

Lập trình Python cơ bản

Giới thiệu thư viện NumPy



VietAI Teaching Team



VietAI

Lập trình Python

Lập trình Python

1. Giới thiệu chung
2. Các kiểu dữ liệu và thao tác cơ bản
 - a) Integer, Float, Boolean, String
 - b) List, Dictionary, Set, Tuple
3. Cấu trúc điều khiển (rẽ nhánh)
4. Cấu trúc lặp
5. Function
6. Class

1 Giới thiệu chung

- Ngôn ngữ lập trình cấp cao
- Tính chất: tuần tự và hướng đối tượng (object-oriented)
- Ngôn ngữ thông dịch (interpreted language)
- Ngắn gọn, cho phép người dùng có thể viết chương trình trong ít dòng code nhất.
- 2 phiên bản chính: Python 2 và Python 3

2 Float và Int

```
x = 3
print(type(x))
print(x)
print(x + 1)
print(x - 1)
print(x * 2)
print(x**2)
x += 1
print(x)
x *= 2
print(x)
print(x, x + 5, x*2, x**2)
print(x%5, x/5, x//5)
```

2 Boolean

```
t = True
f = False
print(type(t))
print(t and f)
print(t or f)
print(not t)
print(t != f)
```

2 String

```
hello = 'hello'
world = 'world'
print(hello)
print(len(hello))
hw = hello + ' ' + world
print(hw)
hw12 = '%s %s %d' % (hello, world, 12)
print(hw12)
```

```
s = "hello"  
print(s.upper())  
print(s.capitalize())  
print(s.replace('l', '(ell)'))  
print(s.find('el'))  
print(' world '.strip())
```


2 List

```
xs = [3, 2, 1]
print(xs, xs[2])
print(xs[-1])
xs[2] = 'foo'
print(xs)
xs.append('bar')
print(xs)
x = xs.pop()
print(x, xs)
```

2 List

```
nums = list(range(5))  
print(nums)  
print(nums[2:4])  
print(nums[2:])  
print(nums[:2])  
print(nums[:])  
print(nums[:-1])  
nums[2:4] = [8, 9]  
print(nums)
```

2 Dictionary

```
d = {'cat': 'cute', 'dog': 'furry'}  
print(d['cat'])  
print('cat' in d)  
d['fish'] = 'wet'  
print(d['fish'])  
print(d.get('rat'))  
print(d.get('monkey', 'N/A'))  
print(d['monkey'])
```

```
animals = {'cat', 'dog'}  
print('cat' in animals)  
print('fish' in animals)  
animals.add('fish')  
print('fish' in animals)  
print(len(animals))  
animals.add('cat')  
print(len(animals))  
animals.remove('cat')  
print(len(animals))
```

2 Tuple

```
t = (5,6)
print(type(t))
a, b = t
print(a, b)
t2 = t + (7, 8)
print(t2, t2[0])
t2[0] = 4
```

Cấu trúc điều khiển (rẽ nhánh)

```
temp = 60
if temp > 100:
    print("REALLY HOT")
elif temp > 85:
    print("Hot")
elif temp >= 60:
    print("Comfortable")
else:
    print("Cold")
```

4 Cấu trúc lặp trên List

```
animals = ['cat', 'dog', 'monkey']  
for item in animals:  
    print(item)  
  
for index, item in enumerate(animals):  
    print("#%d: %s" % (index, item))
```

4

Cấu trúc lặp trên Dictionary

```
d = {'fish': 0, 'cat': 4, 'spider': 8}
print(d['spider'])
print(d.keys())
print(d.values())
for animal, legs in d.items():
    print("A %s has %d legs" % (animal, legs))
```

```
nums = [0, 1, 2, 3, 4, 5, 6]
even_square = {x: x ** 2 for x in nums if x % 2 == 0}
print(even_square)
```


