AI VIETNAM
All-in-One Course

Data Structure

List

Quang-Vinh Dinh
Ph.D. in Computer Science

Year 2022

Outline

- Introduction
- List
- Some algorithms on List
- Addresses
- Common Errors

Abstract Data Types

```
1  # create a set
2  animals = {"cat", "dog", "tiger"}
3
4  print(type(animals))
5  print(animals)
```

```
<class 'set'>
{'dog', 'cat', 'tiger'}
```

tuple_name = (element-1, …, element-n)

dictionary_name = {key-1:value-1, …, key-n:value-n}

bracket

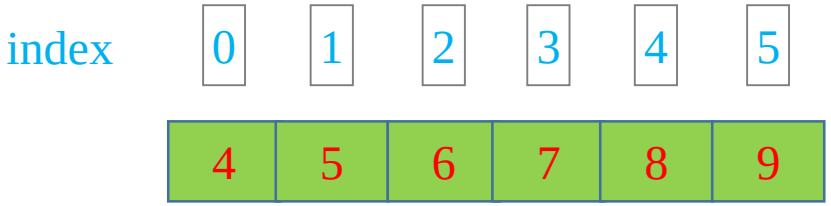
element

comma

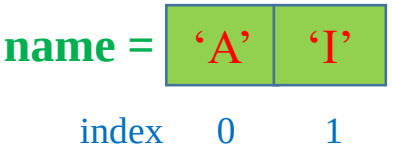
colon

List

data = [4, 5, 6, 7, 8, 9]



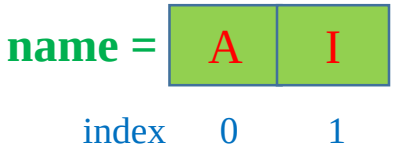
Strings



String

❖ Create and iterate a string

name = 'AI'




```
1  # create a string
2  name = 'AI'
3  print(name)
```

```
AI
```

```
1  # iterate a string
2  name = 'AI'
3  for character in name:
4      print(character)
```

```
A
I
```

```
1  # iterate a string
2  name = 'AI'
3  length = 2
4
5  for index in range(length):
6      print(name[index])
```

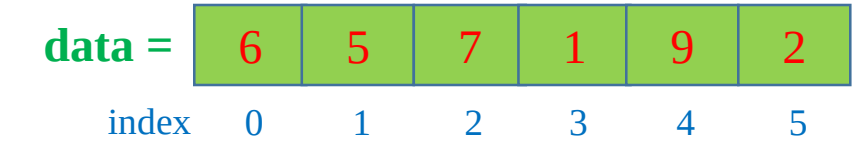
```
A
I
```

List

❖ A container that can contain elements

list_name = [element-1, …, element-n]

```
// create a list
data = [6, 5, 7, 1, 9, 2]
```

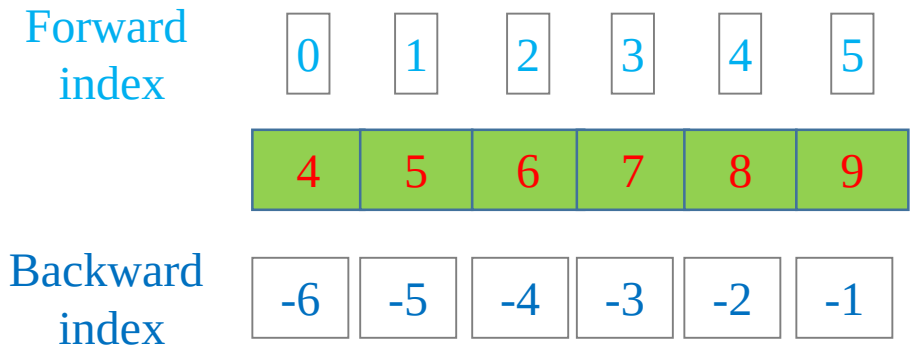



```
1.  # danh sách trống
2.  emty_list = []
3.
4.  # danh sách số tự nhiên nhỏ hơn 10
5.  my_list = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
6.
7.  # danh sách kết hợp nhiều kiểu dữ liệu
8.  mixedList = [True, 5, 'some string', 123.45]
9.  n_list = ["Happy", [2,0,1,5]]
10.
11. #danh sách các loại hoa quả
12. shoppingList = ['táo', 'chuối', 'cherries', 'dâu', 'mận']
```

List

❖ Index

data = [4, 5, 6, 7, 8, 9]



data[0] → 4

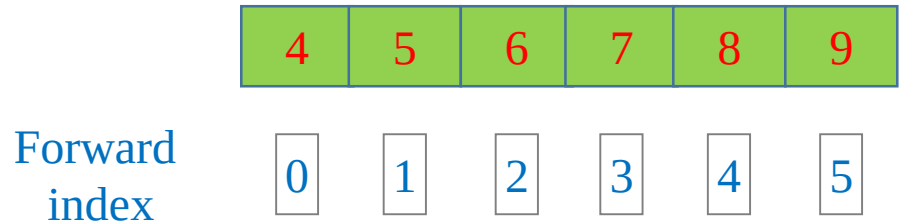
data[3] → 7

data[-1] → 9

data[-3] → 7

❖ Slicing

data = [4, 5, 6, 7, 8, 9]



data[:3]

data[2:4]

data[3:]

List

❖ Add an element

data = | 6 | 5 | 7 | 1 | 9 | 2 |

data.append(4) # thêm 4 vào vị trị cuối list

data = | 6 | 5 | 7 | 1 | 9 | 2 | 4 |

data = | 6 | 5 | 7 | 1 | 9 | 2 |

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

data = | 4 | 6 | 5 | 7 | 1 | 9 | 2 |

```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]
```

```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

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

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

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

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

❖ Updating an element

data = | 6 | 5 | 7 | 1 | 9 | 2 |

thay đổi phần tử thứ 1
data[1] = 4

data = | 6 | 4 | 7 | 1 | 9 | 2 |

❖ Add a list of elements

data = | 6 | 5 | 7 | 1 |

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

data = | 6 | 5 | 7 | 1 | 9 | 2 |

List

❖ + and * operators

data1 = | 6 | 5 | 7 |

data2 = | 1 | 9 | 2 |

nối 2 list
data = data1 + data2

data = | 6 | 5 | 7 | 1 | 9 | 2 |

data = | 6 | 5 |

nhân list với một số nguyên
data_m = data * 3

data_m = | 6 | 5 | 6 | 5 | 6 | 5 |

```
1  data1 = [6, 5, 7]
2  data2 = [1, 9, 2]
3
4  # concatenate
5  data = data1 + data2
6  print(data)
```

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

```
1  data = [6, 5]
2
3  # multiply with a number
4  data_m = data*3
5  print(data_m)
```

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

List

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

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

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

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

❖ sort() – Sắp xếp các phần tử

data =

6	5	7	1	9	2

data.sort()

data =

1	2	5	6	7	9

data =

6	5	7	1	9	2

data.sort(reverse = True)

data =

9	7	6	5	2	1

List

❖ Deleting an element

data = | 6 | 5 | 7 | 1 | 9 | 2 |

data.pop(2) # tại vị trí index = 2

data = | 6 | 5 | 1 | 9 | 2 |

data = | 6 | 5 | 7 | 1 | 9 | 2 |

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

data = | 6 | 7 | 1 | 9 | 2 |

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3  data.pop(2)  # by index
4  print(data)
```

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

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3  data.remove(2) # by value
4  print(data)
```

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

```
1  data = [6, 5, 2, 1, 9, 2]
2  print(data)
3  data.remove(2) # by value
4  print(data)
```

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

List

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

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

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  data.clear()
5  print(data)
```

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

❖ Delete elements

xóa phần tử thứ 1 và 2
del data[1:3]

data.clear()

data = []

List

index() – Trả về vị trí đầu tiên

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

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  indexOf9 = data.index(9)
5  print(indexOf9)
```

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

reverse() – Đảo ngược vị trí các phần tử

data =

6	5	7	1	9	2

data.reserse()

data =

2	9	1	7	5	6

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  data.reverse()
5  print(data)
```

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

List

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  numOf7 = data.count(7)
5  print(numOf7)
```

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

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  aCopy = data.copy()
5  print(aCopy)
```

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

count() – Trả về số lần xuất hiện của một phần tử

data =

6	5	7	1	9	2

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

copy() – copy một list

data =

6	5	7	1	9	2

data_copy = data.copy()

data_copy =

6	5	7	1	9	2

Built-in Functions for List

len(), min(), and 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

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

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

```
1  # get a number of elements
2  length = len(data)
3  print(length)
```

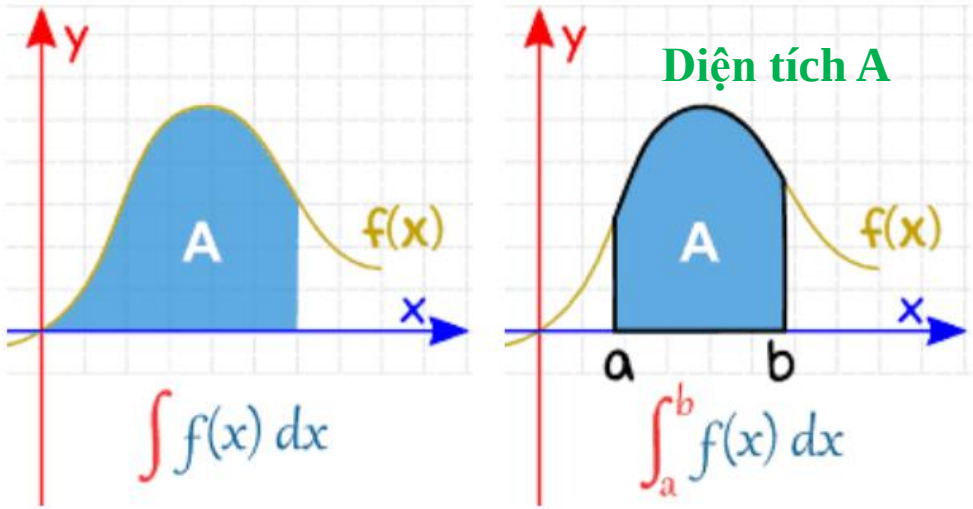
```
6
```

```
1  # get the min and max values
2  print(min(data))
3  print(max(data))
```

```
1
9
```

Example

Công thức



https://www.mathsisfun.com/calculus/integration-introduction.html

$$\mathrm{F(a)} = \int_{-\infty}^{a} f(x)d(x)$$

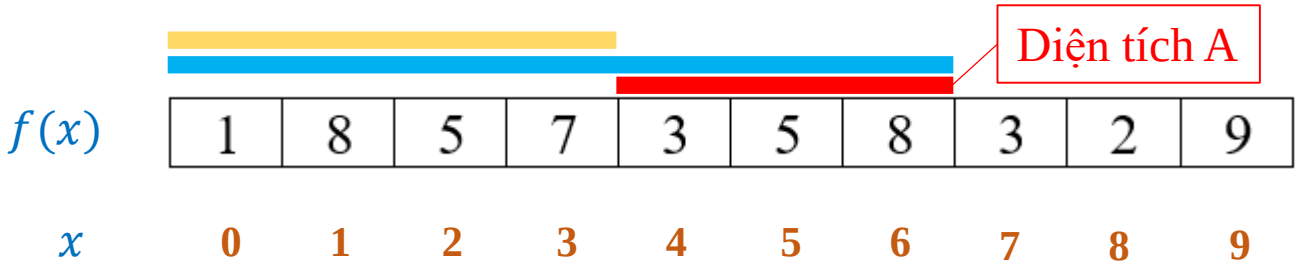
$$\mathrm{F(b)} = \int_{-\infty}^{b} f(x)d(x)$$

Diện tích A

$$\mathrm{A} = \mathrm{F(b)} - \mathrm{F(a)}$$

$$f(x) \geq 0$$

Áp dụng cho hàm rời rạc (1D)



$$\mathrm{F}(3) = \sum_{x \leq 3} f(x) = f(0) + f(1) + f(2) + f(3)$$

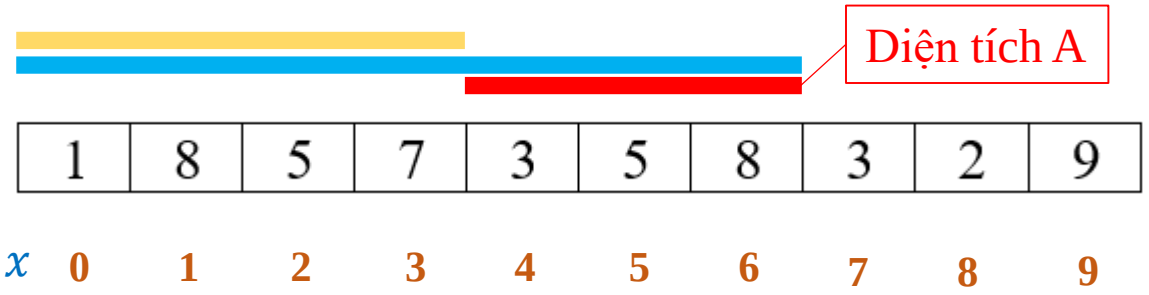
$$= 1 + 8 + 5 + 7 = 21$$

$$\mathrm{F}(6) = \sum_{x \leq 6} f(x) = 1 + 8 + 5 + 7 + 3 + 5 + 8 = 37$$

$$\mathrm{A} = \mathrm{F}(6) - \mathrm{F}(3) = \sum_{4 \leq x \leq 6} f(x) = 3 + 5 + 8 = 16$$

Example

Áp dụng cho hàm rời rạc (1D)



$$F(3) = \sum_{x \le 3} f(x) = f(0) + f(1) + f(2) + f(3)$$
$$= 1 + 8 + 5 + 7 = 21$$

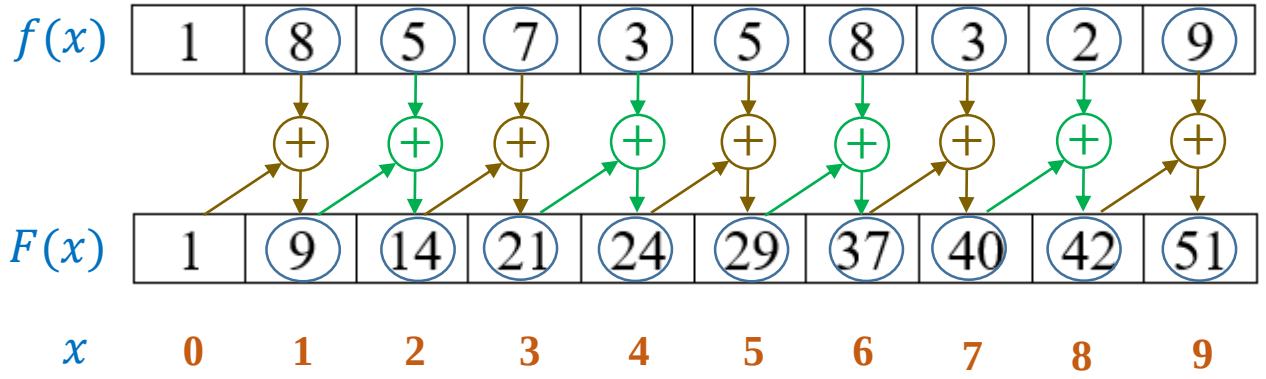
$$F(6) = \sum_{x \le 6} f(x) = 1 + 8 + 5 + 7 + 3 + 5 + 8 = 37$$

$$A = F(6) - F(3) = \sum_{4 \le x \le 6} f(x) = 3 + 5 + 8 = 16$$

Tính chất

$$F(x) = f(x) + F(x-1)$$
$$F(7) = f(7) + F(6) = 3 + 37 = 40$$

Xây dựng integral array dùng tính chất $F(x) = f(x) + F(x-1)$



Tính tổng với độ phức tạp ~ O(1)

$$\sum_{a \le x \le b} f(x) = F(b) - F(a-1)$$

$$\sum_{4 \le x \le 6} f(x) = F(6) - F(3) = 37 - 21 = 16$$

Built-in Functions

❖ sorted(aList) – Sắp xếp các phần tử

sorted(iterable, reverse=reverse)

sorted_data = sorted(data)

sorted_data = sorted(data, reverse=True)


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

sorted_data = sorted(data)
print(sorted_data)
```

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

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

sorted_data = sorted(data, reverse=True)
print(sorted_data)
```

