

# Bài 2: Lập trình Python cơ bản

---

Tuần 2

# Lập trình Python

---

# Lập trình Python

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

# 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

# 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)
```

# Boolean

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

# String

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

# String

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



# 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)
```

# 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)
```

# 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'])
```

# 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))
```

# 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")
```

# 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))
```

# 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)
```



# 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)))
```

# 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)
```

# 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)
```

# Tổng kết

- Giới thiệu chung
- Các kiểu dữ liệu và thao tác cơ bản
- Cấu trúc điều khiển (rẽ nhánh)
- Cấu trúc lặp
- Function
- Class

# Giới thiệu thư viện NumPy

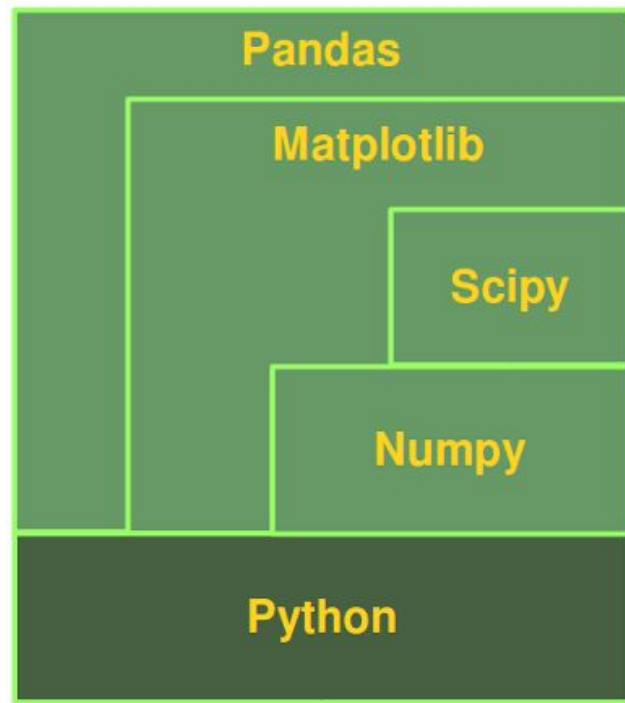
---

# Thư viện NumPy

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

# 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.



# 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])
```



# 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)
```

# 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])
```

# 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]]))
```

# 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>

# 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))
```

# 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>

# 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)
```

# 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)
```

# 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)
```

# 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)
```

# Tổng kết

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



# Trực quan hóa với Matplotlib

---

# Thực quan hóa với Matplotlib

- Cài thư viện Matplotlib (nếu chưa có):

```
conda install -c conda-forge matplotlib
```

- Khởi tạo dữ liệu dùng numpy:

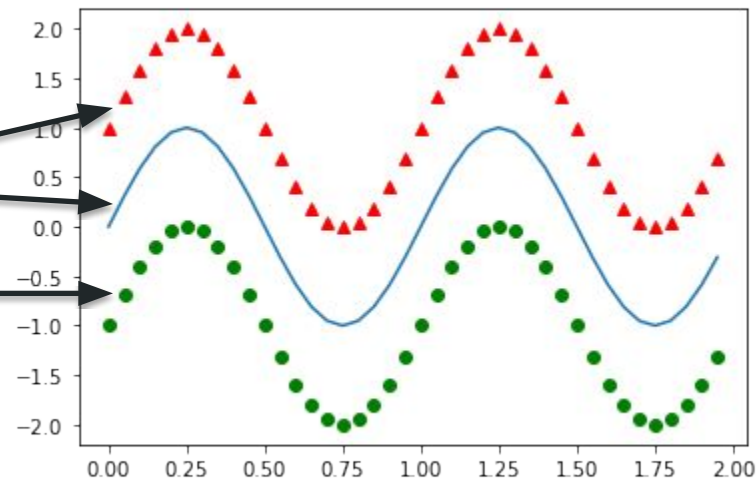
```
import matplotlib.pyplot as plt
import numpy as np

# Tạo dữ liệu hình sin
t = np.arange(0.0, 2.0, 0.05) # t lấy mẫu từ 0 đến 2, bước nhảy 0.05
s = np.sin(2 * np.pi * t)     # s tính theo t: s = sin(2*pi*t)
```

