First-class Python

“You mean I can access that!?”

Dictionaries

```
super_hero = {
    ‘name’ : ‘Wonder Woman’,
    ‘ability’ : ‘hand-to-hand combat’,
    }
    >>>super_hero[‘name’]
    ‘Wonder Woman’
```

Modules

```
# awesome.py


cool = 42
def amazing():
    return “zomg”
def rad():
    return “totally”
```

Modules & Dictionaries

```
>>>awesome.__dict__
```

```
{‘cool’ : 42, ‘amazing’ : <function amazing>,
‘rad’ : <function rad>}
```

```
>>>awesome.rad
<function rad>
```

Objects

```
class SuperHero(object):
    def __init__(self, name, ability):
        self.name = name
        self.ability = ability
```

Objects & Dictionaries

```
ww = SuperHero(‘Wonder Woman’, ‘hand-to-hand combat’)
```

```
>>>ww.__dict__
{‘name’ : ‘Wonder Woman’, ‘ability’ : ‘hand-
to-hand combat’, __init__: <function>}
```

Instances and Classes

Instance:

```
>>>ww.__dict__
{‘name’ : ‘Wonder Woman’, ‘ability’ : ‘hand-to-hand combat’, __init__: <function>}
```

Class:

```
>>>ww.__class__
<class ‘__main__.SuperHero’>
```

Accessing data and methods...

Accessing the name of our instance (ww):

```
>>>name = ww.name
>>> print name
‘Wonder Woman’
```

Accessing & Modifying

```
>>>ww.name
‘Wonder Woman’
>>>ww.transportation = ‘Invisible Plane’
>>>ww.__dict__
{‘name’ : ‘Wonder Woman’, ‘ability’ : ‘hand-
to-hand combat’, ‘transportation’:
‘Invisible Plane’ }
```

Craziness...

```
>>>del ww.name
>>>ww.__dict__
{‘ability’ : ‘hand-to-hand combat’,
‘transportation’: ‘Invisible Plane’ }
>>>ww.new_name = ‘Wonderous Woman’
```

Oh No I didn’t...

```
>>>ww.__dict__
{‘ability’ : ‘hand-to-hand combat’,
‘transportation’: ‘Invisible Plane’,
‘new_name’ : ‘Wonderous Woman’}
```

Yes. You can totally do that.

There Must Be a Way!

Private variables? Please?

Maybe a little ‘data encapsulation’?

ANYTHING? PLEASE!?

Python Data Encapsulation

‘Protected’ Variable:

```
class SuperHero(object):
    def __init__(self, name, ability):
        self.name = name
        self.ability = ability
        self._vehicle = ‘invisible plane’
```

Protected(hah!) Variables

```
>>>ww = SuperHero(‘Wonder Woman’, ‘hand-to-hand combat’)
>>>ww._vehicle
‘invisible plane’
>>>ww._vehicle = ‘batmobile’
(yeah, that does what you think)
```

What the what...

Ok, really now...PRIVATE Variables

```
class SuperHero(object):
    def __init__(self, name, ability):
        self.name = name
        self.ability = ability
        self.__vehicle = ‘invisible plane’
>>>ww.__vehicle
```

```
AttributeError: ‘SuperHero’ object has no
attribute ‘__vehicle’
```

Nope! Just Kidding!

```
>>>ww._SuperHero__vehicle
‘invisible plane’
```

NO! NO! NO! NO!

yeah...

Python Zen

You’re an adult.
You know what ‘_foo’ means.
You understand and respect ‘__foo’.

We trust you to make the right choice.

(we’re all hippies here!)

You try!

