Lecture06

July 6, 2025

Delete apple on middle list Find index of dog on list Sorted 2 dimention list

[1]:
```python
fruits = ["apple", "banana", "cherry"]

print(fruits[0])
print(fruits[1])
print(fruits[2])
```

```
apple
banana
cherry
```

[2]:
```python
fruits = ["apple", "banana", "cherry"]
more_fruits = ["mango", "pineapple"]
for fruit in more_fruits:
    fruits.append(fruit)
print(f"Fruits after apped: {fruits}")
```

```
Fruits after apped: ['apple', 'banana', 'cherry', 'mango', 'pineapple']
```

[4]:
```python
fruits = ["apple", "banana", "cherry"]
more_fruits = ["mango", "pineapple"]
print(fruits + more_fruits)
```

```
['apple', 'banana', 'cherry', 'mango', 'pineapple']
```

[6]:
```python
berries = ["raspberry", "blackberyy"]
berries.insert(1, "strawberyy")
berries.insert(2, "blueberry")
print(f"Berries after insert: {berries}")
```

```
Berries after insert: ['raspberry', 'strawberyy', 'blueberry', 'blackberyy']
```

[3]:
```python
fruits_with_duplicates = ["apple", "banana", "apple", "cherry", "apple", "kiwi"]
while "apple" in fruits_with_duplicates:
    fruits_with_duplicates.remove("apple")
# fruits_with_duplicates.remove("apple")
print(f"Fruits after remove: {fruits_with_duplicates}")
```

```
Fruits after remove: ['banana', 'cherry', 'kiwi']
```

Delete apple on middle list

Delete apple on middle list

Go top

[16]:

```
fruits_with_duplicates = ["apple", "banana", "apple", "cherry", "apple", "kiwi"]
count_apple = fruits_with_duplicates.count("apple")
for i in range(count_apple - 1):
    fruits_with_duplicates.remove("apple")
fruits_with_duplicates.insert(0, "apple")
print(fruits_with_duplicates)
```

```
['apple', 'banana', 'cherry', 'apple', 'kiwi']
```

[17]:

```
grades = [85, 90, 78, 92, 88]
third_grade = grades.pop(2)
grades.append(third_grade)
print(f"Grades after pop: {grades}")
```

```
Grades after pop: [85, 90, 92, 88, 78]
```

[18]:

```
grades = [85, 90, 78, 92, 88]
third_grade = grades.pop()
grades.append(third_grade)
print(f"Grades after pop: {grades}")
```

```
Grades after pop: [85, 90, 78, 92, 88]
```

[26]:

```
animal = ["cat", "dog", "rabbit", "hamster", "dog", "parrot"]
first_dog_index = animal.index("dog")
print(f"The first occurrence of 'dog' is at index: {first_dog_index}")

second_dog_index = animal.index("dog", first_dog_index + 1)
print(f"The second occurrence of 'dog' is at index: {second_dog_index}")

third_dog_index = animal.index("dog", second_dog_index + 1)
print(f"The second occurrence of 'dog' is at index: {third_dog_index}")
```

```
The first occurence of 'dog' is at index: 1
The second occurence of 'dog' is at index: 4
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[26], line 8
      5 second_dog_index = animal.index("dog", first_dog_index + 1)
      6 print(f"The second occurence of 'dog' is at index: {second_dog_index}")
----> 8 third_dog_index = animal.index("dog", second_dog_index + 1)
      9 print(f"The second occurence of 'dog' is at index: {third_dog_index}")
```

```
ValueError: 'dog' is not in list
```

Find position dog on list

Go top

[53]:

```python
animal = [
    "dog",
    "cat",
    "dog",
    "rabbit",
    "hamster",
    "dog",
    "parrot",
    "rabbit",
    "hamster",
    "dog",
    "parrot",
    "rabbit",
    "hamster",
    "dog",
    "parrot",
]
count_dog = animal.count("dog")
current_count_dog = 0
dog_index = 0
while 1:
    if current_count_dog == count_dog:
        break
    if dog_index == 0:
        dog_index = animal.index("dog")
        print(
            f"The number {current_count_dog+1} 'dog' on the list is
            at index: {dog_index}"
        )
        dog_index += 1
    else:
        dog_index = animal.index("dog", dog_index + 1)
        print(
            f"The number {current_count_dog+1} 'dog' on the list is
            at index: {dog_index}"
        )
    current_count_dog += 1
```

```
The number 1 'dog' on the list is at index: 0
The number 2 'dog' on the list is at index: 2
The number 3 'dog' on the list is at index: 5
The number 4 'dog' on the list is at index: 9
The number 5 'dog' on the list is at index: 13
```