4 Function

```
def sign(x):  
    if x > 0:  
        return 'positive'  
    elif x < 0:  
        return 'negative'  
    else:  
        return 'zero'  
  
for x in [0, 5, -1]:  
    print("Input: %d, Output: %s" %(x, sign(x)))
```

4 Function

```
def f(a, b=2, c=3):  
    print('a =', a)  
    print('b =', b)  
    print('c =', c)  
  
f(3, -1, 1.5)  
f(1)  
f(1, 2)  
f(a=0.5, c=4)
```

5 Class

```
class Greeter():  
    # Constructor  
    def __init__(self, name):  
        self.name = name # Create an instance variable  
  
    # Instance method  
    def greet(self, loud=False):  
        if loud:  
            print("HELLO, %s" % self.name)  
        else:  
            print("hello, %s" % self.name)  
  
g = Greeter("VietAI")  
g.greet()  
g.greet(loud=True)
```



VietAI

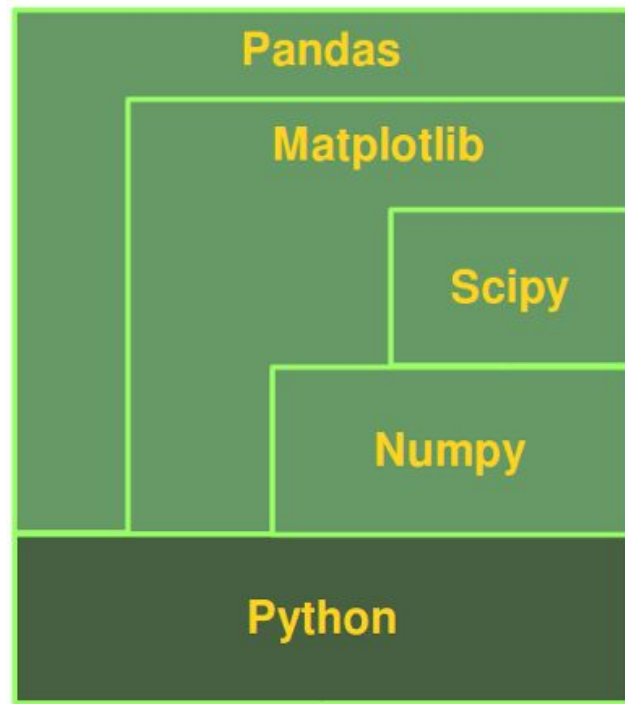
Giới thiệu thư viện NumPy

Thư viện NumPy

1. Giới thiệu chung
2. NumPy Array
3. Array Indexing trong NumPy
4. Các thao tác trên Array trong NumPy

1 Giới thiệu chung

- NumPy (**N**umerical **P**ython) là một thư viện Python hỗ trợ xử lý liên quan tới ma trận, vector.
- Được viết bằng Python và C, nên tốc độ thực thi nhanh.



```
import numpy as np
a = np.array([1, 2, 3])
print(type(a))
print(a.shape)
print(a[0], a[1], a[2])
a[0] = 5
print(a)

b = np.array([[1, 2, 3], [4, 5, 6]])
print(b.shape)
print(b[0, 0], b[0, 1], b[1, 0])
```

2

NumPy Array

```
import numpy as np
a = np.zeros((2,2))
print(a)
b = np.ones((1,2))
print(b)
c = np.full((2,2), 7)
print(c)
d = np.eye(2)
print(d)
e = np.random.random((2,2))
print(e)
```


3 Array Indexing

```
import numpy as np
a = np.array([[1,2,3,4], [5,6,7,8], [9, 10, 11, 12]])
b = a[:2, 1:3]
print(b)
print(a[0, 1])
b[0, 0] = 77
print(a[0, 1])
```

3 Array Indexing

```
import numpy as np
a = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])
row_1 = a[1, :]
row_2 = a[1:2, :]
print(row_1, row_1.shape)
print(row_2, row_2.shape)

col_1 = a[:, 1]
col_2 = a[:, 2]
print(col_1, col_1.shape)
print(col_2, col_2.shape)
```