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

Built-in Functions

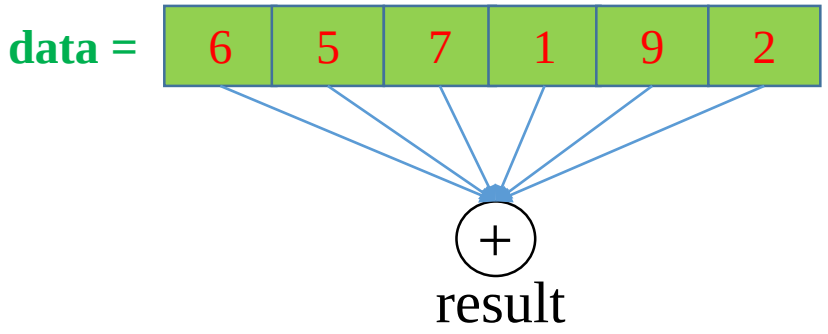
sum()

data = | 6 | 5 | 7 | 1 | 9 | 2 |

tính tổng
sum(data) = 30

```
1  data = [6, 5, 7, 1, 9, 2]
2  print(data)
3
4  summation = sum(data)
5  print(summation)
```

```
[6, 5, 7, 1, 9, 2]
30
```




```
1  # custom summation - way 1
2  def computeSummation(data):
3      result = 0
4
5      for value in data:
6          result = result + value
7
8      return result
9
10 # test
11 data = [6, 5, 7, 1, 9, 2]
12 summation = computeSummation(data)
13 print(summation)
```

```
30
```

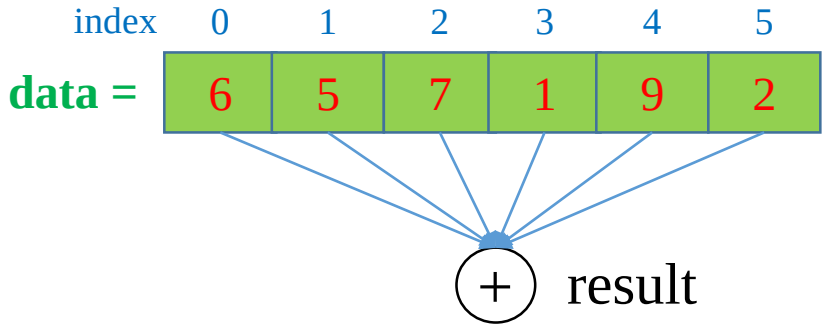
Built-in Functions

sum()

tính tổng
sum(data) = 30

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

summation = sum(data)
print(summation)
```

```
[6, 5, 7, 1, 9, 2]
30
```

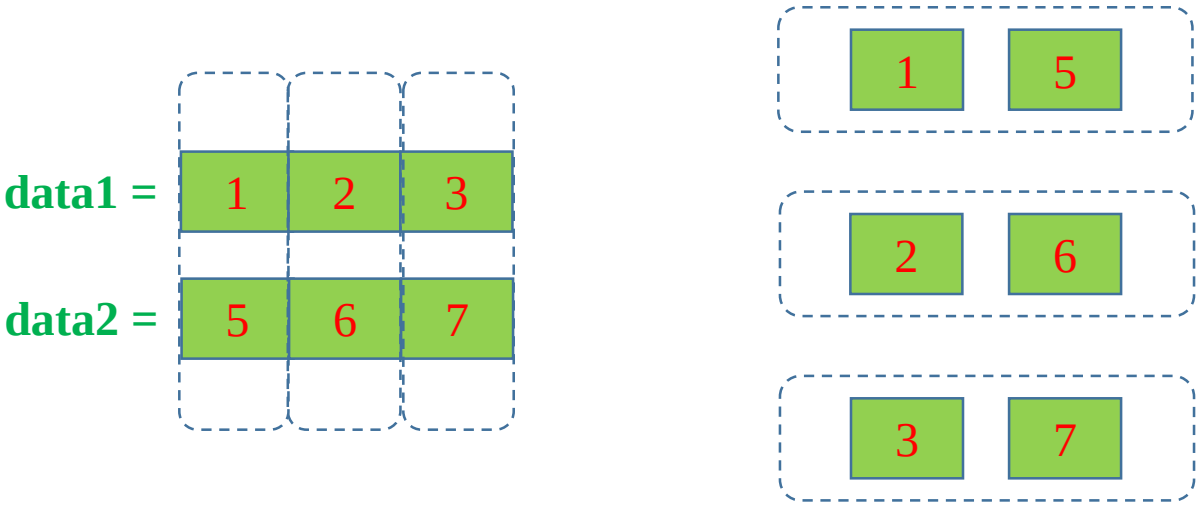



```
# custom summation - way 2
def computeSummation(data):
    result = 0

    length = len(data)
    for index in range(length):
        result = result + data[index]

    return result

# test
data = [6, 5, 7, 1, 9, 2]
summation = computeSummation(data)
print(summation)
```

```
30
```

Built-in Functions

zip()




```python
l1 = [1, 2, 3]
l2 = [5, 6, 7]

# print in pairs
length = len(l1)
for i in range(length):
    print(l1[i], l2[i])
```

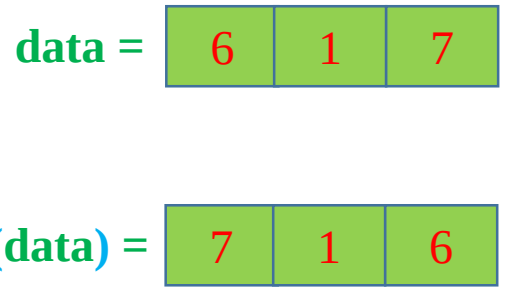
```
1 5
2 6
3 7
```

```python
l1 = [1, 2, 3]
l2 = [5, 6, 7]

# print in pairs
for v1, v2 in zip(l1, l2):
    print(v1, v2)
```

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

Built-in Functions

reversed()




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

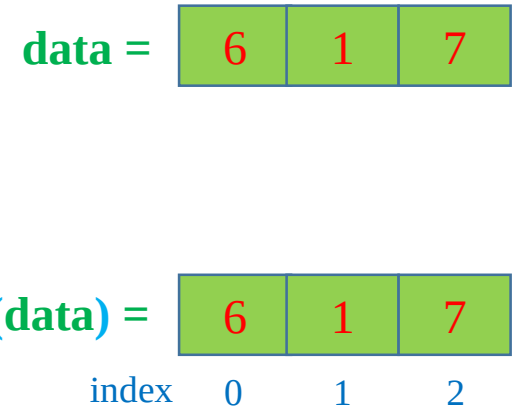
```
6
1
7
```

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

```
7
1
6
```

Built-in Functions

enumerate()




```python
# get index and value
data = [6, 1, 7]

length = len(data)
for index in range(length):
    print(index, data[index])
```

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

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

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

Examples

Sum of even numbers

data = 


```python
# sum of even number
def sum1(data):
    result = 0

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

    return result

# test
data = [6, 5, 7, 1, 9, 2]
summation = sum1(data)
print(summation)
```

```
8
```

Sum of elements with even indices

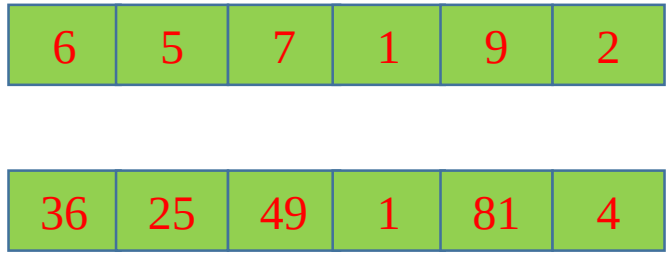
data = 


```python
# sum of numbers with even indices
def sum2(data):
    result = 0

    length = len(data)
    for index in range(length):
        if index%2 == 0:
            result = result + data[index]

    return result

# test
data = [6, 5, 7, 1, 9, 2]
summation = sum2(data)
print(summation)
```

```
22
```

Examples

square(aList)




```python
# square function
def square(data):
    result = []

    for value in data:
        result.append(value*value)

    return result

# test
data = [6, 5, 7, 1, 9, 2]
print(data)
data_s = square(data)
print(data_s)
```

```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

List Comprehension

```python
# square function
def square(data):
    result = []

    for value in data:
        result.append(value*value)

    return result
```

omitted

```python
# using list comprehension
def square(data):
    result = [value*value for value in data]

    return result
```

added

```python
# using list comprehension
def square(data):
    result = [value*value for value in data]

    return result

# test
data = [6, 5, 7, 1, 9, 2]
print(data)
data_s = square(data)
print(data_s)
```

```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

List Comprehension

Sigmoid Function

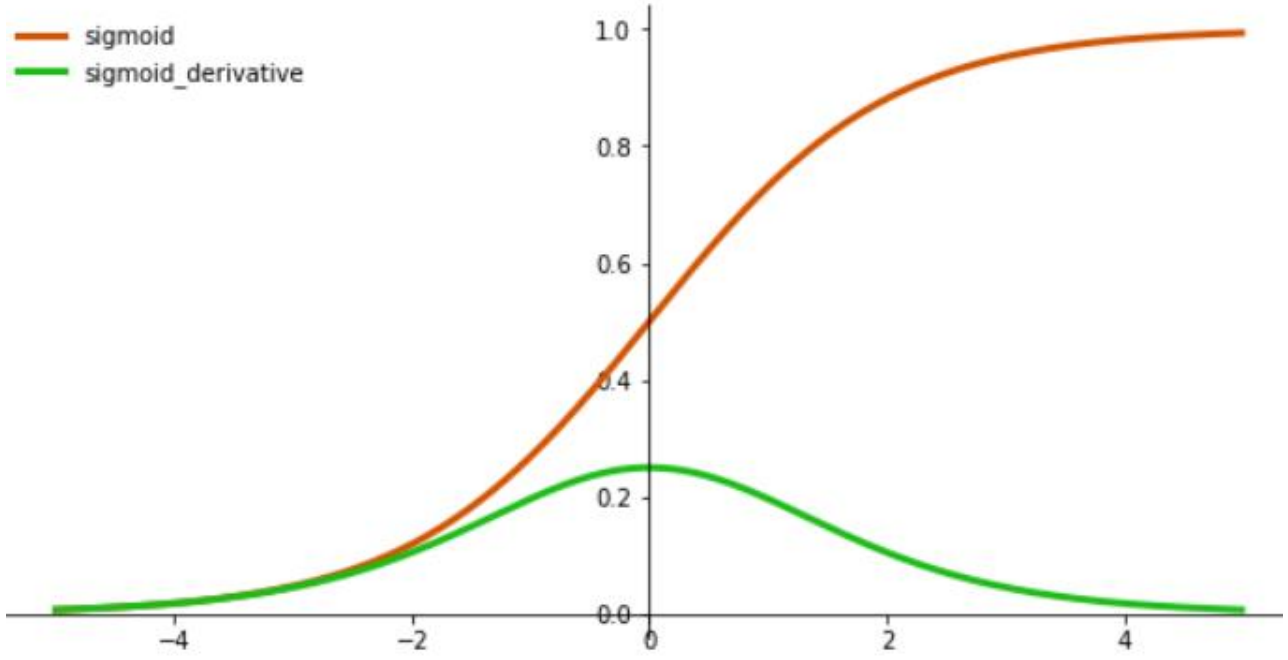
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

data =	1	5	-4	3	-2

data_a = sigmoid(data)

data_a =	0.731	0.993	0.017	0.95	0.119




```python
import math

# sigmoid function
def sigmoid(x):
    result = 1 / (1 + math.exp(-x))
    return result

def sigmoidForList(data):
    result = [sigmoid(x) for x in data]
    return result

# test
data = [1, 5, -4, 3, -2]
print(data)
data_a = sigmoidForList(data)
print(data_a)
```

List Comprehension

ReLU Function

$$\mathrm{ReLU}(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$$

$$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0 \end{cases}$$

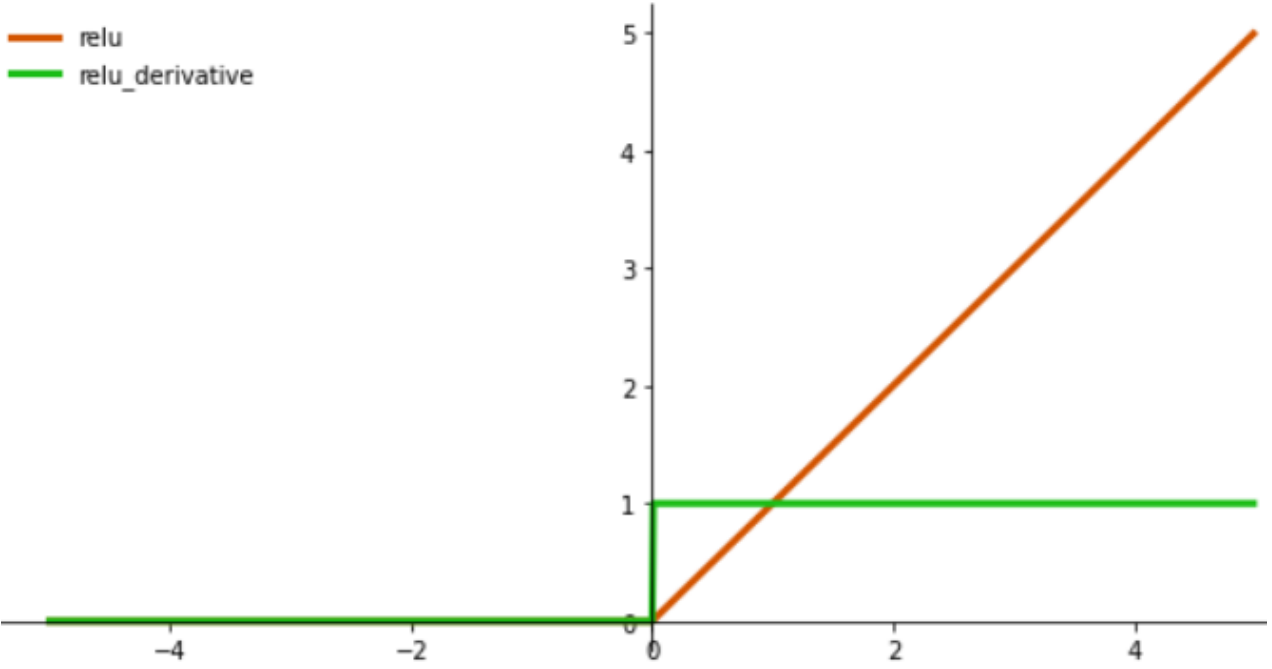
data =

1	5	-4	3	-2

data_a = ReLU(data)

data_a =

1	5	0	3	0




```python
def relu(x):
    result = 0
    if x > 0:
        result = x

    return result

def reluForList(data):
    result = [relu(x) for x in data]
    return result

# test
data = [1, 5, -4, 3, -2]
print(data)
data_a = reluForList(data)
print(data_a)
```

```
[1, 5, -4, 3, -2]
[1, 5, 0, 3, 0]
```

List Comprehension

ReLU Function

$$\mathrm{ReLU}(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x & \text{if } x > 0 \end{cases}$$

$$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{if } x \le 0 \\ 1 & \text{if } x > 0 \end{cases}$$

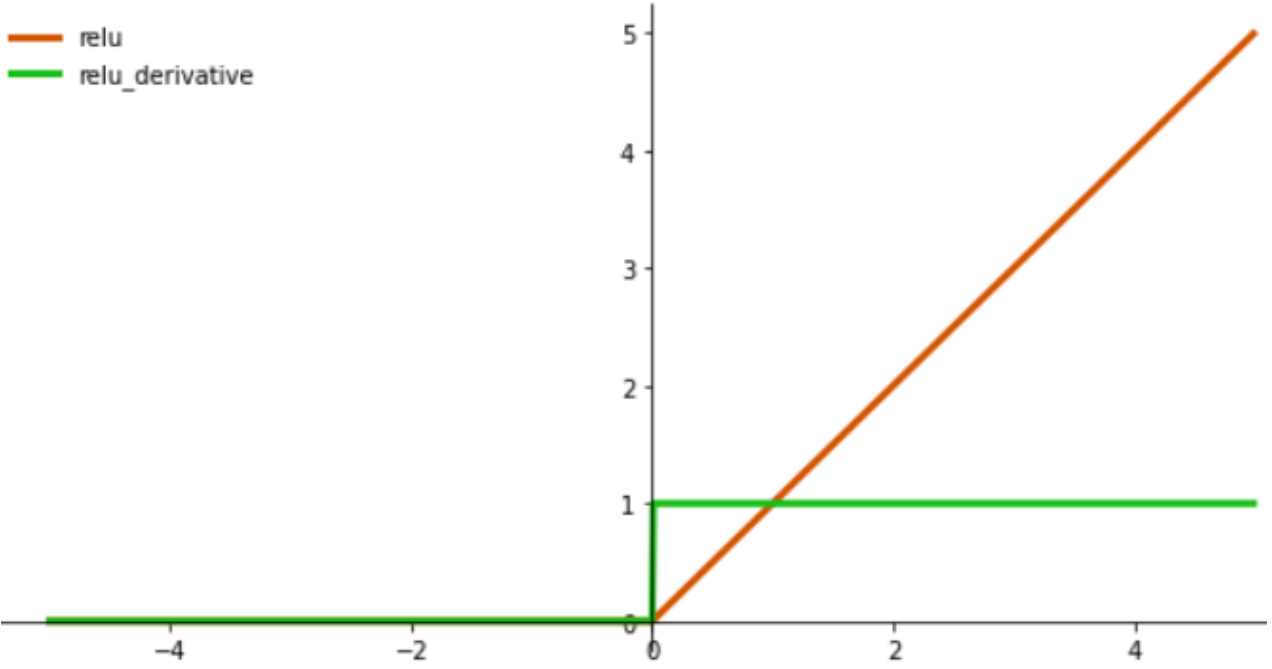
data =

1	5	-4	3	-2

data_a = ReLU(data)

data_a =

1	5	0	3	0