# Trực quan hóa với Matplotlib

- Vẽ dạng đường và điểm
  - Mặc định là dạng đường (line)
  - Tham số để vẽ điểm 'r^' ☐ tam giác đỏ, 'go' ☐ tròn xanh lá

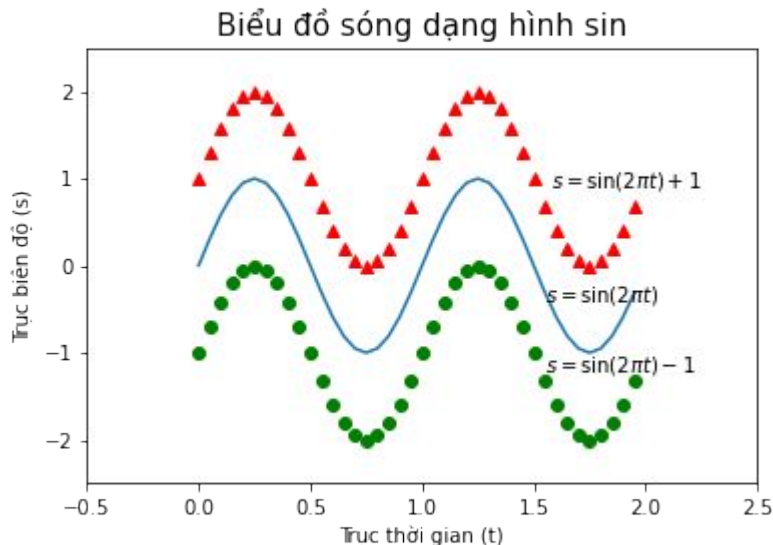
```
# Vẽ dạng đường và điểm  
plt.plot(t, s)  
plt.plot(t, s+1, 'r^')  
plt.plot(t, s-1, 'go')  
plt.show()
```



