adopt_erlang(python) -> awesome.

Garrett Smith
Chicago Erlang User Group
April 21, 2010

Motivations for Python

- Simplicity of language
- Can do attitude!
  - Deep, mature libraries
  - Integration (SWIG, ctypes, etc.)
  - Range of real world applications
- Rich web development

Motivations for Erlang

- Highly concurrent applications
- Faul tolerant, long running processes
- Simple functional language
- Solid systems integration

Adopting Erlang

- Imperative → Functional
- OO → Message Passing
- Hack and Slash → Immutability
- Django → OTP*
- Core modules 1 → Core modules 2
- Ecosystem 1 → Ecosystem 2

* Let me explain

Imperative → Functional

Python:

```
for x in range(10):
    print x
```

Erlang:

```
lists:map(fun(X) -> io:format("~b~n", [X]) end,
          lists:seq(1, 10))
```

or:

```
print_it([]) -> ok;
print_it([X|Rest]) →
    io:format("~b~n", [X]),
    print_it(Rest).
```

OO → Message Passing

Python:

```
> class Adder:
.     def add(self, lhs, rhs):
.         return lhs + rhs

> adder = Adder()
> adder.add(1, 100)
101
```

Erlang*:

```
> adder() →
.     receive
.         {add, From, Lhs, Rhs} -> From ! {ok, Lhs + Rhs}
.     end,
.     adder().
> Adder = spawn(fun adder/0).
> Adder ! {add, self(), 1, 100}.
101
```

* Grossly simplified, do not attempt

Hack and Slash* → Immutability

Python:

```
do_work(x):
    x = x + 2
    x = do_other_work(x)
    return x * 100
```

Erlang:

```
do_work(X) ->
    X2 = X + 2,
    X3 = do_other_work(X2),
    X3 * 100.
```

* Slight exageration

Django → OTP

- High level abstractions
- Application model
- Programming idioms/patterns
- Supporting libraries and tools
- Totally and utterly different, let's discuss

Core Modules

Python

- __builtin__
- os
- os.path
- sys
- re
- httpd (Python 3)

Erlang

- erlang (mostly)
- erlang, file, port, etc.
- filename
- init
- re
- http

etc! (Erlang has impressive core libs)

Ecosystems*

Python

- Huge and robust
- Webby
- App + Scripts
- Tao of "Simplicity"
- Low wart tolerance

Erlang

- Getting there
- Getting there
- Apps
- Tao of "Correctness"
- Moderate wart tolerance

* Running fast and loose with this one

Payoff

- Highly concurrent and/or long running processes
- Totally different world view for application design – you will be changed for the better, mostly!
- Terrific first functional language