

|   |   |
|---|---|
| <div>Today for Tomorrow</div> <div>A D</div> <div>A I</div>   | <b>PHÒNG THÍ NGHIỆM THỰC HÀNH ỨNG DỤNG</b><br><b>KHOA HỌC DỮ LIỆU VÀ TRÍ TUỆ NHÂN TẠO</b><br>Applied Data Science and Artificial Intelligence Lab |
|  <div data-bbox="791 651 1268 792"> <b>LẬP TRÌNH PYTHON CHO KHOA HỌC DỮ LIỆU</b><br/> <b>PYTHON FOR DATA SCIENCE</b> </div> <div data-bbox="930 866 1112 896">Hà Nội ★ 11.2020</div> |   |
| Nothing is faster than Innovation together  |   |

1

|  |                             |   |
|--|-----------------------------|---|
| <div>Today for Tomorrow</div> <div>A D</div> <div>A I</div>  | <b>Thông tin giảng viên</b> |   |
|                             | <b>TS. TẠ QUANG CHIẾU</b>   | <div data-bbox="675 1294 888 1332"><b>Vị trí &amp; Đào tạo</b></div> <ul style="list-style-type: none"> <li>- Trưởng bộ môn Hệ thống thông tin và Tri thức, Khoa Công nghệ Thông tin,</li> <li>- Thành viên HĐQT Hitelli - hitelli.com</li> <li>- PhD &amp; Post-doc in Polytech Tours, France</li> </ul> <div data-bbox="675 1494 1091 1532"><b>Lĩnh vực giảng dạy, nghiên cứu</b></div> <ul style="list-style-type: none"> <li>- Big data, Data science, machine learning, AI</li> <li>- Các phương pháp cho bài toán lập kế hoạch</li> </ul> <div data-bbox="675 1626 799 1659"><b>Contact:</b></div> <div data-bbox="675 1673 1007 1704">(E): <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a></div> <div data-bbox="675 1713 882 1742">(M): 0913 522 275</div> |
| Tạ Quang Chiếu - (E): <a href="mailto:quangchieu.ta@gmail.com">quangchieu.ta@gmail.com</a> - (M): 0913 522 275 |                             | <b>2/55</b>   |

2

| PHẦN 1: LẬP TRÌNH PYTHON |                                |       |
|--------------------------|--------------------------------|-------|
|                          | NỘI DUNG:                      | TRANG |
| 1                        | BIẾN TRONG PYTHON              |       |
| 2                        | CÁC KIỂU DỮ LIỆU và TOÁN TỬ    |       |
| 3                        | CẤU TRÚC ĐIỀU KHIỂN, VÒNG LẶP  |       |
| 4                        | HÀM TRONG PYTHON               |       |
| 5                        | LÀM VIỆC VỚI FILE TRONG PYTHON |       |
| 6                        | MODULE, XỬ LÝ NGOẠI LỆ         |       |
| 7                        | LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG      |       |

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

3/55

3

| BIẾN TRONG PYTHON  |  |  |
|--|--|--|
|  |  |  |

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

4/55

4

A

D

Biến (VARIABLE)

- Trong lập trình, biến (variable) là **tên của một vùng trong bộ nhớ RAM**, được sử dụng để **lưu trữ thông tin**.
- Biến giống như **một chiếc hộp** có thể giúp chúng ta lưu trữ dữ liệu. Ta có thể gán thông tin cho một biến, và có thể lấy thông tin đó ra để sử dụng.
- Khi một biến được khai báo, một vùng trong bộ nhớ sẽ dành cho các biến.
- Biến là một thứ **cực kì quan trọng trong lập trình** mà không thể thiếu trong bất cứ chương trình lớn, nhỏ nào.

```
In [3]: message = 'CHXH' #tạo biến message nhận giá trị là 'CHXH'
        n = 17 # tạo biến n nhận giá trị là 17
        pi = 3.1415926535897931 # tạo biến pi nhận giá trị là 3.14...
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

5/55

5

A

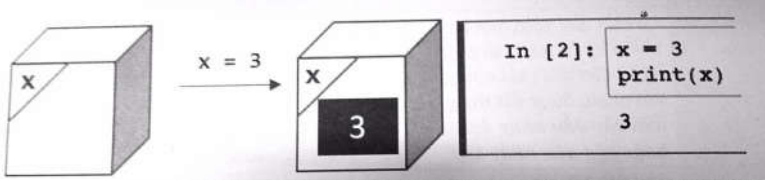
D

Biến (VARIABLE)

## Khai báo Biến (VARIABLE)

Cú pháp: **tên\_biến = <giá trị>**

```
In [3]: message = 'CHXH' #tạo biến message nhận giá trị là 'CHXH'
        n = 17 # tạo biến n nhận giá trị là 17
        pi = 3.1415926535897931 # tạo biến pi nhận giá trị là 3.14...
```



Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

6/55

6

A D  
A I

Biến (VARIABLE)

## Vai trò của Biến

- Biến giúp chúng ta lưu trữ các dữ liệu và cho phép chúng ta lấy các dữ liệu của chúng để tính toán được thuận tiện và chính xác hơn.
- Khi muốn lấy dữ liệu từ một nguồn không cố định từ ngoài chương trình (ví dụ: dữ liệu người dùng nhập vào, dữ liệu load từ file ...) ta cần sử dụng biến số để lưu thông tin của các dữ liệu này

