

Python

FIC - Ta Dang Khoa

Python cơ bản

- Các kiểu dữ liệu
- Giải thuật cơ bản
- Class, function

Python cơ bản

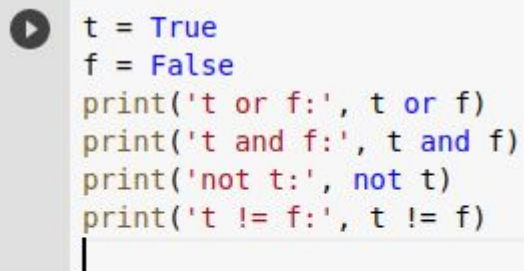
Các kiểu dữ liệu: Integer và float

```
[1] x = 6
    print(type(x))
    print(x, x + 9, x * 2, x ** 2)
    x += 1
    print(x)
    x *= 2
    print(x)
```

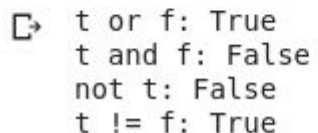
```
↳ <class 'int'>
6 15 12 36
7
14
```

Python cơ bản

Các kiểu dữ liệu: Boolean

A code editor window with a play button icon on the left. It contains Python code that defines two Boolean variables, t and f, and then prints the results of several Boolean operations: 't or f', 't and f', 'not t', and 't != f'.

```
t = True
f = False
print('t or f:', t or f)
print('t and f:', t and f)
print('not t:', not t)
print('t != f:', t != f)
|
```

The output of the Python code, showing the results of the Boolean operations. It is displayed as a list of four lines, each preceded by a small icon of a document with a checkmark.

```
t or f: True
t and f: False
not t: False
t != f: True
```

Python cơ bản

Các kiểu dữ liệu: String

```
▶ f = 'FPT'
  i = 'Innovation'
  c = 'Club'
  print(f, i, c)
  print('-'.join([f, i, c]))
  print('{1} {0} {2}, {3:0.2f}'.format(i, f, c, 111.1234))
  x = '    xxx yy zz    '
  print("'" + x.replace('x', 'a') + "'")
  print("'" + x.strip() + "'")
```

```
↳ FPT Innovation Club
   FPT-Innovation-Club
   FPT Innovation Club, 111.12
   '    aaa yy zz    '
   'xxx yy zz'
```

Python cơ bản

Các kiểu dữ liệu: List

```
l = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
print(l[1:5])
print(l[6:])
print(l[-1])
l[2:4] = ['a', 'b']
print(l)
print("=" * 40)

l.append('z')
print(l)
print('pop:', l.pop())
print(l)
```

```
[1, 2, 3, 4]
[6, 7, 8, 9]
9
[0, 1, 'a', 'b', 4, 5, 6, 7, 8, 9]
=====
[0, 1, 'a', 'b', 4, 5, 6, 7, 8, 9, 'z']
pop: z
[0, 1, 'a', 'b', 4, 5, 6, 7, 8, 9]
```

Python cơ bản

Các kiểu dữ liệu: Dictionary

```
[31] d = {'room':'607', 'time':'7h15'}  
      print(d['room'])  
      print(d.get('a'), 1)  
      d['a'] = 1  
      print(d)
```

```
↳ 607  
   None 1  
   {'room': '607', 'time': '7h15', 'a': 1}
```

Python cơ bản

Các kiểu dữ liệu: Set và Tuple

```
▶ set_ex = {'cat', 'dog'}  
tuple_ex = ('cat', 'dog')  
  
try:  
    tuple_ex[1] = '1'  
except:  
    print('error')
```

📄 error

Python cơ bản

Giải thuật cơ bản: if, else, for, while

```
[34] for i in range(10):  
    if i < 5:  
        print(i, '< 5')  
    else:  
        print(i, '>= 5')
```

```
↳ 0 < 5  
   1 < 5  
   2 < 5  
   3 < 5  
   4 < 5  
   5 >= 5  
   6 >= 5  
   7 >= 5  
   8 >= 5  
   9 >= 5
```

```
[35] i = 0  
    while True:  
        print(i)  
        i += 1  
        if i == 5:  
            break
```

```
↳ 0  
   1  
   2  
   3  
   4
```

