AI VIET NAM
@aivietnam.edu.vn

Module 01 – Extra Class

LIST

Nguyen Quoc Thai

Objectives

Python Basic

- ❖ Variable
- ❖ Operators
- ❖ Condition
- ❖ Function
- ❖ Built-in Function
- ❖ For/While Loop
- ❖ If-else

Data Structure

- ❖ List

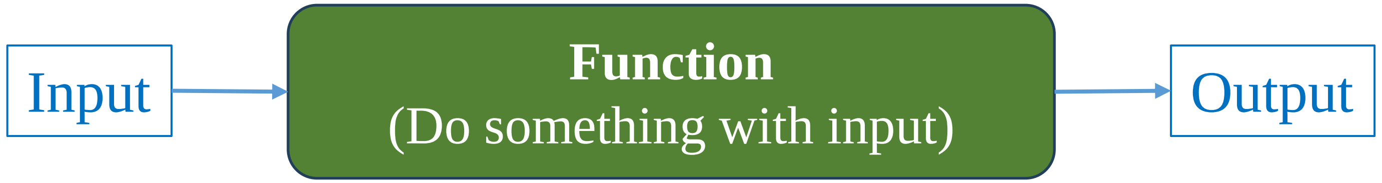
AI VIET NAM
@aivietnam.edu.vn

Outline

AI VIET NAM
@aivietnam.edu.vn

Review

! Function

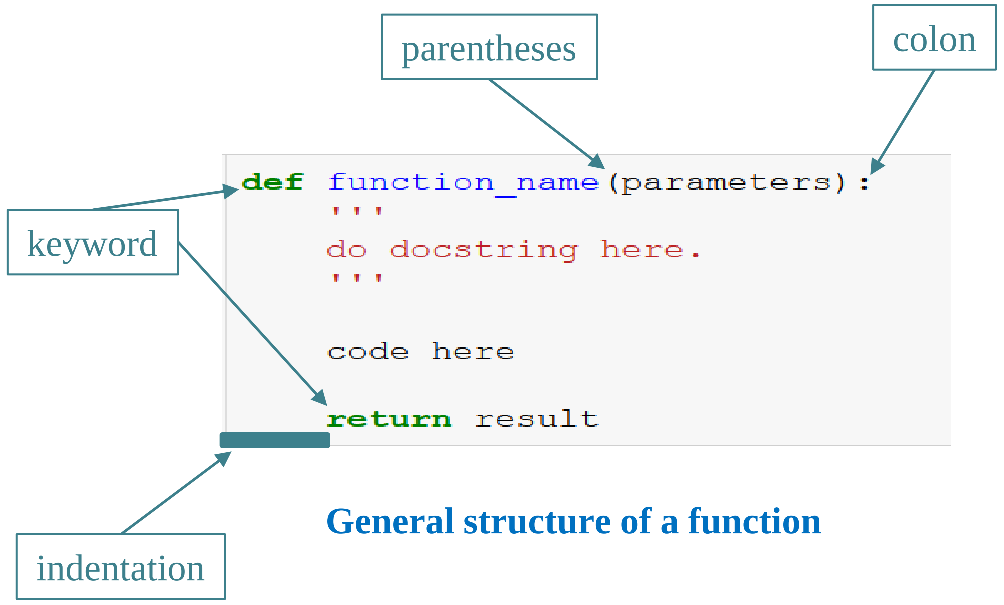


Built-in Functions

User-defined Functions

print(parameters**)**

type(parameter**)**

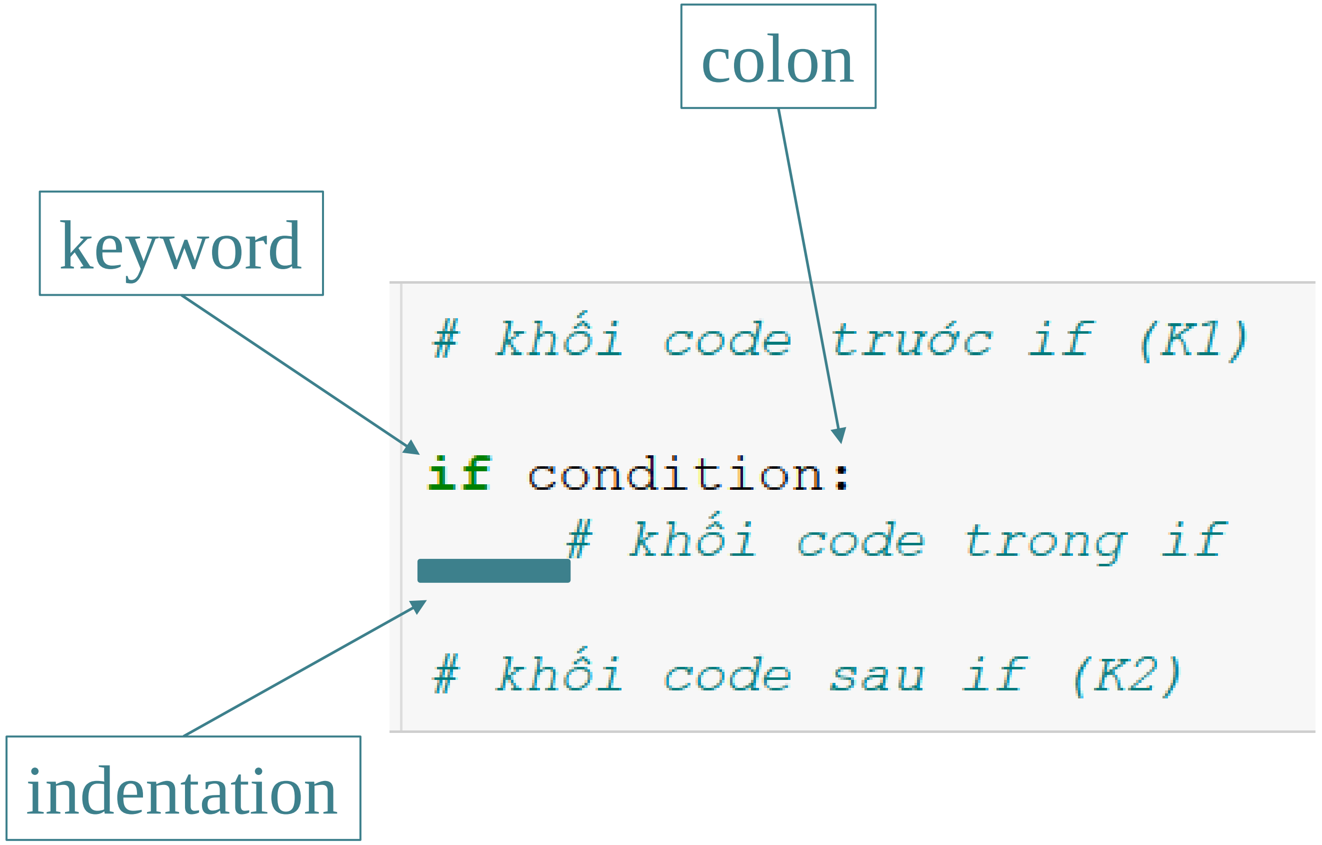


```
def function_name(parameters):
    '''
    do docstring here.
    '''

    code here

    return result
```

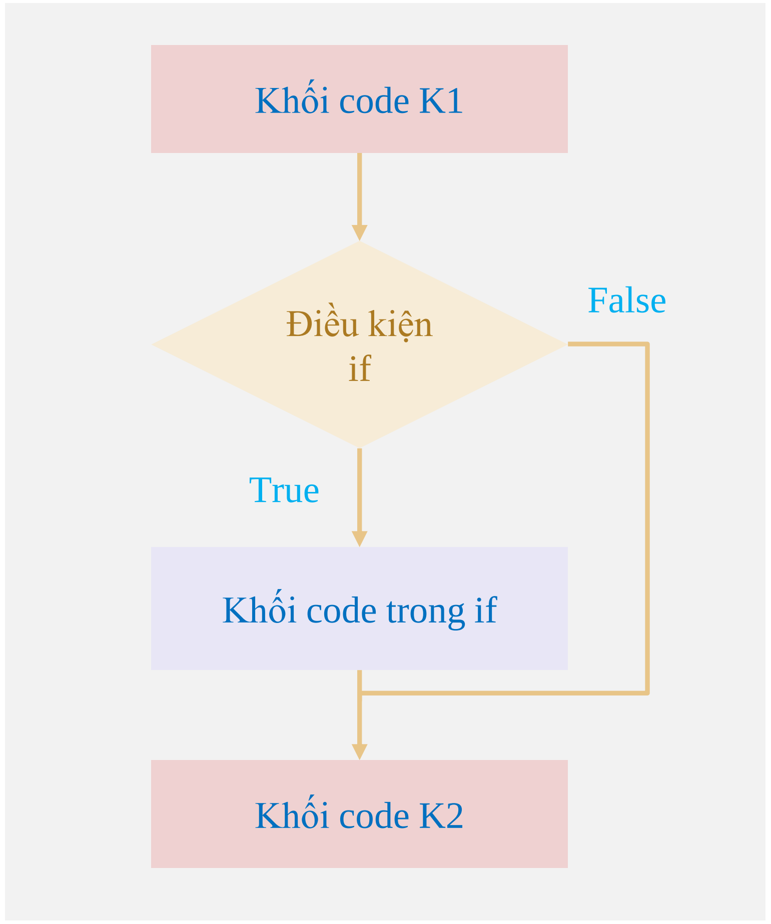
General structure of a function

Review

If-Else



```
# khối code trước if (K1)

if condition:
    # khối code trong if

# khối code sau if (K2)
```