```
2      result = 0
3      if x > 0:
4          result = x
```

```
1  # relu function
2  def relu(data):
3      result = [x if x>0 else 0 for x in data]
4      return result
5
6  # test
7  data = [1, 5, -4, 3, -2]
8  print(data)
9  data_a = relu(data)
10 print(data_a)
```

```
[1, 5, -4, 3, -2]
[1, 5, 0, 3, 0]
```

AI VIETNAM
All-in-One Course

List Comprehension

[condition_to_branch_x for x in data condition_to_filter_x]

```python
# quiz 1
data = [1, 5, -4, 3, -2]
print(data)

data_a = [x if x>0 else 0 for x in data]
print(data_a)
```

```python
# quiz 2
data = [1, 5, -4, 3, -2]
print(data)

data_a = [x if x>0 for x in data]
print(data_a)
```

```python
# quiz 3
data = [1, 5, -4, 3, -2]
print(data)

data_a = [x for x in data if x>0]
print(data_a)
```

```python
# quiz 4
data = [1, 5, -4, 3, -2]
print(data)

data_a = [x for x in data if x>0 else 0]
print(data_a)
```

List Sorting

```python
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]
```

```python
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]
```

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

sorted_data = sorted(data)
print(sorted_data)
```

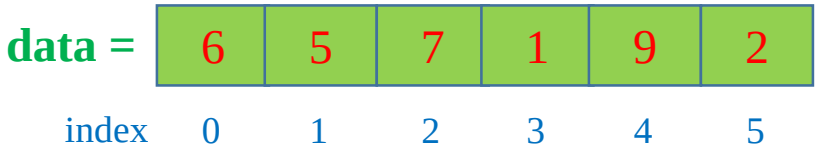
```
[6, 5, 7, 1, 9, 2]
[1, 2, 5, 6, 7, 9]
```

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

sorted_data = sorted(data, reverse=True)
print(sorted_data)
```

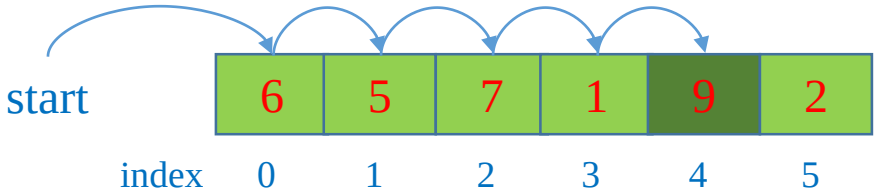
```
[6, 5, 7, 1, 9, 2]
[9, 7, 6, 5, 2, 1]
```

Algorithms on List

❖ Linear searching

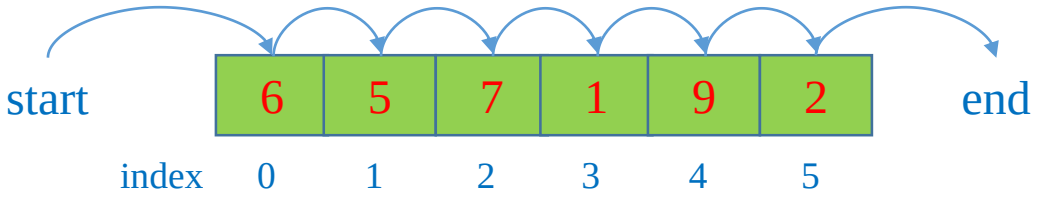


Searching for 9

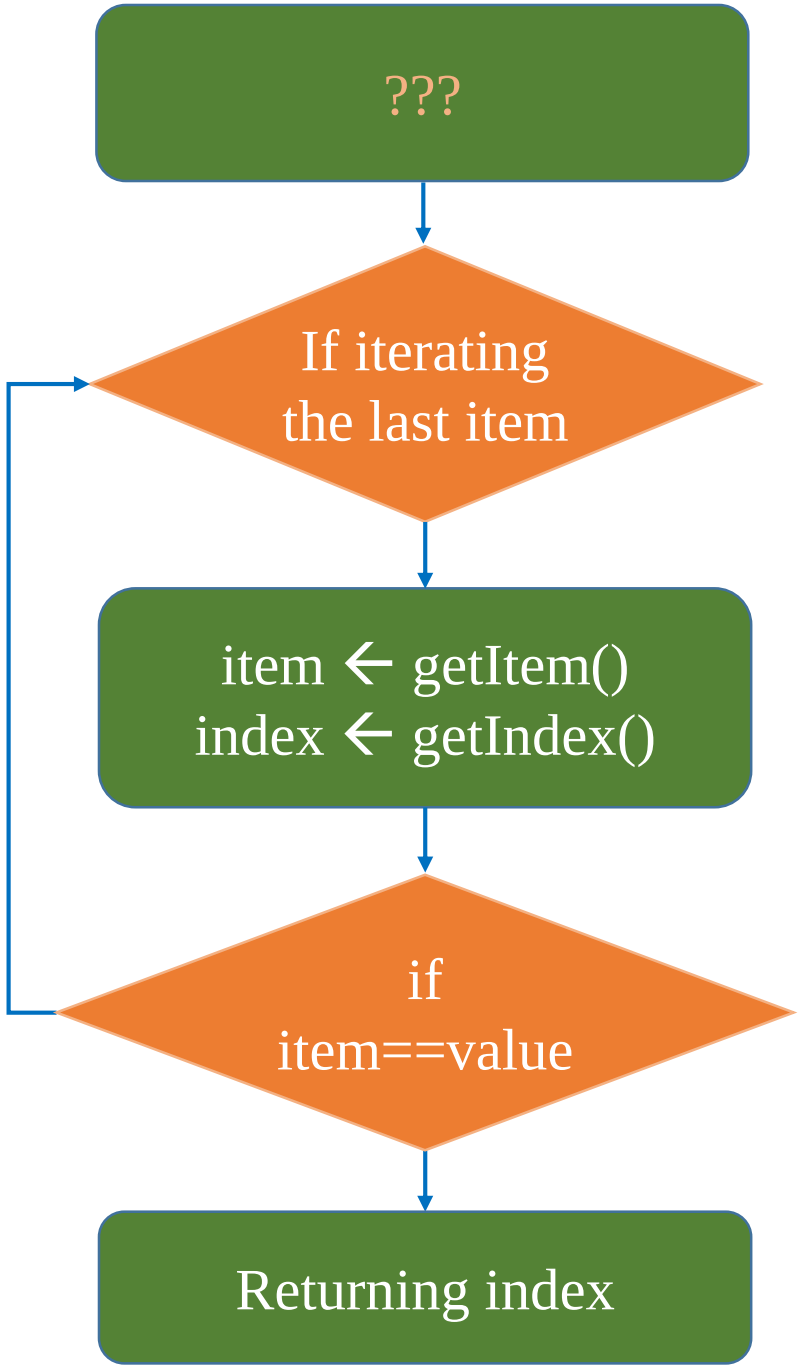


Returning 4

Searching for 8



Returning ?



Algorithms on List

❖ **Sorting using min(), remove(), and append()**

data =

6	5	7	1	9	2

result = [ ]

min(data) = 1

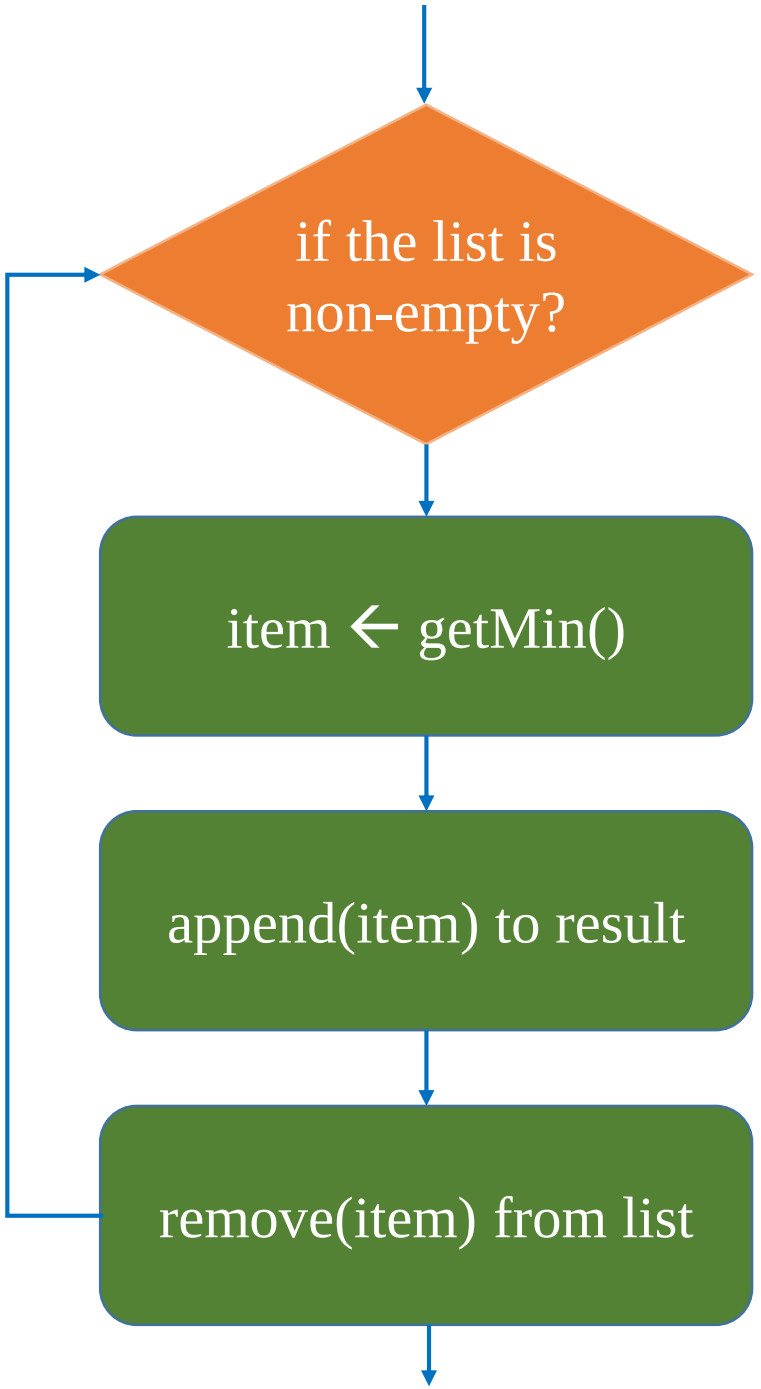
result.append(1) =

1

data.remove(1) =

6	5	7	9	2

. . .



Algorithms on List

❖ **Sorting using min(), remove(), and append()**

data =

6	5	7	9	2

result =

1

min(data) = 2

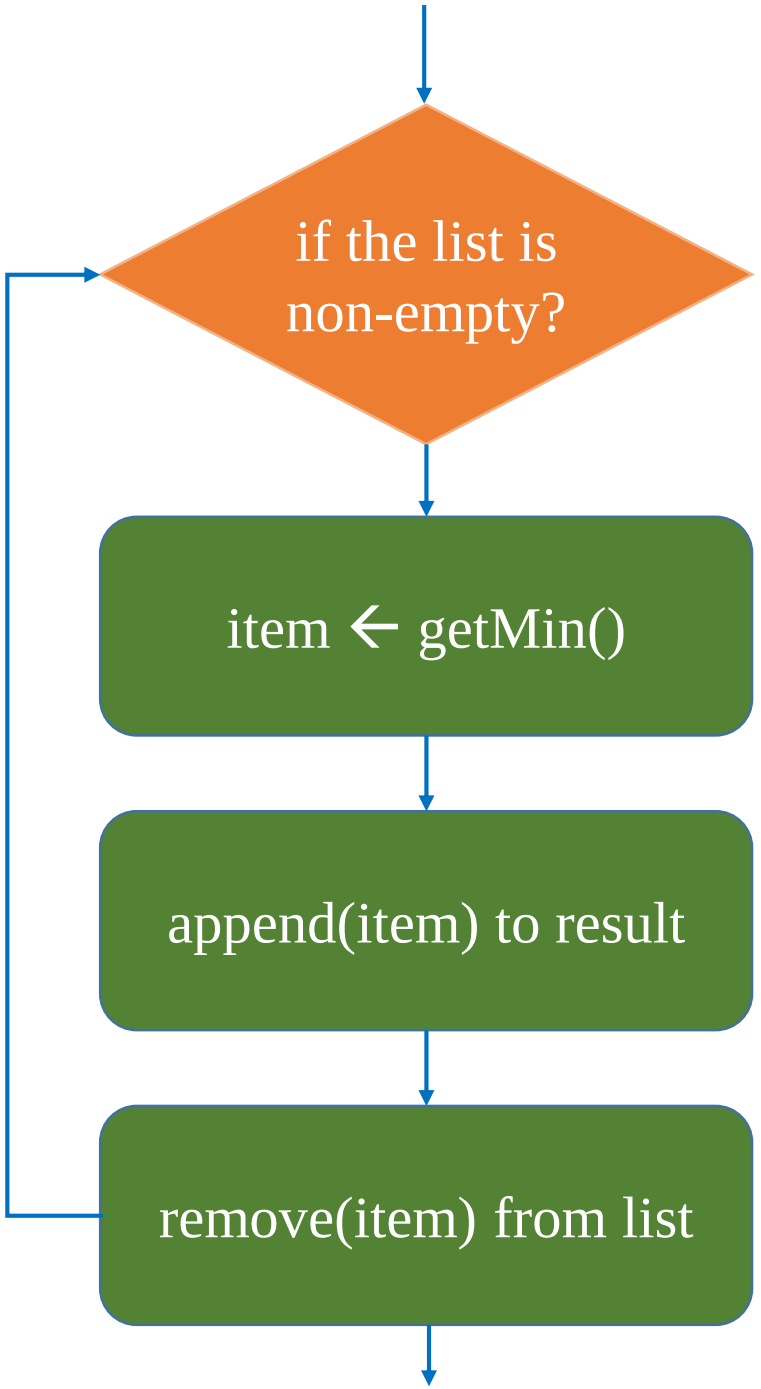
result.append(2) =

1	2

data.remove(2) =

6	5	7	9

. . .



Mutable and Immutable

```python
# immutable
def square(data):
    result = []
    for value in data:
        result.append(value*value)

    return result

# test
data = [6, 5, 7, 1, 9, 2]
print(data)

data_s = square(data)
print(data_s)
```

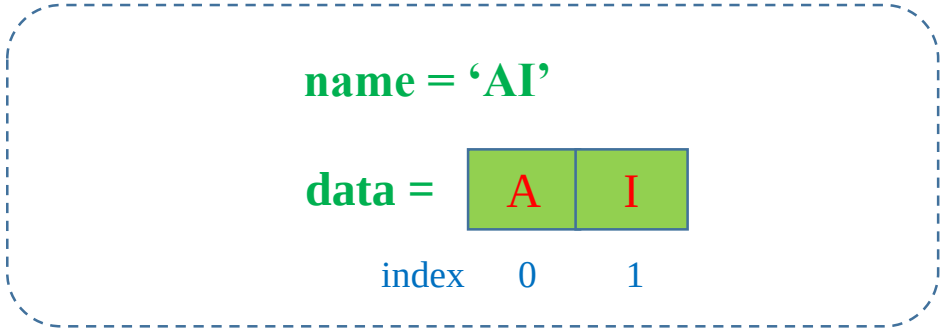
```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