```
In [4]: 52348252408 + 523482034
Out[4]: 52871734442

In [5]: 52871734442 + 545354645577
Out[5]: 598226380019

In [6]: a = 52348252408
        b = 523482034
        c = 545354645577
        d = a + b + c
        print(d)
598226380019
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
7/55

7

A D  
A I

Biến (VARIABLE)

## Quy tắc đặt tên Biến

- Một số quy tắc bắt buộc khi đặt tên biến, nếu vi phạm sẽ bị báo lỗi:
  - Tên biến chỉ được chứa:
    - Các ký tự chữ cái (a → z, A → Z)
    - Các ký tự số (0 → 9)
    - Dấu gạch dưới ( \_ )
  - Tên biến không được bắt đầu bằng ký tự số
  - Phân biệt chữ hoa chữ thường (a # A)
  - Tên biến không được chứa các ký tự đặc biệt như "@,&..." hay các keywords trong Python.
  - Không được sử dụng dấu cách khi đặt tên biến
  - Tên biến nên được đặt theo một tên có ý nghĩa, có tính gợi nhớ.
  - Trường hợp tên biến gồm nhiều tiếng ghép lại, nên sử dụng dấu gạch nối.

```
In [9]: 76ab = 'asd'
File "cipython-input-9-d8355cbf35f4", line 1
76ab = 'asd'
      ^
SyntaxError: invalid syntax

In [8]: then@ = 1000000
File "cipython-input-8-0de45931b145", line 1
then@ = 1000000
      ^
SyntaxError: invalid syntax

In [10]: class = 'Advanced Theoretical Herpetology'
File "cipython-input-10-9630266a2671", line 1
class = 'Advanced Theoretical Herpetology'
      ^
SyntaxError: invalid syntax
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
8/55

8

A
D
A
I
**Biến (VARIABLE)**

## Quy tắc đặt tên Biến

```

[help> keywords

Here is a list of the Python keywords.  Enter any keyword to get more help.

False          def          if          raise
None           del          import       return
True           elif         in          try
and            else        is          while
as            except      lambda      with
assert        finally   nonlocal    yield
break         for        not
class         from       or
continue      global    pass

help>


```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
9/55

9

A
D
A
I

# DATA TYPE IN PYTHON



Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
10/55

10

A

D

2.1. Numeric data types (1)

• Python hỗ trợ nhiều kiểu dữ liệu khác nhau

| Type                      | Example   |
|---------------------------|---|
| • Numeric: Integer, Float | <code>x = 10   x = 1.0</code>                                   |
| • String                  | <code>x= 'Mike'</code>  |
| • Boolean                 | <code>y = True   x = False</code>                               |
| • List                    | <code>my_list = [10, 20, 30]</code>                             |
| • Tuple                   | <code>my_tuple = ('Brett', 'Cisco', 'Cary', 2015)</code>        |
| • Dictionary              | <code>my_dict = {"one":1, "two":2}</code>                       |
| • Lists in Lists          | <code>my_list2=[[10,20,30], ['Cisco Live', 'May', 2016]]</code> |

**Python  
Data types**

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

11/55

11

A

D

2.1. Numeric data types (2)

**a) Kiểu dữ liệu số**

Bao gồm 2 kiểu:

- Số nguyên – int (ví dụ: 1, 4, 111)
- Số thực – float (ví dụ: 1.1, 3.23, 11.01)

In [13]:

```
a=11
b=12
c=a/b
print(c)
```

0.9166666666666666

In [14]:

```
a=11
b=12
c=int(a/b)
print(c)
```

0

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

12/55

12

A
D
A
I
2.1. Numeric data types (3)

## Các phép toán với số

```

1 a = 10
2 b = 8
3 #-----
4 tong = a + b           # Tổng của hai số (+)
5 hieu = a - b           # Hiệu của hai số (-)
6 tích = a*b             # Tích của hai số (*)
7 thuong = a/b           # Thương của hai số (/)
8 thuong_nguyen = a//b   # Phép chia Lấy phần nguyên (//)
9 thuong_du = a % b      # Phép chia Lấy phần dư (%)
10 mu = a**b             # Tính giá trị a Lũy thừa b (**)

```

- Học viên nhập code và đọc kết quả của các phép toán ở trên!

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
13/55

13

A
D
A
I
2.1. Numeric data types (4)

## Kiểm tra kiểu dữ liệu của biến

- Biến trong Python rất linh hoạt, có thể chứa giá trị thuộc nhiều kiểu dữ liệu khác nhau.

Cú pháp: **type(biến)**  
sẽ giúp ta biết được kiểu dữ liệu của biến

```

1 #Kiểm tra kiểu dữ liệu của biến
2 x = 1985
3 y = 3.1415926535
4 z = 'Đại học Mo Địa chất'
5 n = [5, 7, 9, 8]
6 b = True
7 #-----
8 print('Kiểu dữ liệu biến x: ', type(x))
9 print('Kiểu dữ liệu biến y: ', type(y))
10 print('Kiểu dữ liệu biến z: ', type(z))
11 print('Kiểu dữ liệu biến n: ', type(n))
12 print('Kiểu dữ liệu biến b: ', type(b))

```

```

Kiểu dữ liệu biến x: <class 'int'>
Kiểu dữ liệu biến y: <class 'float'>
Kiểu dữ liệu biến z: <class 'str'>
Kiểu dữ liệu biến n: <class 'list'>
Kiểu dữ liệu biến b: <class 'bool'>

```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
14/55

14

A

D

2.2 String type (1)

- Bên cạnh dữ liệu kiểu số, Python cũng có thể thao tác với chuỗi ký tự (string)
- Chuỗi có thể được để trong dấu nháy đơn ('...') hoặc kép ("...")