Python cơ bản

Function, Class

```
▶ def plus(a, b):  
    print(a + b)  
    return a + b
```

```
x = plus(1, 2)
```

```
↳ 3  
   3
```

```
▶ class FIC():  
    def __init__(self, a, b):  
        self.a = a  
        self.b = b  
        print('(a, b) = ({}, {})'.format(a, b))
```

```
    def plus(self):  
        print('Sum:', self.a + self.b)  
        return self.a + self.b
```

```
a = FIC(1, 2)  
x = a.plus()
```

```
↳ (a, b) = (1, 2)  
   Sum: 3
```

Numpy cơ bản

- Numpy array
- Numpy indexing
- Các phép toán cơ bản

Numpy cơ bản

Numpy array

```
[3] # Import thư viện
import numpy as np

# array
array_ex = np.array([1, 2, 3])
print(array_ex.shape)
print(array_ex[0], array_ex[1])
print('-'*20)

# matrix
matrix_ex = np.array([[1, 2, 3], [4, 5, 6]])
print(matrix_ex.shape)
print(matrix_ex)
```

```
↳ (3,)
   1 2
   -----
   (2, 3)
   [[1 2 3]
    [4 5 6]]
```

Numpy cơ bản

Numpy array

```
▶ # Một số mảng đặc biệt
a = np.zeros((2, 3))
print(a)
a = np.ones((2, 3))
print(a)
a = np.full((2, 3), 69)
print(a)
a = np.random.random((2, 3))
print(a)
a = np.eye(5)
print(a)
```

```
↳ [[0. 0. 0.]
    [0. 0. 0.]]
[[1. 1. 1.]
 [1. 1. 1.]]
[[69 69 69]
 [69 69 69]]
[[0.13702422 0.0750927 0.00524166]
 [0.15722961 0.44394305 0.57991953]]
[[1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0.]
 [0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0.]
 [0. 0. 0. 0. 1.]]
```

Numpy cơ bản

Array indexing

```
▶ a = np.zeros((5, 5))  
  b = a[3:, 3:]  
  print(a)  
  print(b)  
  print('-'*20)  
  
  b[0, 0] = 1  
  print(b)  
  print(a)
```

```
↳   
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]]  
[[0. 0.]  
 [0. 0.]]  
-----  
[[1. 0.]  
 [0. 0.]]  
[[0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 0. 0.]  
 [0. 0. 0. 1. 0.]  
 [0. 0. 0. 0. 0.]]
```

Numpy cơ bản

Array indexing

```
[5] a = np.array([
                [1, 2, 3],
                [1, 2, 3],
                [1, 2, 3]
            ])

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

```
↳ [[False False  True]
    [False False  True]
    [False False  True]]
[3 3 3]
```

```
[17] a = np.array([
                [1, 2, 3],
                [1, 2, 3],
                [1, 2, 3]
            ])

b = np.where(a > 2, a, a*10)
print(b)
```

```
↳ [[10 20  3]
    [10 20  3]
    [10 20  3]]
```

Numpy cơ bản

Phép toán cơ bản

```
▶ A = np.array([  
    [1, 2],  
    [3, 4]  
])  
B = np.array([  
    [5, 6],  
    [7, 8]  
])
```

```
print(A + B)  
print(A - B)  
print(A * B)  
print(A / B)
```

```
↵ [[ 6  8]  
   [10 12]]  
   [[-4 -4]  
   [-4 -4]]  
   [[ 5 12]  
   [21 32]]  
   [[0.2      0.33333333]  
    [0.42857143 0.5      ]]
```


Numpy cơ bản

Phép toán cơ bản

```
▶ A = np.array([
    [1, 2],
    [3, 4]
])
B = np.array([
    [5, 6],
    [7, 8]
])

print(A.dot(B))
print(B.dot(A))
```

```
↳ [[19 22]
    [43 50]]
   [[23 34]
    [31 46]]
```

```
▶ A = np.array([
    [1, 2, 3, 4],
    [5, 6, 7, 8]
])

print(A.T)
print('-'*20)
print(A.reshape(2,2,2))
```

```
↳ [[1 5]
    [2 6]
    [3 7]
    [4 8]]
-----
   [[[1 2]
     [3 4]]
    [[5 6]
     [7 8]]]
```

Numpy cơ bản

Phép toán cơ bản: Broadcasting

```
▶ A = np.array([1, 2, 3])  
print(A + 1)
```

```
↳ [2 3 4]
```

```
▶ A = np.asarray([  
    [4, 5, 6],  
    [7, 8, 9]  
])  
  
B = np.asarray([  
    [4],  
    [7]  
])  
  
print(A * B)
```

```
↳ [[16 20 24]  
    [49 56 63]]
```

Bài tập