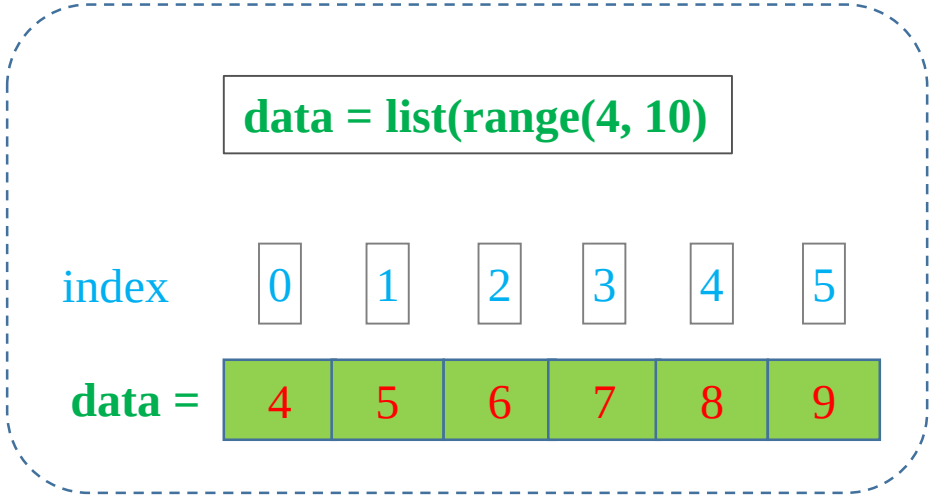
```python
# mutable
def square(data):
    length = len(data)

    for i in range(length):
        value = data[i]
        data[i] = value*value

# test
data = [6, 5, 7, 1, 9, 2]
print(data)

square(data)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

Converting to List

aList ← list(iterable)






```
1  name = 'AI'
2  data = list(name)
3
4  print(name)
5  print(data)
```

```
AI
['A', 'I']
```

```
1  data = list(range(4, 10))
2  print(data)
```

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

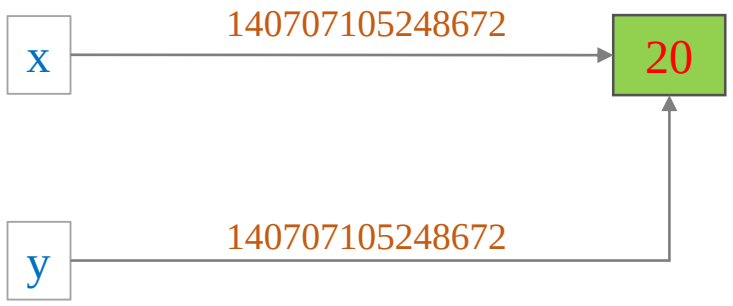
Outline

- Introduction
- List
- Some algorithms on List
- Addresses
- Common Errors

Variables and Addresses

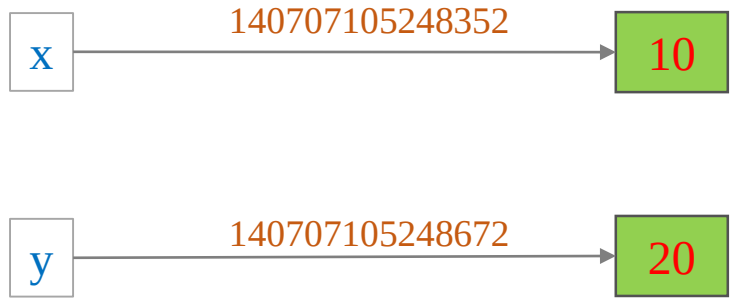
```
1  x = 20
2  y = x
3
4  print(x)
5  print(y)
6  print(id(x))
7  print(id(y))
```

```
20
20
140707105248672
140707105248672
```



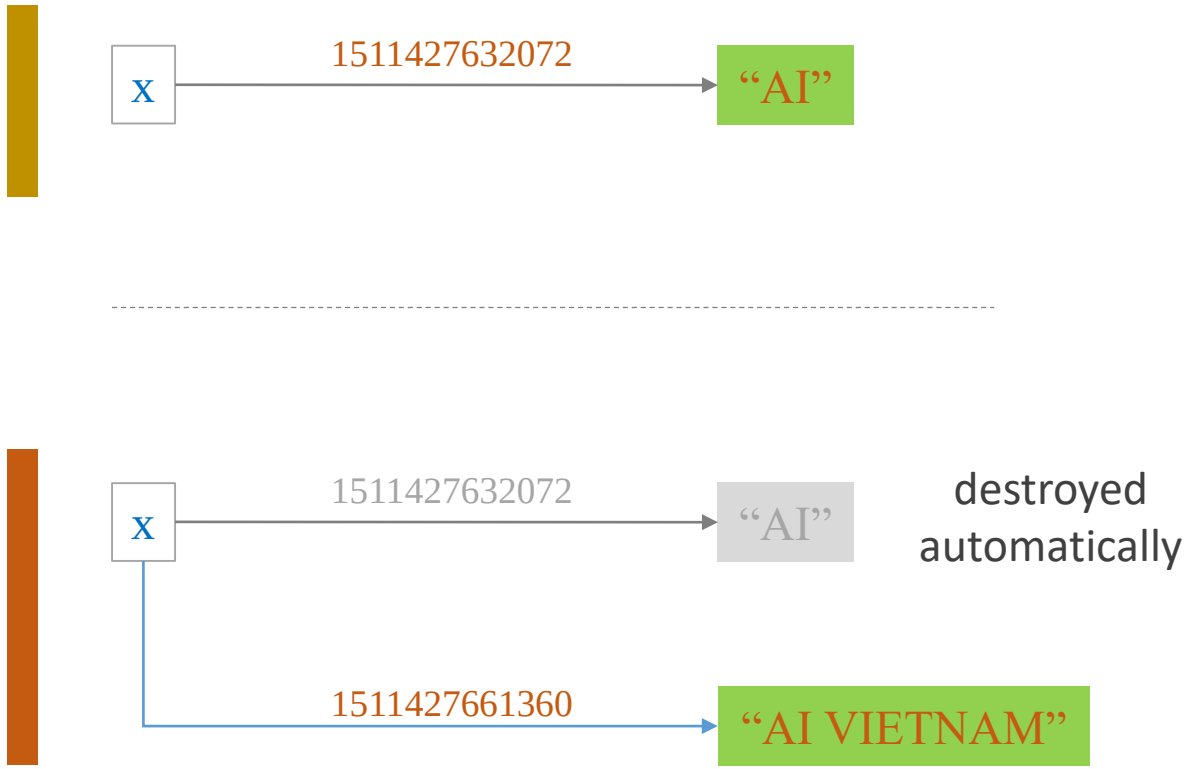

```
1  x = 20
2  y = x
3  x = 10
4
5  print(x)
6  print(y)
7  print(id(x))
8  print(id(y))
```

```
10
20
140707105248352
140707105248672
```



Variables and Addresses

- Immutable types: Cannot be changed in place
- Including ints, floats, strings, and tuples



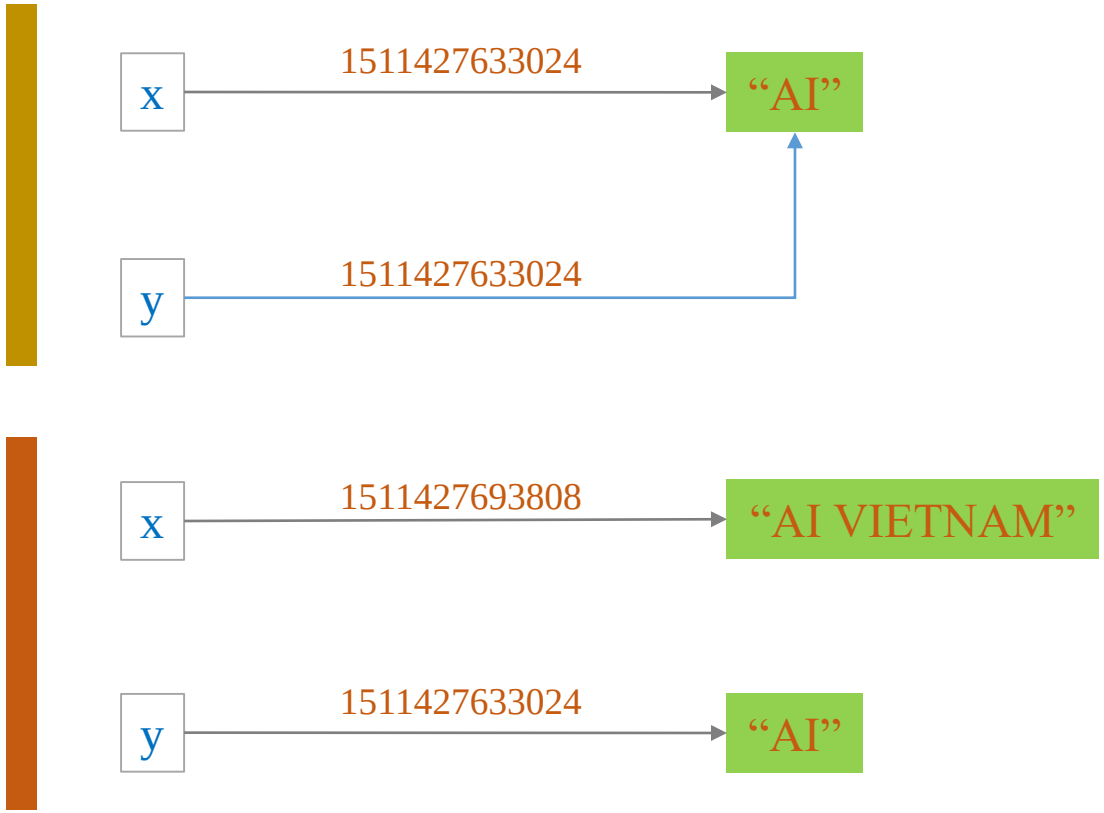

```
1  # immutable
2
3  x = "AI"
4  print(x)
5  print(id(x))
6  print()
7
8  x = x + " VIETNAM"
9  print(x)
10 print(id(x))
```

```
AI
1511427632072
```

```
AI VIETNAM
1511427661360
```

Variables and Addresses

- Immutable types: Cannot be changed in place
- Including ints, floats, strings, and tuples




```python
# immutable

x = "AI"
y = x
print('x: ', x)
print('y: ', y)
print('x address: ', id(x))
print('y address: ', id(y))
print()

x = x + " VIETNAM"
print('x: ', x)
print('x address: ', id(x))
print('y: ', y)
print('y address: ', id(y))
```

```
x:  AI
y:  AI
x address:  1511427633024
y address:  1511427633024
```

```
x:  AI VIETNAM
x address:  1511427693808
y:  AI
y address:  1511427633024
```

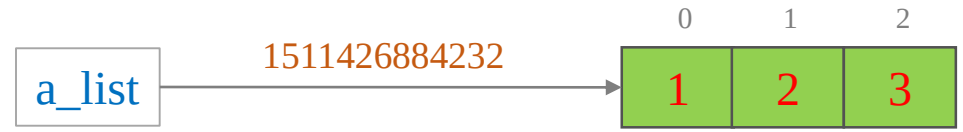
Variables and Addresses

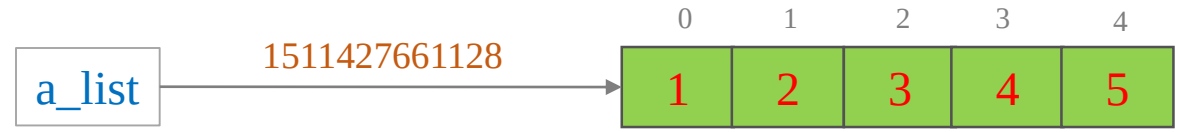
```
1  # aivietnam
2
3  a_list = [1, 2, 3]
4  print('a_list: ', a_list)
5  print('a_list address: ', id(a_list))
```

```
a_list:  [1, 2, 3]
a_list address:  1511426884232
```

```
1  a_list = a_list + [4, 5]
2  print('a_list: ', a_list)
3  print('a_list address: ', id(a_list))
```

```
a_list:  [1, 2, 3, 4, 5]
a_list address:  1511427661128
```

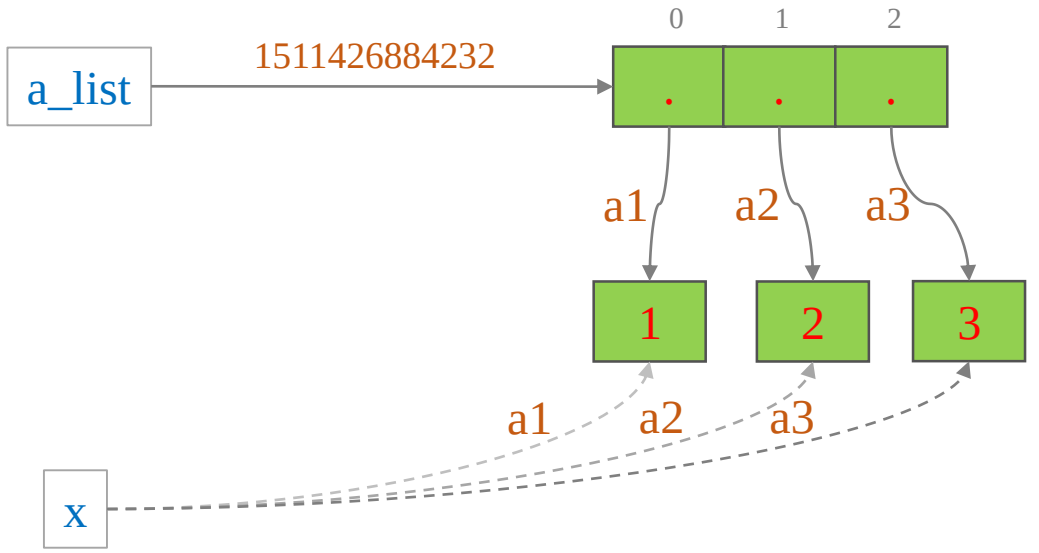




Variables and Addresses

```python
# aivietnam

a_list = [1, 2, 3]
for i in range(3):
    print(f'a_list[{i}] address is {id(a_list[i])}')

print()
for x in a_list:
    print(f'x address is {id(x)}')
```

```
a_list[0] address is 140707105248064
a_list[1] address is 140707105248096
a_list[2] address is 140707105248128

x address is 140707105248064
x address is 140707105248096
x address is 140707105248128
```

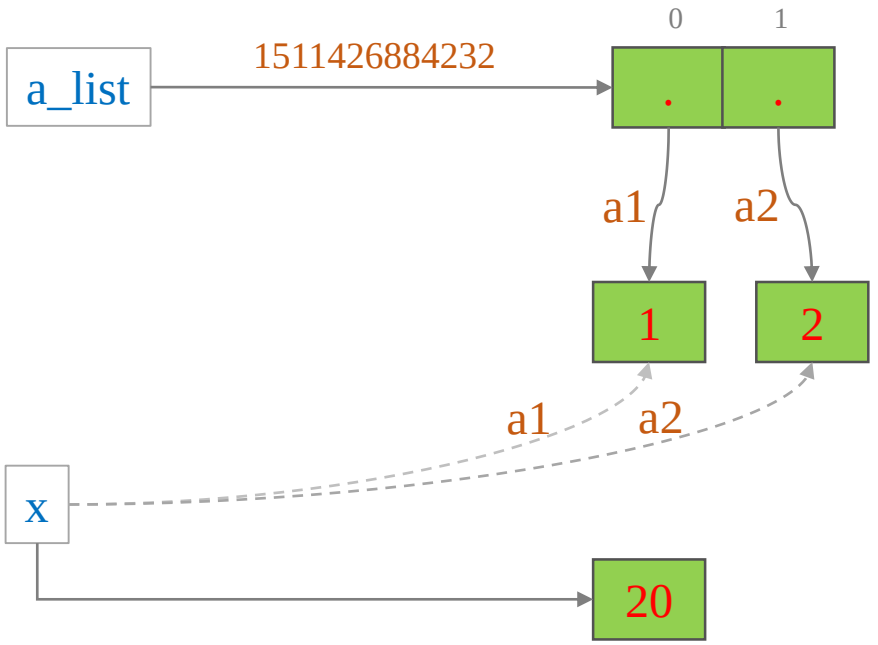


a1 = 140707105248064
a2 = 140707105248096
a3 = 140707105248128

Variables and Addresses

```
1  # aivietnam
2
3  a_list = [1, 2]
4  print(f'a_list is {a_list}')
5
6  for x in a_list:
7      x = 20
8
9  print(f'a_list is {a_list}')
```