```

1 s='Sinh viên Việt Nam chịu khó, thông minh'
2 print(s)
3 print(type(s))
4 print("Ta Quang Chieu")
5 print('1 2 3 4 5 6')
6 print("@ # % $ & *")
7 name="Chieu"
8 print("Xin chào", name)
```

```

Sinh viên Việt Nam chịu khó, thông minh
<class 'str'>
Ta Quang Chieu
1 2 3 4 5 6
@ # % $ & *
Xin chào Chieu
```

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

15/55

15

A

D

2.2 String type (2)

- Cần **phân biệt** kỹ giữa **kiểu dữ liệu chuỗi và số**:

```

1 #Kiểu dữ liệu số
2 a=123
3 print('Số = ', a, type(a))
4
5 #Kiểu dữ liệu chuỗi
6 s="123"
7 print('Chuỗi số = ', s, type(s))
```

```

Số = 123 <class 'int'>
Chuỗi số = 123 <class 'str'>
```

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

16/55

16



A D I I

## 2.2 String type (3)

### Truy cập phần tử trong chuỗi (1)

- Các chuỗi có thể được lập chỉ mục với ký tự đầu tiên được đánh số 0.
- Chỉ số cũng có thể là số âm, bắt đầu đếm từ bên phải

|     |     |     |     |     |    |    |    |    |    |    |    |    |    |
|-----|-----|-----|-----|-----|----|----|----|----|----|----|----|----|----|
| T   | a   |     | Q   | u   | a  | n  | g  |    | C  | h  | i  | e  | u  |
| 0   | 1   | 2   | 3   | 4   | 5  | 6  | 7  | 8  | 9  | 10 | 11 | 12 | 13 |
| -14 | -13 | -12 | -11 | -10 | -9 | -8 | -7 | -6 | -5 | -4 | -3 | -2 | -1 |

```

1 name="Ta Quang Chieu"
2 print(name, len(name))
3 print(name[0])
4 print(name[-5])

Ta Quang Chieu 14
T
C
```

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
17/55

17

A D I I

## 2.2 String type (4)

### Truy cập phần tử trong chuỗi (2)

- Ngoài việc lấy từng ký tự thì chúng ta có thể lấy nhiều phần tử cùng lúc:
- Slicing: dùng để lấy chuỗi ký tự từ i đến j-1, ký hiệu s[i:j]

```

1 s='Sinh viên Việt Nam chịu khó, thông minh'
2 print(s)
3 print(type(s))
4 print(s[0])
5 print(s[0:4])
6 print(s[:9])
7 print(s[10])
8 print(s[15])
9 print(s[10:18])

Sinh viên Việt Nam chịu khó, thông minh
<class 'str'>
S
Sinh
Sinh viên
V
N
Việt Nam
```

Tạ Quang Chiếu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
18/55

18

A

D

A

I

## 2.2 String type (5)

### Truy cập phần tử trong chuỗi (3)

- Stride: dùng để lấy chuỗi ký tự từ l đến j-1 và mỗi ký tự cách nhau bước nhảy k, ký hiệu s[i:j:k]

```

1 s='Sinh viên Việt Nam'
2 print(len(s))
3 print(s[:5])
4 print(s[10:17:5])

```

18

SvVN

VN

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

19/55

19

A

D

A

I

## 2.2 String type (6)

### String operations

- Cho biết số ký tự của chuỗi: **len(st)**

```

1 s='Sinh viên Việt Nam chịu khó, thông minh'
2 print('Độ dài của chuỗi: ',len(s))
3 print('Độ dài chuỗi Việt Nam: ',len(s[10:18]))

```

Độ dài của chuỗi: 39

Độ dài chuỗi Việt Nam: 8

- Chuyển chuỗi st thành chữ hoa (upper) – chữ thường (lower):  
**st.upper() – st.lower()**

```

1 s='Sinh viên Việt Nam chịu khó, thông minh'
2 print('Chữ hoa: ',s.upper())
3 print('Chữ thường: ',s.lower())

```

Chữ hoa: SINH VIÊN VIỆT NAM CHỊU KHÓ, THÔNG MINH

Chữ thường: sinh viên việt nam chịu khó, thông minh

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

20/55

20

A

D

A

I

## 2.2 String type (7)

### String type - String operations

- Loại bỏ khoảng trắng ở đầu và cuối chuỗi: **st.strip()**

```

1 #Loại bỏ khoảng trắng ở đầu và cuối chuỗi: st.strip()
2 st=' Tôi yêu Việt Nam '
3 st.strip()

'Tôi yêu Việt Nam'
```

- Thay thế một chuỗi s thành s1 trong chuỗi st: **st.replace(s,s1)**

```

1 #Thay thế một chuỗi s thành s1 trong chuỗi st: st.replace(s,s1)
2 s='Sinh viên Việt Nam chịu khó, thông minh'
3 s_new=s.replace('chịu khó', 'chăm chỉ')
4 print(s_new)

Sinh viên Việt Nam chăm chỉ, thông minh
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
21/55