[27]:

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for sublist in nested_list:
    sublist.clear()
print(f"Nested list after clear: {nested_list}")
```

```
Nested list after clear: [[], [], []]
```

[36]:

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for sublist in nested_list:
    sublist.pop(1)
print(f"Nested list after clear: {nested_list}")
```

```
Nested list after clear: [[1, 3], [4, 6], [7, 9]]
```

Sorted 2 dimention list

Go top

[39]:

```
nested_list = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
for sublist in nested_list:
    sublist.reverse()
nested_list.reverse()
print(f"Nested list after clear: {nested_list}")
```

```
Nested list after clear: [[9, 8, 7], [6, 5, 4], [3, 2, 1]]
```

[44]:

```
heroes = ["Ironman", "Thor", "Hulk", "Superman", "Spiderman"]
h2 = ["Dr. Strange", "Cpt. America", "Black Panther", "Ant Man"]

heroes.insert(0, h2[0])
print(heroes.index("Thor"))
heroes.insert(heroes.index("Thor"), h2[1])
print(heroes)
heroes.remove("Superman")
heroes.append("Ant Man")
print(heroes)
heroes.sort()
print(heroes)
heroes.reverse()
print(heroes)
newheroes = heroes
newheroes[0] = "Wonder Women"
print(heroes)
copyheroes = [] + heroes
print(copyheroes)
copyheroes[0] = "Hanuman"
print(heroes)
print(copyheroes)
```

```
2
['Dr. Strange', 'Ironman', 'Cpt. America', 'Thor', 'Hulk', 'Superman',
'Spiderman']
['Dr. Strange', 'Ironman', 'Cpt. America', 'Thor', 'Hulk', 'Spiderman', 'Ant
Man']
['Ant Man', 'Cpt. America', 'Dr. Strange', 'Hulk', 'Ironman', 'Spiderman',
'Thor']
['Thor', 'Spiderman', 'Ironman', 'Hulk', 'Dr. Strange', 'Cpt. America', 'Ant
Man']
['Wonder Women', 'Spiderman', 'Ironman', 'Hulk', 'Dr. Strange', 'Cpt. America',
'Ant Man']
['Wonder Women', 'Spiderman', 'Ironman', 'Hulk', 'Dr. Strange', 'Cpt. America',
'Ant Man']
['Wonder Women', 'Spiderman', 'Ironman', 'Hulk', 'Dr. Strange', 'Cpt. America',
'Ant Man']
['Hanuman', 'Spiderman', 'Ironman', 'Hulk', 'Dr. Strange', 'Cpt. America', 'Ant
Man']
```

[49]:

```python
data = list(range(100))
slice_data = data[10:51:5]
print(f"Sliced data: {slice_data}")
```

```
Sliced data: [10, 15, 20, 25, 30, 35, 40, 45, 50]
```

[52]:

```python
ss = "Sammy Shark!"
print(ss[4])
print(ss[6:11])
print(ss[:5])
print(ss[7:])
print(ss[-4:-1])
print(ss[6:11])
print(ss[6:11:1])
print(ss[0:12:2])
print(ss[0:12:4])
print(ss[::4])
print(ss[::-1])
print(ss[::-2])
```

```
y
Shark
Sammy
hark!
ark
Shark
Shark
SmySak
Sya
Sya
!krahS ymmaS
```

```
!rh ma
```

[54]:

```
even_number = [2, 4, 6, 8, 10]
heroes = ["Ironman", "Thor", "Hulk", "Spiderman"]
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
print(numbers[-5:])
numbers[8]  = 99
print(numbers)
pluslist = heroes + even_number
print(pluslist)
print(len(numbers))
```

```
[6, 7, 8, 9, 10]
[1, 2, 3, 4, 5, 6, 7, 8, 99, 10]
['Ironman', 'Thor', 'Hulk', 'Spiderman', 2, 4, 6, 8, 10]
10
```

