Perl vs Python

Martyn Smith
Questions?
perl is easier to type

and so is pii
Perl has CPAN

Python has ... I’m not really sure.
Modules

- Perl has CPAN
- Python has pip and easy_install
- Python has a much bigger core
- Some of Python’s core is a bit dumb
DateTime

- Perl has many modules, but I tend to use DateTime (it’s not core)
- Python has datetime (it’s core)
- `datetime.now()` vs `datetime.utcnow()`
- Naive datetime objects
Scoping/Classes

Perl

```perl
package Martyn;

use Moose;

has 'wearing' => ( isa => 'Str' )

sub get_dressed {
    my ($self) = @_;
    $self->wearing('pants');
}

sub scope_example {
    my $a = 'pants'; # local var
    if ( 1 ) {
        my $a = 'other pants'; # "more" local var
    }
    say $a; # "pants"
}
```

Python

```python
class Martyn(object):
    wearing = None

    def get_dressed(self):
        self.wearing = 'pants'

    def scope_example(self):
        a = 'pants'
        if True:
            a = 'other pants'
        print a # "other pants"
```
## Map/Grep and data

### Perl

```perl
sub munge {
    my ($data, $filter_field, $search) = @_; 
    my @matches = grep {
        $_->{$filter_field} ~~ $search 
    } @{$data}; 
    return map {
        @$_->{first} . ' ' . @$_->{last} 
    } @matches;
}
```

### Python

```python
def munge(data, filter_field, search):
    matches = [
        x for x in data
        if x[filter_field] == search
    ]

    return ["%s %s" % (x['first'], x['last'])
            for x in matches]

    # or

    return ["%(first)s %(last)s" % x
            for x in matches]
```
Exceptions

Perl

use some random module from CPAN that the author of the 3rd party module you’re using didn’t use, so if you want to catch his exceptions you’re screwed!

Python

def attempt_stuff():
    try:
        do_something()
    except DidntWork, e:
        # deal with it
    except IOError:
        # something different
    except:
        # all other exceptions
Anonymous subs

Perl

```perl
my $sub = sub { more_munge(munge($_[0])) };
my $sub = sub {
    my ($in) = @_;

    $in = munge($in);
    $in = more_munge($in);
    return $in;
};
```

Python

```python
sub = lambda x: more_munge(munge(x))

def temp_name(x):
    x = munge(x)
    x = more_munge(x)
    return x

sub = temp_name
```
**Importing**

---

**Perl**

```perl
use MyProject::Model::MyClass;

$a = MyProject::Model::MyClass->new(@args);

if ($a->isa('MyProject::Model::MyClass')) {
    do_stuff()
}
```

**Python**

```python
from myproject.model import MyClass

a = MyClass(args)

if isinstance(a, MyClass):
    do_stuff()
```
Assorted stuff

- Regular expressions
- Static analysis
- .pyc files
- Code auto-reloading
- Python web framework debugging

```
[console ready]
>>> kol.id
ObjectId('504e55e8b43b1c3ab35c1c7f')
>>> kol.profile_image
<GridFSPProxy: None>
>>> kol.profile_image._id

Traceback (most recent call last):

File "<debugger>", line 1, in <module>
    kol.profile_image._id

File "/home/martyn/src/bayer/lib/python2.7/site-packages/mongoengine-0.6.18-py2.7.egg/mongoengine/fields.py", line 893, in __getattr__
    raise AttributeError

AttributeError

>>> 

File "/home/martyn/src/bayer/bayer/model/__init__.py", line 40, in profile_image_url
    raise Exception("No URL for you")

Exception: No URL for you
```
Questions?