

# LAB 00: REVIEW PYTHON

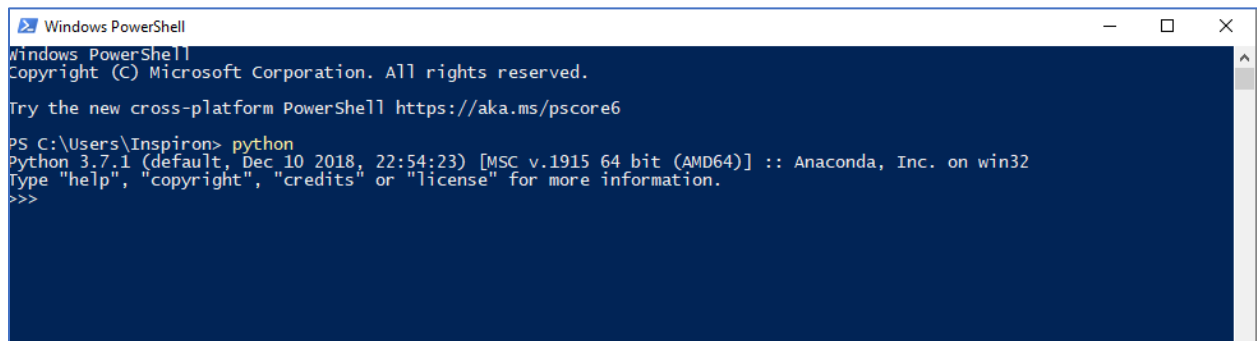
Giảng viên: thầy Bùi Tiến Lên

Trợ giảng: thầy Lê Ngọc Thành

Họ và tên: Huỳnh Minh Huấn

MSSV: 1612858

## Cài đặt Anaconda + jupyter và kiểm tra thử



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

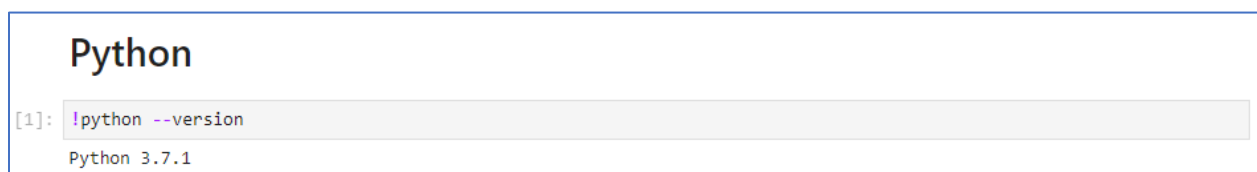
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Inspiron> python
Python 3.7.1 (default, Dec 10 2018, 22:54:23) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Hình 1 Kiểm tra cài đặt

## Python

### Python version



```
Python

[1]: !python --version
Python 3.7.1
```

Hình 2 Xem version python

### Basic data types

- Primitive

## Basic data types

```
[2]: x = 3
[3]: type(x)
[3]: int
[4]: print(x)
      print(x + 1, x - 1)
      print(x * 2, x**2)
      x += 1
      print(x)
      x *= 2
      print(x)
```

```
3
4 2
6 9
4
8
```

Hình 3 kiểu dữ liệu int

```
[5]: y = 2.5
      print(type(y))
      print(y, y + 1, y * 2, y**2)
<class 'float'>
2.5 3.5 5.0 6.25
```

Hình 4 kiểu dữ liệu float

```
[7]: t, f = True, False
      print(type(t))
      print(t and f) # AND
      print(t or f) # OR
      print(t != f) # XOR
      print(not t) # NOT
<class 'bool'>
False
True
True
False
```

Hình 5 string

- string

```
[8]: s1 = 'hello'
      s2 = 'world'
      print(s1, len(s1))
      hw = s1 + ' ' + s2
      print(hw)
      hw12 = '%s %s %d' % (s1, s2, 12) # sprintf style string formatting
      print(hw12) # prints "hello world 12"

hello 5
hello world
hello world 12

[9]: print(s1.capitalize())
      print(s1.upper())
      print(s1.lower())
      print(s1.rjust(8))
      print(s1.center(7))
      print(s1.replace('l','ell'))
      print(' world '.strip())

Hello
HELLO
hello
    hello
    hello
he(ell)(ell)o
world
```

Hình 6 Các thao tác với string

## Containers

- Lists

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

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

Hình 7 các phương thức cơ bản với list

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

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

Hình 8 slicing list

```
[26]: animals = ['cat', 'dog', 'pig', 'dolphin']
      for animal in animals:
          print(animal)
```

```
cat
dog
pig
dolphin
```

```
[27]: for idx, animal in enumerate(animals):
      print('%d: %s' % (idx + 1, animal))
```

```
#1: cat
#2: dog
#3: pig
#4: dolphin
```

Hình 9 vòng lặp và enumerate với list

```
[28]: nums = [0, 1, 2, 3, 4]
      squares = [x ** 2 for x in nums]
      print(squares) # Prints [0, 1, 4, 9, 16]
```