21

A

D

A

I

## 2.2 String type (8)

### String type - String operations

Để kiểm tra một chuỗi con có trong chuỗi st hay không? hãy dùng từ khóa **in**

```

1 #Để kiểm tra một chuỗi con có trong chuỗi st hay không? hãy dùng từ khóa in
2 #Trả về giá trị True nếu s1 có trong s, ngược lại trả về giá trị False
3 s='Sinh viên Việt Nam chịu khó, thông minh'
4 s1='chịu khó' in s
5 s2='chăm chỉ' in s
6 print("Kết quả kiểm tra s1 có trong s? ", s1)
7 print("Kết quả kiểm tra s2 có trong s? ", s2)

Kết quả kiểm tra s1 có trong s? True
Kết quả kiểm tra s2 có trong s? False
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
22/55

22

A
D
A
I

## 2.2 String type (9)

### String type - String operations

- Tách một chuỗi thành một danh sách (mảng 1 chiều):  
**`string.split('seperator')`**

```

1 #Tách một chuỗi thành một danh sách (mảng 1 chiều): string.split('seperator')
2 s='Tôi yêu Việt Nam'
3 #Tách chuỗi tại vị trí dấu cách
4 list_s=s.split()
5 print(list_s)

```

`['Tôi', 'yêu', 'Việt', 'Nam']`

```

1 #Tách một chuỗi thành một danh sách (mảng 1 chiều): string.split('seperator')
2 s='Sinh,viên,Việt,Nam,chịu,khó'
3 #Tách chuỗi tại vị trí dấu ','
4 list_s=s.split(',')
5 print(list_s)

```

`['Sinh,viên,Việt,Nam,chịu,khó']`

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
23/55

23

A
D
A
I

## 2.2 String type (10)

### String type - String operations

- Concatenate: Dùng để cộng 2 chuỗi lại với nhau

```

1 #Concatenate: Dùng để cộng 2 chuỗi lại với nhau
2 name=' Ta Quang Chieu'
3 s1 = name + ' is teaching for his student'
4 s2=2*name
5 print(s1)
6 print(s2)

```

Ta Quang Chieu is teaching for his student  
Ta Quang Chieu Ta Quang Chieu

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
24/55

24

A

D

A

I

## 2.2 String type (11)

### Phương thức kiểm tra với chuỗi

- st.isalpha()** : Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự thuộc bảng chữ cái hay không (a-z, A-Z)?
- st.isdigit()** : Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự chữ số hay không (0-9)?
- st.isalnum()** : Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự chữ cái (a-z, A-Z) và ký tự số (0-9) hay không?
- st.isupper()** : Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự chữ được viết hoa hay không (A-Z)?
- st.islower()** : Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự chữ được viết hoa hay không (a-z)?

```

1 #st.isalpha():Kiểm tra chuỗi ký tự st có phải chỉ chứa ký tự
2 #thuộc bảng chữ cái hay không (a-z, A-Z)?
3 alpha='ViệtNam'
4 print('Chuỗi st chỉ chứa các chữ cái?',alpha.isalpha())
5 alpha1='Tôi yêu Việt Nam'
6 print('Chuỗi st chỉ chứa các chữ cái?',alpha1.isalpha())
```

Chuỗi st chỉ chứa các chữ cái? True  
Chuỗi st chỉ chứa các chữ cái? False

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

25/55

25

A

D

A

I

## BÀI TẬP THỰC HÀNH

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

26/55


26

A D I I

## Bài 1: Chia kẹo cho học sinh lớp 1

**Cô có n cái kẹo, có m học sinh trong lớp. Hãy giúp cô chia đều số kẹo cho tất cả học sinh.**

- **Yêu cầu:**
  - Nhập vào số kẹo của cô – N
  - Nhập vào số học sinh trong lớp – M
- **Cho biết:**
  - Mỗi học sinh được nhận bao nhiêu cái kẹo?
  - Cô còn lại bao nhiêu cái kẹo?



Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

27/55

27

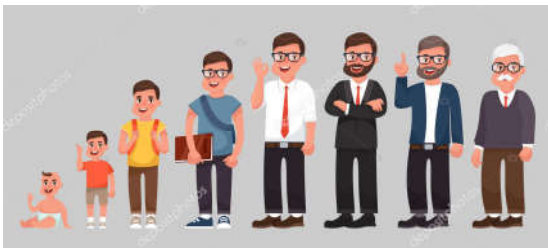
A D I I

## Bài 2: Tính tuổi

**Nhập vào tên và năm sinh của một người, tính tuổi của người đó và hiển thị ra màn hình thông báo**

- **Yêu cầu:**
  - Nhập vào họ tên: mai phuong thuy
  - Nhập vào năm sinh: 2000
- **Hiển thị thông báo kết quả:**

**Bạn “MAI PHƯƠNG THÚY” năm nay 19 tuổi!**



Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

28/55

28

A D  
A I

## Bài 3: Cho biết số thỏ trong rừng

Ban đầu trong rừng có một cặp thỏ, biết rằng cứ sau 1 tháng thì số lượng thỏ trong rừng tăng lên gấp đôi. Hỏi sau  $x$  tháng thì trong rừng có bao nhiêu con thỏ (Giả thiết các con thỏ không chết)

- Yêu cầu:**
- Nhập vào số tháng  $x$ : 3
- Hiển thị thông báo kết quả:**

Trong rừng có: 16 con thỏ



Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

29/55


29

A D  
A I

## Bài 4: Chuỗi văn bản

**“Nước Việt Nam là một, dân tộc Việt Nam là một. Sông có thể cạn núi có thể mòn, song chân lý ấy không bao giờ thay đổi. (HỒ CHÍ MINH, 1890 – 1969)”**

- Yêu cầu:**
- Cho biết số ký tự có trong đoạn văn trên.
- Cho biết trong đoạn có chứa từ nào sau đây không (Không phân biệt chữ hoa, chữ thường)?
  - “hồ chí minh”
  - “non sông”
- Tách đoạn văn thành các câu bởi dấu .
- Cho biết trong đoạn văn trên có ký tự nào khác ký tự chữ và số hay không?
- Thay thế các từ ‘Việt Nam’ bằng ‘VIỆT NAM’ trong đoạn văn trên.



Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

30/55

30

A

D

2.3 Tuples data type (1)

- Indexing
- Slicing
- Sorting
- Nested Tuples

▪ Tuples là kiểu dữ liệu có thể chứa nhiều biến có các kiểu dữ liệu khác nhau. Sử dụng dấu ngoặc ()