AI VIET NAM
@aivietnam.edu.vn

Review

Example

Built-in Functions

```python
sentence = 'I love AI'
print(sentence)
print(type(sentence))
```

```
I love AI
<class 'str'>
```

User-defined Functions

```python
def add_numbers(num_1, num_2):
    total = num_1 + num_2
    return total

num_1 = 10
num_2 = 8
print(add_numbers(num_1, num_2))
```

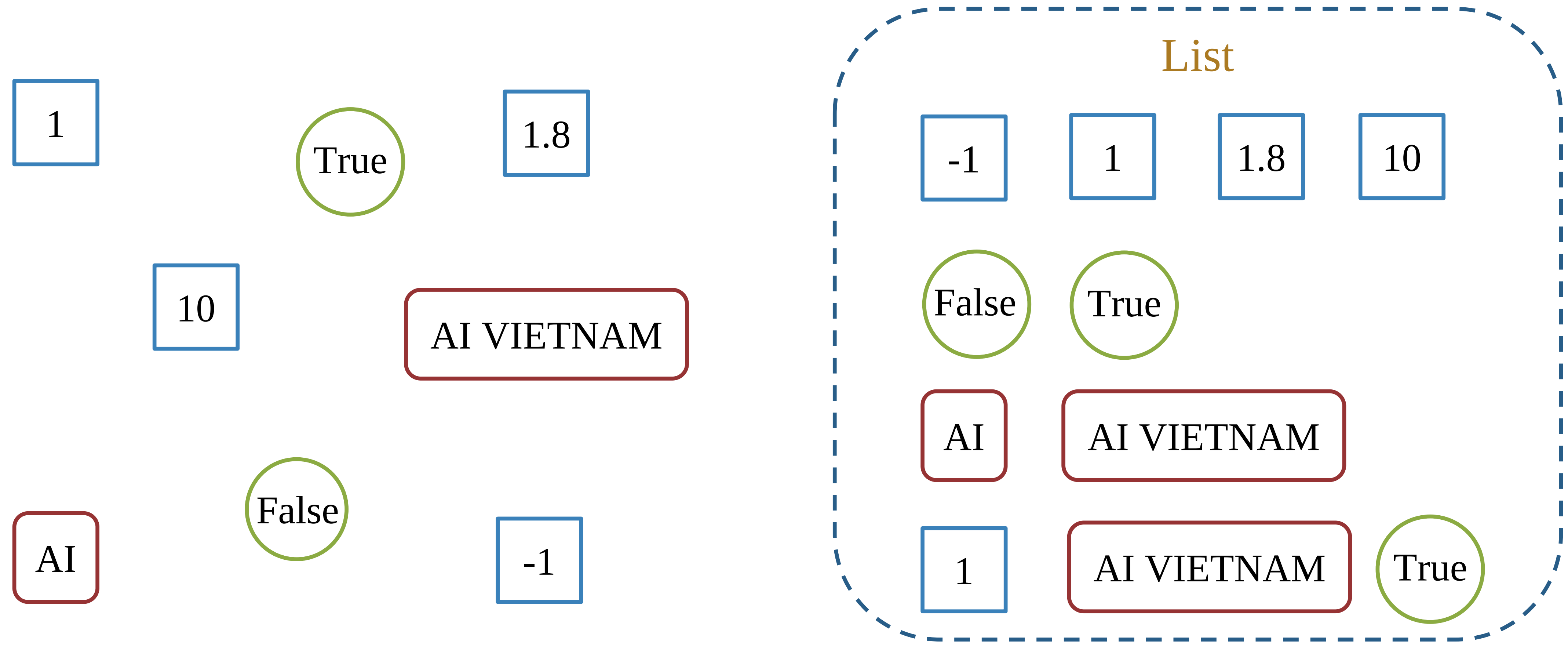
```
18
```

AI VIET NAM
@aivietnam.edu.vn

Outline

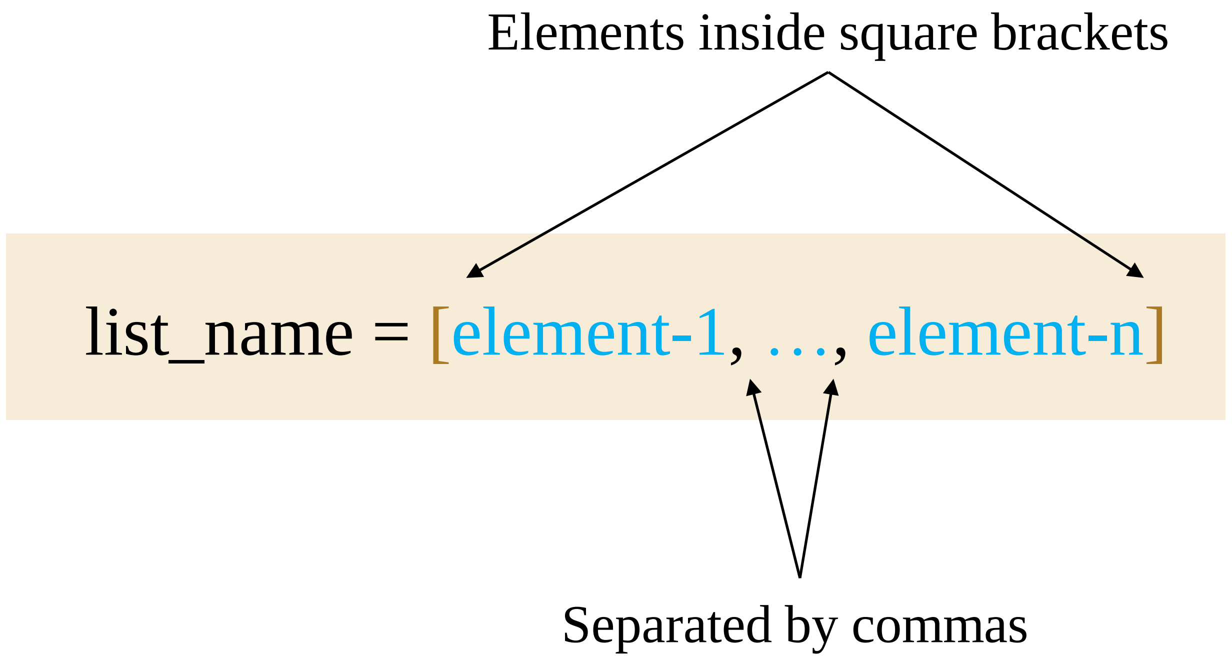
List

! **Lists are used to store multiple items in a single variable**



List

A container that can contain elements


Elements inside square brackets
list_name = [element-1, …, element-n]
Separated by commas

AI VIET NAM
@aivietnam.edu.vn

List

A container that can contain elements

```python
# create a list
data = [4, 5, 6, 7, 8, 9]
print(data)
print(type(data))
print(len(data))
```

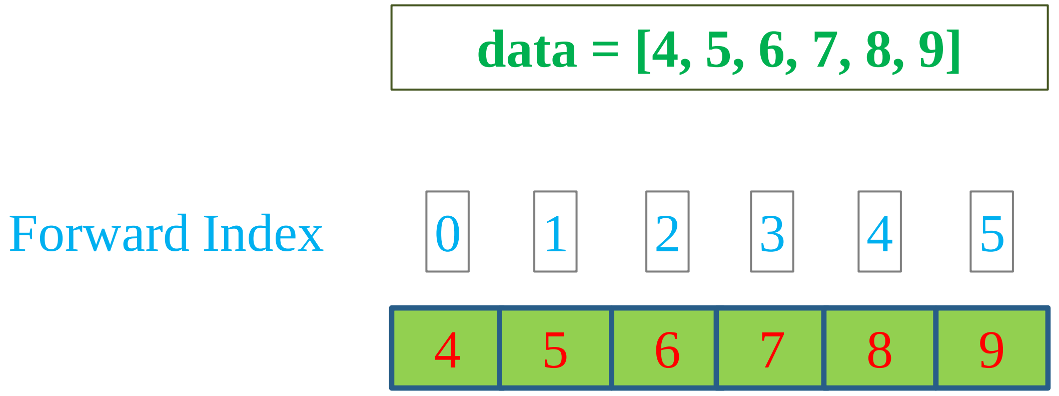
```
[4, 5, 6, 7, 8, 9]
<class 'list'>
6
```

```python
# danh sách trống
emty_list = []

# danh sách số tự nhiên nhỏ hơn 10
my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

# danh sách kết hợp nhiều kiểu dữ liệu
mixed_list = [True, 5, 'some string', 123.45]
n_list = ["Happy", [2,0,1,5]]

# danh sách các loại hoa quả
shopping_list = ['táo', 'chuối', 'cherries', 'dâu', 'mận']
```