# Trực quan hóa với Matplotlib

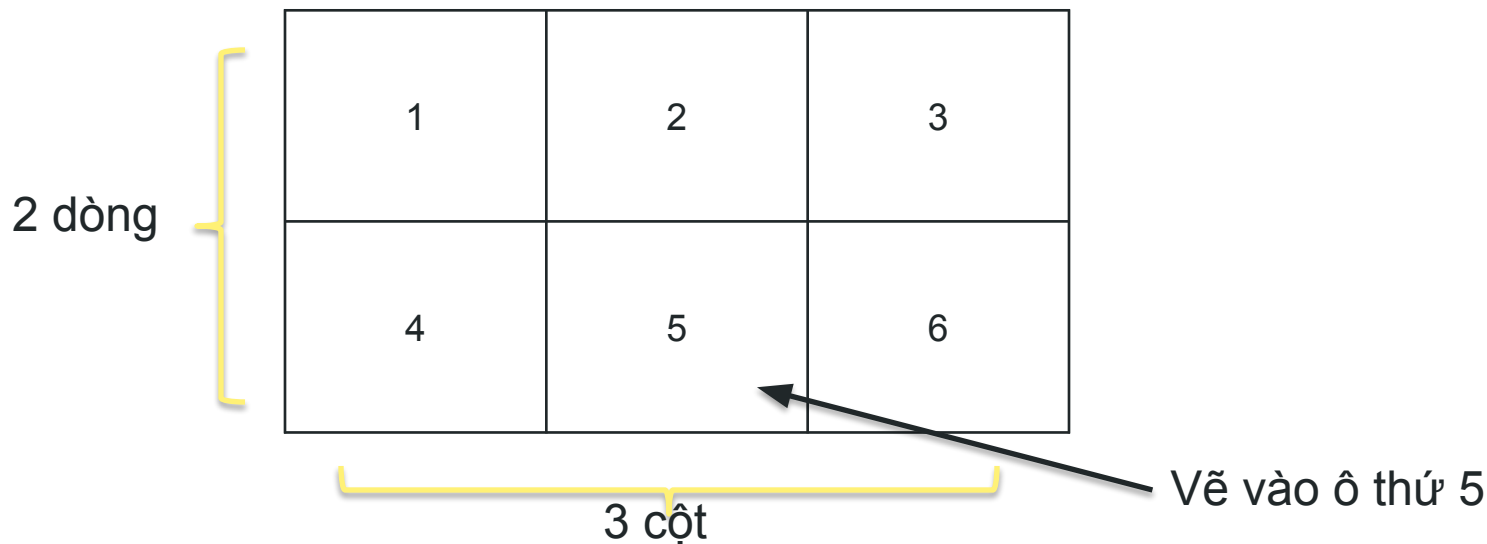
- Cấu hình biểu đồ:

```
plt.plot(t, s)
plt.plot(t, s+1, 'r^')
plt.plot(t, s-1, 'go')
plt.title('Biểu đồ sóng dạng hình sin', fontsize=15)
plt.xlabel('Trục thời gian (t)')
plt.ylabel('Trục biên độ (s)')
plt.text(1.55, -0.4, r'$s=\mathrm{sin}(2 \pi t)$')
plt.text(1.58, 0.9, r'$s=\mathrm{sin}(2 \pi t) + 1$')
plt.text(1.55, -1.2, r'$s=\mathrm{sin}(2 \pi t) - 1$')
plt.xlim(-0.5, 2.5)
plt.ylim(-2.5, 2.5)
plt.show()
```



# Trực quan hóa với Matplotlib

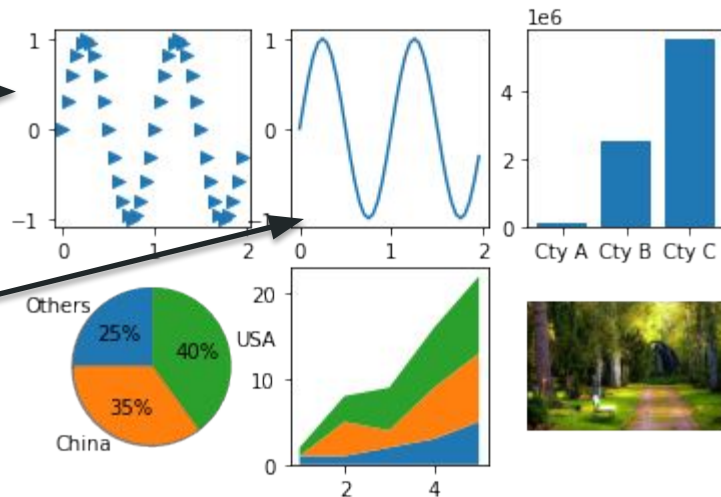
- Vẽ nhiều biểu đồ với hàm: `subplot (<nrow><ncol><index>)`
- Ví dụ: `subplot (235)`



# Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 1 trong bảng 2x3  
# Biểu đồ dạng điểm  
plt.subplot(231)  
plt.scatter(t, s, marker=">")
```

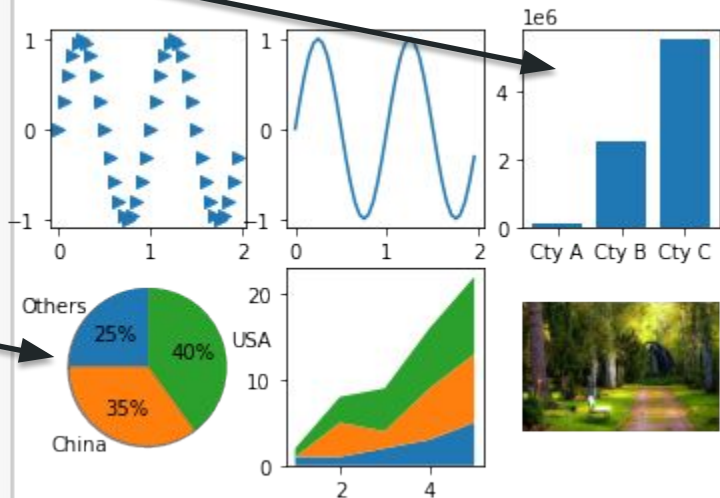
```
# Vẽ ô thứ 2 trong bảng 2x3  
# Biểu đồ dạng đường  
plt.subplot(232)  
plt.plot(t, s)
```



# Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 3 trong bảng 2x3  
# Biểu đồ cột  
plt.subplot(233)  
x = np.arange(3)  
money = [1.5e5, 2.5e6, 5.5e6]  
plt.bar(x, money)  
plt.xticks(x, ('Cty A', 'Cty B', 'Cty C'))
```

```
# Vẽ ô thứ 4 trong bảng 2x3  
# Biểu đồ tròn  
plt.subplot(234)  
labels = 'Others', 'China', 'USA'  
sizes = [25, 35, 40]  
plt.pie(sizes, labels=labels, autopct='%1.0f%%',  
        shadow=True, startangle=90)
```

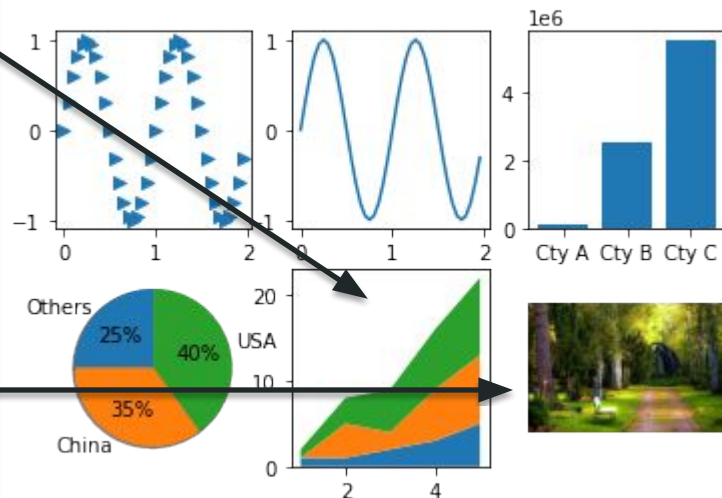


# Trực quan hóa với Matplotlib

```
# Vẽ ô thứ 5 trong bảng 2x3  
# Biểu đồ xếp chồng  
plt.subplot(235)  
x = [1, 2, 3, 4, 5]  
y1 = [1, 1, 2, 3, 5]  
y2 = [0, 4, 2, 6, 8]  
y3 = [1, 3, 5, 7, 9]  
y = np.vstack([y1, y2, y3])  
plt.stackplot(x, y1, y2, y3, labels=labels)
```

```
# Vẽ ô thứ 6 trong bảng 2x3  
# Vẽ ảnh  
plt.subplot(236)  
image = plt.imread('scene.jpg')  
plt.imshow(image)  
plt.axis('off')
```

```
plt.show()
```





# Tổng kết

- Cài đặt thư viện matplotlib
- Các lệnh cơ bản để vẽ biểu đồ
- Vẽ biểu đồ phức tạp và tùy chỉnh tham số

# Xử lý dữ liệu với Pandas

---

# Xử lý dữ liệu với Pandas

- Cài thư viện Pandas (nếu chưa có):

```
conda install -c anaconda pandas
```

# Xử lý dữ liệu với Pandas

- Tạo DataFrame
  - Khai báo dữ liệu theo **cột**

```
import pandas as pd

df = pd.DataFrame({
    "X" : [13, 30, 'A'],
    "Y" : [15, 32, 'B'],
    "Z" : [10, 29, 'O'],
    "T" : [12, 28, 'AB']},
    index = [1, 2, 3]
)
```

