolv.conf  
[ "search babel.it",  
  "nameserver 127.0.0.1" ]
```

```
#ret.s == ' '.join(ret)  
True
```

```
#ret.nlstr == '\n'.join(ret)  
True
```

```
#, ret.grep babel.it
```

THINK OUTSIDE THE IT BOX



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



iPython expansion

Pass python stuff to the shell

```
# import os  
# ret = !cat /proc/{os.getpid()}/status
```

Consider the following steps during expansion

1. ipython parses the line and expands {contents}
2. prepare the command to be executed to the shell
3. fork/exec the shell
4. the shell interprets the command!

```
# from os import getpid as PID  
# ret = !cat /proc/{PID()}/* # guess what?
```



Parsing system files

Get ipython Memory usage

```
# ret = !cat /proc/{os.getpid()}/status
```

```
VmPeak:      27308 kB  
VmSize:      27244 kB  
VmHWM:       7080 kB  
VmRSS:       7072 kB
```

Replace AWK

```
# ret.fields(0)           # awk '{print $1;}'  
# ret.fields(2,1,3)      # awk '{print $2, $1, $3;}'  
# ret.fields(*range(1,10)) # print 2..9th fields  
# ret.fields(-1)        # awk '{print $NF;}'
```




B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Unpack data

**SList.fields() just splits
by white-space**

**fieldize() is more
generic!**

**iPython profiles loads
custom files under
./startup/ directory.**

```
# 99-fieldize.py
# save under the ./startup/
# of the rofile directory:
# ~/.ipython/profile_xx/

def fieldize(ret, sep=":"):
    """Let's be generic"""
    return dict([
        map(str.strip,
            x.split(sep,1))
        for x in ret ])

```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



monitoring.ipynb

```
#!/usr/bin/ipython --profile=pysh
# Important: save as .ipy to run with ipython
from time import sleep
from os import getpid

fields = '{VmSize},{VmRSS},{VmSwap}'
while sleep(1) == None:
    ret = !grep ^Vm /proc/{getpid()}/status
    d = fieldize(ret)
    print(fields.format(**d))
```



\$PITFALLS - 1

Beware of bash expansion/substitution

```
ret = !echo /proc/$$/cmdline
```

```
ls /proc/$$/fn
```

```
ls /proc/$$/cwd
```

os.system

```
fork  
exec /bin/sh
```

```
ls /proc/98/cwd
```

1. `!python` replaces `fn` with its python value - eg. "cwd"
2. then uses `os.system`
3. `fork()` happens **before** bash expansion
4. the shell interprets `$$` as the pid of the **current** process (the forked one!)



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



\$PITFALLS - 1

Expand Early and in Python

unless you know what you're doing!

```
GOOD: ! echo /proc/{os.getpid()}/cmdline
```

```
EVIL: ! echo /proc/$$/cmdline
```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



\$PITFALLS - 2

os.system|subprocess use /bin/shell

```
# this = !{echo does; echo >&2 not; } |& grep work
```

We can work it out ;)

```
edit _process_common.py  
add the `executable` argument to  
subprocess.Popen (  
    executable= '/bin/bash',  
    shell=True, ...)
```



IMHO: Don't you trust `os.environ.get('SHELL')`? Don't trust `os.system` too!



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Plotting system data

A sysadmin must plot...

```
# from matplotlib import pyplot
```

Collect data...

```
#ret = !ping -t 100 -w 100 foo.com
```

```
#ret = ret.fields(7).grep('time')
```

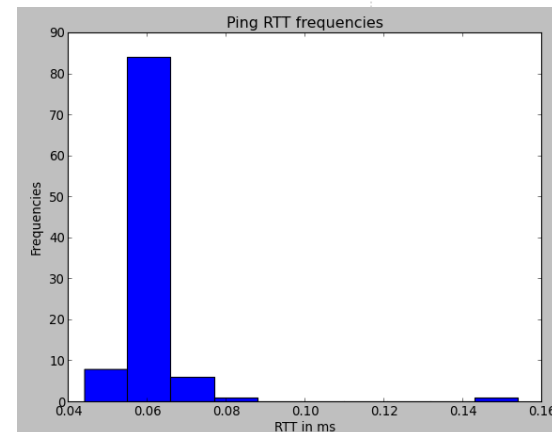
Mangle it... (get just millisecs)

```
#ret = [ x.split("=")[1]  
        for x in ret ]
```

And show!