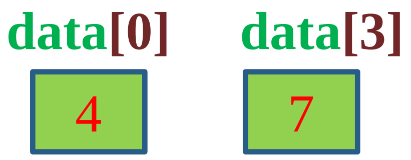
List

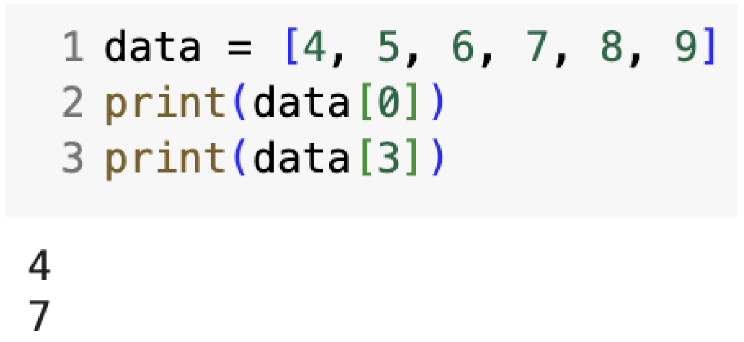
Index

➢ Each element in a list is associated with a number, known as a index



Access elements using index





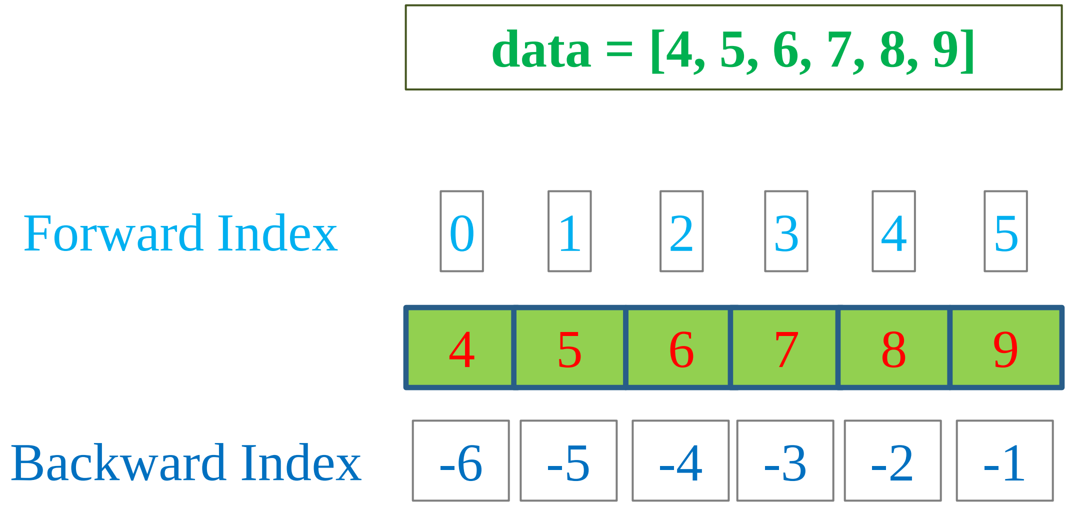
```
data = [4, 5, 6, 7, 8, 9]
print(data[0])
print(data[3])
```

```
4
7
```

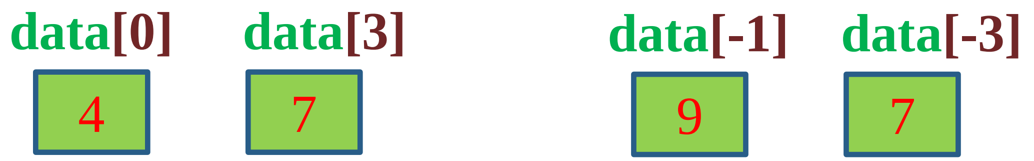
List

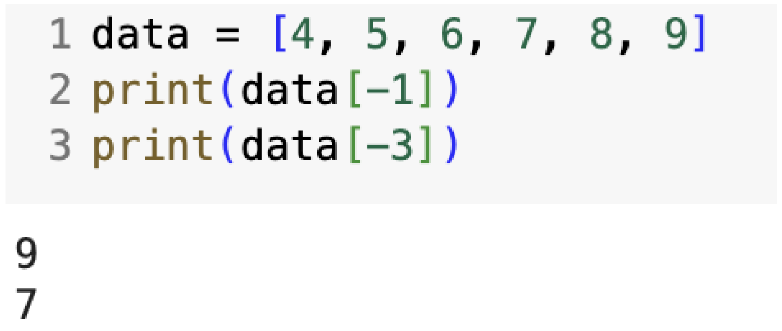
Index

➢ Each element in a list is associated with a number, known as a index



Access elements using index





```
data = [4, 5, 6, 7, 8, 9]
print(data[-1])
print(data[-3])
```

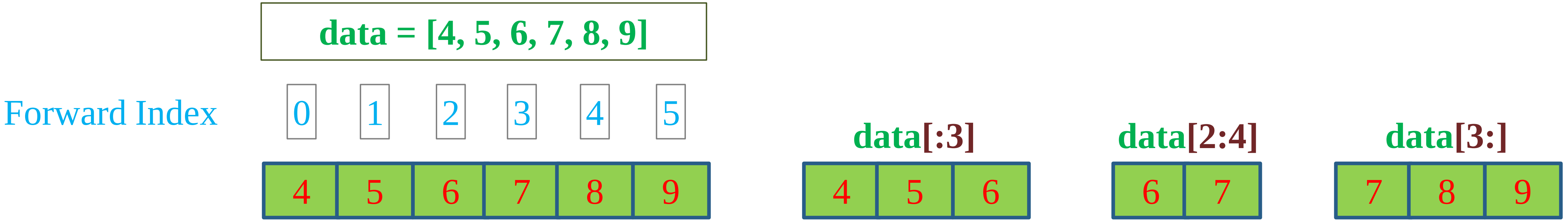
```
9
7
```

List

Slicing

➢ Access a section of items from list using the slicing operator.

list[start:end:step]



```
data = [4, 5, 6, 7, 8, 9]
print(data[:3])
print(data[2:4])
print(data[3:])
```

```
[4, 5, 6]
[6, 7]
[7, 8, 9]
```

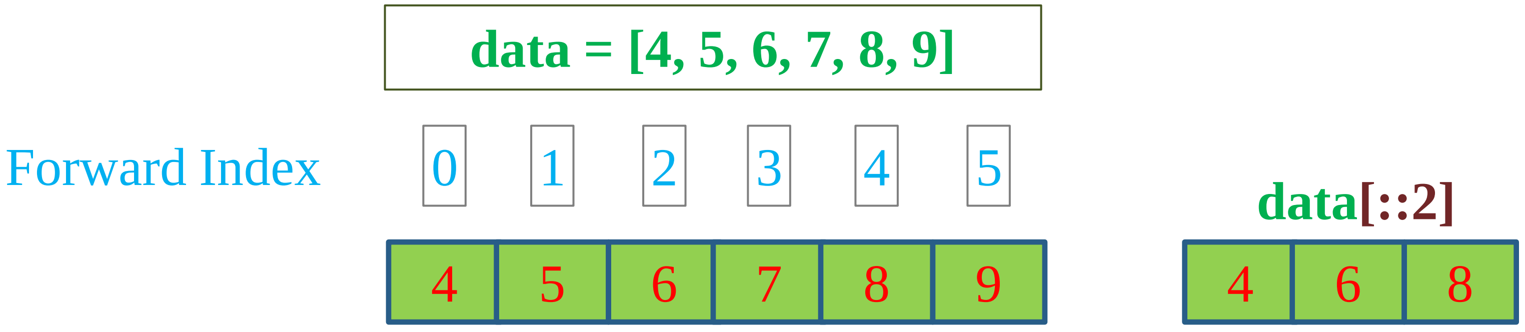
AI VIET NAM
@aivietnam.edu.vn

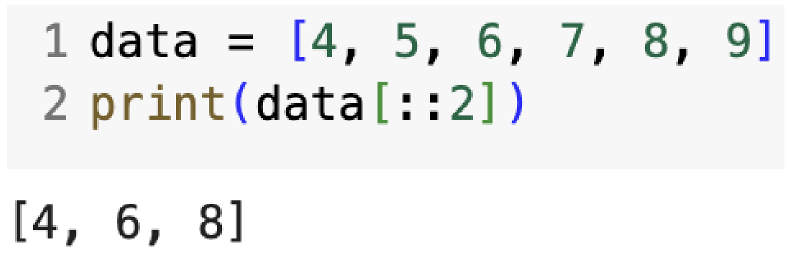
List

Slicing

- Access a section of items from list using the slicing operator.