Open Python Interpreter:

$ python

```
Python 2.7.2 (default, Oct 11 2012, 20:14:37)
[GCC 4.2.1 Compatible Apple Clang 4.0 (tags/Apple/clang-418.0.60)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>import math
>>>math.__dict__
```

Something Like This...

```
{'pow': <built-in function pow>, 'fsum': <built-in function fsum>, 'cosh': <built-in function
cosh>, 'ldexp': <built-in function ldexp>, 'hypot': <built-in function hypot>, 'acosh':
<built-in function acosh>, 'tan': <built-in function tan>, 'asin': <built-in function asin>,
'isnan': <built-in function isnan>, 'log': <built-in function log>, 'fabs': <built-in
function fabs>, 'floor': <built-in function floor>, 'atanh': <built-in function atanh>,
'modf': <built-in function modf>, 'sqrt': <built-in function sqrt>, '__package__': None,
'frexp': <built-in function frexp>, 'degrees': <built-in function degrees>, 'lgamma': <built-
in function lgamma>, 'log10': <built-in function log10>,  '__doc__': 'This module is always
available.  It provides access to the\nmathematical functions defined by the C standard.',
'asinh': <built-in function asinh>, 'fmod': <built-in function fmod>, 'atan': <built-in
function atan>, 'factorial': <built-in function factorial>, '__file__':
'/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.7/lib-dynload/math.so',
'copysign': <built-in function copysign>, 'expm1': <built-in function expm1>, 'ceil': <built-
in function ceil>, 'isinf': <built-in function isinf>, 'sinh': <built-in function sinh>,
'__name__': 'math', 'trunc': <built-in function trunc>, 'cos': <built-in function cos>, 'pi':
3.141592653589793, 'e': 2.718281828459045, 'tanh': <built-in function tanh>, 'radians':
<built-in function radians>, 'sin': <built-in function sin>, 'atan2': <built-in function
atan2>, 'erf': <built-in function erf>, 'erfc': <built-in function erfc>, 'exp': <built-in
function exp>, 'acos': <built-in function acos>, 'log1p': <built-in function log1p>, 'gamma':
<built-in function gamma>}
```

Typing!

```
>>>math.__doc__="huzzah!"
>>>math.__dict__
{'pow': <built-in function pow>, 'fsum': <built-in function fsum>, 'cosh': <built-in function cosh>,
'ldexp': <built-in function ldexp>, 'hypot': <built-in function hypot>, 'acosh': <built-in function acosh>,
'tan': <built-in function tan>, 'asin': <built-in function asin>, 'isnan': <built-in function isnan>, 'log':
<built-in function log>, 'fabs': <built-in function fabs>, 'floor': <built-in function floor>, 'atanh':
<built-in function atanh>, 'modf': <built-in function modf>, 'sqrt': <built-in function sqrt>,
'__package__': None, 'frexp': <built-in function frexp>, 'degrees': <built-in function degrees>, 'lgamma':
<built-in function lgamma>, 'log10': <built-in function log10>,'__doc__': 'huzzah!', 'asinh': <built-in
function asinh>, 'fmod': <built-in function fmod>, 'atan': <built-in function atan>, 'factorial': <built-in
function factorial>, '__file__': '/System/Library/Frameworks/Python.framework/Versions/2.7/lib/python2.
7/lib-dynload/math.so', 'copysign': <built-in function copysign>, 'expm1': <built-in function expm1>,
'ceil': <built-in function ceil>, 'isinf': <built-in function isinf>, 'sinh': <built-in function sinh>,
'__name__': 'math', 'trunc': <built-in function trunc>, 'cos': <built-in function cos>, 'pi':
3.141592653589793, 'e': 2.718281828459045, 'tanh': <built-in function tanh>, 'radians': <built-in function
radians>, 'sin': <built-in function sin>, 'atan2': <built-in function atan2>, 'erf': <built-in function
erf>, 'erfc': <built-in function erfc>, 'exp': <built-in function exp>, 'acos': <built-in function acos>,
'log1p': <built-in function log1p>, 'gamma': <built-in function gamma>}
```

Freestyle!

- Find a module
- Import it
- Examine it
- Change it

<heads explode>