```
a_list is [1, 2]
a_list is [1, 2]
```

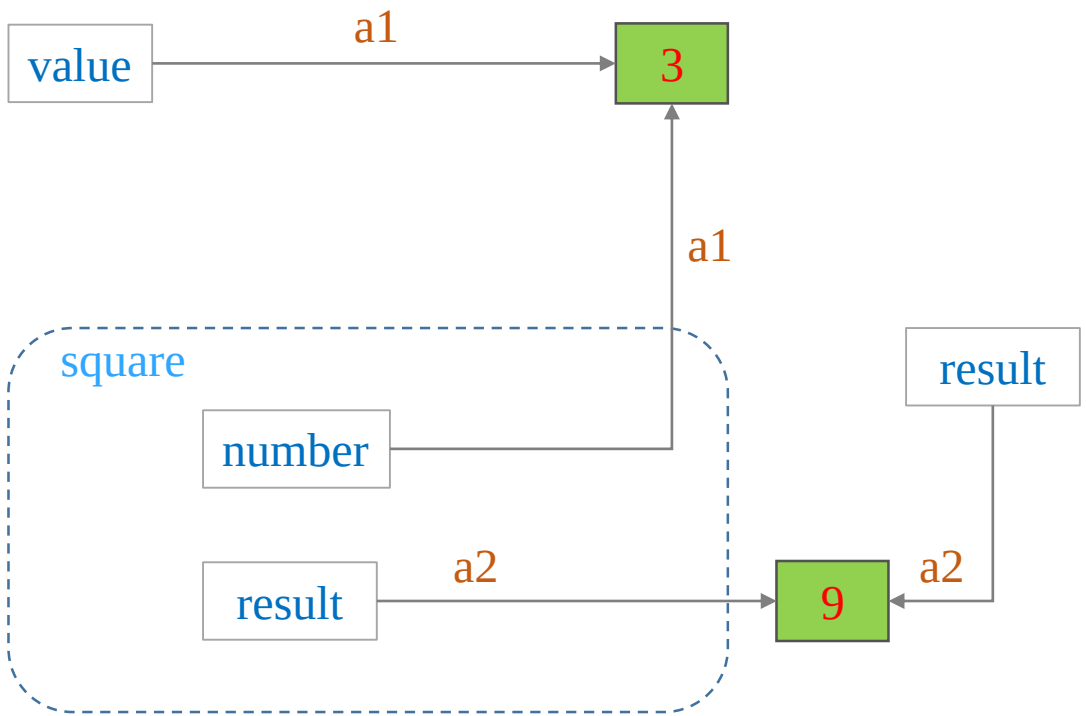


a1 = 140707105248064
a2 = 140707105248096

Variables and Addresses

```python
# aivietnam

def square(number):
    result = number*number

    print(f'number address is {id(number)}')
    print(f'result address is {id(result)}')

    return result

# test the function square
value = 3
print(f'value address is {id(value)}')

result = square(value)
print(f'result address is {id(result)}')
```

```
value address is 140707105248128
number address is 140707105248128
result address is 140707105248320
result address is 140707105248320
```

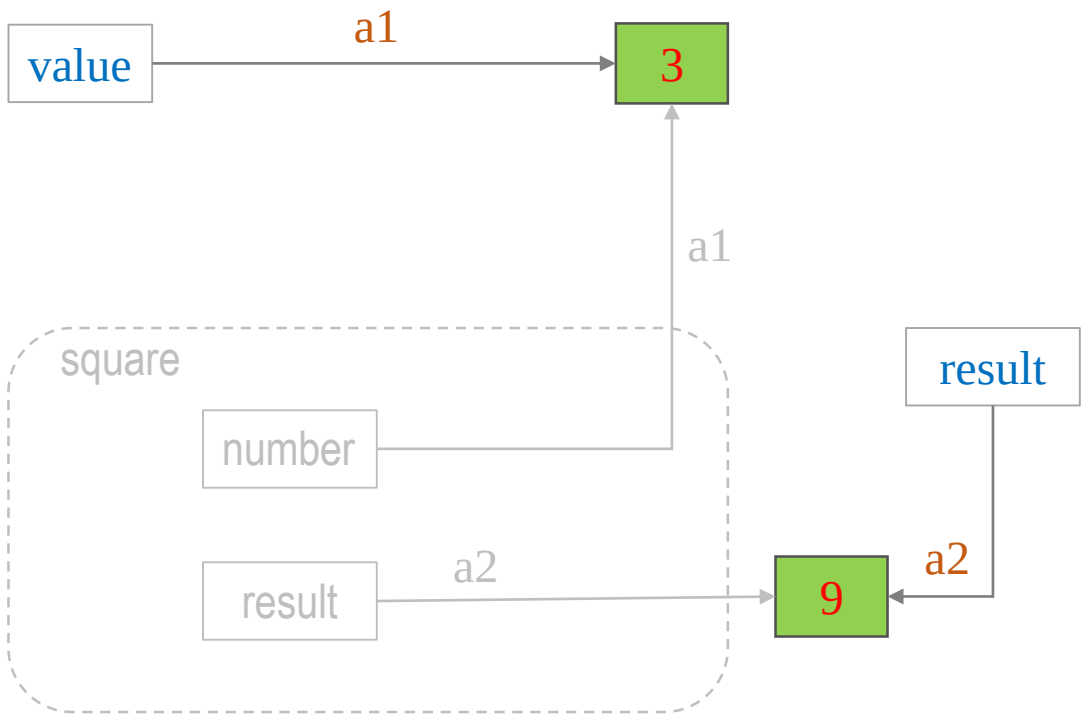


a1 = 140707105248128
a2 = 140707105248320

Variables and Addresses

```python
# aivietnam

def square(number):
    result = number*number

    print(f'number address is {id(number)}')
    print(f'result address is {id(result)}')

    return result

# test the function square
value = 3
print(f'value address is {id(value)}')

result = square(value)
print(f'result address is {id(result)}')
```

```
value address is 140707105248128
number address is 140707105248128
result address is 140707105248320
result address is 140707105248320
```



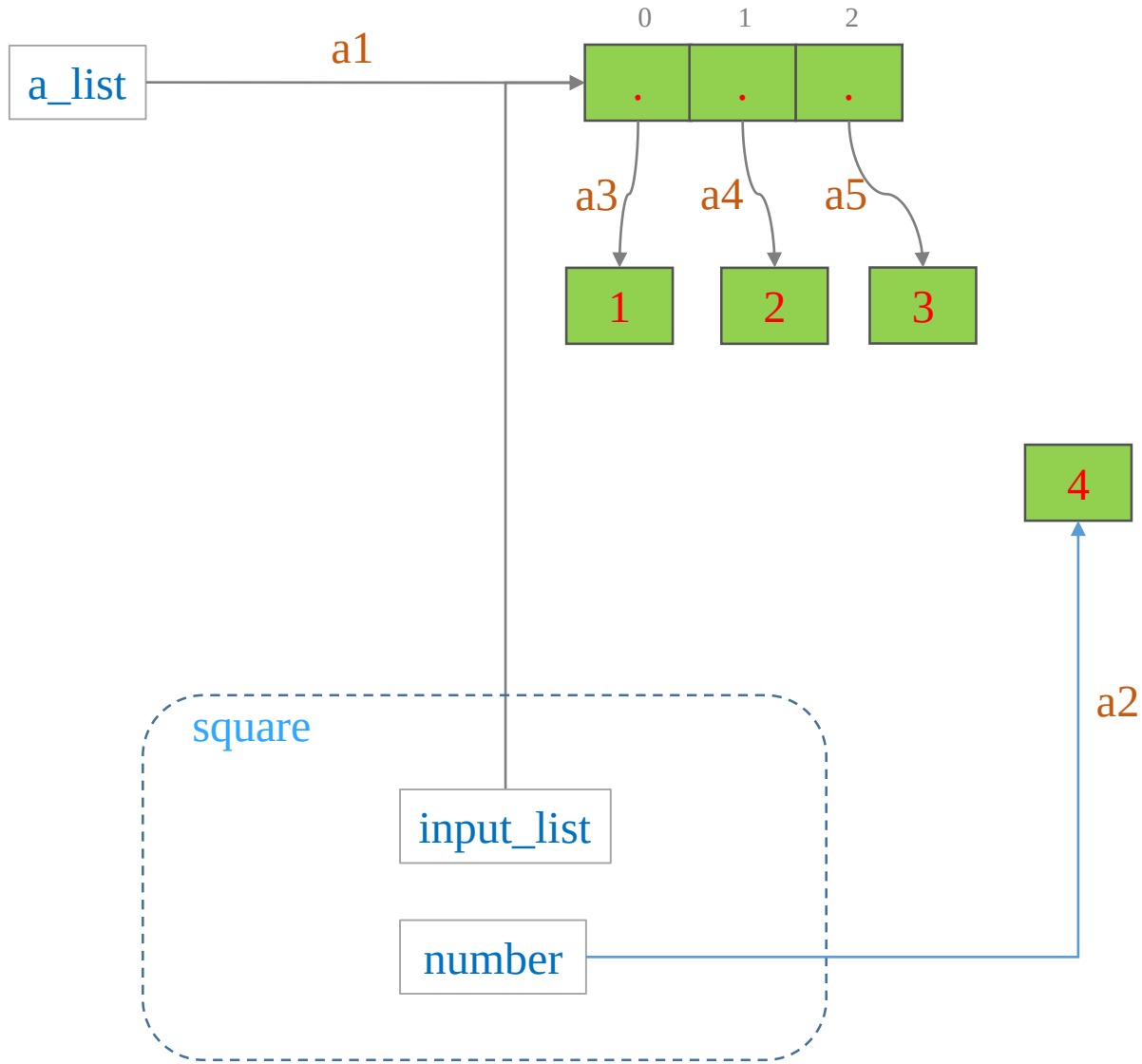
a1 = 140707105248128
a2 = 140707105248320

```python
# aivietnam

a_list = [1, 2, 3]
print(f'a_list address is {id(a_list)}')
print(f'a_list is {a_list} \n')

def add(input_list, number):
    input_list.append(number)
    print(f'input_list address is {id(input_list)}')

# test the add function
add(a_list, 4)
print(f'a_list is {a_list}')
```

```
a_list address is 1511427661128
a_list is [1, 2, 3]

input_list address is 1511427661128
a_list is [1, 2, 3, 4]
```

```python
# aivietnam

a_list = [1, 2, 3]

def add(input_list, number):
    input_list.append(number)
    print(f'number address is {id(number)}')

# test the add function
add(a_list, 4)
print(f'a_list is {a_list}')

print(f'a_list[3] address is {id(a_list[3])}')
```

```
number address is 140707105248160
a_list is [1, 2, 3, 4]
a_list[3] address is 140707105248160
```



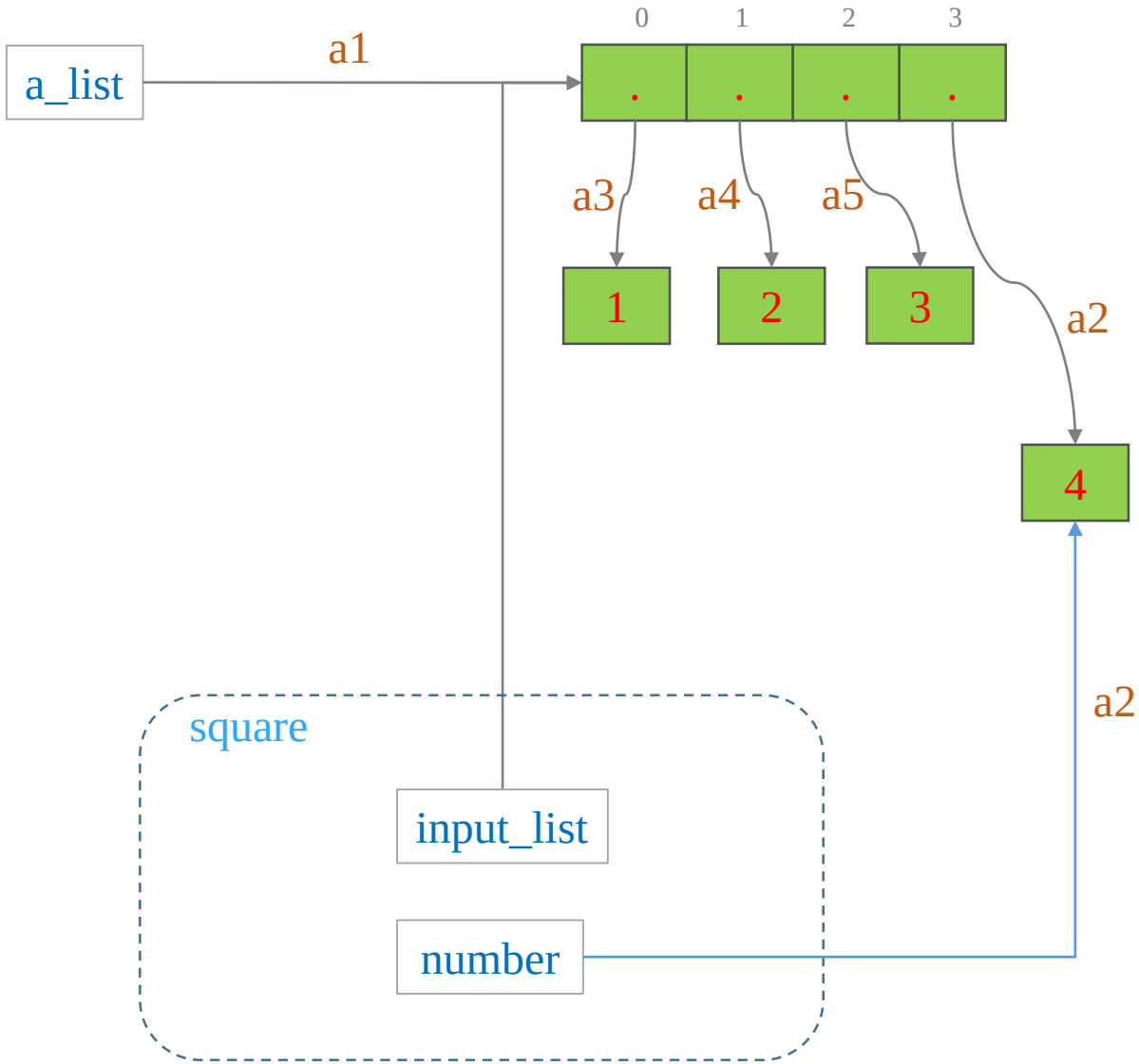
a1 = 1511428290376
a2 = 140707105248160

```python
# aivietnam

a_list = [1, 2, 3]
print(f'a_list address is {id(a_list)}')
print(f'a_list is {a_list} \n')

def add(input_list, number):
    input_list.append(number)
    print(f'input_list address is {id(input_list)}')

# test the add function
add(a_list, 4)
print(f'a_list is {a_list}')
```

```
a_list address is 1511427661128
a_list is [1, 2, 3]

input_list address is 1511427661128
a_list is [1, 2, 3, 4]
```

```python
# aivietnam

a_list = [1, 2, 3]

def add(input_list, number):
    input_list.append(number)
    print(f'number address is {id(number)}')

# test the add function
add(a_list, 4)
print(f'a_list is {a_list}')

print(f'a_list[3] address is {id(a_list[3])}')
```

```
number address is 140707105248160
a_list is [1, 2, 3, 4]
a_list[3] address is 140707105248160
```

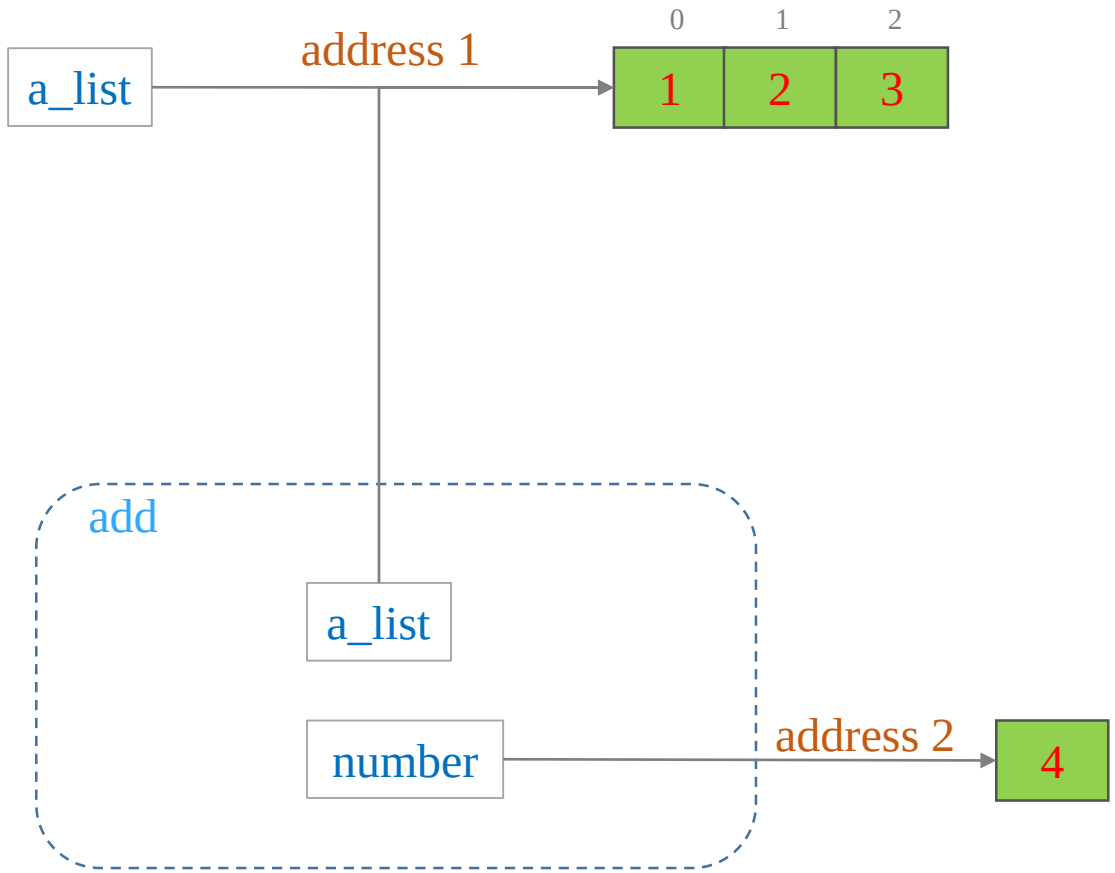


a1 = 1511428290376
a2 = 140707105248160

Variables and Addresses