list[start:end:step]





```
data = [4, 5, 6, 7, 8, 9]
print(data[::2])
```

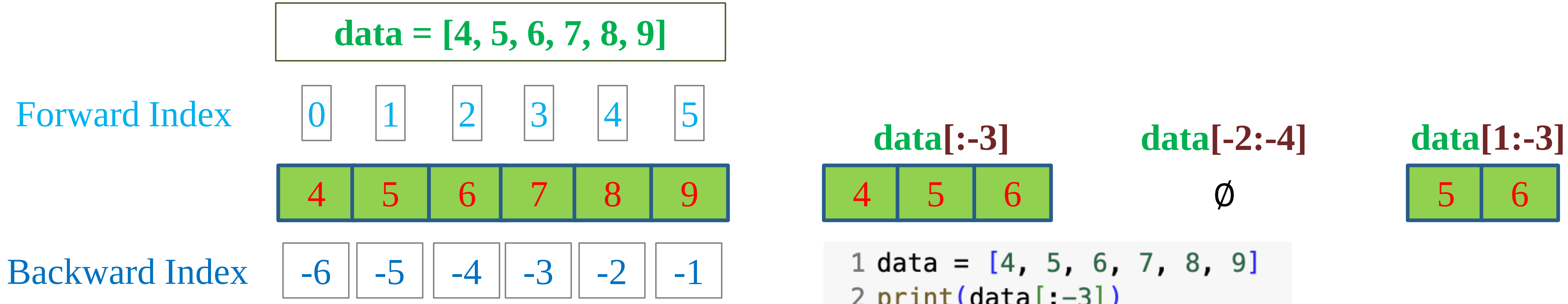
```
[4, 6, 8]
```

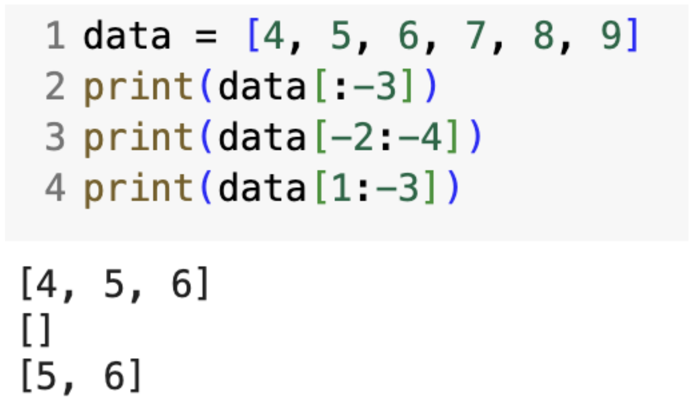
List

Slicing

- Access a section of items from list using the slicing operator.

list[start:end:step]





```
data = [4, 5, 6, 7, 8, 9]
print(data[:-3])
print(data[-2:-4])
print(data[1:-3])
```

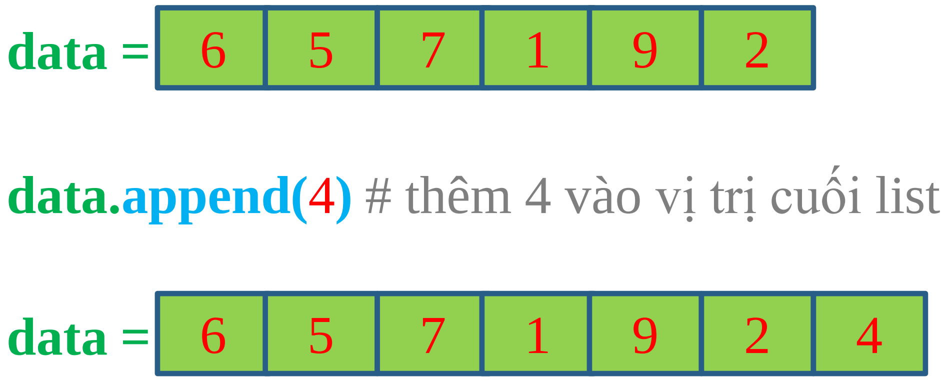
```
[4, 5, 6]
[]
[5, 6]
```

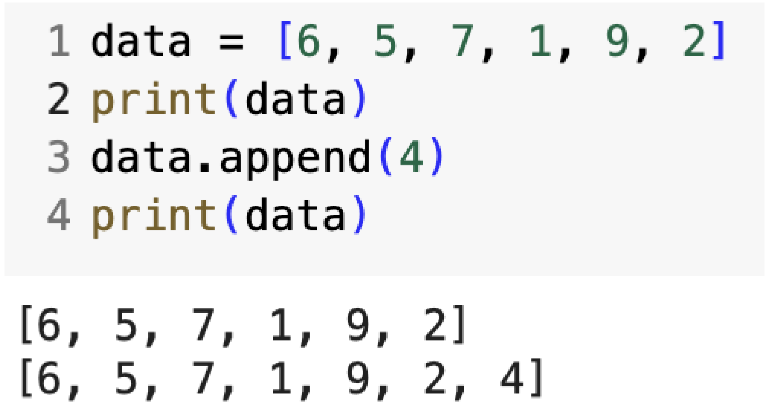
AI VIET NAM
@aivietnam.edu.vn

List

Add elements to a Python List

- Use the **append()** method to add an element to the end of a Python list.





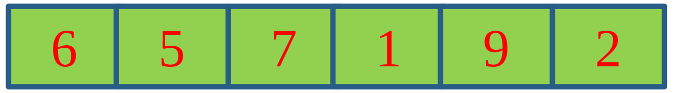
```python
data = [6, 5, 7, 1, 9, 2]
print(data)
data.append(4)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[6, 5, 7, 1, 9, 2, 4]
```

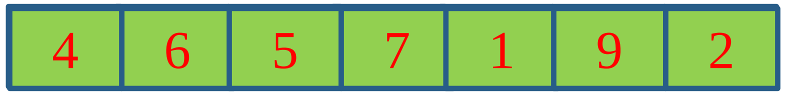
List

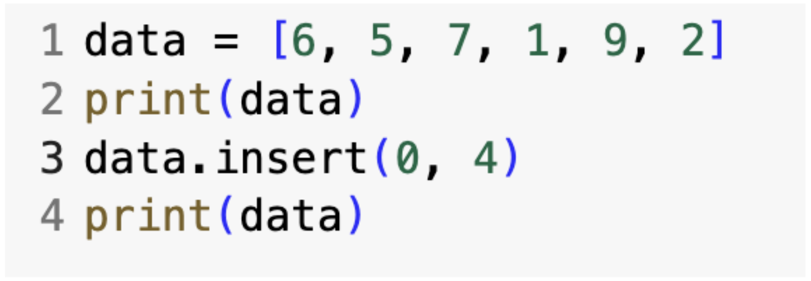
Add elements to a Python List

- Use the **insert()** method to add an element at the specified index of a Python list.

data = 

data.insert(0, 4) # thêm 4 vào vị trị index=0

data = 



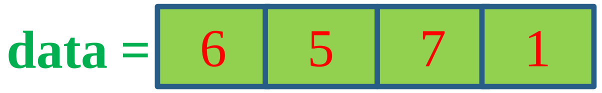
```python
data = [6, 5, 7, 1, 9, 2]
print(data)
data.insert(0, 4)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[4, 6, 5, 7, 1, 9, 2]
```

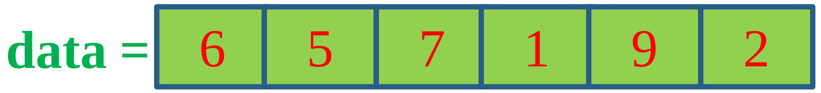
List

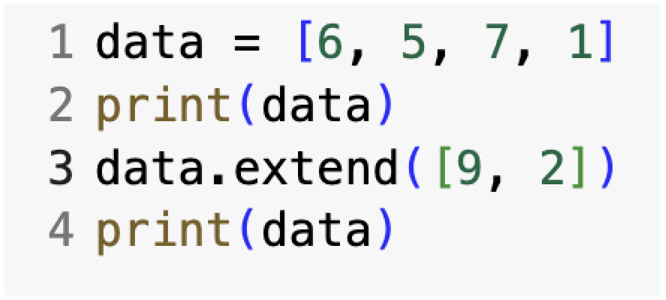
Add elements to a Python List

➢ Use the **extend()** method to add elements to a list from other iterables.