```

1  tpl=('tuples', 1,3.14,True)
2  print(tpl)
3  print('Kiểu dữ liệu: ',type(tpl))
4  print(tpl[0])
5  print('Độ dài của tuples:',len(tpl))
6  print(tpl[-1])

('tuples', 1, 3.14, True)
Kiểu dữ liệu:  <class 'tuple'>
tuples
Độ dài của tuples: 4
True
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

31/55

31

A

D

2.3 Tuples data type (2)

▪ Có thể cộng (+) 2 tuple

```

1  tpl=('tuples', 1,3.14,True)
2  print(type(tpl[0]))
3  print(type(tpl[1]))
4  print(type(tpl[2]))
5  print(type(tpl[3]))
6  tpl2=tpl+('add','xau',1,2,3)
7  print(tpl2)

<class 'str'>
<class 'int'>
<class 'float'>
<class 'bool'>
('tuples', 1, 3.14, True, 'add', 'xau', 1, 2, 3)
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

32/55

32



A

D

I

## 2.3 Tuples data type (3)

- Slicing dùng để lấy một số phần tử từ **i** đến **j-1**, kí hiệu là **tuples[i:j]**
- Nếu **i** không điền giá trị thì lấy giá trị **mặc định là 0**
- Nếu **j** không điền giá trị thì lấy giá trị **mặc định là length của tuple**

```

1  tpl=('tuples', 1,3.14,True)
2  print(tpl[:3])
3  print(tpl[2:])
4  print(tpl[1:3])
```

('tuples', 1, 3.14)  
(3.14, True)  
(1, 3.14)

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

33/55

33

A

D

I

## 2.3 Tuples data type (4)

- Tuples có thể được sắp xếp theo thứ tự nào đó thông qua hàm **sorted**

```

1  tpl=(3,-9,8,4,6)
2  print('Tuple trước khi sắp xếp:',tpl)
3  tpl=sorted(tpl)
4  print('Tuple sau khi sắp xếp:',tpl)
```

Tuple trước khi sắp xếp: (3, -9, 8, 4, 6)  
Tuple sau khi sắp xếp: [-9, 3, 4, 6, 8]

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

34/55

34

A
D
A
I

## 2.3 Tuples data type (5)

▪ Các phần tử của tuple có thể là một tuple khác. Người ta gọi là lồng nhau:

```

1  tpl=('tuples',(3,-9,8,4,6),(3.14, 5.12),(True, False))
2  print(tpl)
3  print(tpl[:3])
4  print(tpl[2:])
5  print(tpl[1:3])

```

```

('tuples', (3, -9, 8, 4, 6), (3.14, 5.12), (True, False))
('tuples', (3, -9, 8, 4, 6), (3.14, 5.12))
((3.14, 5.12), (True, False))
((3, -9, 8, 4, 6), (3.14, 5.12))

```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

35/55

35

A
D
A
I

## 2.4. List data type (1)

- Indexing
- List content
- List Operations
- Copy and Clone list

List là danh sách các đối tượng có thể có nhiều kiểu dữ liệu khác nhau (như số, logic, string, hay cũng có thể là list khác hoặc các cấu trúc dữ liệu khác) và sử dụng dấu ngoặc [] để biểu diễn.

```

1  #List type
2  lst=['số nguyên',3,5,'hello',True,3.14,False,1000]
3  print(lst)
4  print(len(lst))
5  print(lst[3])
6  print(type(lst[3]))

```

```

['số nguyên', 3, 5, 'hello', True, 3.14, False, 1000]
8
hello
<class 'str'>

```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

36/55

36

A

D

I

2.4. List data type (2)

- Giả sử ta có nhu cầu lưu danh sách tên tất cả học sinh trong lớp học. Nếu lớp học ít:
 

```
1 #khai báo danh sách học sinh
2 hoc_sinh1 = 'Lê Thùy Dung'
3 hoc_sinh2 = 'Trần Đức Hùng'
4 hoc_sinh3 = 'Nguyễn Lan Anh'
```
- Nhưng nếu trong lớp có hàng chục, hàng trăm học sinh thì điều đó thật sự rất khó khăn, khó kiểm soát, quản lý.
- Kiểu dữ liệu danh sách (list) ra đời giúp chúng ta có thể khai báo, lưu trữ, truy xuất một dãy các đối tượng.

'foo'

'bar'

'baz'

'qux'

'quux'

'corge'

012345

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

37/55

37

A

D

I

2.4. List data type (3)

- Khai báo list:
 

$\langle \text{tên\_danh\_sách} \rangle = [\langle \text{Phần tử 1} \rangle, \langle \text{Phần tử 2} \rangle, \dots, \langle \text{Phần tử n} \rangle]$