```python
# aivietnam

a_list = [1, 2, 3]
print(f'a_list is {a_list}')

def add(a_list, number):
    a_list = a_list + [number]

    return a_list

# test the add function
a_list = add(a_list, 4)
print(f'a_list is {a_list}')
```

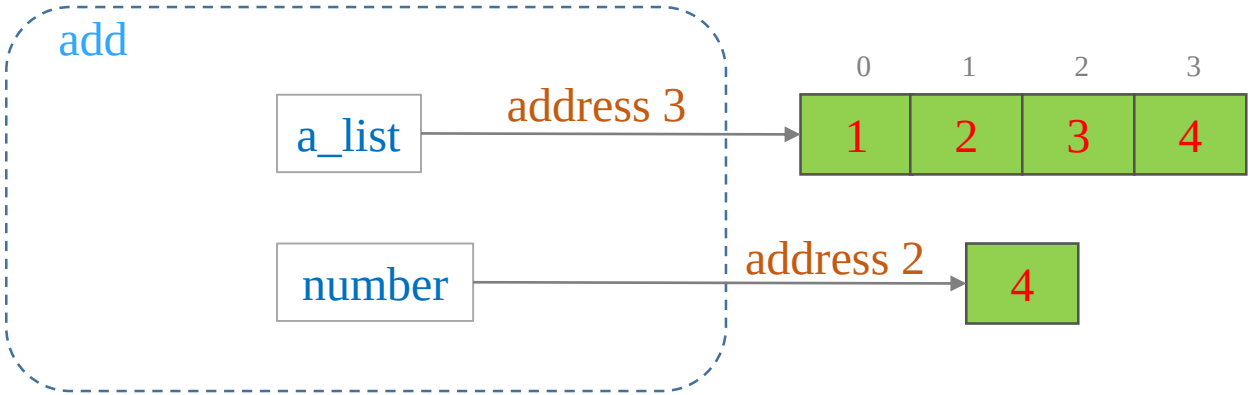
```
a_list is [1, 2, 3]
a_list is [1, 2, 3, 4]
```



Variables and Addresses

```python
# aivietnam

a_list = [1, 2, 3]
print(f'a_list is {a_list}')

def add(a_list, number):
    a_list = a_list + [number]

    return a_list

# test the add function
a_list = add(a_list, 4)
print(f'a_list is {a_list}')
```

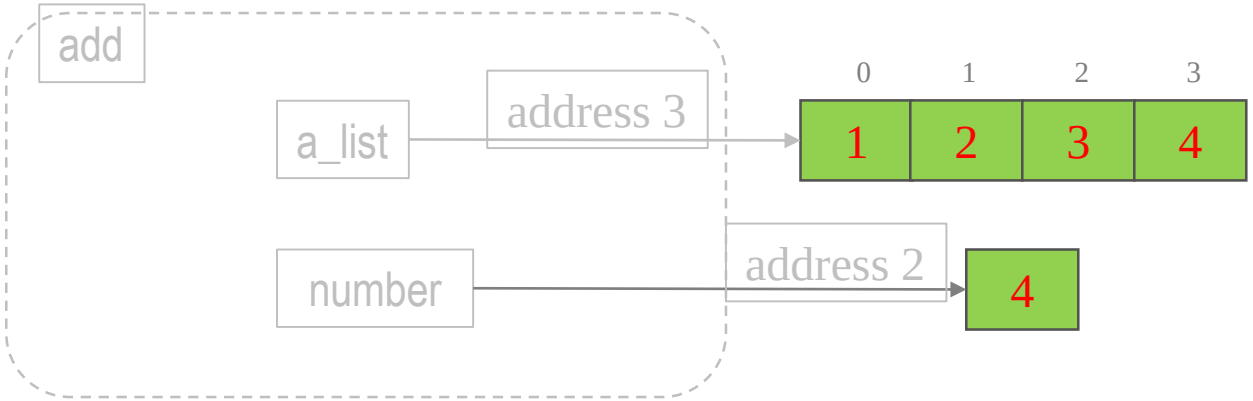
```
a_list is [1, 2, 3]
a_list is [1, 2, 3, 4]
```



Variables and Addresses

```python
# aivietnam

a_list = [1, 2, 3]
print(f'a_list is {a_list}')

def add(a_list, number):
    a_list = a_list + [number]

    return a_list

# test the add function
a_list = add(a_list, 4)
print(f'a_list is {a_list}')
```

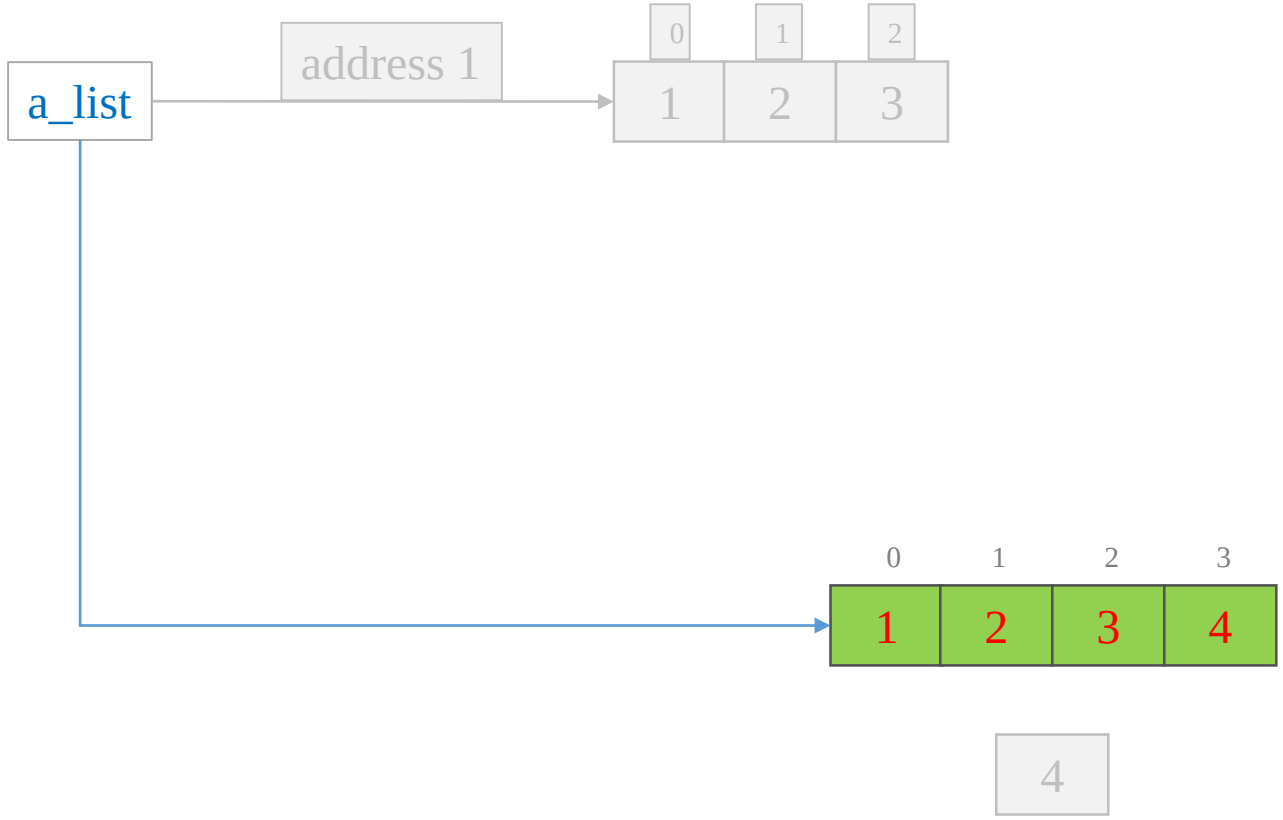
```
a_list is [1, 2, 3]
a_list is [1, 2, 3, 4]
```



Variables and Addresses

```
1  # aivietnam
2
3  a_list = [1, 2, 3]
4  print(f'a_list is {a_list}')
5
6  def add(a_list, number):
7      a_list = a_list + [number]
8
9      return a_list
10
11 # test the add function
12 a_list = add(a_list, 4)
13 print(f'a_list is {a_list}')
```

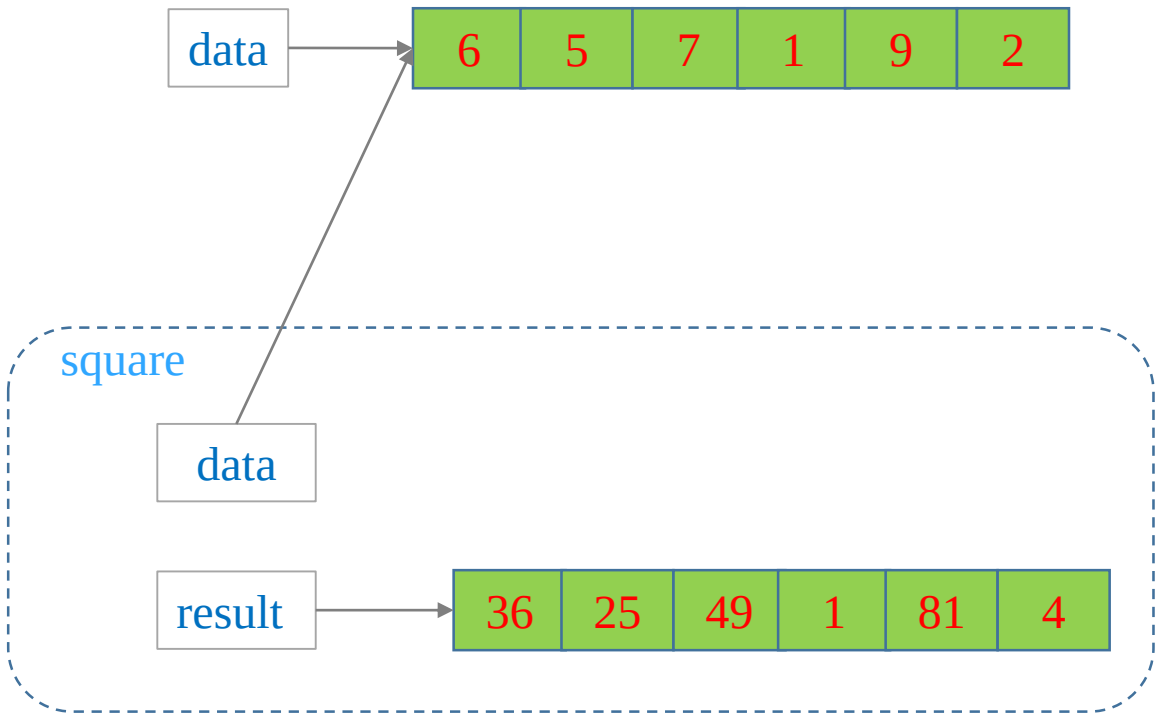
```
a_list is [1, 2, 3]
a_list is [1, 2, 3, 4]
```



Mutable and Immutable

```python
# immutable
def square(data):
    result = []
    for value in data:
        result.append(value*value)

    return result

# test
data = [6, 5, 7, 1, 9, 2]
print(data)

data_s = square(data)
print(data_s)
```

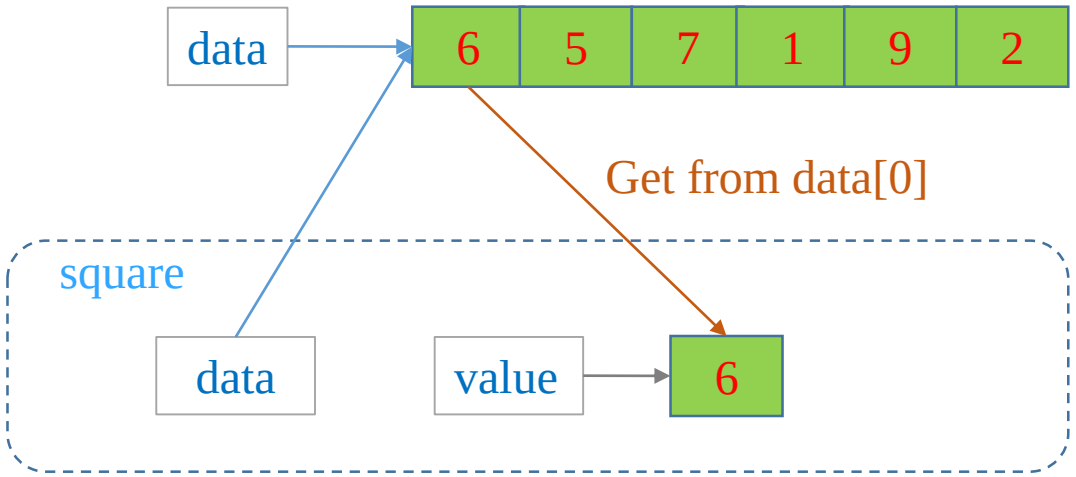
```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

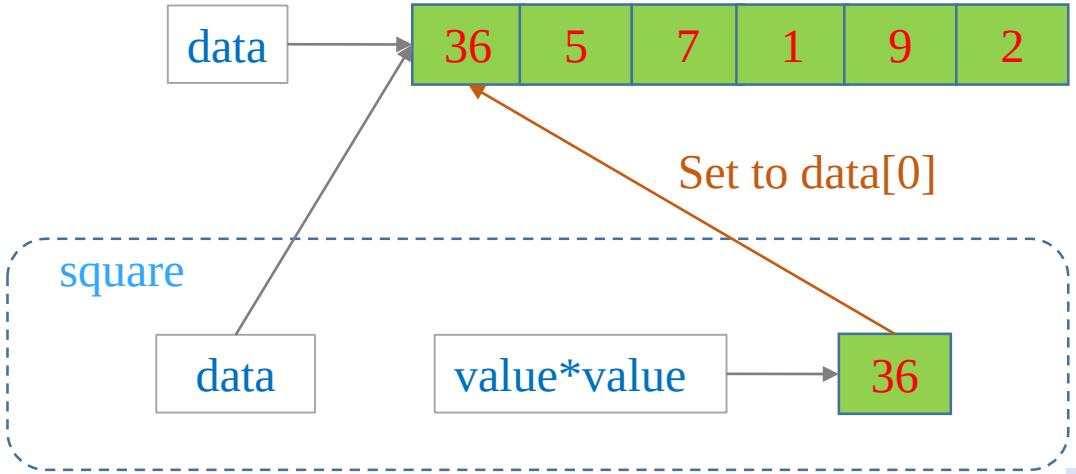


Mutable and Immutable

```python
# mutable
def square(data):
    length = len(data)

    for i in range(length):
        value = data[i]
        data[i] = value*value

# test
data = [6, 5, 7, 1, 9, 2]
print(data)

square(data)
print(data)
```

```
[6, 5, 7, 1, 9, 2]
[36, 25, 49, 1, 81, 4]
```