data.extend([9, 2]) # thêm 9 và 2 vào vị trị cuối list





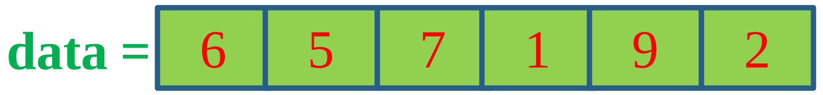
```python
data = [6, 5, 7, 1]
print(data)
data.extend([9, 2])
print(data)
```

```
[6, 5, 7, 1]
[6, 5, 7, 1, 9, 2]
```

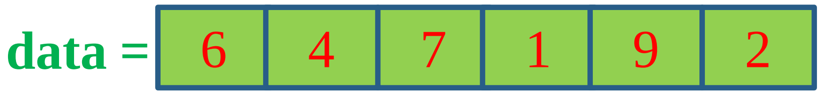
List

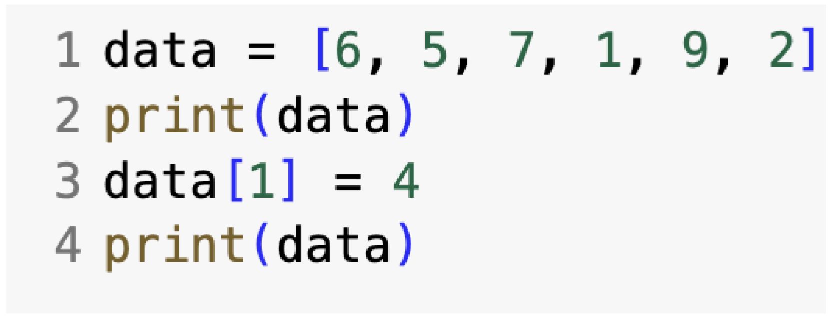
Updating an element

➢ Change the items of a list by assigning new values using the = operator .



# thay đổi phần tử thứ 1
data[1] = 4





```python
data = [6, 5, 7, 1, 9, 2]
print(data)
data[1] = 4
print(data)
```

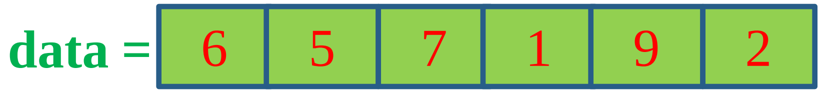
```
[6, 5, 7, 1, 9, 2]
[6, 4, 7, 1, 9, 2]
```

AI VIET NAM
@aivietnam.edu.vn

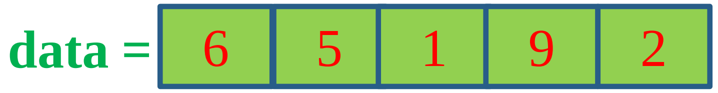
List

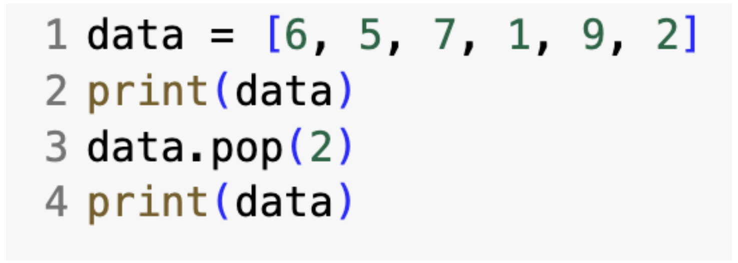
Delete an element from a list

- Using the **remove()** and **pop()** method.



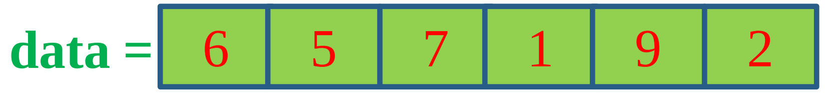
data.pop(2) # tại vị trí index = 2



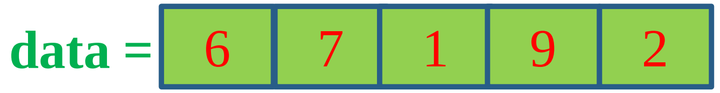


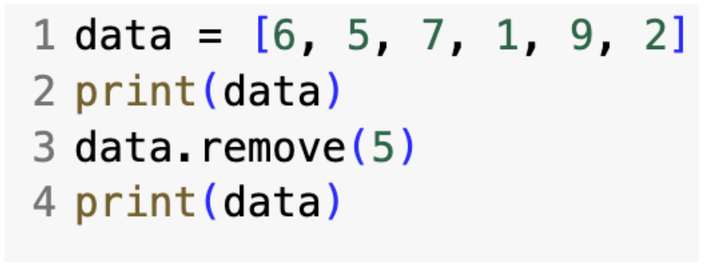
```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.pop(2)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[6, 5, 1, 9, 2]
```



data.remove(5) # xóa phần tử đầu tiên
# có giá trị là 5





```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.remove(5)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[6, 7, 1, 9, 2]
```

List

Delete elements from a list

- Using ‘del’ keyword to delete objects or **clear()** to removal elements.

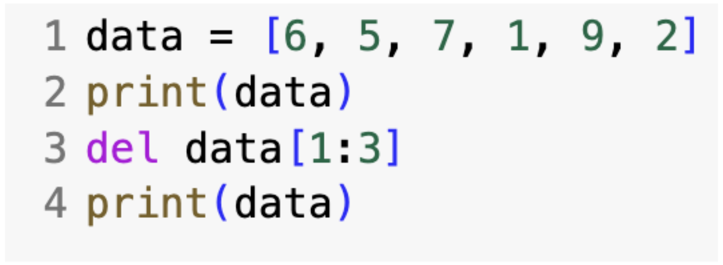
data =

6	5	7	1	9	2

# xóa phần tử thứ 1 và 2
del data[1:3]

data =

6	1	9	2



```
data = [6, 5, 7, 1, 9, 2]
print(data)
del data[1:3]
print(data)
```

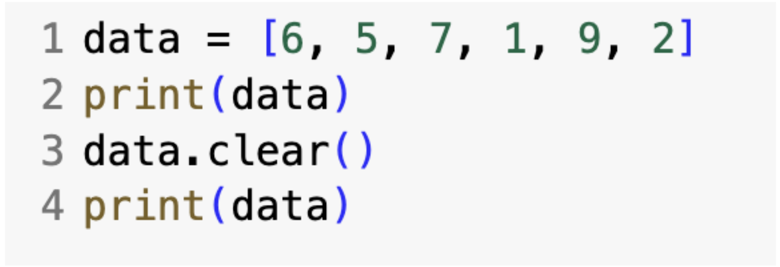
```
[6, 5, 7, 1, 9, 2]
[6, 1, 9, 2]
```

data =

6	5	7	1	9	2

data.clear()

data = []



```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.clear()
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[]
```

List

Index() method: Returns the index of the first matched item

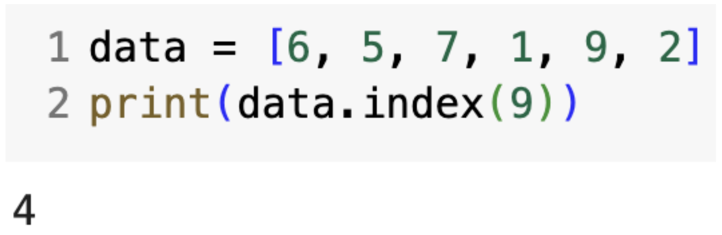
data =

6	5	7	1	9	2

# trả về vị trí của phần tử đầu tiên có giá trị là 9

data.index(9)

