Code With Style

Clayton Parker

PyOhio 2010

six feet up

nowhere to go but
open source

sixfeetup.com/deploy2010

Who am I?

- claytron on IRC
- Python dev since 2003

six feet up

sixfeetup.com/deploy2010

What will we learn?

- Zen
- PEP8
- Tools

six feet up

sixfeetup.com/deploy2010

Don’t dissapoint

http://www.flickr.com/photos/docsearls/199700939/

six feet up

sixfeetup.com/deploy2010

The Zen of Python

```
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

Important Zen

- Beautiful is better than ugly.
- Readability counts.
- Now is better than never.
- Although never is often better than right now.

What is PEP8?

- A style guide for Python
- Common sense

six feet up

sixfeetup.com/deploy2010

The importance of PEP8

- Code is read more than it is written
- Consistency

six feet up

sixfeetup.com/deploy2010

When to break the rules

- Less readable
- Be aware of your surroundings

six feet up

sixfeetup.com/deploy2010

Code layout

- 4 space indents
- Never mix tabs and spaces!
- Maximum line length 79 characters
- Use blank lines sparingly

Examples

six feet up

sixfeetup.com/deploy2010

Maximum line length BAD

```
# Example 1
things = ['overwrite', 'photobathic', 'tranquillization', 'resiny', 'runt', 'elpidite', 'Siganus', 'upplough', 'coed']
# This list comprehension is insane and should probably be split up into multiple statements
special_things = [special_thing for special_thing in special_things if special_thing == 'elpidite']

#                                                                 79 columns ->

# Example 2
if event.new_state.id == 'offline' and (state == 'published' or state == 'external'):
    workflow.doActionFor(content, 'reject', workflow='my_custom_workflow', comment='Rejecting content automatically')
```

Maximum line length GOOD

```
# Example 1
things = [
    'overwrite',
    'photobathic',
    'tranquillization',
    'resiny',
    'runt',
    'elpidite',
    'Siganus',
    'upplough',
    'coed']


# Instead of using a list comprehension, we'll use the filter built-in
# to make the code have more clarity.
def my_checker(item):
    if item == "elpidite":
        return item


special_things = filter(my_checker, things)
```

six feet up

sixfeetup.com/deploy2010

Maximum line length
GOOD

```
# Example 2
public_state = state in ['published', 'external']
if event.new_state.id == 'offline' and public_state:
    workflow.doActionFor(
        content,
        'reject',
        workflow='my_custom_workflow',
        comment='Rejecting content automatically')
```

Blank lines

```
import random

ASCII_CAT1 = """\
 /\\_/\\\\
( o.o )
 > ^ <
"""
ASCII_CAT2 = """\
  _ _/|
 \\'o.0'
 =(___)=
    U
"""
CATS = [ASCII_CAT1, ASCII_CAT2]


class CatMadness(object):
    """Cats are curious animals. This is a silly example"""

    def __init__(self, num_cats=0):
        self.num_cats = num_cats

    def make_it_rain(self):
        """Just cats, no dogs yet."""
        count = self.num_cats
        while count > 0:
            count -= 1
            print random.choice(CATS)
```

six feet up

Imports
BAD

```
import os, sys
import config
from my.package.content import *
```

GOOD

```
import os
import sys
# explicit is better than implicit
from my.package import config
from my.package.content import Octopus, Blowfish
```

six feet up

sixfeetup.com/deploy2010

Whitespace BAD

```
counter           =5
another_counter =15
more_cowbell= counter+10
my_dict ={'spam':'eggs','ham':'parrot'}


def complex (real, imag = 0.0):
    return magic(r = real, i = imag)


my_list=[1, 2,3]
another_list = [4,5,6]
combined_list=my_list+another_list
```

Whitespace GOOD

```
counter = 5
another_counter = 15
more_cowbell = counter + 10
my_dict = {'spam': 'eggs', 'ham': 'parrot'}
```

```
def complex(real, imag=0.0):
    return magic(r=real, i=imag)
```

```
my_list = [1, 2, 3]
another_list = [4, 5, 6]
combined_list =  my_list + another_list
```

Guido taking a look

http://www.flickr.com/photos/7901942@N04/3528995869/

Comments

```
# Comments start with a space after the comment symbol. Use complete
# sentences and proper grammar when writing comments. Comments should
# be in English unless you are certain the readers will *not* be
# English speaking.

