PYTHON

Chapter: Pythonic Coding

Pythonic coding -1

- PEP -8 https://www.python.org/dev/peps/pep-0008/
- Written by Rossum

#Naming Style – there is much more to pep-0008

joined_lower for functions, methods, attributes
joined_lower or ALL_CAPS for constants
StudlyCaps for classes
camelCase **only** to conform to pre-existing conventions

Pythonic coding -2

- Compound statements – decreases readablity

GOOD

```
a=10
b=10
c=fun(1)
```

BAD

```
a=10;b=10;c=fun(1)
```

Pythonic coding -3

- Strings
- The algorithm of join does only a single pass through the list to arrive at allcountries
- Bad memory usage with loops as at every step an object is discarded

GOOD

```
>>>countries =[‘IN’,’KA’,’RU’,’BN’,’PK]
>>>allcountries = “ “.join(countries)
```

BAD

```
>>>allcountries = ‘’
>>> for i in countries:
            allcountries +=I
```

Pythonic coding -4

- Testing for truth values

GOOD

```
>>> if a:
            ...do something here
```

BAD

```
>>>if a == ‘True’:
            ... do something here
```

Pythonic coding -5

- Use list comprehension and sum

GOOD

```
>>> total = sum([num * num for num in range(1,
101)])
```

BAD

```
>>> total = 0
>>>for num in range(1, 101):
        total += num * num
```

PYTHON

Pythonic coding -6

- Wild card import

GOOD

reference names through their module (fully qualified identifiers),

import a long module using a shorter name (alias; recommended),

or explicitly import just the names you need.

BAD

```
>>>from module import *
```

Pythonic coding -7

- Many other languages
- int a = 10 means it creates space for int called “a” and then stores value 10 in it

- In Python
- It creates an object with value of 10 and then assigns a name called “a” to it

PYTHON

Pythonic coding -8

- Use dictionaries get

GOOD

```
>>>somedict.get('what','Not Available')
```

BAD

```
>>> somedict['what']
```

Pythonic coding – Many more at

http://python.net/~goodger/projects/pycon/2007/idiomatic/handout.html