=> 4



```
data = [6, 5, 7, 1, 9, 2]
print(data.index(9))
```

```
4
```

Reverse() method: Reverses the item of the list

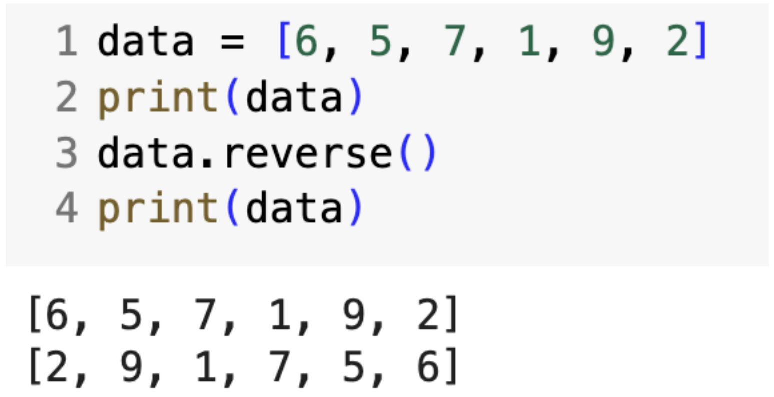
data =

6	5	7	1	9	2

data.reserse()

data =

2	9	1	7	5	6



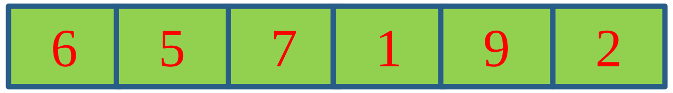
```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.reverse()
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[2, 9, 1, 7, 5, 6]
```

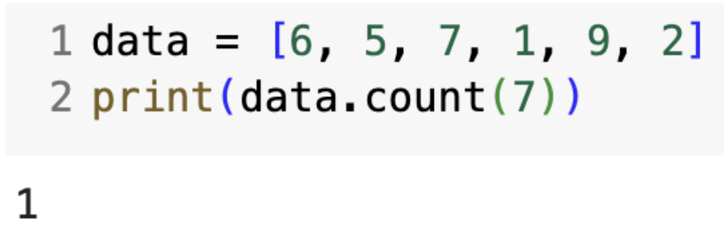
AI VIET NAM
@aivietnam.edu.vn

List

Count() method: Returns the count of the specified item in the list

data =


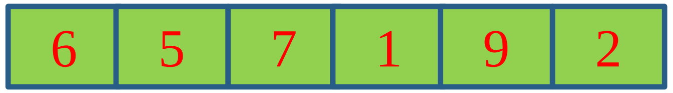
# trả về số lần phần tử 7 xuất hiện trong list
data.count(7) = 1



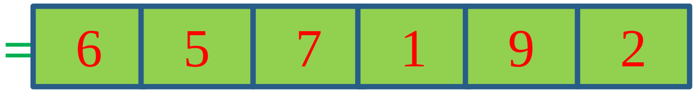
```python
data = [6, 5, 7, 1, 9, 2]
print(data.count(7))
```

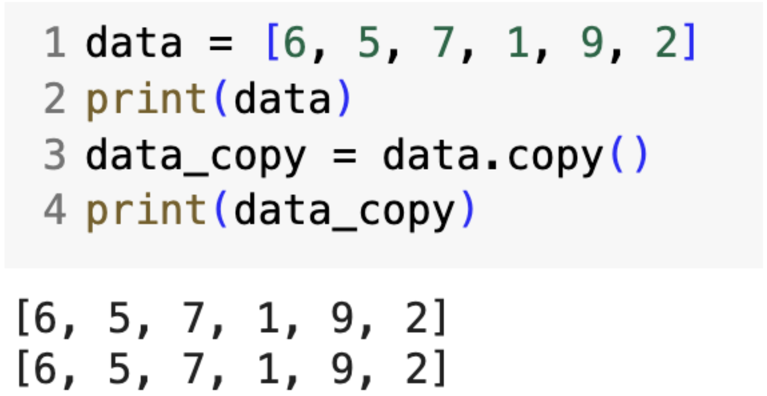
```
1
```

Copy() method: Returns the shallow copy of the list

data =


data_copy = data.copy()

data_copy =




```python
data = [6, 5, 7, 1, 9, 2]
print(data)
data_copy = data.copy()
print(data_copy)
```

```
[6, 5, 7, 1, 9, 2]
[6, 5, 7, 1, 9, 2]
```

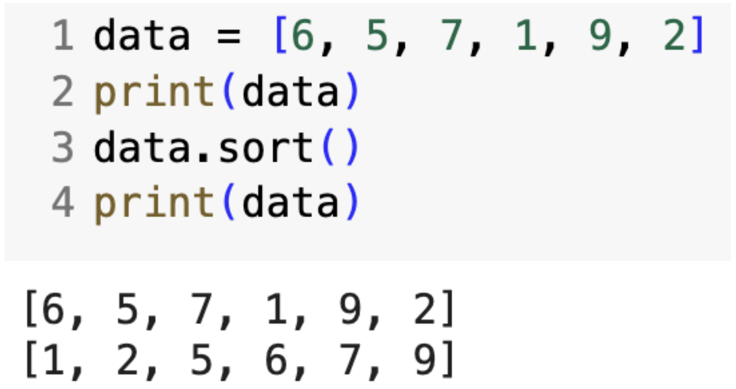
List

Sort() method: Sorts the list in ascending/descending order

data = 6 5 7 1 9 2

data.sort()

data = 1 2 5 6 7 9



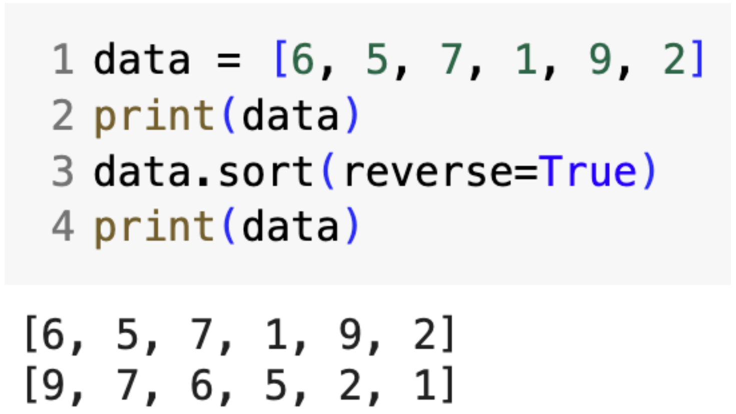
```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.sort()
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[1, 2, 5, 6, 7, 9]
```

data = 6 5 7 1 9 2

data.sort(reverse = True)

data = 9 7 6 5 2 1

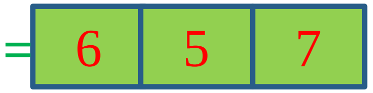


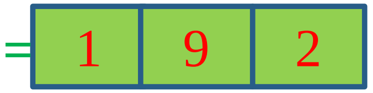
```
data = [6, 5, 7, 1, 9, 2]
print(data)
data.sort(reverse=True)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[9, 7, 6, 5, 2, 1]
```

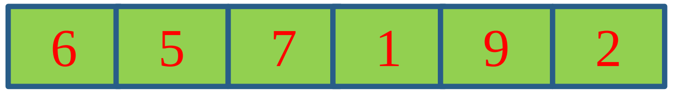
List

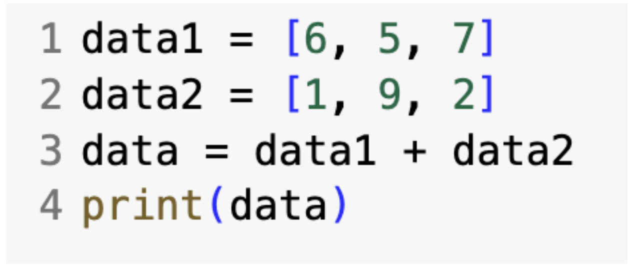
+ and * operators

data1 =


data2 =


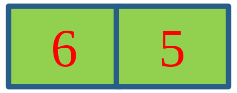
data = data1 + data2 # nối 2 list

data =


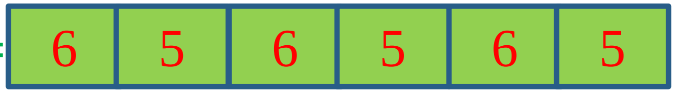


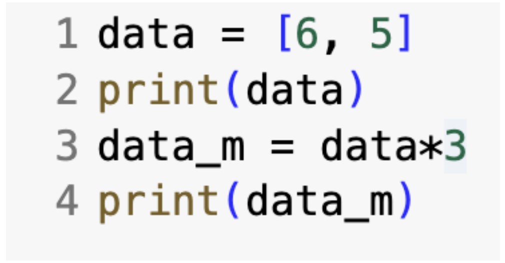
```
data1 = [6, 5, 7]
data2 = [1, 9, 2]
data = data1 + data2
print(data)
```

```
[6, 5, 7, 1, 9, 2]
```

data =


# nhân list với một số nguyên
data_m = data * 3

data_m =




```
data = [6, 5]
print(data)
data_m = data*3
print(data_m)
```

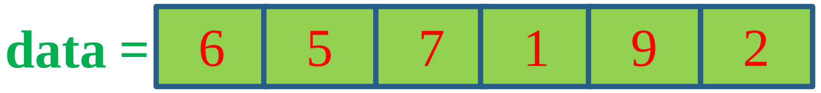
```
[6, 5]
[6, 5, 6, 5, 6, 5]
```

AI VIET NAM
@aivietnam.edu.vn

Outline

Built-in Functions for List

len(), min(), max()