	X	Y	Z	T
1	13	15	10	12
2	30	32	29	28
3	A	B	O	AB

# Xử lý dữ liệu với Pandas

- Tạo DataFrame
  - Khai báo dữ liệu theo **dòng**

```
import pandas as pd

df = pd.DataFrame(
    [
        [13, 15, 10, 12],
        [30, 32, 29, 28],
        ['A', 'B', 'O', 'AB']
    ],
    index=[1, 2, 3]
)
```

	X	Y	Z	T
1	13	15	10	12
2	30	32	29	28
3	A	B	O	AB

# Xử lý dữ liệu với Pandas

- Tạo DataFrame
  - Load dữ liệu từ file csv (bảng)

```
stocks = pd.read_csv('stocks.csv')
```

	date	symbol	open	high	low	close	volume
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001

# Xử lý dữ liệu với Pandas

- Quy ước:

Chỉ mục (index)

Cột (column)

	date	symbol	open	high	low	close	volume
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001

Mẫu quan sát (observation)

Cột dữ liệu (Variable)

# Xử lý dữ liệu với Pandas

- Gom nhóm dữ liệu với phương thức *pivot*

```
stocks.pivot(index='date', columns='symbol', values='close')
```

symbol	AAPL	AMZN	GOOG
date			
2019-03-01	174.97	1671.73	1140.99
2019-03-04	175.85	1696.17	1147.80
2019-03-05	175.53	1692.43	1162.03
2019-03-06	174.52	1668.95	1157.86
2019-03-07	172.50	1625.95	1143.30



# Xử lý dữ liệu với Pandas

- Gom nhóm dữ liệu với phương thức *pivot*

```
stocks.pivot(index='date', columns='symbol', values=['close', 'volume'])
```

	close			volume		
symbol	AAPL	AMZN	GOOG	AAPL	AMZN	GOOG
date						
2019-03-01	174.97	1671.73	1140.99	25886167.0	4974877.0	1450316.0
2019-03-04	175.85	1696.17	1147.80	27436203.0	6167358.0	1446047.0
2019-03-05	175.53	1692.43	1162.03	19737419.0	3681522.0	1443174.0
2019-03-06	174.52	1668.95	1157.86	20810384.0	3996001.0	1099289.0
2019-03-07	172.50	1625.95	1143.30	24796374.0	4957017.0	1166559.0

# Xử lý dữ liệu với Pandas

- Gom nhóm dữ liệu với phương thức *pivot\_table*

```
import numpy as np
stocks.pivot_table(index='symbol', values=['close', 'volume'],
                    aggfunc=np.mean)
```

	close	volume
symbol		
<b>AAPL</b>	174.674	23733309.4
<b>AMZN</b>	1671.046	4755355.0
<b>GOOG</b>	1150.396	1321077.0

# Xử lý dữ liệu với Pandas

- Nối dữ liệu theo chiều **dọc** với *concat* (mặc định axis=0)

```
# Nối hai data frame theo chiều dọc
df1 = pd.DataFrame({
    "X" : ['A', 'B', 'O', 'AB'],
    "Y" : [15, 12, 10, 12],
    "Z" : [30, 28, 23, 29]},
    index = [1, 2, 3, 4])
df2 = pd.DataFrame({
    "X" : ['O', 'A', 'B'],
    "Y" : [20, 21, 22],
    "Z" : [32, 30, 20],
    "T" : [1, 0, 1]},
    index = [1, 2, 3])

df_new = pd.concat([df1, df2])
```

	X	Y	Z	T
1	A	15	30	NaN
2	B	12	28	NaN
3	O	10	23	NaN
4	AB	12	29	NaN
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0