```
1 #KHAI BÁO DANH SÁCH LIST
2 #khai báo danh sách học sinh gồm các chuỗi tên của học sinh
3 hoc_sinh=['Lê Thùy Dung','Trần Đức Hùng','Nguyễn Lan Anh','Mai Phương Thúy']
4 print(hoc_sinh)
5
6 #khai báo danh sách điểm chu gồm các chuỗi ký tự
7 diem = ['A+','A','B+','B','C+','C','D+','D','F']
8 print(diem)
9
10 #khai báo danh sách gồm các số nguyên
11 list_so = [9,5,8,13,0,4,7,-9,11]
12 print(list_so)
13
14 #khai báo danh sách với nhiều kiểu dữ liệu khác nhau
15 person_info = ['Mary',1998,'Tokyo, Japan', 1.70, 65]
16 print(person_info)
17
18 #khai báo danh sách rỗng
19 list_rong = []
20 print(list_rong)
```

```
[ 'Lê Thùy Dung', 'Trần Đức Hùng', 'Nguyễn Lan Anh', 'Mai Phương Thúy' ]
[ 'A+', 'A', 'B+', 'B', 'C+', 'C', 'D+', 'D', 'F' ]
[ 9, 5, 8, 13, 0, 4, 7, -9, 11 ]
[ 'Mary', 1998, 'Tokyo, Japan', 1.7, 65 ]
[ ]
```

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

38/55

38

A D  
A I

## 2.4. List data type (4)

- Truy cập phần tử của danh sách

<tên\_danh\_sách> [<chỉ số>]

```

1 lst=['số nguyên',3,5,'hello',True,3.14,False,1000]
2 print(lst)
3 print(lst[0])
4 print(lst[2])
5 print(lst[1:6])
6 print(lst[:4])
7 print(lst[4:])
```

```

['số nguyên', 3, 5, 'hello', True, 3.14, False, 1000]
số nguyên
5
[3, 5, 'hello', True, 3.14]
['số nguyên', 3, 5, 'hello']
[True, 3.14, False, 1000]
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

39/55

39

A D  
A I

## 2.4. List data type (5)

### List operations

- Thay đổi giá trị phần tử trong list

<tên\_danh\_sách> [<chỉ số>] = <giá trị mới>

```

1 lst=['số nguyên',3,5,'hello',True,3.14,False,1000]
2 print('List ban đầu: ',lst)
3 lst[3]='Xin chào'
4 print('List thay đổi: ',lst)
```

```

List ban đầu:  ['số nguyên', 3, 5, 'hello', True, 3.14, False, 1000]
List thay đổi:  ['số nguyên', 3, 5, 'Xin chào', True, 3.14, False, 1000]
```

- Nối hai danh sách (+): kết quả trả về một danh sách mới.

```

1 #Nối hai danh sách (+)
2 lst1=[1,2,3]
3 lst2=[4,5,6,7]
4 lst3=lst1 + lst2
5 print('Kết quả nối 2 danh sách: ',lst3)
```

Kết quả nối 2 danh sách: [1, 2, 3, 4, 5, 6, 7]

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

40/55

40

A D  
A I

## 2.4. List data type (6)

### List operations

- **Kiểm tra một phần tử có thuộc danh sách hay không (in):** kết quả trả về một danh sách mới.

```

1  #Kiểm tra một phần tử có thuộc danh sách hay không (in):
2  lst=[2,5,6,7,9,1,0,3,]
3  print(lst)
4  #Kiểm tra phần tử 0 có thuộc danh sách
5  int_0 =0 in lst
6  print('Phần tử 0 có thuộc lst không?', int_0)
7
8  #Kiểm tra phần tử 8 có thuộc danh sách
9  int_8=8 in lst
10 print('Phần tử 8 có thuộc lst không?', int_8)

```

[2, 5, 6, 7, 9, 1, 0, 3]  
 Phần tử 0 có thuộc lst không? True  
 Phần tử 8 có thuộc lst không? False

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275 41/55

41

A D  
A I

## 2.4. List data type (7)

### List operations

- **Thêm một phần tử vào danh sách:** có hai cách
  - Thêm phần tử vào cuối danh sách:  
**<danh\_sách>.append(<giá trị>)**
  - Thêm phần tử vào vị trí bất kỳ trong danh sách:  
**<danh\_sách>.insert(<vị trí>,<giá trị>)**

```

1  lst=[2,5,6,7,9,1,0,3]
2  print('Danh sách ban đầu:',lst)
3
4  #Thêm phần tử vào cuối danh sách
5  lst.append(8)
6  print('Danh sách sau thêm vào cuối:',lst)
7
8  #Thêm vào vị trí thứ 2 giá trị 10
9  lst.insert(1,10)
10 print('Danh sách sau thêm vào 2 giá trị 10:',lst)

```

Danh sách ban đầu: [2, 5, 6, 7, 9, 1, 0, 3]  
 Danh sách sau thêm vào cuối: [2, 5, 6, 7, 9, 1, 0, 3, 8]  
 Danh sách sau thêm vào 2 giá trị 10: [2, 10, 5, 6, 7, 9, 1, 0, 3, 8]

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275 42/55

42

A D  
A I

## 2.4. List data type (8)

### List operations

- **Xóa phần tử khỏi danh sách:**
  - Xóa phần tử tại vị trí cuối cùng trong danh sách:  
`<danh_sách>.pop()`
  - Xóa phần tử tại vị trí xác định trong danh sách:  
`del <danh_sách>[<vị trí>]`
  - Xóa phần tử có giá trị xuất hiện đầu tiên trong danh sách:  
`<danh_sách>.remove(<giá trị>)`

