

# Bài 2: Lập trình Python cơ bản. Giới thiệu thư viện NumPy

---

# Lập trình Python

---

# Lập trình Python

## Outline

- Giới thiệu chung
- Các kiểu dữ liệu và phép toán cơ bản
  - Integer, Float, Boolean, String
  - List, Dictionary, Set, Tuple
- Cấu trúc điều khiển (rẽ nhánh)
- Cấu trúc lặp
- Function
- Class

# Lập trình Python

## Giới thiệu chung

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

# Lập trình Python

Các kiểu dữ liệu và phép toán cơ bản - Integer và Float

```
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)
```

# Lập trình Python

Các kiểu dữ liệu và phép toán cơ bản - Boolean

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

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - String

```
hello = 'hello'
world = 'world'
print(hello)
print(world)
hw = hello + world
print(hw)
hw2019 = "%s %s %0.0f" % (hello, world, 2019.123)
print(hw2019)
print("{1} {0} {2:0.2f}".format(hello, world, 168.123))
```

```
hello
world
helloworld
hello world 2019
world hello 168.12
```

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - String

```
s = 'hello world'
print(s.capitalize())
print(s.upper())
print("|{}|".format(s.rjust(15)))
print("|{}|".format(s.center(15)))
print(s.replace('l', "L"))
s_space = "    hello world    "
print("|{}|, |{}|".format(s_space, s_space.strip()))
```

Hello world

HELLO WORLD

| hello world|

| hello world |

heLLo worLd

| hello world |, |hello world|



# Lập trình Python

Các kiểu dữ liệu và phép toán cơ bản - 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)
```

```
[3, 2, 1] 1
1
[3, 2, 'foo']
[3, 2, 'foo', 'bar']
bar [3, 2, 'foo']
```

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - List

```
nums = [0,1,2,3,4,5]
print(nums)
print(nums[2:4])
print(nums[2:])
print(nums[:2])
print(nums[:])
print(nums[:-1])
nums[2:4] = [8, 9, 10]
print(nums)
```

```
[0, 1, 2, 3, 4, 5]
[2, 3]
[2, 3, 4, 5]
[0, 1]
[0, 1, 2, 3, 4, 5]
[0, 1, 2, 3, 4]
[0, 1, 8, 9, 10, 4, 5]
```

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - 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'])
```

```
cute  
True  
wet  
None  
N/A
```

```
-----  
KeyError                                Traceback (most recent call last)  
<ipython-input-111-fb7d56b7f5c7> in <module>()  
      6 print(d.get('rat'))  
      7 print(d.get('monkey', 'N/A'))  
----> 8 print(d['monkey'])
```

```
KeyError: 'monkey'
```

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - Set

```
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))
```

```
True  
False  
True  
3  
3  
2
```

# Lập trình Python

## Các kiểu dữ liệu và phép toán cơ bản - Tuple

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

```
<class 'tuple'>
(5, 6, 7, 8) 5
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-63-6331d5359e8c> in <module>()
      3 t2 = t + (7, 8)
      4 print(t2, t2[0])
----> 5 t2[0] = 4
```

```
TypeError: 'tuple' object does not support item assignment
```

# Lập trình Python

## 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")
```

Comfortable

# Lập trình Python

## Cấu trúc lặp trên List

```
animals = ['cat', 'dog', 'monkey']  
for item in animals:  
    print(item)  
  
for index, item in enumerate(animals):  
    print("{} {}".format(index, item))
```

```
cat  
dog  
monkey  
0, cat  
1, dog  
2, monkey
```

# Lập trình Python

## Cấu trúc lặp trên List

```
nums = [0, 1, 2, 3, 4, 5]
squares = []
for item in nums:
    squares.append(item ** 2)
print(squares)

squares_inline = [item ** 2 for item in nums]
print(squares_inline)
```

```
[0, 1, 4, 9, 16, 25]
```

```
[0, 1, 4, 9, 16, 25]
```



# Lập trình Python

## Cấu trúc lặp trên Dictionary

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

```
8
dict_keys(['fish', 'cat', 'spider']) ['fish', 'cat', 'spider']
dict_values([0, 4, 8]) [0, 4, 8]
A fish has 0 legs
A cat has 4 legs
A spider has 8 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)
```