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

```
[29]: nums = [0, 1, 2, 3, 4]
      even_squares = [x ** 2 for x in nums if x % 2 == 0]
      print(even_squares) # Prints "[0, 4, 16]"
```

```
[0, 4, 16]
```

Hình 10 List comprehensions

- Dictionaries

```
[30]: d = {'cat': 'cute', 'dog': 'furry'}
      print(d['cat'])
      print('cat' in d)
      d['fish'] = 'wet'
      print(d)
      print(d.get('monkey', 'N/A')) # Get an element with a default; prints "N/A"
      print(d.get('fish', 'N/A'))   # Get an element with a default; prints "wet"
      del d['fish']                  # Remove an element from a dictionary
      print(d.get('fish', 'N/A'))    # "fish" is no longer a key; prints "N/A"
```

```
cute
True
{'cat': 'cute', 'dog': 'furry', 'fish': 'wet'}
N/A
wet
N/A
```

Hình 11 Khởi tạo dictionary và một số phương thức

```
[31]: for animal in d:
      print(animal, d[animal])
```

```
cat cute
dog furry
```

```
[32]: d = {'person': 2, 'cat': 4, 'spider': 8}
      for animal, legs in d.items():
          print('A %s has %d legs' % (animal, legs))
```

```
A person has 2 legs
A cat has 4 legs
A spider has 8 legs
```

Hình 12 Lặp trên dictionary

- Sets

```
[34]: animals = {'cat', 'dog', 'pig', 'cat'}
print(animals)
print('cat' in animals) # Check if an element is in a set; prints "True"
print('fish' in animals) # prints "False"
animals.add('fish') # Add an element to a set
print('fish' in animals) # Prints "True"
print(len(animals)) # Number of elements in a set; prints "3"
animals.add('cat') # Adding an element that is already in the set does nothing
print(len(animals)) # Prints "3"
animals.remove('cat') # Remove an element from a set
print(len(animals)) # Prints "2"

{'pig', 'cat', 'dog'}
True
False
True
4
4
3
```

```
[35]: for idx, animal in enumerate(animals):
      print('#%d: %s' % (idx + 1, animal))

#1: pig
#2: fish
#3: dog
```

- Tuples

```
[36]: d = {(x, x + 1): x for x in range(10)} # Create a dictionary with tuple keys
t = (5, 6) # Create a tuple
print(type(t))
print(d[t])
print(d[(4, 5)])

<class 'tuple'>
5
4
```

## Function

```
Function

[10]: def sign(x):
      if x > 0:
          return 'positive'
      elif x == 0:
          return 'zero'
      else:
          return 'negative'

[11]: for x in [-1, 0, 1]:
      print(sign(x))

negative
zero
positive
```

## Numpy

- Arrays

```
[37]: import numpy as np

[38]: 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])

      <class 'numpy.ndarray'>
      (3,)
      1 2 3
      [5 2 3]
      (2, 3)
      1 2 4
```

Hình 13 numpy array cơ bản

```
[39]: a = np.zeros((2, 2))
      print(a)

      b = np.ones((1, 3))
      print(b)

      c = np.full((1, 3), 7)
      print(c)

      d = np.eye(2)
      print(d)

      e = np.random.random((2,2))
      print(e)

      [[0. 0.]
       [0. 0.]]
      [[1. 1. 1.]]
      [[7 7 7]]
      [[1. 0.]
       [0. 1.]]
      [[0.97691427 0.7768843 ]
       [0.54080315 0.25273469]]
```

Hình 14 các phương thức khởi tạo

- Array indexing

```
[62]: a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
      b = a[:2, 1:3]
      print(a[0, 1])
      b[0, 0] = 86
      print(a[0,1])

      2
      86
```

```
[64]: a = np.array([[1,2,3,4], [5,6,7,8], [9,10,11,12]])
row_r1 = a[1, :]
row_r2 = a[1:2, :]
print(row_r1, row_r1.shape)
print(row_r2, row_r2.shape)

col_r1 = a[:, 1]
col_r2 = a[:, 1:2]
print(col_r1, col_r1.shape)
print(col_r2, col_r2.shape)

[5 6 7 8] (4,)
[[5 6 7 8]] (1, 4)
[ 2  6 10] (3,)
[[ 2]
 [ 6]
 [10]] (3, 1)
```

Hình 15 indexing tạo dòng và cột

```
[78]: 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]]))

[1 4 5]
[1 4 5]
[2 2]
[2 2]
```

Hình 16 subarray

- Data types

```
[61]: 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)

int32
float64
int64
```

Hình 17 Các loại dữ liệu

- Array math

```
[67]: 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))
      print(np.sqrt(x))
```

```
[[ 6.  8.]
 [10. 12.]]
[[ 6.  8.]
 [10. 12.]]
[[-4. -4.]
 [-4. -4.]]
[[-4. -4.]
 [-4. -4.]]
```

Hình 18 Các phép tính cơ bản trên array

```
[68]: 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))

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

```
219
219
[29 67]
[29 67]
[[19 22]
 [43 50]]
[[19 22]
 [43 50]]
```

Hình 19 phép nhân ma trận