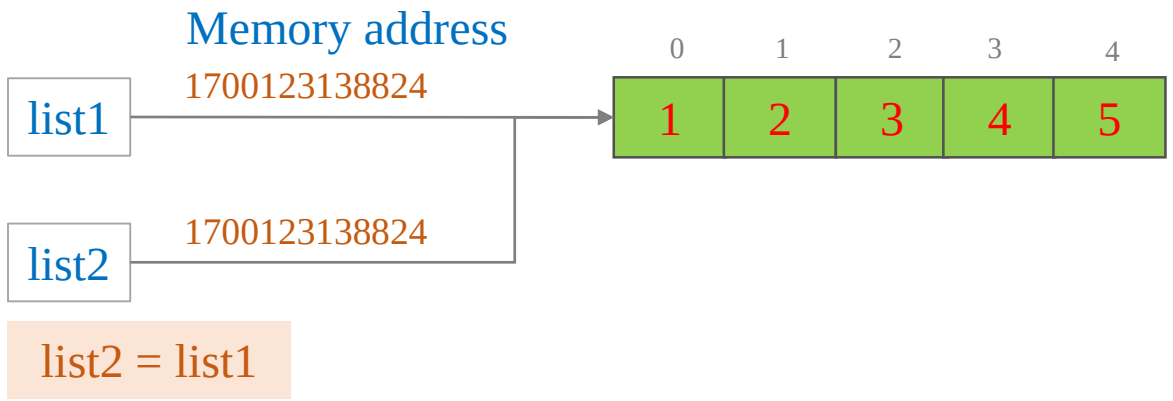




List Copy

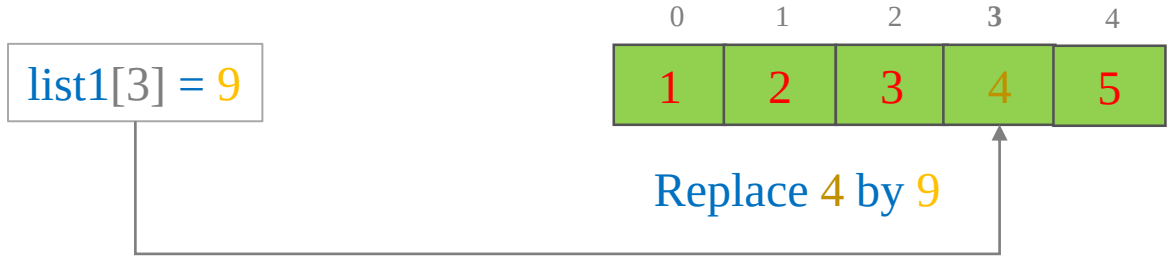
```python
# aivietnam - copy in python

list1 = [1, 2, 3, 4, 5]
list2 = list1

print('List 1: ', list1)
print('List 2: ', list2)
```

```
List 1:  [1, 2, 3, 4, 5]
List 2:  [1, 2, 3, 4, 5]
```



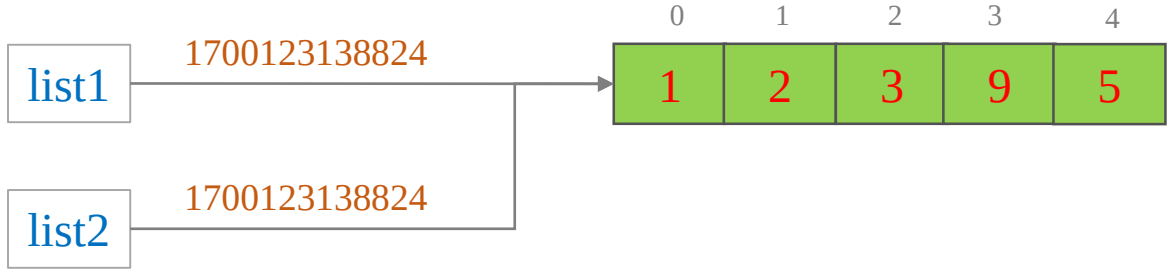

```python
list1[3] = 9
print('List 1: ', list1)
print('List 2: ', list2)
```

```
List 1:  [1, 2, 3, 9, 5]
List 2:  [1, 2, 3, 9, 5]
```




```python
print('List1 address: ', id(list1))
print('List2 address: ', id(list2))
```

```
List1 address:  1700123138824
List2 address:  1700123138824
```



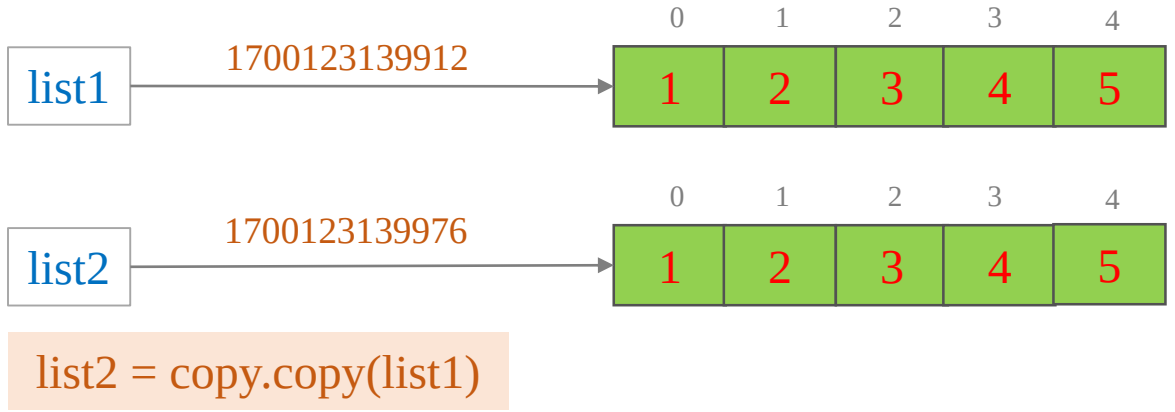
List Copy

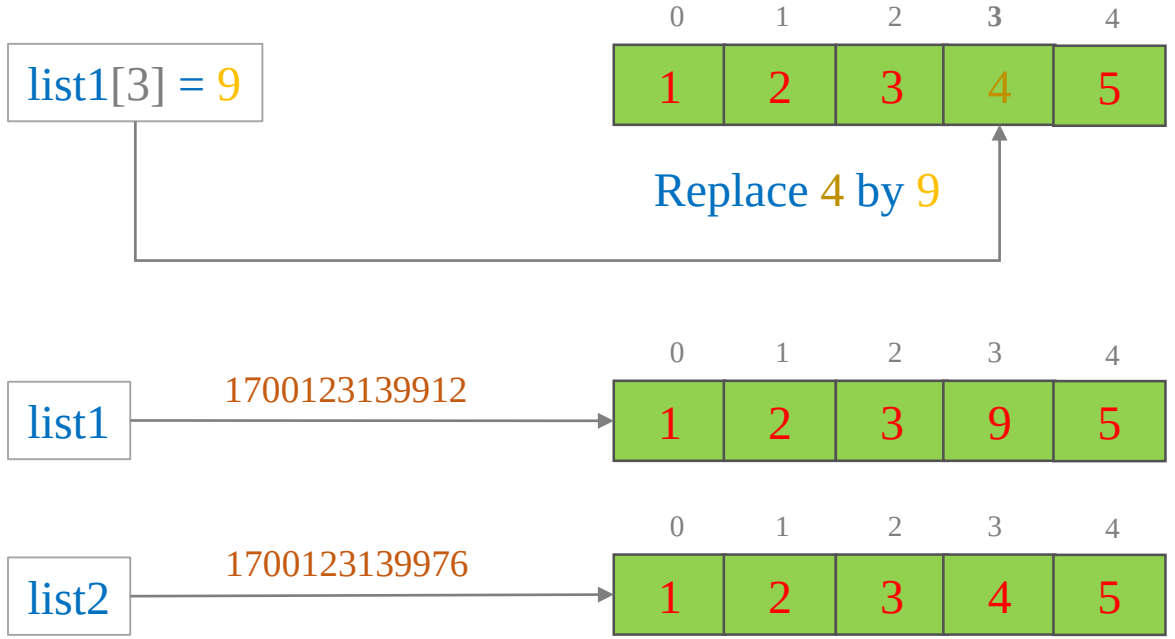
```python
# aivietnam - copy in python

import copy

list1 = [1, 2, 3, 4, 5]
list2 = copy.copy(list1)

print('List 1: ', list1)
print('List1 address: ', id(list1))

print()
print('List 2: ', list2)
print('List2 address: ', id(list2))
```

```
List 1:  [1, 2, 3, 4, 5]
List1 address:  1700123139912

List 2:  [1, 2, 3, 4, 5]
List2 address:  1700123139976
```

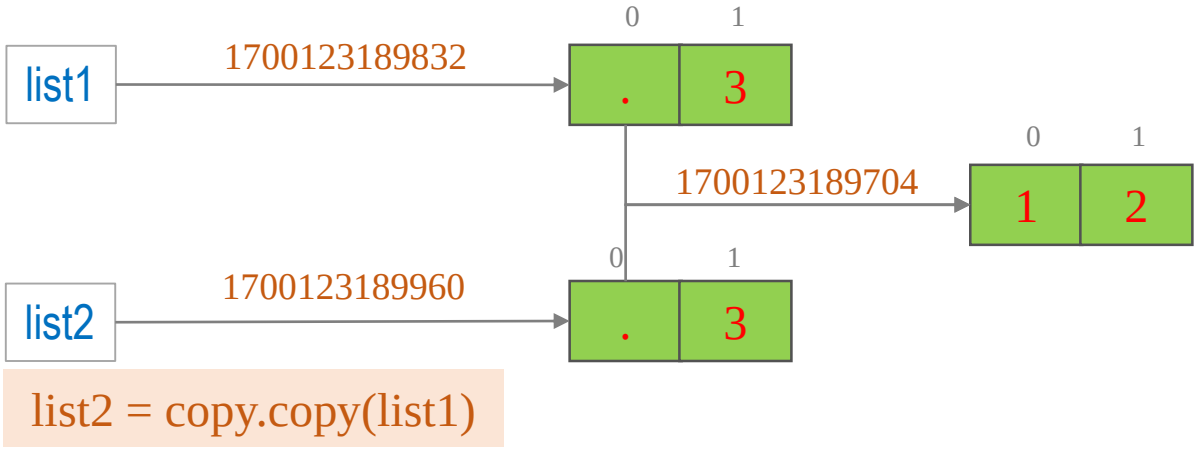
```python
list1[3] = 9
print('List 1: ', list1)
print('List 2: ', list2)
```

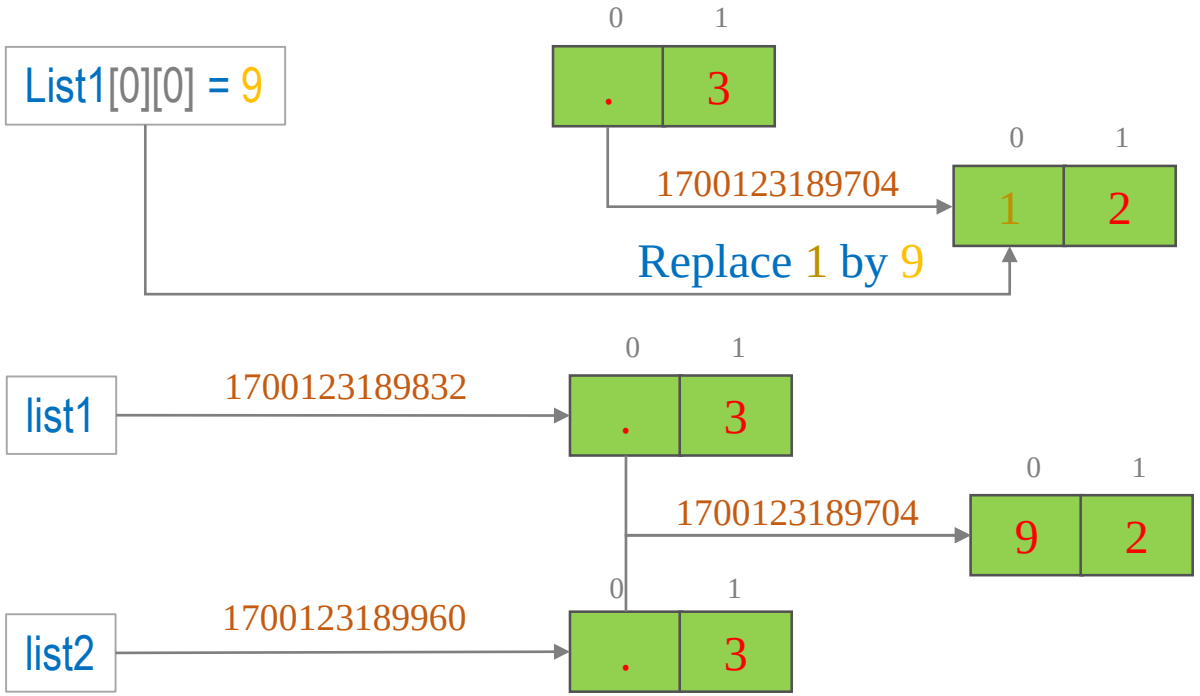
```
List 1:  [1, 2, 3, 9, 5]
List 2:  [1, 2, 3, 4, 5]
```





List Copy






```python
# aivietnam - copy in python

import copy

list1 = [[1, 2], 3]
list2 = copy.copy(list1)

print('List 1: ', list1)
print('List1 address: ', id(list1))

print()
print('List 2: ', list2)
print('List2 address: ', id(list2))
```

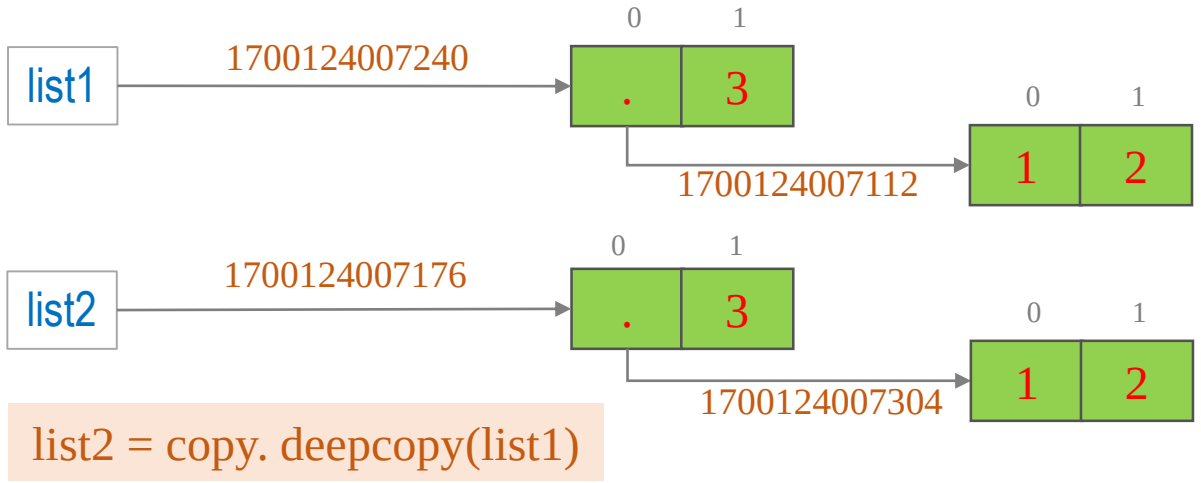
```
List 1:  [[1, 2], 3]
List1 address:  1700123189832

List 2:  [[1, 2], 3]
List2 address:  1700123189960
```