# Long flowing text should be kept to under 72 characters like above.

x = 5  # Use inline comments sparingly.
```

Naming conventions

```
# my/package/camelcase.py
UPPERCASE_UNDERSCORES = "foo"



class CamelCase(object):

    def separated_with_underscores(self):
        my_variable = "foo"
        pass

    def mixedCaseInPlone(self):
        pass

    def _private_method(self):
        pass
```

Recommendations

BAD

```
if type(obj) is type(1)

if my_variable == None

if not len(my_list)

if boolean_value == True
```

GOOD

```
if isinstance(obj, int):

if my_variable is None

if not my_list

if boolean_value
```

six feet up

sixfeetup.com/deploy2010

pep8.py

- Check your code against PEP8
- Brought back from the dead!

pep8.py

```
$ pep8 example-pep8.py
example-pep8.py:1:10: E401 multiple imports on one line
example-pep8.py:3:1: E302 expected 2 blank lines, found 1
example-pep8.py:4:80: E501 line too long (91 characters)
example-pep8.py:9:16: E225 missing whitespace around operator
example-pep8.py:10:5: E301 expected 1 blank line, found 0
example-pep8.py:10:35: E251 no spaces around keyword / parameter equals
example-pep8.py:11:26: W601 .has_key() is deprecated, use 'in'
```

pep8.py

```
$ pep8 --ignore=W601,E301 example-pep8.py
example-pep8.py:1:10: E401 multiple imports on one line
example-pep8.py:3:1: E302 expected 2 blank lines, found 1
example-pep8.py:4:80: E501 line too long (91 characters)
example-pep8.py:9:16: E225 missing whitespace around operator
example-pep8.py:10:35: E251 no spaces around keyword / parameter equals
```

Pyflakes

- Fast
- Catch common mistakes

six feet up

sixfeetup.com/deploy2010

Pyflakes

```
$ pyflakes example-pyflakes.py
example-pyflakes.py:4: invalid syntax
for directory in os.listdir('.')
                                ^
```

Pylint

- More in depth
- Knows about your code

Pylint

```
$ pylint example-pep8.py
************* Module example-pep8
C:  4: Line too long (91/80)
C:  1: Missing docstring
C:  1: Invalid name "example-pep8" (should match (([a-z_][a-z0-9_]*)|([A-Z][a-zA-Z0-9]+))$)
C:  9:WidgetMaker.__init__: Operator not preceded by a space
        whatzit=None
               ^
W:  9:WidgetMaker.__init__: Unused variable 'whatzit'
W:  8:WidgetMaker.__init__: Unused variable 'blerg'
C: 10:WidgetMaker.blerg_list: Missing docstring
E: 11:WidgetMaker.blerg_list: Instance of 'WidgetMaker' has no 'blerg' member
E: 12:WidgetMaker.blerg_list: 'continue' not properly in loop
E: 13:WidgetMaker.blerg_list: Instance of 'WidgetMaker' has no 'blerg' member
R:  3:WidgetMaker: Too few public methods (1/2)
W:  1: Unused import sys
W:  1: Unused import os

Report
======
10 statements analysed.

