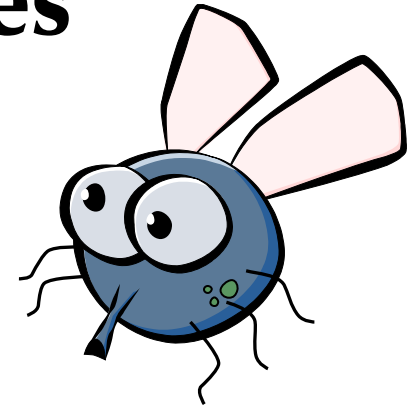


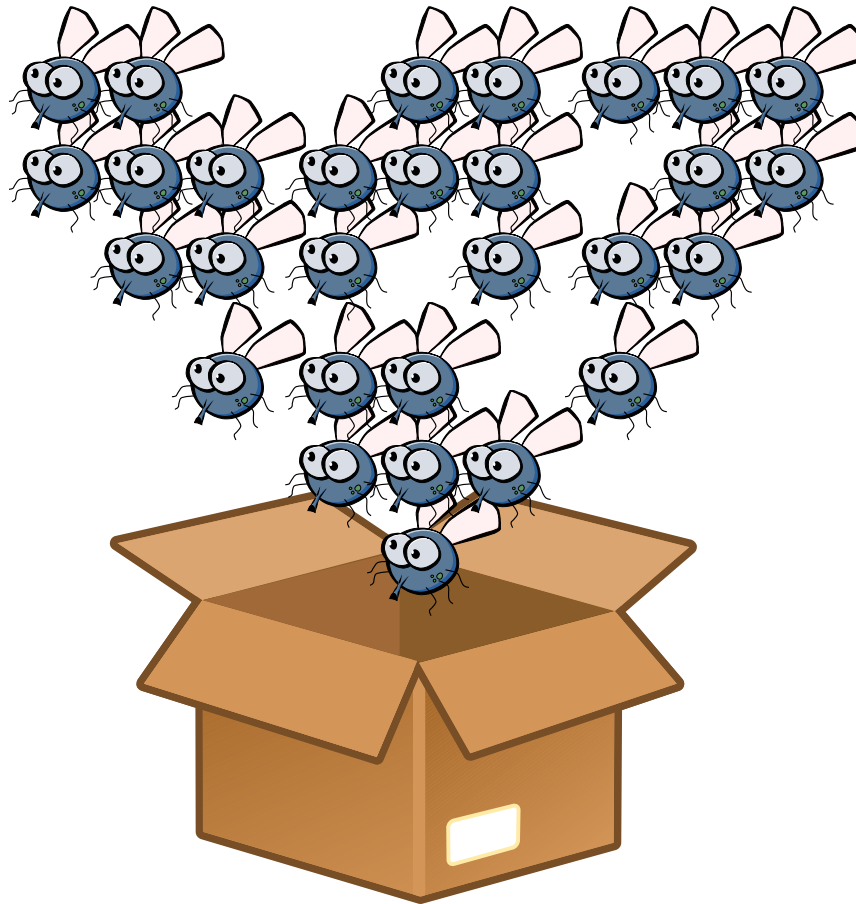
Pythons and Flies



PyCon UK 2014
Coventry, UK

Marc-André Lemburg

Storing lots of objects



Storing lots of objects in memory

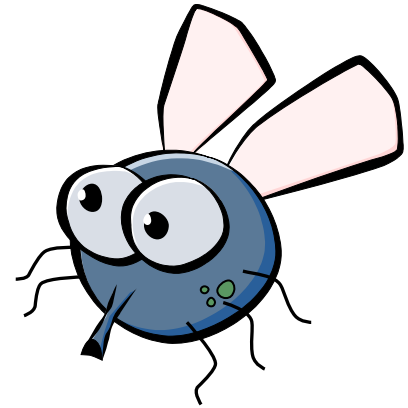
- Class definition:

```
class Fly:  
    def __init__(self, a, b):  
        self.a = a  
        self.b = b
```

- Object creation:

```
obj = Fly()
```

- Now create 1000 of these ...



