What Python can learn from Erlang?

benoît chesneau
@benoitc

• Mostly a functional language
• Concurrent & reliable programs
• the OTP framework
• Performance is a result
• http://learnyousomeerlang.com
OOP to me means only messaging, local retention and protection and hiding of state-process, and extreme late-binding of all things. It can be done in Smalltalk and in LISP. There are possibly other systems in which this is possible, but I'm not aware of them.

Alan Kay

http://userpage.fu-berlin.de/~ram/pub/pub_jf47ht81Ht/doc_kay_oop_en
It’s about reliability
What is a reliable program?

- Resistant to failures
- Recover easily
- Hot-Upgrade
Problem?
Stay alive
Reliability in Erlang

- Process isolation
- Pattern matching ➔ assertion
- Message passing
- No Shared memory
Process isolation
Each process is isolated
Failure

The system continues to operate
Recover

Detect the failures and relaunch
Process isolation is hard in Python

- Everything run in 1 thread
- GIL
- Use OS processes (multiprocessing, gunicorn, ...)
- PyParallel?
- PyPy STM?
Program to Systems

- Supervisor
- Load-balancer & Proxy
- Containers
Immutability
Immutability

- real world is mutable
- but at some time, each mutation is immutable

A snapshot can be easily stored
Easier to think about it

• For one input we will always get the same output
• Easy to test
• Easy to share even between different processes
• Thread-Safe, no locking
immutable data structures in python

- Funktown: https://github.com/zhemao/funktown
- Pysistence https://pythonhosted.org/pysistence
- fn.py https://github.com/kachayev/fn.py
- Need more
- it’s about discipline
class Action(object):
    ...

class SendMessage(Action):
    ...

class Log(Action):
    ...

def send_message(to, msg):
    if is_exists(to):
        SendMessage(to, msg)
    else:
        Log("error %s not found" % to)

run():
    for action in actions:
        action.do()}
A problem has been detected and Windows has been terminated.

If this is the first time you've seen this stop error message, restart your computer. If this screen appears again, do the following:

Check to make sure any new hardware or software you added before the problem started is correctly installed.

If problems continue, disable or remove any new hardware or software you added before the problem started, such as Drivers, BIOS updates, or custom applications.

If you need to use Safe Mode to remove or disable your computer, press F8 when you see the Windows logo, select Advanced Start, and select Safe Mode.

Technical information:

*** STOP: 0x000000f4 (0x000000000000003,0xfffff

Collecting data for crash dump ...

Initializing disk for crash dump ...
Let it crash

- Fail fast
- Crash early
- Crash in a way it can recover fast
try...except but not so much

- Try to fail fast, do not catch everything
- reraise once logged
- Expect content: pattern matching
Pattern matching in Erlang

\{ok, Socket\} = connect() \{error, Error\} will crash

```erlang
receive
    \{say, Msg\} -> say(Msg);
    quit -> quit();
    _ ->
        % other value will crash
        throw(\{error, Error\});
end
```

```erlang
case msg() of
    \{say, Msg\} -> say(Msg);
    quit -> quit()
    % other value will crash
end
```
from patterns import patterns, Mismatch

@patterns
def factorial():
    if 0: 1
    if n is int: n * factorial(n-1)
    if []: []
    if [x] + xs: [factorial(x)] + factorial(xs)
    if {'n': n, 'f': f}: f(factorial(n))

assert factorial(0) == 1
assert factorial(5) == 120
assert factorial([3,4,2]) == [6, 24, 2]
assert factorial({'n': [5, 1], 'f': sum}) == 121

factorial('hello')  # raises Mismatch

https://github.com/Suor/patterns