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
43/55

43

A D  
A I

## 2.4. List data type (9)

### List operations

```

1  #Xóa phần tử khỏi danh sách
2  list_ab=[8,100,4,8,2,3,0,7,6,5,10]
3  print('Danh sách ban đầu: ', list_ab)
4
5  #Xóa phần tử ở cuối
6  list_ab.pop()
7  print('Danh sách sau xóa phần tử cuối:',list_ab)
8
9  #Xóa phần tử ở vị trí số 5
10 del list_ab[5]
11 print('Danh sách sau xóa phần tử vị trí số 5:',list_ab)
12
13 #Xóa phần tử có giá trị 8 xuất hiện đầu tiên
14 list_ab.remove(8)
15 print('Danh sách sau xóa phần tử có giá trị 8 đầu tiên:',list_ab)

```

Danh sách ban đầu: [8, 100, 4, 8, 2, 3, 0, 7, 6, 5, 10]

Danh sách sau xóa phần tử cuối: [8, 100, 4, 8, 2, 3, 0, 7, 6, 5]

Danh sách sau xóa phần tử vị trí số 5: [8, 100, 4, 8, 2, 0, 7, 6, 5]

Danh sách sau xóa phần tử có giá trị 8 đầu tiên: [100, 4, 8, 2, 0, 7, 6, 5]

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
44/55

44

A

D

A

I

## 2.4. List data type (10)

### List operations

- Đếm số lần xuất hiện của một phần tử trong danh sách:

<danh\_sách>.count(<giá trị>)

```

1  #Đếm số lần xuất hiện của một phần tử trong danh sách:
2  print('Danh sách ban đầu: \n', list_ab)
3
4  #Số lần xuất hiện số 8 trong danh sách
5  print(' Số 8 xuất hiện: ', list_ab.count(8))
6
7  #Số lần xuất hiện số 1 trong danh sách
8  print(' Số 1 xuất hiện: ', list_ab.count(1))

```

Danh sách ban đầu:  
[8, 4, 8, 2, 3, 7, 6, 5]  
Số 8 xuất hiện: 2  
Số 1 xuất hiện: 0

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

45/55

45

A

D

A

I

## 2.4. List data type (11)

### List operations

- Lưu ý về việc gán danh sách bởi 1 danh sách khác

```

1  #Trường hợp số, chuỗi
2  a = 10      #Khai báo biến a có giá trị =10
3  b = a      # Gán giá trị của biến a cho biến b
4  b = 5      # thay đổi giá trị của biến b, bằng giá trị mới
5  #-----
6  print('Giá trị của biến a: ', a)
7  print('Giá trị của biến b: ', b)

```

Giá trị của biến a: 10  
Giá trị của biến b: 5

```

1  #Trường hợp danh sách:
2  ds_a = [4,5,8,9] #Khai báo danh sách ds_a
3  ds_b = ds_a      #Gán giá trị của biến ds_a cho ds_b
4  ds_b[1] = 10     #Thay đổi giá trị vị trí số 2 trong ds_b
5  #-----
6  print('Biến ds_a: ', ds_a)
7  print('Biến ds_b: ', ds_b)

```

Biến ds\_a: [4, 10, 8, 9]  
Biến ds\_b: [4, 10, 8, 9]

Với mỗi sự thay đổi ở một trong 2 danh sách quản lý bởi biến ds\_a, ds\_b đều dẫn đến sự thay đổi ở danh sách còn lại.

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

46/55

46

A D  
A I

## 2.4. List data type (5)

### List operations

- Để tạo một danh sách độc lập với danh sách hiện có, các phần tử trong danh sách mới được sao chép giống hoàn toàn các phần tử trong danh sách hiện có:

<danh sách>.copy()

```

1 #Sao chép một danh sách độc lập:
2 ds_a = [4,5,8,9] #Khởi tạo danh sách ds_a
3 ds_b = ds_a.copy() #Sao chép ds_a cho ds_b
4 ds_b[1] = 10 #Thay đổi giá trị vị trí số 2 trong ds_b
5 #-----
6 print('Biến ds_a: ', ds_a)
7 print('Biến ds_b: ', ds_b)

```

Biến ds\_a: [4, 5, 8, 9]  
Biến ds\_b: [4, 10, 8, 9]

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

47/55

47

A D  
A I

## 2.5 Boolean data types

- Kiểu dữ liệu Boolean chỉ có hai giá trị True (đúng) và False (sai):

```

1 #Khởi tạo biến kiểu dữ liệu Boolean:
2 x = True
3 y = False
4 #Khởi tạo biến kiểu dữ liệu boolean qua biểu thức
5 z = 5 > 8
6 w = 12 == 12
7 #-----
8 print('Kiểu dữ liệu của biến x:', type(x), ', Giá trị: ', x)
9 print('Kiểu dữ liệu của biến y:', type(y), ', Giá trị: ', y)
10 print('Kiểu dữ liệu của biến z:', type(z), ', Giá trị: ', z)
11 print('Kiểu dữ liệu của biến w:', type(w), ', Giá trị: ', w)

```

Kiểu dữ liệu của biến x: <class 'bool'> , Giá trị: True  
 Kiểu dữ liệu của biến y: <class 'bool'> , Giá trị: False  
 Kiểu dữ liệu của biến z: <class 'bool'> , Giá trị: False  
 Kiểu dữ liệu của biến w: <class 'bool'> , Giá trị: True

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275

48/55

48



A
D

A
I

## Bài tập thực hành

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

49/55

49


A
D

A
I

## Bài 5: Thống kê điểm sinh viên

**Khởi tạo một danh sách gồm các điểm thi môn “Khoa học dữ liệu” của lớp CNTT01 (Điểm chữ: A, B, C, D, F)**

- Yêu cầu:**
  - 1) Cho biết số sinh viên trong lớp
  - 2) Có bao nhiêu sinh viên phải học lại môn này.
  - 3) Có bao nhiêu sinh viên có điểm từ B trở lên.
  - 4) Sinh viên đầu tiên và cuối cùng trong lớp đã nghỉ học, tạo một bảng điểm mới và loại bỏ điểm của các sinh viên này ra khỏi danh sách điểm.