# Xử lý dữ liệu với Pandas

- Điền giá trị khuyết với `fillna(value)`

```
df_new.fillna(-1)
```

	X	Y	Z	T
1	A	15	30	NaN
2	B	12	28	NaN
3	O	10	23	NaN
4	AB	12	29	NaN
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0



	X	Y	Z	T
1	A	15	30	-1.0
2	B	12	28	-1.0
3	O	10	23	-1.0
4	AB	12	29	-1.0
1	O	20	32	1.0
2	A	21	30	0.0
3	B	22	20	1.0

# Xử lý dữ liệu với Pandas

- Nối dữ liệu theo chiều **ngang** với *concat* (*axis=1*)

```
# Nối hai data frame theo chiều ngang
df1 = pd.DataFrame({
    "X" : ['A', 'B', 'O', 'AB'],
    "Y" : [15, 12, 10, 12],
    "Z" : [30, 28, 23, 29]},
    index = [1, 2, 3, 4])

df2 = pd.DataFrame({
    "U" : [0, 1, 0],
    "V" : [20, 1, 6]},
    index = [1, 2, 3])

pd.concat([df1, df2], axis=1)
```

	X	Y	Z	U	V
1	A	15	30	0.0	20.0
2	B	12	28	1.0	1.0
3	O	10	23	0.0	6.0
4	AB	12	29	NaN	NaN

# Xử lý dữ liệu với Pandas

- Lấy tập con theo dòng

```
# Lấy tập con theo dòng  
sub_df = df1[df1.Y > 10]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Y	Z
1	A	15	30
2	B	12	28
4	AB	12	29

# Xử lý dữ liệu với Pandas

- Lấy tập con theo dòng

```
sub_df = df1[df1.X.isin(['AB', 'A'])]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Y	Z
1	A	15	30
4	AB	12	29

# Xử lý dữ liệu với Pandas

- Lấy tập con theo cột

```
# Lấy tập con gồm nhiều cột  
columns = df1[['X', 'Z']]
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29



	X	Z
1	A	30
2	B	28
3	O	23
4	AB	29



# Xử lý dữ liệu với Pandas

- Lấy tập con theo cột

```
# Lấy tập con của một cột  
colX = df1.X  
# hoặc  
colX = df1['X']
```

	X	Y	Z
1	A	15	30
2	B	12	28
3	O	10	23
4	AB	12	29




```
1    A  
2    B  
3    O  
4   AB  
Name: X, dtype: object
```

# Xử lý dữ liệu với Pandas

- Tạo mới cột

