

#### [Indentation]

Mixing use of tab and blank indentation is not allowed in Python 3. Please fix to use either tab or blank only.

#### [print]

The print function needs parentheses in Python3.

Python2:

```
print "ABC", "123"
```

Python3:

```
print("ABC", "123")
```

#### [print without a line feed (continuation line)]

Python2:

```
print "ABC",
```

Python3:

```
print("ABC", end="")
```

#### [string functions]

The string functions were changed. The followings need to be fixed.

Python2:

```
string.atof(s)
```

```
string.atoi(s[, base])
```

```
string.atol(s[, base])
```

```
string.find(s, sub[, start[, end]])
```

```
string.index(s, sub[, start[, end]])
```

```
string.split(s[, sep[, maxsplit]])
```

```
string.join(words[, sep])
```

```
string.upper(s)
```

```
string.replace(s, old, new[, maxreplace])
```

Python3:

```
float(s)
```

```
int(s[, base])
int(s[, base])
s.find(sub[, start[, end]])
s.index(sub[, start[, end]])
s.split([, sep[, maxsplit]])
sep.join(words)
s.upper()
s.replace(old, new[, maxreplace])
```

[`p`] (backquote returns text expression in Python2)

Python2:

```
`p`
```

Python3:

```
repr(p)
```

[/(division)]

Integer division yields float in Python3. It is need to be modified to return integer.  
Replaced to "//".

`range(N/2+1)` -> `range(N//2+1)`

[range] (If you need a list type value.)

The range returns an object not a list. Use a list function.

Python2:

```
>>> range(10)
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

Python3:

```
>>> range(10)
range(0, 10)
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[map.keys()] (If you need a list type value.)

The keys() method returns an object not a list in Python3. Use a list function.

Python2:

```
>>> m = {'A':1, 'B':2, 'C':3}
>>> m.keys()
['A', 'C', 'B']
```

Python3:

```
>>> m = {'A':1, 'B':2, 'C':3}
>>> m.keys()
dict_keys(['A', 'B', 'C'])
>>> list(m.keys())
['A', 'B', 'C']
```

[map.has\_key(key)]

No has\_key method in Python3.

Python2:

```
map.has_key(key)
```

Python3:

```
key in map
```

[list.sort()] (How to sort a list of objects, etc)

Python2:

The sort operation uses def \_\_cmp\_\_(self, other): method of the sorted class in Python2. The \_\_cmp\_\_ needs to return an integer type value. Or the compare function is passed with the cmp argument parameter. There is no built-in cmp function in Python3.

Python3:

The sort operation uses def \_\_lt\_\_(self, other): method of the sorted class in Python3. The \_\_lt\_\_ needs to return an boolean type value. Or the compare function is passed with key=cmp\_to\_key(function) argument parameter.

```
from functools import cmp_to_key
list = [(3,5), (4,2), (7,4), (5,1), (2,3)]
def compare_second(x, y):
    return x[1] - y[1]
list.sort(key=cmp_to_key(compare_second))
```

[except]

The "except" statement is rewritten as below.

Python2:

```
except IOError, message:
    raise IOError, message
```

Python3:

```
except IOError as message:
    raise IOError(message)
```

[raise]

Python2: (It is not bad that a string follows the raise statement)

```
PlotterError = 'PlotterError'
raise PlotterError, 'No labels'
```

Python3: (a subclass of Exception class follows the raise statement)

```
class PlotterError(Exception):
    pass
raise PlotterError('No labels')
```

[import \*]

It is not able to write "from module import \* " in a function.

Python3:

```
def func():
    from random import *
```

-> SyntaxError: import \* only allowed at module level

[\_\_builtin\_\_]

Python2:

```
import __builtin__
```

Python3:

```
import builtins
```

[Encoding]

For the first line of the script with double-byte characters. Please comment in English.

Shift-JIS

```
# -*- coding: cp932 -*-
```

UTF8

```
# -*- coding: utf-8 -*-
```