Global evaluation
-----------------
Your code has been rated at -15.00/10
```

Editor Integration

six feet up

sixfeetup.com/deploy2010

Vim

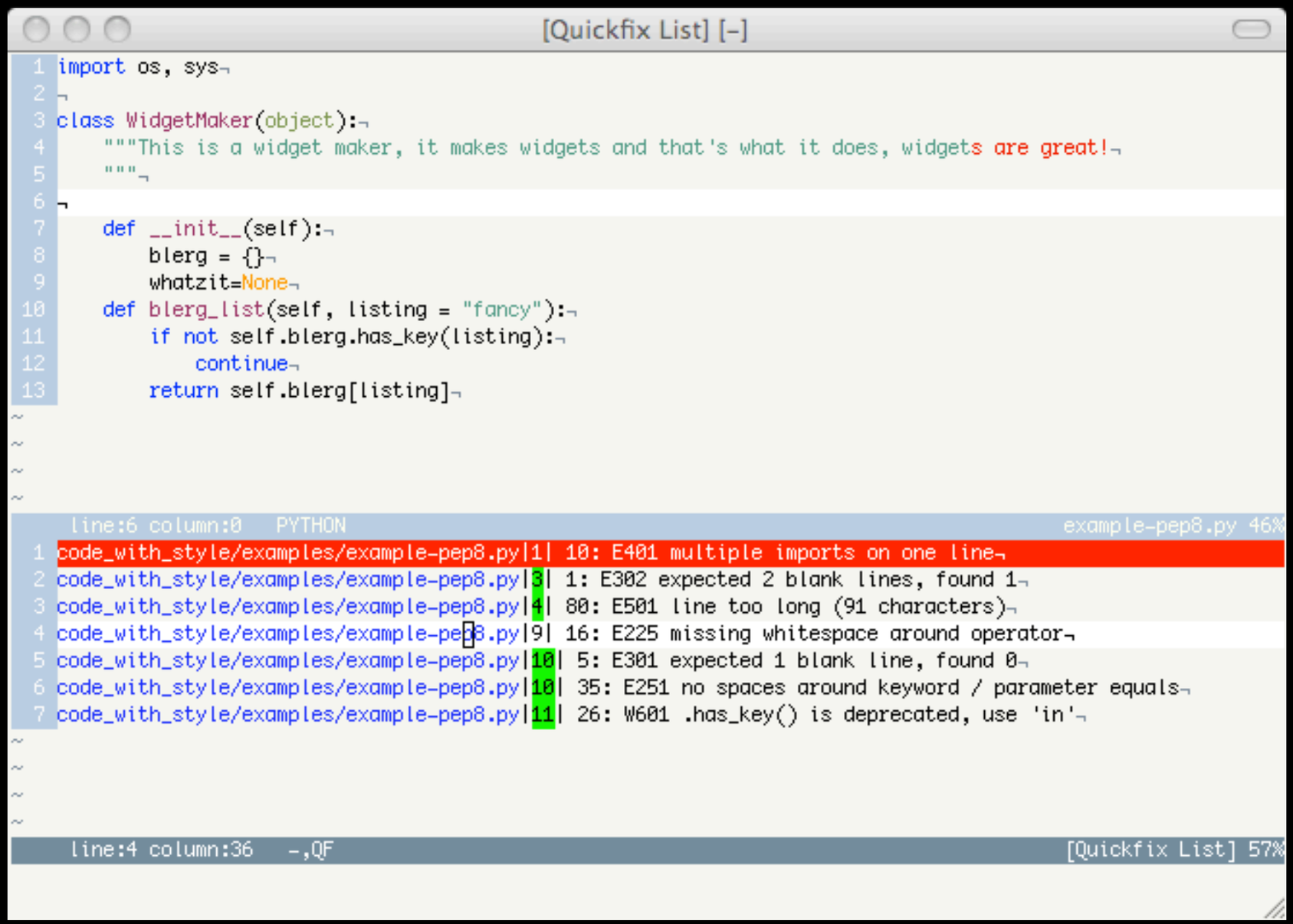
code_with_style/examples/example-pyflakes.py