```
stocks['value'] = stocks.close*stocks.volume
```



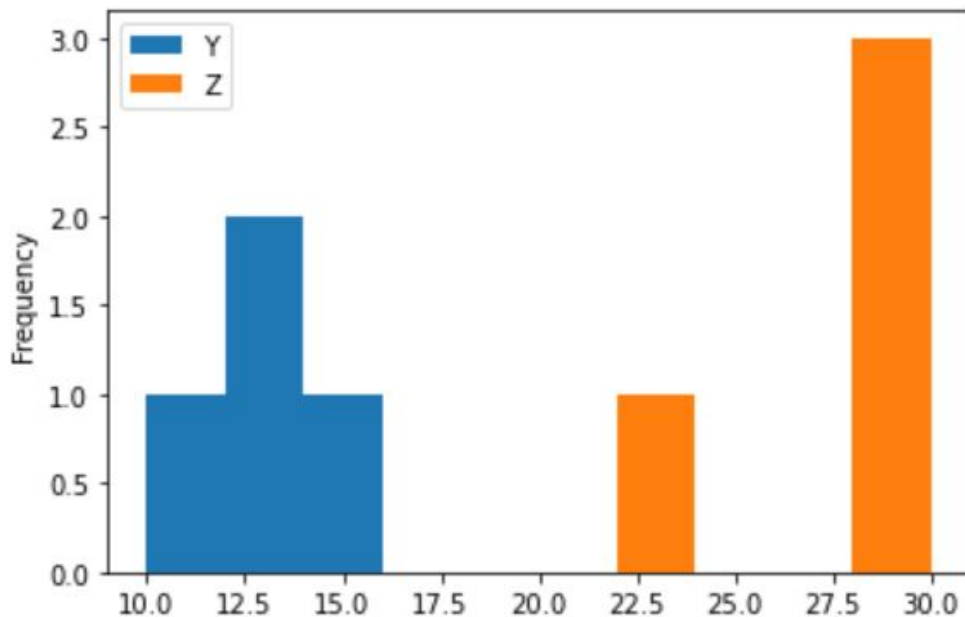
	date	symbol	open	high	low	close	volume	value
0	2019-03-01	AMZN	1655.13	1674.26	1651.00	1671.73	4974877	8.316651e+09
1	2019-03-04	AMZN	1685.00	1709.43	1674.36	1696.17	6167358	1.046089e+10
2	2019-03-05	AMZN	1702.95	1707.80	1689.01	1692.43	3681522	6.230718e+09
3	2019-03-06	AMZN	1695.97	1697.75	1668.28	1668.95	3996001	6.669126e+09
4	2019-03-07	AMZN	1667.37	1669.75	1620.51	1625.95	4957017	8.059862e+09

# Xử lý dữ liệu với Pandas

- Vẽ biểu đồ với plot

```
df.plot.hist()
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2545b11a7c0>

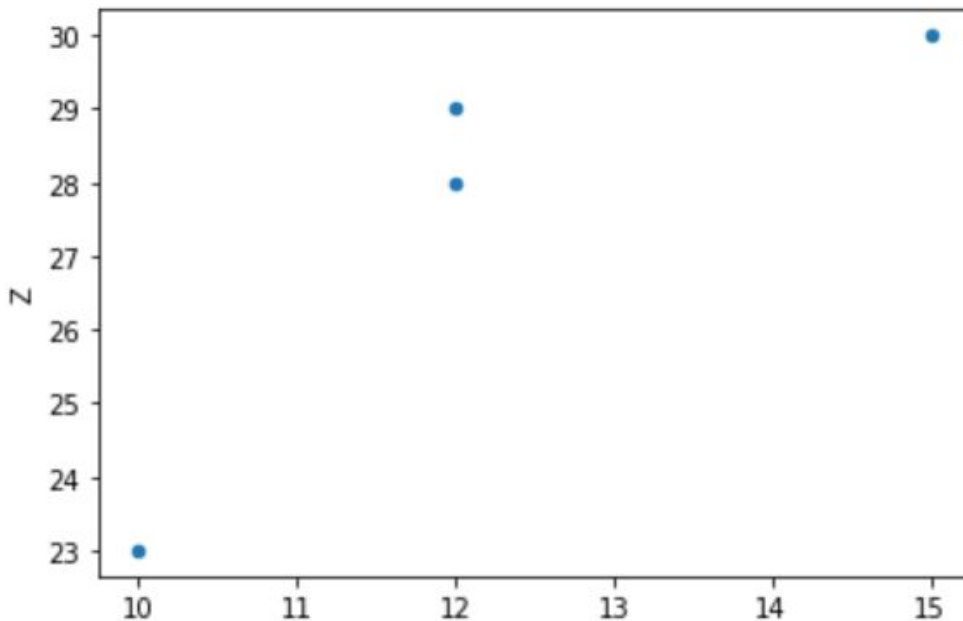


# Xử lý dữ liệu với Pandas

- Vẽ biểu đồ với plot

```
df.plot.scatter(x='Y', y='Z')
```

<matplotlib.axes.\_subplots.AxesSubplot at 0x2545b1bc730>



# Tổng kết

- Cài đặt thư viện pandas
- Các khái niệm cơ bản
- Các thao tác với trường dữ liệu

# Tài liệu tham khảo

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