# trả về số phần tử
len(data) = 6

# trả về số phần tử có giá trị nhỏ nhất
min(data) = 1

# trả về số phần tử có giá trị lớn nhất
max(data) = 9

```python
# get a number of elements

data = [6, 5, 7, 1, 9, 2]
length = len(data)
print(length)
```

```
6
```

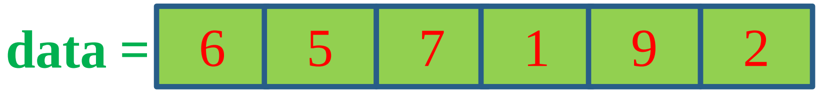
```python
# get the min and max values

data = [6, 5, 7, 1, 9, 2]
print(min(data))
print(max(data))
```

```
1
9
```

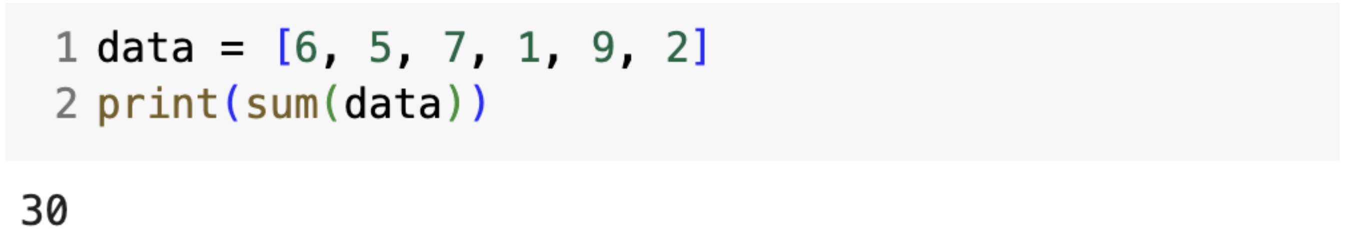
Built-in Functions for List

sum(): returns a number, the sum of all elements in a list

sum(iterable, start)

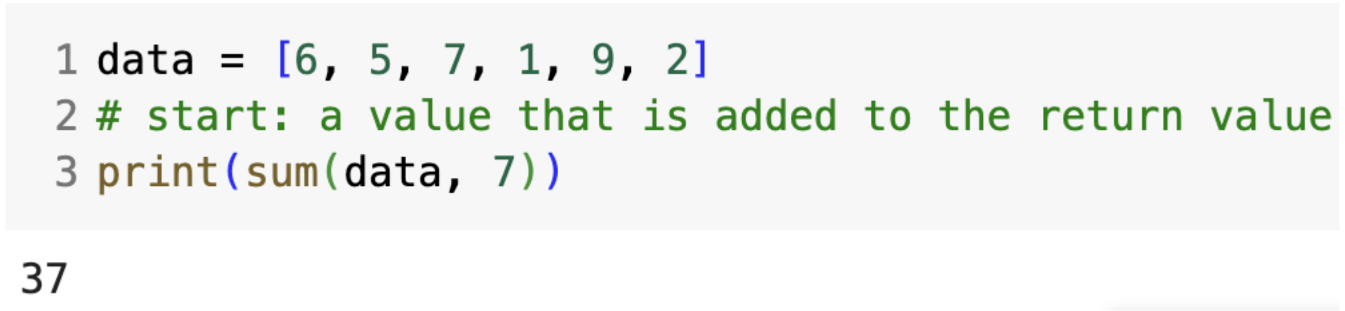


sum(data)
=> 30



```
data = [6, 5, 7, 1, 9, 2]
print(sum(data))
```

```
30
```



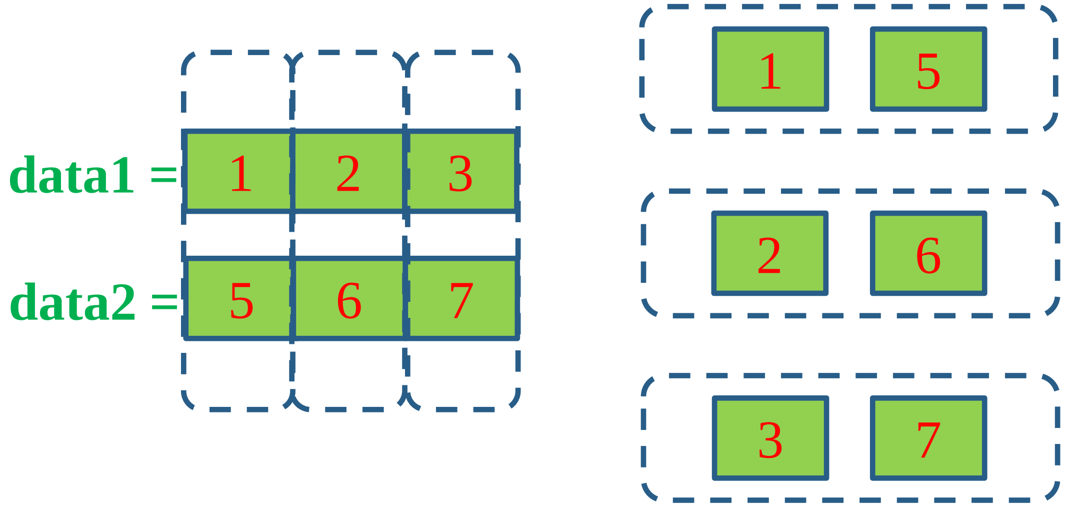
```
data = [6, 5, 7, 1, 9, 2]
# start: a value that is added to the return value
print(sum(data, 7))
```

```
37
```

Built-in Functions for List

zip(): takes iterable containers and returns a single iterator object, having mapped values from all the container.

zip(*iterators)



```python
data1 = [1, 2, 3]
data2 = [5, 6, 7]

for v1, v2 in zip(data1, data2):
    print(v1, v2)
```

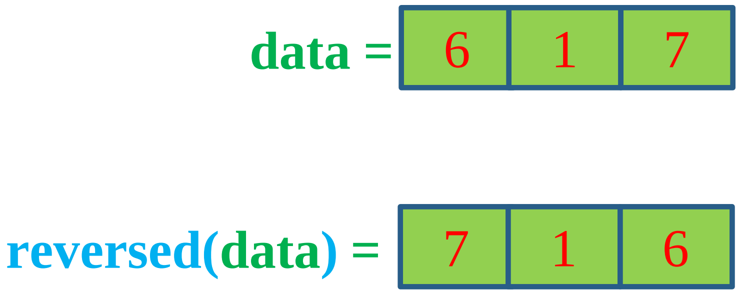
```
1 5
2 6
3 7
```

AI VIET NAM
@aivietnam.edu.vn

Built-in Functions for List

! **reversed(): returns a reversed iterator object**

reversed(iterable)



```python
data = [6, 1, 7]
for value in reversed(data):
    print(value)
```