Array Indexing

```
a = np.array([[1, 2], [3, 4], [5, 6]])  
print(a[[0, 1, 2], [0, 1, 0]])  
print(np.array([a[0, 0], a[1, 1], a[2, 0]]))  
print(a[[0, 0], [1, 1]])  
print(np.array([a[0, 1], a[0, 1]]))
```

3

Array Indexing

```
import numpy as np
a = np.array([[1,2,3], [4,5,6], [7,8,9], [10,11,12]])
print(a)

b = np.array([0,2,0,1])
print(a[np.arange(4), b])

a[np.arange(4), b] += 10
print(a)
```

Array Indexing

```
import numpy as np
a = np.array([[1,2], [3,4], [5,6]])

bool_idx = (a > 2)

print(bool_idx)
print(a[bool_idx])
print(a[a > 2])
```

<http://docs.scipy.org/doc/numpy/reference/arrays.indexing.html>

Array Indexing

```
import numpy as np

x = np.array([1, 2])
print(x.dtype)

x = np.array([1.0, 2.0])
print(x.dtype)

x = np.array([1, 2], dtype=np.int64)
print(x.dtype)
```

<https://docs.scipy.org/doc/numpy/reference/arrays.dtypes.html>

4

Các thao tác trên Array

```
import numpy as np

x = np.array([[1,2], [3,4]], dtype=np.float64)
y = np.array([[5,6], [7,8]], dtype=np.float64)

print(x + y)
print(np.add(x, y))

print(x - y)
print(np.subtract(x, y))

print(x * y)
print(np.multiply(x, y))

print(x / y)
print(np.divide(x, y))
```

Các thao tác trên Array

```
import numpy as np
x = np.array([[1, 2], [3, 4]])
y = np.array([[5, 6], [7, 8]])

v = np.array([9, 10])
w = np.array([11, 12])

print(v.dot(w))
print(np.dot(v, w))

print(x.dot(v))
print(np.dot(x, v).shape)

print(x.dot(y))
print(np.dot(x, y))
```


4

Các thao tác trên Array

```
import numpy as np
x = np.array([[1,2,3], [3,4,5]])

print(np.sum(x))
print(np.sum(x, axis=0))
print(np.sum(x, axis=1))
```

<https://docs.scipy.org/doc/numpy/reference/routines.math.html>

Các thao tác trên Array

```
import numpy as np
x = np.array([[1, 2], [3, 4]])

print(x)
print(x.T)

v = np.array([1, 2, 3])
print(v)
print(v.T)
```

<https://docs.scipy.org/doc/numpy/reference/routines.array-manipulation.html>

4

Các thao tác trên Array

```
import numpy as np
x = np.array([[1,2,3], [4,5,6], [7,8,9], [10,11,12]])
v = np.array([1,0,1])
y = np.empty_like(x)

for i in range(4):
    y[i, :] = x[i, :] + v

print(y)
```

4 Các thao tác trên Array

```
import numpy as np
x = np.array([[1,2,3], [4,5,6], [7,8,9], [10,11,12]])
v = np.array([1,0,1])
vv = np.tile(v, (4,1))

print(vv)

vv2 = np.tile(v, (4,2))
print(vv2)

y = x + vv
print(y)
```

4

Các thao tác trên Array

```
import numpy as np

x = np.array([[1,2,3], [4,5,6], [7,8,9], [10,11,12]])
v = np.array([1,0,1])

print(v+x)
```

4

Các thao tác trên Array

```
import numpy as np

v = np.array([1,2,3])
w = np.array([4,5])

x = np.array([[1,2,3], [4,5,6]])

print(x + v)
print((x.T + w).T)
print(x * 2)
```

Tài liệu tham khảo

1. Python 3.6 documentation - <https://docs.python.org/3/>
2. NumPy documentation - <https://docs.scipy.org/doc/numpy/reference/>