```
{0: 0, 2: 4, 4: 16, 6: 36}
```

# Lập trình Python

## 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: {}, Output: {}".format(x, sign(x)))
```

Input: 0, Output: zero

Input: 5, Output: positive

Input: -1, Output: negative

# Lập trình Python

## Function

```
def plus(a, b=2):
    print('a + b =', a + b)
```

```
plus(1)
plus(10, b=500)
plus('hello ', 'world')
plus('hello', 100)
```

```
a + b = 3
a + b = 510
a + b = hello world
```

```
-----
TypeError                                Traceback (most recent call last)
<ipython-input-103-0b36f165b96f> in <module>()
      5 plus(10, b=500)
      6 plus('hello ', 'world')
----> 7 plus('hello', 100)

<ipython-input-103-0b36f165b96f> in plus(a, b)
      1 def plus(a, b=2):
----> 2     print('a + b =', a + b)
      3
      4 plus(1)
      5 plus(10, b=500)
```

**TypeError:** must be str, not int

# Lập trình Python

## 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, {}'.format(self.name))  
        else:  
            print('hello, {}'.format(self.name))  
  
g = Greeter("Binh")  
g.greet()  
g.greet(loud=True)
```

```
hello, Binh!  
HELLO, Binh!
```

# Giới thiệu thư viện NumPy

---

<https://bit.ly/2DmNPYA>

# Giới thiệu thư viện NumPy

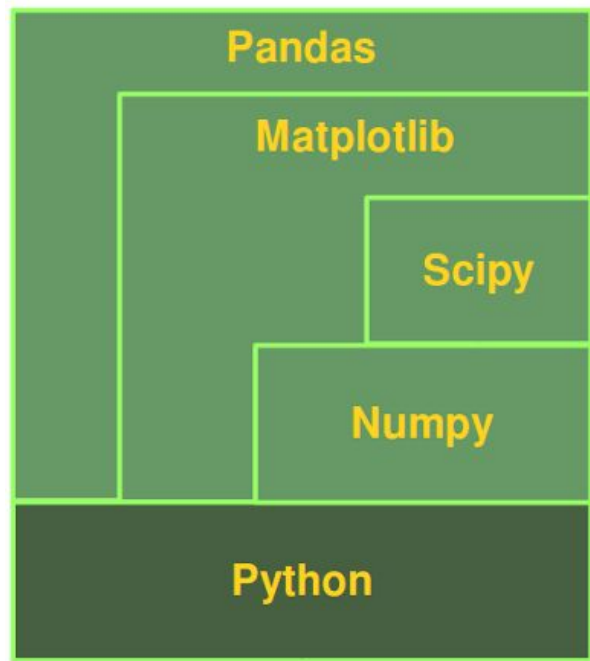
## Outline

- Giới thiệu chung
- NumPy Array
- Array Indexing trong NumPy
- Các thao tác trên Array trong NumPy

# Giới thiệu thư viện NumPy

## Giới thiệu chung

- NumPy (Numerical Python) 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.



# Giới thiệu thư viện NumPy

## Numpy array

```
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])
```



# Giới thiệu thư viện NumPy

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)
```

# Giới thiệu thư viện NumPy

## 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])
```

# Giới thiệu thư viện NumPy

## 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)
```

# Giới thiệu thư viện NumPy

## 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]]))
```

# Giới thiệu thư viện NumPy

## 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)
```

# Giới thiệu thư viện NumPy

## 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>

# Giới thiệu thư viện NumPy

## 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>

# Giới thiệu thư viện NumPy

## Các thao tác trên Array trong NumPy

```
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))
```



# Giới thiệu thư viện NumPy

## Các thao tác trên Array trong NumPy

```
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))
```

# Giới thiệu thư viện NumPy

Các thao tác trên Array trong NumPy

```
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>

# Giới thiệu thư viện NumPy

Các thao tác trên Array trong NumPy

```
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>

# Giới thiệu thư viện NumPy

Các thao tác trên Array trong NumPy

```
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)
```

# Giới thiệu thư viện NumPy

## Các thao tác trên Array trong NumPy

```
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)
```

# Giới thiệu thư viện NumPy

Các thao tác trên Array trong NumPy

```
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)
```

# Giới thiệu thư viện NumPy

Các thao tác trên Array trong NumPy

```
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/>