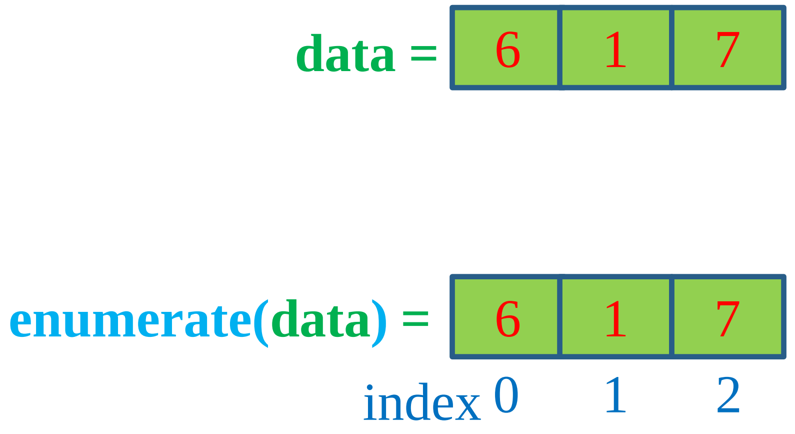
```
7
1
6
```

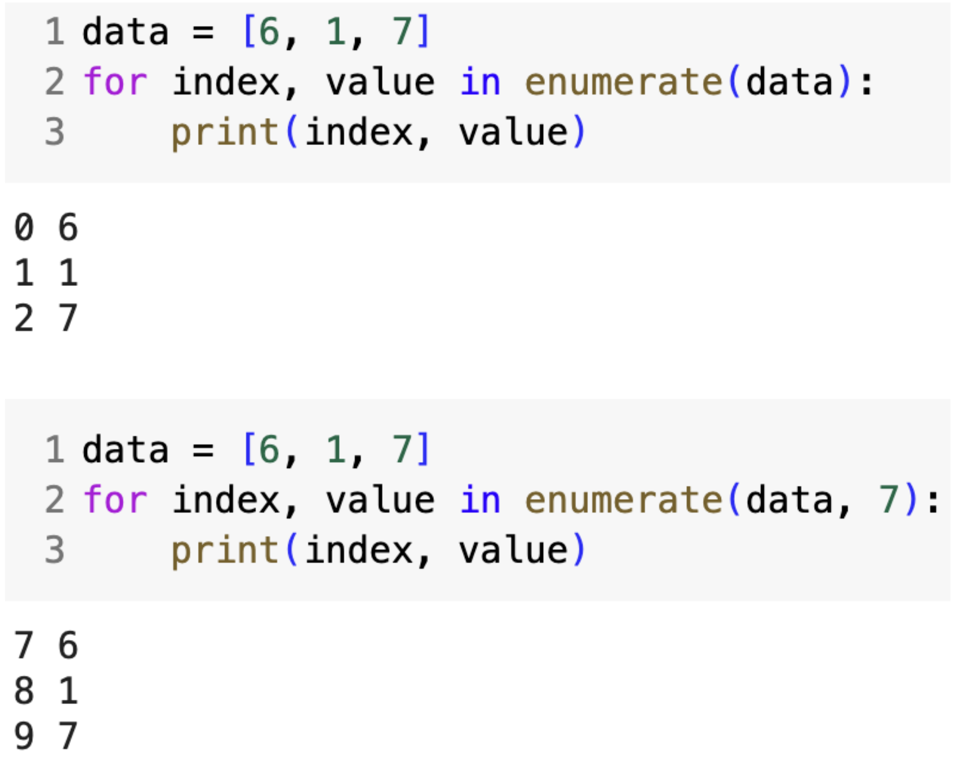
AI VIET NAM
@aivietnam.edu.vn

Built-in Functions for List

enumerate(): adds a counter to an iterable and returns it as an enumerate object (iterator with index and the value)

reversed(iterable, start)





```
data = [6, 1, 7]
for index, value in enumerate(data):
    print(index, value)
```

```
0 6
1 1
2 7
```

```
data = [6, 1, 7]
for index, value in enumerate(data, 7):
    print(index, value)
```

```
7 6
8 1
9 7
```

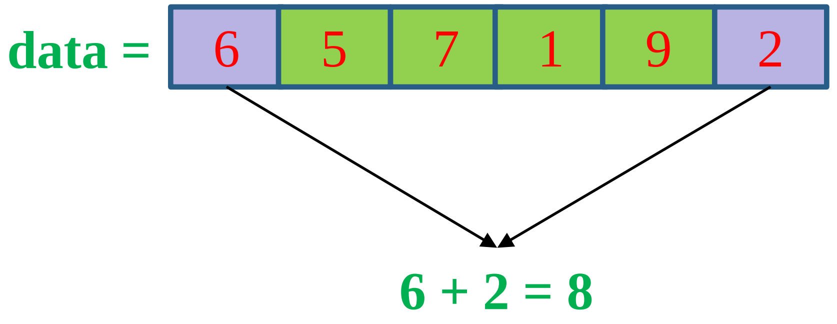
AI VIET NAM
@aivietnam.edu.vn

Outline

AI VIET NAM
@aivietnam.edu.vn

Practice

Sum of even numbers



AI VIET NAM
@aivietnam.edu.vn

Practice

Sum of even numbers

data =

6	5	7	1	9	2

For loop

```
# sum of even numbers
def even_sum_1(data):
    result = 0

    for value in data:
        if value%2 == 0:
            result = result + value

    return result

data = [6, 5, 7, 1, 9, 2]
sum_data = even_sum_1(data)
print(sum_data)
```

```
8
```

AI VIET NAM
@aivietnam.edu.vn

Practice

Sum of even numbers

data =

6	5	7	1	9	2

While loop

```
def even_sum_2(data):
    index = 0
    result = 0

    while index < len(data):
        if data[index]%2 == 0:
            result = result + data[index]
        index = index + 1

    return result

data = [6, 5, 7, 1, 9, 2]
sum_data = even_sum_2(data)
print(sum_data)
```

```
8
```

AI VIET NAM
@aivietnam.edu.vn

Practice

Two sum

- Given an array of integers *data* and an integer *target*, return indices of the two numbers such that they add up to *target*

data =

6	5	7	1	9	2

target = 8

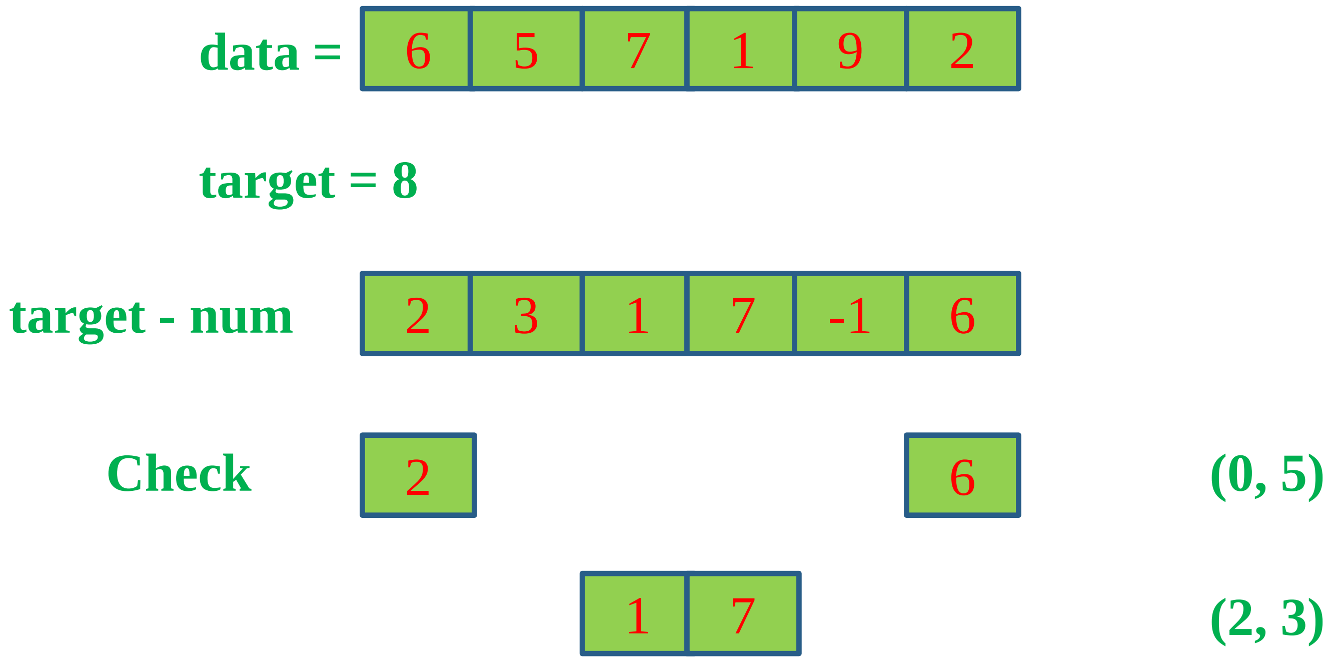
=> [2, 3] # [0, 5]

AI VIET NAM
@aivietnam.edu.vn

Practice

Two sum

- Given an array of integers *data* and an integer *target*, return indices of the two numbers such that they add up to *target*



AI VIET NAM
@aivietnam.edu.vn

Practice

Two sum

```python
def two_sum(data, target):
        num_indices = {}

        for i, num in enumerate(data):
            complement = target - num
            if complement in num_indices:

                return [num_indices[complement], i]

            num_indices[num] = i

        return []

data = [6, 5, 7, 1, 9, 2]
target = 8
result = two_sum(data, target)
print(result)
```

```
[2, 3]
```

Summary

List

- Create: nums = [1, 2, 3]
- Index: nums[0] => 1
- Slicing: nums[:2] => [1, 2]
- Add an element: nums.append(3)
- Update: nums[0] = 2
- Delete: nums.remove(3), nums.pop(0)
- Reverse: nums.reverse()
- Count: nums.count(1)
- Copy: new_nums = nums.copy()
- Sort: nums.sort(reverse=True/False)

Built-in Functions

- len(nums)
- min(nums)
- max(nums)
- sum(nums)
- reversed(nums)
- enumerate
- zip

Thanks!

Any questions?