11/08/2021

Python pour l'ingénieur

Morceaux de code

Annexe

Morceaux de code

```
# Fonctions lambda
liste = [('a',1,'dummy'), ('x',2), ('y',2)]
print(liste)
liste.sort(key=lambda item: item[1], reverse=True)
print(liste)


# isinstance
a = 5
if isinstance(a, float):
    print('a -> float')
elif isinstance (a, int):
    print('a -> int')
print(isinstance ('', str))
```

Morceaux de code

```
# Opérateur +
class Book:
    def __init__(self, title, nbpages):
        self.title = title
        self.nbpages = nbpages
    def __repr__(self):
        return '{} : {:d} pages'.format(self.title, self.nbpages)
    def __add__(self, other):
        return Book(' / '.join([self.title, other.title]),
                    self.nbpages + other.nbpages)
a = Book('Pavé', 1000)
b = Book('Résumé', 1)
print(a+b)
```

Morceaux de code

```
# * et **
def myfunc(*args, **kwargs):
    s = ' '.join(args)
    for k,v in kwargs.items():
      s += ' {}={}'.format(k,v)
    return s

print(myfunc('a'))
print(myfunc('a', 'b'))
print(myfunc('a', 'b', hello='world', good='bye'))


# set
a = {1,2,99,4,1,1,2,1,6,1,1,1,1,1,1,9,99,2}
print(a)
print(len(a))

b = [1,2,99,4,1,1,2,1,6,1,1,1,1,1,1,9,99,2]
c = list(set(b))
c.sort()
print(c)
```

Morceaux de code

```
# Générateurs
#   En lecture seule une seule fois et de longueur inconnue !
generator = (x**2 for x in range(3))
for val in generator:
    print(val)
for val in generator:  # vide...
    print(val)


# yield
#   Dans une fonction, la transforme en générateur !
def compute(exp):
    mylist = range(3)
    for i in mylist:
      yield i**exp

generator1 = compute(2)
generator2 = compute(3)
for val in generator1:
    print(val)
for val in generator2:
    print(val)
```

Morceaux de code

```
# for/else
def search(data, value):
    for v in data:
        if v == value:
            print('FOUND')
            break
    else:
        print('NOT FOUND')
        v = None
    return v

print(search([1, 42, 99, 51], 42))
print(search([1, 42, 99, 51], 43))


# with
with open('expourfinir3.py') as f:
    for i,line in enumerate(f):
        if i == 19:
            print(line.strip())
            break
```

Morceaux de code

```
# enum
from enum import Enum
class Color(Enum):
    RED = 1
    GREEN = 2
    BLUE = 3

for const in Color:
    print(const.value)

print(Color.RED is Color.RED)
print(Color.RED is Color.GREEN)


# itertools
import itertools
data = [3, 4, 6, 2, 1, 9, 0, 7, 5, 8]
print(data)
print(list(itertools.accumulate(data, max)))
print(list(itertools.compress(data, [1,0,1,0,1,1])))
```

Morceaux de code

```
# next
def gencolours():
    while(True):
        yield 'RED'
        yield 'GREEN'
        yield 'BLUE'
colours = gencolours()
for i in range(10):
    print(next(colours))
```