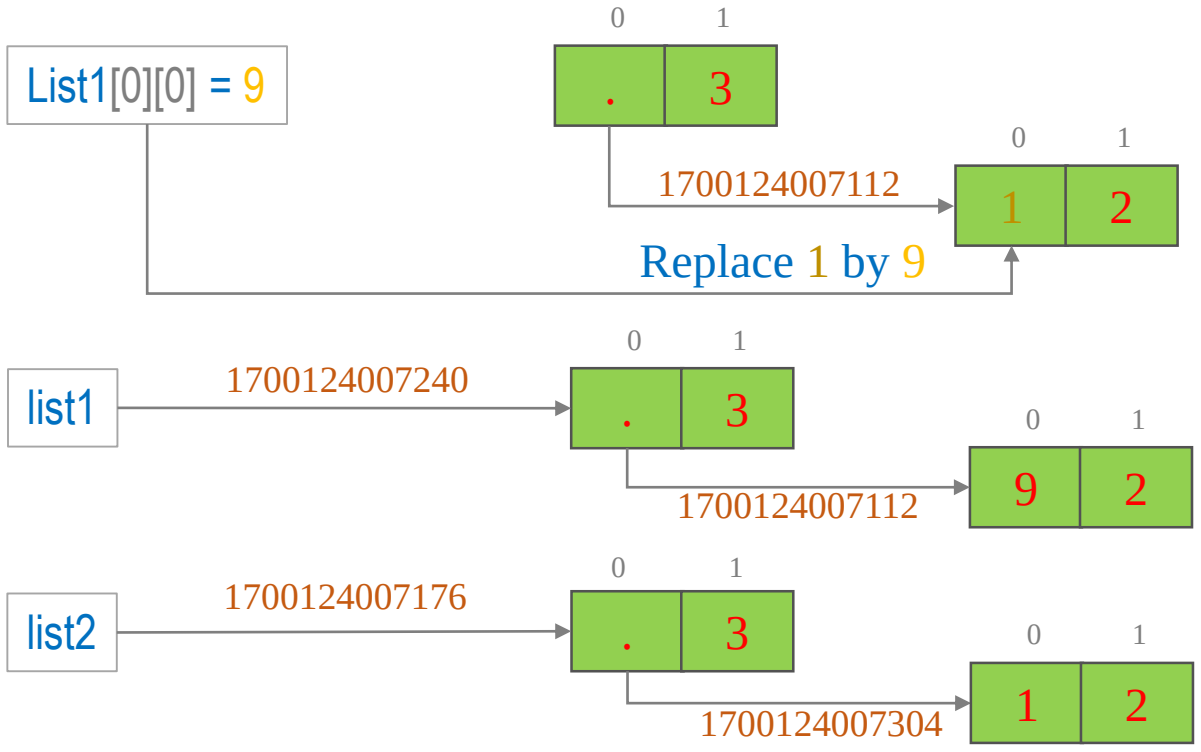
```python
print('List1[0] address: ', id(list1[0]))
print('List2[0] address: ', id(list2[0]))
```

```
List1[0] address:  1700123189704
List2[0] address:  1700123189704
```

```python
list1[0][0] = 9
print('List 1: ', list1)
print('List 2: ', list2)
```

```
List 1:  [[9, 2], 3]
List 2:  [[9, 2], 3]
```






```python
# aivietnam - copy in python

import copy

list1 = [[1, 2], 3]
list2 = copy.deepcopy(list1)

print('List 1: ', list1)
print('List1 address: ', id(list1))

print()
print('List 2: ', list2)
print('List2 address: ', id(list2))
```

```
List 1:  [[1, 2], 3]
List1 address:  1700124007240

List 2:  [[1, 2], 3]
List2 address:  1700124007176
```

```python
print('List1[0] address: ', id(list1[0]))
print('List2[0] address: ', id(list2[0]))
```

```
List1[0] address:  1700124007112
List2[0] address:  1700124007304
```

```python
list1[0][0] = 9
print('List 1: ', list1)
print('List 2: ', list2)
```

```
List 1:  [[9, 2], 3]
List 2:  [[1, 2], 3]
```

Outline

- Introduction
- List
- Some algorithms on List
- Addresses
- Common Errors

Common Errors

❖ Error 1

```
4.  # khai báo biến a = 5
5.  a = 5
6.
7.  # thực hiên a + b, sau đó lưu vào biến c
8.  c = a + b
9.
10. # in giá trị c
11. print(c)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-1-eae96ee94f9f> in <module>
      6
      7 # thực hiên a + b, sau đó lưu vào biến c
----> 8 c = a + b
      9
     10 # in giá trị c

NameError: name 'b' is not defined
```

Common Errors

❖ Error 2

```
4.  # khai báo biến a = 5
5.  a = 5
6.
7.  # in giá trị a
8.  Print(a)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-f09db6b2bf7e> in <module>
      6
      7 # in giá trị a
----> 8 Print(a)

NameError: name 'Print' is not defined
```

Common Errors

❖ Error 3

```
4.  # khai báo biến chuỗi s
5.  s = 'Hello AIVIETNAM"
6.
7.  # in giá trị s
8.  print(s)
```

```
  File "<ipython-input-3-96feed73c6b1>", line 5
    s = 'Hello AIVIETNAM"
                        ^
SyntaxError: EOL while scanning string literal
```

Common Errors

❖ Error 4

```
4.  # khai báo biến a và b
5.  a = 5
6.  b = 0
7.
8.  # tính giá trị c bằng a chia cho b
9.  c = a / b
10.
11. # in giá trị c
12. print(c)
```

```
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call last)
<ipython-input-4-298e1112d534> in <module>
      7
      8 # tính giá trị c bằng a chia cho b
----> 9 c = a / b
     10
     11 # in giá trị c

ZeroDivisionError: division by zero
```

Common Errors

❖ Error 5

```
4.  # khai báo biến chuỗi s
5.  s = 'AI'
6.
7.  # khai báo biến n có kiểu integer
8.  n = 5
9.
10. # tính giá trị c
11. c = s + n
12.
13. # in giá trị c
14. print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-f1e2455fae51> in <module>
      9
     10 # tính giá trị c
---> 11 c = s + n
     12
     13 # in giá trị c

TypeError: must be str, not int
```

Common Errors

❖ Error 6

```
4.  # khai báo biến chuỗi s
5.  s = 'AI'
6.
7.  # khai báo biến n có kiểu integer
8.  n = 5
9.
10. # tính giá trị c
11. c = n + s
12.
13. # in giá trị c
14. print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-8-cecd5289546d> in <module>
      9
     10 # tính giá trị c
---> 11 c = n + s
     12
     13 # in giá trị c

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Common Errors

❖ Error 7

```
4.  # khai báo biến a và b
5.  a = 5
6.    b = 6
7.
8.  # thực hiên a + b, sau đó lưu vào biến c
9.  c = a + b
10.
11. # in giá trị c
12. print(c)
```

```
  File "<ipython-input-7-31f64166c395>", line 6
    b = 6
    ^
IndentationError: unexpected indent
```

Common Errors

❖ Error 8

```
4.  import math
5.
6.  number = 20.2
7.  print(math.floor(number)
8.  print(math.pi)
```

```
  File "<ipython-input-15-920005110c33>", line 8
    print(math.pi)
          ^
SyntaxError: invalid syntax
```

Common Errors

❖ Error 9

```
3.
4.  print "aivietnam.ai"
```

```
  File "<ipython-input-3-a46b1c9e05ed>", line 4
    print "aivietnam.ai"
                       ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("aivietnam.ai")?
```

Common Errors

❖ Error 10

```
4.  import mymodule
5.
6.  print("aivietnam.ai")
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-5-1b242c59080b> in <module>
      2 # Lỗi khai báo module không tồn tại
      3
----> 4 import mymodule
      5
      6 print("aivietnam.ai")

ModuleNotFoundError: No module named 'mymodule'
```

Common Errors

❖ Error 11

```
3.
4.  l = [1, 2, 3, 4, 5]
5.  print(l[0])
6.  print(l[5])
```

```
1

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-6-b8c53176edc0> in <module>
      4 l = [1, 2, 3, 4, 5]
      5 print(l[0])
----> 6 print(l[5])

IndexError: list index out of range
```

Common Errors

❖ Error 12

```
4.  name = "aivietname.ai"
5.  print(name[0])
6.  print(name[50])
```

```
a

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-7-357ff533411a> in <module>
      4 name = "aivietname.ai"
      5 print(name[0])
----> 6 print(name[50])

IndexError: string index out of range
```

Common Errors

❖ Error 13

```
4.  number = 15
5.  if number < 10
6.      print("A small number")
7.  else:
8.      print("A large number")
```

```
  File "<ipython-input-15-fedf173614ac>", line 5
    if number < 10
                ^
SyntaxError: invalid syntax
```

Common Errors

❖ Error 14

```
4.  number = 15
5.  if number < 10:
6.      print("A small number")
7.  else
8.      print("A large number")
```

```
  File "<ipython-input-16-699752908646>", line 7
    else
       ^
SyntaxError: invalid syntax
```

Common Errors

❖ Error 15

```
1.  import math
2.
3.  number = -4
4.  print(math.sqrt(number))
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-8-f25b4b744f6e> in <module>
      2
      3 number = -4
----> 4 print(math.sqrt(number))

ValueError: math domain error
```

Common Errors

❖ Error 16

```
4.  # import pytorch
5.  import torch
6.
7.  print(torch.cuda.is_available())
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-14-680f8ea2b256> in <module>
      1 # import thư viện pytorch
----> 2 import torch
      3
      4 print(torch.cuda.is_available())

ModuleNotFoundError: No module named 'torch'
```

Common Errors

❖ Error 17

```
4  index = 5
5  def my_function():
6      index += 1
7      print(index)
8
9  my_function()
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-4-63ee67900288> in <module>
      7     print(index)
      8
----> 9 my_function()

<ipython-input-4-63ee67900288> in my_function()
      4 index = 5
      5 def my_function():
----> 6     index += 1
      7     print(index)
      8

UnboundLocalError: local variable 'index' referenced before assignment
```

Common Errors

❖ Error 18

```
4 a_number = 5
5 a_string = 'value '
6 result = a_string + a_number
7
8 print(result)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-7-9772d690ff0d> in <module>
      4 a_number = 5
      5 a_string = 'value '
----> 6 result = a_string + a_number
      7
      8 print(result)

TypeError: can only concatenate str (not "int") to str
```

Common Errors

❖ Error 19

```
3  def a_function(x):
4      a_variable = 4
5      result = x*a_variable
6
7      return result
8
9  print(a_variable)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-43-2cce0357dce7> in <module>
      6
      7
----> 8 print(a_variable)

NameError: name 'a_variable' is not defined
```

Common Errors

❖ Error 20

```
4 str1 = '5'
5 str2 = 'hello'
6
7 value1 = int(str1)
8 value2 = int(str2)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-9-1f0b23b26eb1> in <module>
      6
      7 value1 = int(str1)
----> 8 value2 = int(str2)

ValueError: invalid literal for int() with base 10: 'hello'
```

Common Errors

❖ Error 21

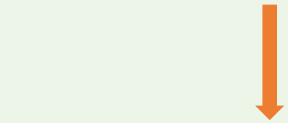
```
3
4  def a_function(n):
5      return a_function(n)
6
7  a_function(5)
```

```
---------------------------------------------------------------------------
RecursionError                            Traceback (most recent call last)
<ipython-input-10-bda7ef50bf68> in <module>
      5     return a_function(n)
      6
----> 7 a_function(5)

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

... last 1 frames repeated, from the frame below ...

<ipython-input-10-bda7ef50bf68> in a_function(n)
      3
      4 def a_function(n):
----> 5     return a_function(n)
      6
      7 a_function(5)

RecursionError: maximum recursion depth exceeded
```

Common Errors

❖ Error 22

```
1  # aivietnam - example
2
3  # create a list of numbers
4  numbers = [1, 2, 3, 4, 5]
5
6  for num in numbers:
7      # print the num
8      print(num)
9
10     # check if the num is equal to 3
11     if (num == 3):
12         # add 3 to the list
13         numbers.append(3)
```



numbers = [1, 2, 3, 4, 5]

Output

1

numbers = [1, 2, 3, 4, 5]

Output

1, 2

numbers = [1, 2, 3, 4, 5]

numbers = [1, 2, 3, 4, 5, 3]

Output

1, 2, 3

Common Errors

❖ Error 22

```
1  # aivietnam - example
2
3  # create a list of numbers
4  numbers = [1, 2, 3, 4, 5]
5
6  for num in numbers:
7      # print the num
8      print(num)
9
10     # check if the num is equal to 3
11     if (num == 3):
12         # add 3 to the list
13         numbers.append(3)
```

numbers = [1, 2, 3, 4, 5, 3]

numbers = [1, 2, 3, 4, 5, 3, 3]

Output

1, 2, 3, 4, 5, 3

Common Errors

❖ Error 22

```
1  # aivietnam - example
2
3  # create a list of numbers
4  numbers = [1, 2, 3, 4, 5]
5
6  for num in numbers:
7      # print the num
8      print(num)
9
10     # check if the num is equal to 3
11     if (num == 3):
12         # add 3 to the list
13         numbers.append(3)
```

numbers = [1, 2, 3, 4, 5, 3]

Output

1, 2, 3, 4, 5, 3

numbers = [1, 2, 3, 4, 5, 3, 3]

Output

1, 2, 3, 4, 5, 3, 3

numbers = [1, 2, 3, 4, 5, 3, 3, 3]

o o o

Common Errors

❖ Errors 25

```
1  # common error
2
3  name = 'John'
4  age  = 26
5
6  print('Hello {name}. Are you {age}?')
```

```
Hello {name}. Are you {age}?
```

```
1  # common error
2
3  fruits = {'banana': 2}
4  fruits['apple'] += 10
```

```
---------------------------------------------------------------
KeyError                                  Traceback
<ipython-input-1-139246a7a818> in <module>
      2
      3 fruits = {'banana': 2}
----> 4 fruits['apple'] += 10

KeyError: 'apple'
```

Common Errors

❖ Errors 26

```
# common error

fruits = ['apple', 'banana', 'peach']

for fruit in fruits:
    if (fruit == 'peach'):
        fruits.append('peach')

    print('Do something with this fruit -', fruit)
```

```
# common error

fruits = ['apple', 'banana', 'peach']

for fruit in fruits:
    if (fruit == 'banana'):
        fruits.remove('banana')

    print('Do something with this fruit -', fruit)
```

???

Common Errors

❖ Errors 27

```
# common error

a = 0.1 + 0.1 + 0.1
if (a == 0.3):
    print('Yes')
else:
    print('No')
```

Quizzes

```python
# quiz 1
index = 5
def my_function():
    print(index)


my_function()
```

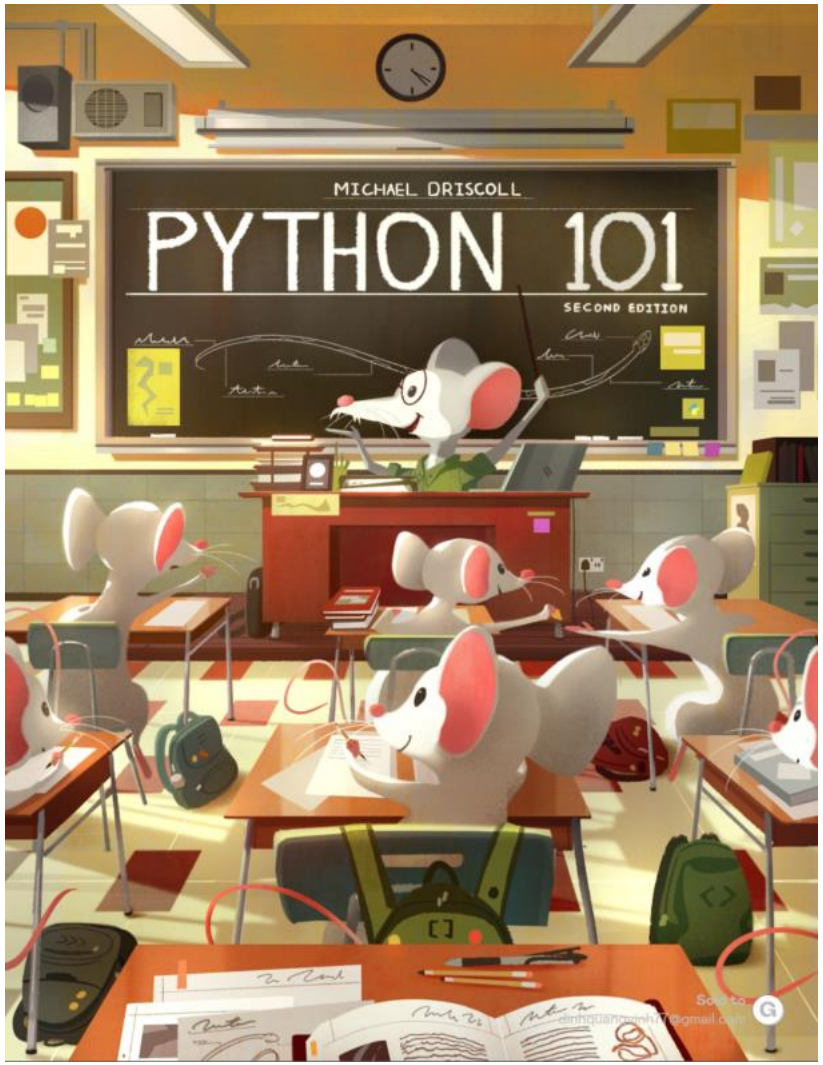
```python
# quiz 2
index = 5
def my_function():
    data = index + 1
    print(index)

my_function()
```

```python
# quiz 3
index = 5
def my_function():
    index = index + 1
    print(index)

my_function()
```

```python
# quiz 4
index = 5
def my_function(index):
    index = index + 1
    print(index)

my_function(7)
```

Further Reading

❖ List (Chapters 6 and 13)



❖ Binary Searching and Sorting