```
# pyplot.hist(map(float, ret))
```

```
# pyplot.show()
```



Roberto Polli - roberto.polli@babel.it



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Flaskize

Scripting in python you'll collect a lot of useful snippets in a very short time.

```
%history #use history magic
```

Flask is a web microframework you can use to convert them in webservices!

Can't use the !command syntax with Flask...(can I?)

```
"""Things are easy even without
the !command stuff
"""
from flask import Flask
import simplejson
from fieldize import fieldize

app = Flask(__name__)

@app.route('/r/mem.view/<pid>')
def monitor(pid):
    """Return some process info"""
    int(pid) # trigger ValueError
    fpath = '/proc/%s/status' % pid
    # a file is iterable ;)
    with open(fpath) as ret
        d = fieldize(fpath, ":")
    return simplejson.dumps(d)

app.run(debug=True)
```



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Nosetests: speed up your tests!

Three simple steps

- put your test files in `./test/`
- run `#nosetests ./test/`
- nose discovers and runs them all

...really! That's all!

```
"""Test is easy with nose.
   Docstrings are used in reports!
"""
from telnetlib import Telnet

def mping(ip):
    # ICMP requires root privileges ;)
    cmd = '/bin/ping -w 2 -t 2 '+ip
    return os.system(cmd)

def test_ping_gw():
    "ping default gw"
    assert mping('192.168.1.254')==0

def test_ping_remote():
    "ping remote host"
    assert mping('10.0.11.1')==0

def test_tcp_remote():
    "port 80 remote"
    port, timeout = 80, 1
    Telnet('10.0.11.1', port, timeout).close()
```




B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Bonus tracks: Ovirt and 389ds

iPython auto-complete helps learning new API

- Ovirt open-source virtualization infrastructure
- 389 LDAP directory server

```
#!/sudo pip install ovirt-engine-sdk
from ovirtsdk.api import API

client = API(**{'url':..., 'username':...})
for vm in client.vm.list():
    name = vm.get_name()
    try:    vm.stop() # PEP8 who? ;)
    except: log.exception("Error stopping " + name)
```

Roberto Polli - roberto.polli@babel.it



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Bonus tracks: Ovirt and 389ds

389 API are under development: feedback is important!

```
#git clone https://github.com/ioggstream/dsadmin
```

```
from dsadmin import DSAdmin
# connect to the server
ds = DSAdmin(**{'binddn':..., 'bindpw': ...})
# configure 389 to use SSL
ds.setupSSL()
# weird name? iPython will auto-complete!
ds.getReplStatus()
```

Development releases includes method name refactoring!

```
ds.replica.status()
ds.replica.agreements()
```

Roberto Polli - roberto.polli@babel.it



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Will iPython replace Bash?

Not in the near future ;)

Can lead to a different use of bash

**Compete with weakly interactive
toos like perl, awk, sed**

It is a tool that you shouldn't miss!



B · A · B · E · L

EUROPYTHON
FLORENCE, JULY 1-7 | 1-5 TALKS
6-7 SPRINTS



Code

```
http://ipython.org/  
http://flask.pocoo.org/  
http://nose.readthedocs.org/  
https://github.com/ioggstream/dsadmin/
```

Babel

```
http://www.babel.it  
http://vaunaspada.babel.it/blog
```

Thank You!

```
roberto.polli@babel.it
```

Roberto Polli - roberto.polli@babel.it