[56]:

```
numbers = [4, 2, 9, 1, 5, 6]
length = len(numbers)
print(f"Length of the list: {length}")

total_sum = sum(numbers)
print(f"Sum of all elements: {total_sum}")

max_value = max(numbers)
print(f"Maximum value: {max_value}")

min_value = min(numbers)
print(f"Minimun value: {min_value}")

sorted_numbers = sorted(numbers)
print(f"Sorted list: {sorted_numbers}")

bool_list = [False, True, False]
any_true = any(bool_list)
print(f"Is any element True? {any_true}")

all_true = all(bool_list)
print(f"Are all element True? {all_true}")

string = "hello"
char_list = list(string)
print(f"List of charecters: {char_list}")

reversed_numbers = list(reversed(numbers))
print(f"Reversed list: {reversed_numbers}")

enumerate_numbers = list(enumerate(numbers))
```

```
print(f"Enumerate list: {enumerate_numbers}")
```

```
Length of the list: 6
Sum of all elements: 27
Maximum value: 9
Minimun value: 1
Sorted list: [1, 2, 4, 5, 6, 9]
Is any element True? True
Are all element True? False
List of charecters: ['h', 'e', 'l', 'l', 'o']
Reversed list: [6, 5, 1, 9, 2, 4]
Enumerate list: [(0, 4), (1, 2), (2, 9), (3, 1), (4, 5), (5, 6)]
```

[58]:

```
NUM_EMPLOYEES = 6


def main():
    hours = [0] * NUM_EMPLOYEES

    for index in range(NUM_EMPLOYEES):
        print("Enter the hours worked by employee ",
              index + 1, ": ", sep="", end="")
        hours[index] = float(input())

    pay_rate = float(input("Enter the hourly pay rate: "))

    for index in range(NUM_EMPLOYEES):
        gross_pay = hours[index] * pay_rate
        print(
            "Gross pay for employee ",
            index + 1,
            ": $",
            format(gross_pay, ",.2f"),
            sep="",
        )


main()
```

```
Enter the hours worked by employee 1: Enter the hours worked by employee 2:
Enter the hours worked by employee 3: Enter the hours worked by employee 4:
Enter the hours worked by employee 5: Enter the hours worked by employee 6:
Gross pay for employee 1: $247,200.00
Gross pay for employee 2: $3,120,900.00
Gross pay for employee 3: $8,114,340.00
Gross pay for employee 4: $3,157,980.00
Gross pay for employee 5: $80,903,616.00
Gross pay for employee 6: $27,192.00
```

[62]:

```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

matrix[0][1] = 10
print(matrix)

for row in matrix:
    for element in row:
        print(element, end=" ")
    print()
```

```
[[1, 10, 3], [4, 5, 6], [7, 8, 9]]
1 10 3
4 5 6
7 8 9
```

[63]:

```
import random

ROWS = 3
COLS = 4


def main():
    values = [
        [0, 0, 0, 0],
        [0, 0, 0, 0],
        [0, 0, 0, 0],
    ]

    for r in range(ROWS):
        for c in range(COLS):
            values[r][c] = random.randint(1, 100)

    print(values)


main()
```

```
[[97, 52, 56, 58], [33, 52, 52, 18], [4, 44, 94, 74]]
```

[69]:

```
my_tuple = 1, 2, 3, 4
print(my_tuple)

a, b, c = my_tuple
print(a)
print(b)
print(c)
```

```
(1, 2, 3, 4)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[69], line 4
      1 my_tuple = 1, 2, 3, 4
      2 print(my_tuple)
----> 4 a, b, c = my_tuple
      5 print(a)
      6 print(b)

ValueError: too many values to unpack (expected 3)
```

[91]:

```
inventory = [
    ["Apple", 50, 0.75],
    ["Banana", 100, 0.5],
    ["Orange", 75, 0.8],
    ["Coconut", 20, 0.8],
]


def display(inventory):
    for items in inventory:
        print(items)


def update_inventory(inventory, item_name, quantity_sold):
    for items in inventory:
        if item_name in items:
            items[1] = items[1] - quantity_sold


def calculate_total_value(inventory):
    # sum_val = 0
    # for items in inventory:
    #     sum_val += items[1]*items[2]
    # return sum_val
    return sum([items[1] * items[2] for items in inventory])


def find_most_expensive(inventory):
    value_expensive = max([item_price[2] for item_price in inventory])
    # for item in inventory:
    #     if item[2] == value_expensive:
    #         expensive_list.append(item[0])
    return [items[0] for items in inventory if items[2] == value_expensive]
```

```
def add_item(inventory, item_name, quantity, price):
    find = False
    for items in inventory:
        if item_name in items:
            items[1] = quantity
            items[2] = price
            find = True
    if not find:
        inventory.append([item_name, quantity, price])


# update_inventory(inventory, "Apple", 30)

print(calculate_total_value(inventory))

# print(find_most_expensive(inventory))

# add_item(inventory, "Banana", 50, 0.46)
# add_item(inventory, "Rasberry", 65, 0.76)

# display(inventory)
```

```
163.5
```