```
[69]: x = np.array([[1,2],[3,4]])

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

```
10
[4 6]
[3 7]
```

Hình 20 phép cộng theo cột tương ứng



```
[70]: x = np.array([[1,2], [3,4]])
      print(x)
      print(x.T)

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

      [[1 2]
       [3 4]]
      [[1 3]
       [2 4]]
      [1 2 3]
      [1 2 3]
```

Hình 21 phép chuyển vị

- Broadcasting

```
[73]: 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)

      [[ 2  2  4]
       [ 5  5  7]
       [ 8  8 10]
       [11 11 13]]
```

Hình 22 phương thức empty\_like tạo ma trận có dạng như ma trận ch trước

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

      [76]: array([[1, 0, 1],
                  [1, 0, 1],
                  [1, 0, 1],
                  [1, 0, 1]])
```

Hình 23 phương thức tile

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

      [77]: array([[ 2,  2,  4],
                  [ 5,  5,  7],
                  [ 8,  8, 10],
                  [11, 11, 13]])
```

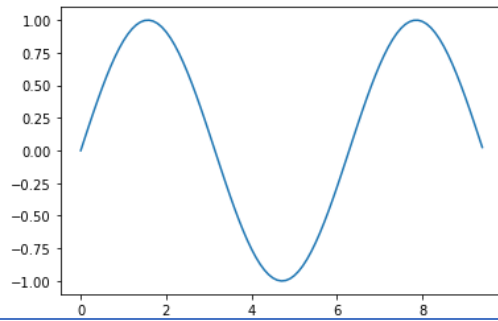
Hình 24 broadcast

## Matplotlib

```
[40]: import matplotlib.pyplot as plt

[41]: # Compute the x and y coordinates for points on a sine curve
x = np.arange(0, 3 * np.pi, 0.1)
y = np.sin(x)

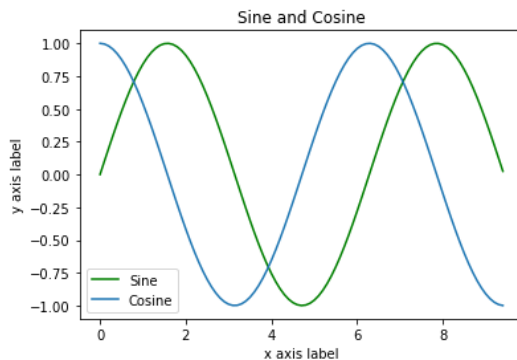
# Plot the points using matplotlib
plt.plot(x, y)
plt.show() # You must call plt.show() to make graphics appear.
```



Hình 25 Vẽ đồ thị

```
[43]: # Compute the x and y coordinates for points on sine and cosine curves
x = np.arange(0, 3 * np.pi, 0.1)
y_sin = np.sin(x)
y_cos = np.cos(x)

# Plot the points using matplotlib
plt.plot(x, y_sin, color='green')
plt.plot(x, y_cos)
plt.xlabel('x axis label')
plt.ylabel('y axis label')
plt.title('Sine and Cosine')
plt.legend(['Sine', 'Cosine'])
plt.show()
```



Hình 26 Vẽ nhiều đồ thị trên cùng một plot

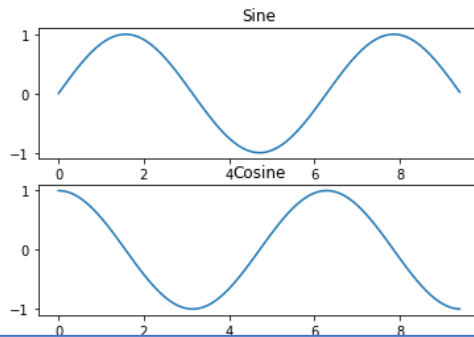
```
[49]: x = np.arange(0, 3 * np.pi, 0.1)
      y_sin = np.sin(x)
      y_cos = np.cos(x)

      plt.subplot(2, 1, 1)

      plt.plot(x, y_sin)
      plt.title('Sine')

      plt.subplot(2, 1, 2)
      plt.plot(x, y_cos)
      plt.title('Cosine')

      plt.show()
```



Hình 27 Dùng subplot vẽ nhiều đồ thị

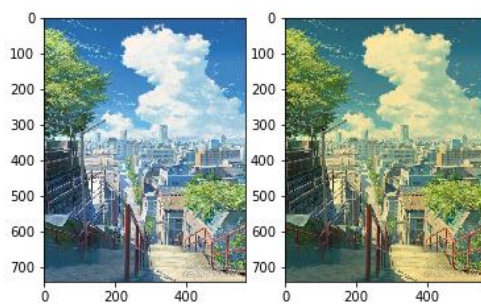
```
[59]: from imageio import imread

[60]: img = imread('image.jpg')
      img_tinted = img * [1, 0.95, 0.7]

      plt.subplot(1, 2, 1)
      plt.imshow(img)

      plt.subplot(1, 2, 2)
      plt.imshow(np.uint8(img_tinted))

      plt.show()
```



Hình 28 hiển thị hình ảnh