

**BÀI TẬP THỰC HÀNH**  
**MÔN TRÍ TUỆ NHÂN TẠO**  
**LAB 2: KỸ THUẬT LẬP TRÌNH VỚI PYTHON**

- <https://www.w3schools.com/python/default.asp>
- <https://www.programiz.com/python-programming/first-program>

**A. Cú pháp:**

**Python List:**

**List Methods**

Python has a set of built-in methods that you can use on lists.

Method	Description
<code>append()</code>	Adds an element at the end of the list
<code>clear()</code>	Removes all the elements from the list
<code>copy()</code>	Returns a copy of the list
<code>count()</code>	Returns the number of elements with the specified value
<code>extend()</code>	Add the elements of a list (or any iterable), to the end of the current list
<code>index()</code>	Returns the index of the first element with the specified value
<code>insert()</code>	Adds an element at the specified position
<code>pop()</code>	Removes the element at the specified position
<code>remove()</code>	Removes the item with the specified value
<code>reverse()</code>	Reverses the order of the list
<code>sort()</code>	Sorts the list

**Python tuple**

**Tuple Methods**

Python has two built-in methods that you can use on tuples.

Method	Description
<code>count()</code>	Returns the number of times a specified value occurs in a tuple
<code>index()</code>	Searches the tuple for a specified value and returns the position of where it was found

**Python sets**

## Set Methods

Python has a set of built-in methods that you can use on sets.

Method	Description
<code>add()</code>	Adds an element to the set
<code>clear()</code>	Removes all the elements from the set
<code>copy()</code>	Returns a copy of the set
<code>difference()</code>	Returns a set containing the difference between two or more sets
<code>difference_update()</code>	Removes the items in this set that are also included in another, specified set
<code>discard()</code>	Remove the specified item
<code>intersection()</code>	Returns a set, that is the intersection of two other sets
<code>intersection_update()</code>	Removes the items in this set that are not present in other, specified set(s)
<code>isdisjoint()</code>	Returns whether two sets have a intersection or not
<code>issubset()</code>	Returns whether another set contains this set or not
<code>issuperset()</code>	Returns whether this set contains another set or not
<code>pop()</code>	Removes an element from the set
<code>remove()</code>	Removes the specified element
<code>symmetric_difference()</code>	Returns a set with the symmetric differences of two sets
<code>symmetric_difference_update()</code>	inserts the symmetric differences from this set and another
<code>union()</code>	Return a set containing the union of sets
<code>update()</code>	Update the set with the union of this set and others

## Python dictionary

### Dictionary Methods

Python has a set of built-in methods that you can use on dictionaries.

Method	Description
<code>clear()</code>	Removes all the elements from the dictionary
<code>copy()</code>	Returns a copy of the dictionary
<code>fromkeys()</code>	Returns a dictionary with the specified keys and value
<code>get()</code>	Returns the value of the specified key
<code>items()</code>	Returns a list containing a tuple for each key value pair
<code>keys()</code>	Returns a list containing the dictionary's keys
<code>pop()</code>	Removes the element with the specified key
<code>popitem()</code>	Removes the last inserted key-value pair
<code>setdefault()</code>	Returns the value of the specified key. If the key does not exist: insert the key, with the specified value
<code>update()</code>	Updates the dictionary with the specified key-value pairs
<code>values()</code>	Returns a list of all the values in the dictionary

### Function:

#### The def statement:

The `def` statement is used to define functions and methods.

## Returning values

The `return` statement is used to return values from a `function`.

The `return` statement takes zero or more values, separated by commas. Using commas actually returns a single tuple.

The default value is `None`.

## B. Bài tập

1. Cho 2 mảng giá trị như sau:

```
suits = ['Hearts', 'Diamonds', 'Clubs', 'Spades']
```

```
values = ['Two', 'Three', 'Four', 'Five', 'Six', 'Seven', 'Eight', 'Nine', 'Ten', 'Jack', 'Queen', 'King', 'Ace']
```

In ra tên tất cả quân bài trong bộ bài tây, ví dụ: “Two of Hearts”, “Two of Diamonds”, ...

2. Cho tập dữ liệu bên dưới. Thực hiện các yêu cầu sau:

- Lưu tập dữ liệu này vào 1 biến kiểu list
- Đếm số phần tử trong list
- Đếm tần số xuất hiện của từng giá trị và lưu chúng trong 1 biến kiểu dictionary.

```
M    M    M    A    H    M    S    M    H    M
S    M    M    M    M    A    M    M    A    M
M    M    H    M    M    M    H    M    H    M
A    M    M    M    H    M    M    M    M    M
```

3. Viết hàm nhận vào chiều dài và rộng hình chữ nhật, xuất ra hình chữ nhật với số dấu \* chiều dài và rộng ứng với đầu vào. Ví dụ hình chữ nhật dài 4, rộng 3 như sau:

```
****
****
****
```

4. Viết hàm nhận vào chiều cao tam giác vuông, xuất ra hình tam giác như sau:  
Chiều cao là 5:

```
*
**
* *
* *
*****
```

5. Viết hàm xuất giá trị Fibonacci thứ n. Biết rằng dãy Fibonacci bắt đầu: 0 1 1 2 3 5 ...

6. Dùng numpy để xây dựng các hàm:

- Tạo mảng 1 chiều gồm n phần tử có giá trị ngẫu nhiên (n chia hết cho 5 và n lớn hơn 20)

- Xuất mảng
- Chuyển mảng 1 chiều thành mảng 2 chiều số cột là 5.
- In các thông tin: shape, ndim, dtype, itemsize, size
- Đếm tần số xuất hiện của 1 giá trị x trong mảng 2 chiều.
- Tính tổng 2 mảng
- Tính tích 2 mảng

-----**Hết**-----