Danh sách điểm KHDL của lớp CNTT01 là:  
 ['C', 'B', 'D', 'A', 'F', 'A', 'B', 'F', 'B', 'B', 'C', 'A', 'D', 'F', 'B']  
 -----THỐNG KÊ-----

- 1) Tổng số học sinh của lớp CNTT01: 15
- 2) Số học sinh phải học lại là: 3
- 3) Số học sinh phải học có điểm từ B trở lên là: 8
- 4) Bảng điểm sau khi đã loại bỏ sinh viên đầu tiên và cuối cùng nghỉ học là:  
 ['B', 'D', 'A', 'F', 'A', 'B', 'F', 'B', 'B', 'C', 'A', 'D', 'F']

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

50/55

50

A

D

A

I

2.6. Dictionary data type (1)

- What are dictionaries
- Keys and Values
- Append, Delete and Verify key in dictionaries

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

51/55

51

A

D

A

I

2.6. Dictionary data type (2)

### What are dictionaries??

- Một Dictionaries chứa keys và Values để truy cập đến từng Indexed thông qua Key của nó. Dùng dấu ngoặc {}

```

1 #Dictionary data type
2 dic={'key1':1, 'key2':'hello', 'key3':('tuples', 'Welcome', 'DS'), (0,1):True}
3 print(dic)
4 print(dic['key3'])
5 dic={'key1':1, 'key2':'hello', 'key3':3}
6 print(dic)
7 print(dic['key1'])

```

```

{'key1': 1, 'key2': 'hello', 'key3': ('tuples', 'Welcome', 'DS'), (0, 1): True}
('tuples', 'Welcome', 'DS')
{'key1': 1, 'key2': 'hello', 'key3': 3}
1

```

Tạ Quang Chiêu – [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) – [M]: 0913 522 275

52/55

52

A
D
A
I

## 2.6. Dictionary data type (3)

### Keys and Values

- keys của dictionary có thể là các kiểu: strings, numbers, và tuple

```

1 dic={'key1':1, 2:'hello',(1,'python','3'):(('tuples','Welcome','DS'),(0,1):True}
2 print(dic.keys())
3 print(type(dic.keys()))
4 print(dic.values())
5 print(type(dic.keys()))

dict_keys(['key1', 2, (1, 'python', '3'), (0, 1)])
<class 'dict_keys'>
dict_values([1, 'hello', ('tuples', 'Welcome', 'DS'), True])
<class 'dict_keys'>
```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
53/55

53

A
D
A
I

## 2.6. Dictionary data type (4)

### Keys

☐ Thuộc tính của khóa

Có ba điểm quan trọng trong khi sử dụng khóa

- Các khóa không được trùng lặp nhau.
- Các giá trị trong từ điển có thể thuộc bất kỳ loại nào trong khi các khóa phải là bất biến như số hoặc chuỗi.
- Các khóa từ điển có phân biệt chữ hoa chữ thường.

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
54/55

54

A D  
A I

2.6. Dictionary data type (5)

• Append, delete and verify key in dictionaries:

```

1 dic={'key1':1, 2:'hello', (1, 'python', '3'): ('tuples', 'Welcome', 'DS'), (0,1):True}
2 print(dic)
3 # Adding a new key value pair
4 dic['keyn']='NewValue'
5 print(dic)
6 dic.update( {'before' : 23} )
7 print(dic)
8 #Delete a key in dictionaries
9 del(dic['keyn'])
10 print(dic)
11 #verify a key in dictionaries (IN operator)
12 print(('key1') in dic)
13 print(('key23') in dic)

```

```

{'key1': 1, 2: 'hello', (1, 'python', '3'): ('tuples', 'Welcome', 'DS'), (0, 1): True}
{'key1': 1, 2: 'hello', (1, 'python', '3'): ('tuples', 'Welcome', 'DS'), (0, 1): True, 'keyn': 'NewValue'}
{'key1': 1, 2: 'hello', (1, 'python', '3'): ('tuples', 'Welcome', 'DS'), (0, 1): True, 'keyn': 'NewValue', 'before': 23}
{'key1': 1, 2: 'hello', (1, 'python', '3'): ('tuples', 'Welcome', 'DS'), (0, 1): True, 'before': 23}
True
False

```

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
55/55

55

A D  
A I

Bài tập

**Bài 6.** Nhập một số n, hãy tạo ra **dictionary** chứa các phần tử dạng i:i2 với (i chạy từ 1 đến n) và in ra **dictionary** đó. Ví dụ với n là 3 thì đầu ra sẽ là: {1: 2, 2: 4, 3: 6}

**Bài 7.** Nhập vào một dãy số ngăn cách bởi dấu phẩy, hãy tạo và in ra một **list** và một tuple từ các số đó. Sự khác nhau giữa **list** và **tuple** là gì?

Tạ Quang Chiêu - [E]: [quangchieu.ta@gmail.com](mailto:quangchieu.ta@gmail.com) - [M]: 0913 522 275
56/55

56