```
#!/usr/bin/env python
import os

for directory in os.listdir('.')
    print directory
```

```
line:4 column:13    PYTHON                example-pyflakes.py 80%
could not compile: invalid syntax (example-pyflakes.py, line 4)
```

Vim



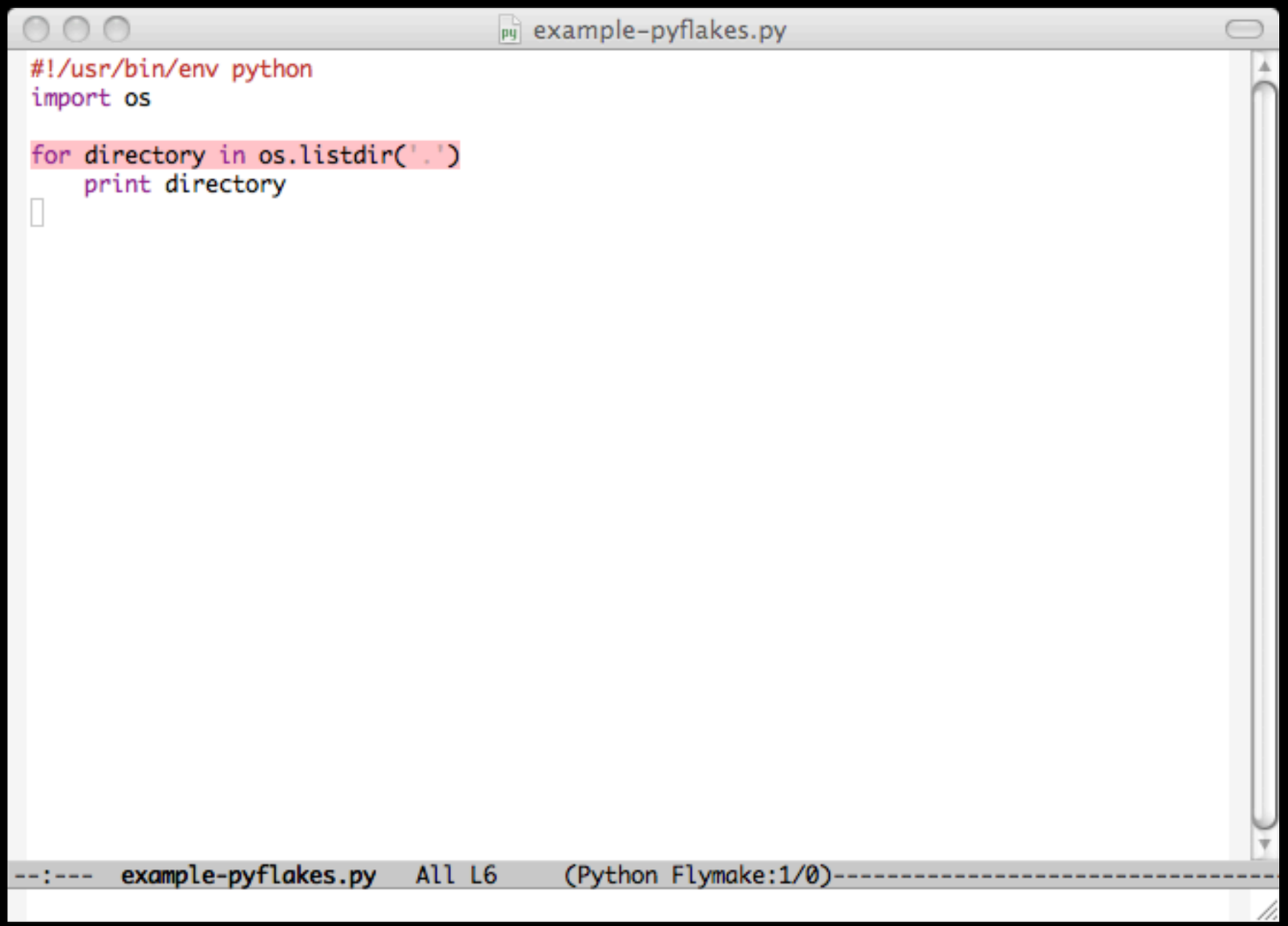
six feet up

sixfeetup.com/deploy2010

Vim

- pyflakes.vim
  - http://www.vim.org/scripts/script.php?script_id=2441
- pep8.vim
  - http://www.vim.org/scripts/script.php?script_id=2914

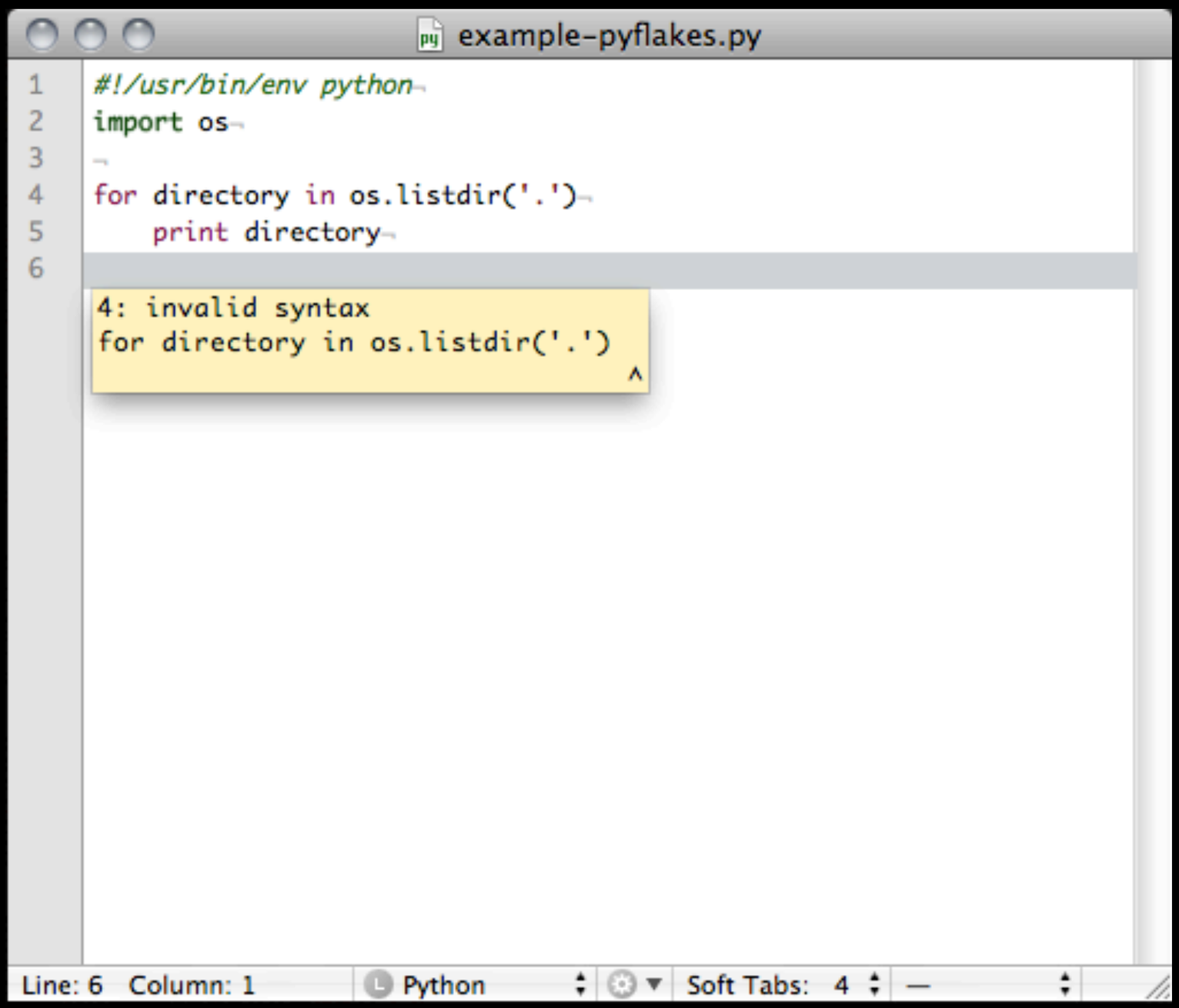
Emacs


example-pyflakes.py
#!/usr/bin/env python
import os
for directory in os.listdir('.')
print directory
--:--- example-pyflakes.py All L6 (Python Flymake:1/0)

Emacs

- Pyflakes
  - http://www.plope.com/Members/chrism/flymake-mode
- I'm not an Emacs dude, better ways to do this?

TextMate



six feet up

sixfeetup.com/deploy2010

TextMate

PEP-8 Python

PEP-8 Python

Theme: default

Python style checker

File : example-pep8.py

view source view PEP

1:9	*E401* : multiple imports on one line
3:0	*E302* : expected 2 blank lines, found 1
4:79	*E501* : line too long (91 characters)
10:4	*E301* : expected 1 blank line, found 0
11:25	*W601* : .has_key() is deprecated, use 'in'

1	*E301* : expected 1 blank line, found 0
1	*E302* : expected 2 blank lines, found 1
1	*E401* : multiple imports on one line
1	*E501* : line too long (91 characters)
1	*W601* : .has_key() is deprecated, use 'in'

example-pep8.py