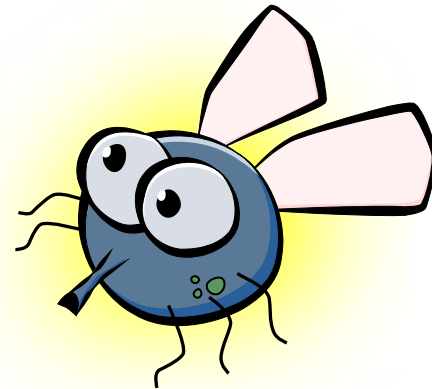
355 usec

353 kB

Storing lots of objects: Slightly more efficient

- Class definition with new style classes:

```
class FlyObject(object):  
    def __init__(self, a, b):  
        self.a = a  
        self.b = b
```



- Object creation:

```
obj = FlyObject()
```

- Now create 1000 of these ...

316 usec
345 kB

Storing lots of objects: More memory efficient

- **Fly weight pattern:**
 - Store only the data
 - Create objects on demand when needed
 - **Goal: Reduce memory load**
- Create a FlyStorage()
 - Works as list of Fly objects
 - Access by index
 - Recreates objects based on index



Storing lots of objects: Fly weights

- Create list of Fly objects in FlyStorage

606 usec
80 kB

- Create list of FlyObject objects in FlyStorage

627 usec
80 kB



80 kB vs. 345 kB: **Reduction by 76%**

Storing lots of objects: Even better ...

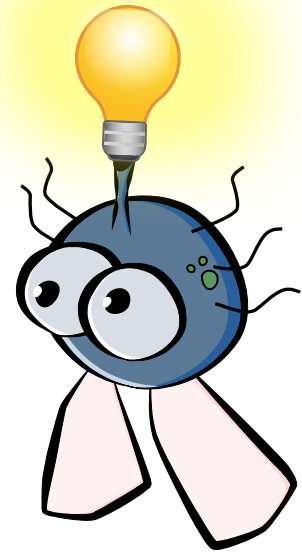
- Class definition with (read-write) slots:

```
class FlySlots(object):  
    __slots__ = ('a', 'b')  
    def __init__(self, a, b):  
        self.a = a  
        self.b = b
```

- Object creation:

```
obj = FlySlots()
```

- Now create 1000 of these ...



243 usec
72 kB

Storing lots of objects: Summary

- Create list of Fly objects in FlyStorage

606 usec
80 kB

- Create list of FlyObject objects in FlyStorage

627 usec
80 kB

- Create list of FlySlots objects



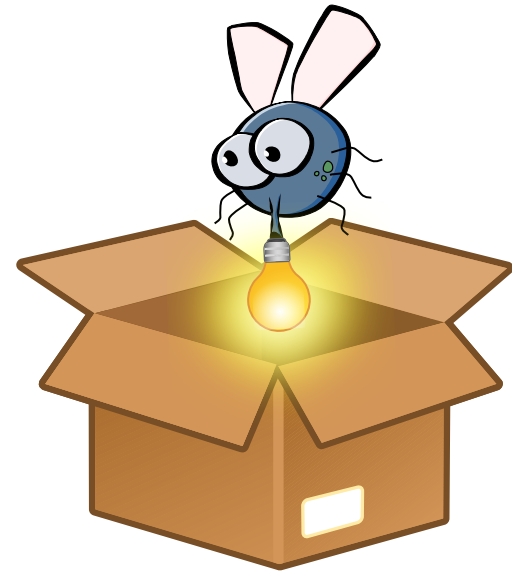
243 usec
72 kB

Resources

- Implementations available on Github:
 - <https://github.com/egenix/when-performance-matters>
 - Look for [test_flyweight.py](#)

No fly was harmed during the making of this lightning talk

... wherever it is now, anyway.



Fly SVG: <https://openclipart.org/detail/145735/cartoon-fly-by-printerkiller>

Performance Talk: Question... and answer

- Typical use case: **attribute introspection**

```
try:
```

```
    x = obj.attribute
```

```
except AttributeError:
```

```
    pass
```

- Exception triggers

629 usec

with 20 levels deep call stack: 631 usec

- Exception does not trigger

43 usec

with 20 levels deep call stack: 45 usec

Thank you for listening



Beautiful is better than ugly.

Contact

eGenix.com Software, Skills and Services GmbH

Marc-André Lemburg

Pastor-Löh-Str. 48

D-40764 Langenfeld

Germany

eMail: mal@egenix.com

Phone: +49 211 9304112

Fax: +49 211 3005250

Web: <http://www.egenix.com/>