Python２→３バージョンアップ時のスクリプト　主な修正箇所

V 0.14

OCTA の Python スクリプトを２→３にコンバートした時の修正箇所をまとめました。

【インデント】

Python3 ではタブと空白の混在が許されない。タブのみ、空白のみのどちらかに統一。
混在していると、IndentationError が出力される。

【print】

Python3 では( )が必要。

Python2:

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

Python3:

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

【改行しない（継続行）print】

Python2:

```
print "ABC",
```

Python3:

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

【string】

Python3 では、string 関数の多くは使えない。以下はよく使われているので修正が必要。

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`】（バッククォート：オブジェクトの文字列表現を返す）
Python2:
  `p`
Python3:
  repr(p)

【/ （割り算）】
Python3 では整数の割り算でも実数が返される。整数を返す場合は修正が必要。//に替える。
range(N/2+1) → range(N//2+1)

【range】（リストデータが必要な場合）
Python3 ではオブジェクトが返される。list(...)で囲む必要がある。
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()】（リストデータが必要な場合）

Python3 ではオブジェクトが返される。list(...)で囲む必要がある。

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)】

Python3 では has_key なし。

Python2:

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

Python3:

```
key in map
```

【list.sort()】（クラス）オブジェクト等のリストのソート

Python2:

対象クラスの def __cmp__(self, other):メソッドが比較に使われていた。__cmp__の戻り値は整数。または、cmp パラメーター引数で比較関数を渡す。

なお、Python3 ではビルトイン関数の cmp はなくなった。

Python3:

対象クラスの def __lt__(self, other):メソッドが比較に使われる。__lt__の戻り値はブール値。または、比較関数を key=cmp_to_key(比較関数)で渡す。

```
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))
```

【例外 except】

except 文の書き換え例。

Python2:

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

Python3:

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

【raise】

Python2:（raise の次の式が文字列でも通っていた）

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

Python3:（raise の次の式は Exception クラスのサブクラスまたはインスタンス）

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

【import *】

関数内に from モジュール import * は書けない。

Python3:

```
def func():
        from random import *
-> SyntaxError: import * only allowed at module level
```

【__builtin__】

Python2:

```
  import __builtin__
```

Python3:

```
  import builtins
```

【エンコーディング】
日本語コメントのあるスクリプトの１行目に必要。コメントは英語でお願いします。

Shift-JIS の場合
```
# -*- coding: cp932 -*-
```

UTF8 の場合
```
# -*- coding: utf-8 -*-
```