```
1  import os, sys
2
3  class WidgetMaker(object):
4      """This is a widget maker, it makes widgets and that's what it
   does, widgets are great!
5      """
6
7      def __init__(self):
8          blerg = {}
9          whatzit=None
10     def blerg_list(self, listing = "fancy"):
11         if not self.blerg.has_key(listing):
12             continue
13         return self.blerg[listing]
14
```

Line: 3 Column: 1 Python Soft Tabs: 4 WidgetMa...

six feet up

sixfeetup.com/deploy2010

TextMate

- Zope.tmbundle (pyflakes on save)
  - http://github.com/tomster/zope.tmbundle
- PEP8 Bundle (requires python2.5+)
  - http://github.com/ppierre/python-pep8-tmbundle

Buildout

```
[buildout]
...
parts =
    ...
    pylint

[pylint]
recipe = zc.recipe.egg
eggs =
    pylint
    ${instance:eggs}
entry-points = pylint=pylint.lint:Run
scripts = pylint
arguments = [
    '--output-format=colorized',
    '--zope=y',
    '--reports=no',
# Suppress certain errors (interfaces missing __init__,
invalid imports etc)
    '--disable=E0611,F0401,W0232',
    ] + sys.argv[1:]
```

six feet up

sixfeetup.com/deploy2010

What did we learn

- Zen
- PEP8
- Style
- Development workflow

six feet up

sixfeetup.com/deploy2010

Happiness!

http://www.flickr.com/photos/docsearls/199700290/

six feet up

sixfeetup.com/deploy2010

Links

- PEP8 - http://www.python.org/dev/peps/pep-0008/
- Pyflakes - http://divmod.org/trac/wiki/DivmodPyflakes
- pylint - http://www.logilab.org/857
- pyflakes.vim - http://www.vim.org/scripts/script.php?script_id=2441
- PEP8 and pyflakes in emacs - http://github.com/akaihola/flymake-python
- TextMate Zope Bundle - http://github.com/tomster/zope.tmbundle

More info at:
sixfeetup.com/deploy2010

six feet up
sixfeetup.com/deploy2010