migo_5_list

June 27, 2024

List

A list is an ordered collection of items (also known as elements) that can be of any type

Ordered,Mutable,Dinamic size,Heterogeneous(contain elements of different types)

Creating Lists

[1]:
```
l1=[]
l2=[3,5,1,2]
l3=[2,"lala",3+4j,-7]
print(l1,l2,l3,sep="\n")
```

```
[]
[3, 5, 1, 2]
[2, 'lala', (3+4j), -7]
```

Indexing 0,1,2…….len-1 -len,-len+1,…..-1

[2]:
```
first_l2=l2[0]
last_l3=l3[-1]
print(first_l2,last_l3,sep="\n")
```

```
3
-7
```

Modifiying

[3]:
```
l3[1]="lolo"
print(l3)
```

```
[2, 'lolo', (3+4j), -7]
```

Adding and Insert

[4]:
```
l2.append(152)        #to end of the list
l3.insert(2,"lili")   #insert at specific position
print(l2,l3,sep="\n")
```

```
[3, 5, 1, 2, 152]
[2, 'lolo', 'lili', (3+4j), -7]
```

Removing

[5]:
```
l2.remove(5)  #by value
del(l3[-1])   #by index
print(l2,l3,sep="\n")
```

```
[3, 1, 2, 152]
[2, 'lolo', 'lili', (3+4j)]
```

Pop(remove and return)

[6]:
```
last_element=l2.pop()
specific_element=l3.pop(1)
print(last_element,specific_element)
```

```
152 lolo
```

Slicing

[7]:
```
l=[1,2,3,4,5,6,7]
sub_list1=l[1:4]
sub_list2=l[:3]
sub_list3=l[1::2]
sub_list4=l[::-1]
print(sub_list1,sub_list2,sub_list3,sub_list4,sep="\n")
```

```
[2, 3, 4]
[1, 2, 3]
[2, 4, 6]
[7, 6, 5, 4, 3, 2, 1]
```

Iterating

[8]:
```
l=[1,2,3,4,5,6,7]
for item in l:
    print(item,end=" ") # item copy of the element
print()
for i in range(len(l)):
    print(l[i],end=" ") #l[i] the element
print()
for i,_ in enumerate(l): #'_' when we do not using
    l[i]*=2
print(l)
```

```
1 2 3 4 5 6 7
1 2 3 4 5 6 7
[2, 4, 6, 8, 10, 12, 14]
```

List Operations:Concatenation and Repetition

[2]:
```
l1=[1,2,3,4]
l2=[5,6]
l3=l1+l2
```

```
print(l3)
l3=l2*3
print(l3)
```

```
[1, 2, 3, 4, 5, 6]
[5, 6, 5, 6, 5, 6]
```

Enter the amount of precipitation for each month in the last year and print the months with the minimum and maximum precipitation.

[9]:
```
month=["jan","feb","mar","apr","may","jun","jul","aug","sep","oct","nov","dec"]
rainAmount=[]
for i in range(12):
    x=int(input(f"Enter mounth {month[i]}:"))
    rainAmount.append(x)
print(rainAmount)
mn=mx=rainAmount[0]
sum=0
for elm in rainAmount:
    if elm>mx:
        mx=elm
    if elm<mn:
        mn=elm
    sum+=elm
print(f"min={mn} max={mx} avg={sum/len(rainAmount)}")
mnl=[]
mxl=[]
for i in range(len(rainAmount)):
    if rainAmount[i] <= mn:
        mnl.append(month[i])
    if rainAmount[i] >= mx:
        mxl.append(month[i])
print(f"min:{mnl}\nmax:{mxl}")
```

```
Enter mounth jan: 20
Enter mounth feb: 25
Enter mounth mar: 30
Enter mounth apr: 25
Enter mounth may: 20
Enter mounth jun: 50
Enter mounth jul: 60
Enter mounth aug: 22
Enter mounth sep: 26
Enter mounth oct: 60
Enter mounth nov: 60
Enter mounth dec: 60
[20, 25, 30, 25, 20, 50, 60, 22, 26, 60, 60, 60]
min=20 max=60 avg=38.166666666666664
```

```
min:['jan', 'may']
max:['jul', 'oct', 'nov', 'dec']
```

Nested list (list in list)

contains other lists as its elements. This allows the creation of complex, multi-dimensional data structures

[10]:
```
nested_list=[
    [1,2,3],
    [4,5,6],
    [7,8,9]
]
print(nested_list)
```

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

Accessing and Modifing

[11]:
```
val=nested_list[1][2]
nested_list[0][0]=100
print(val)
print(nested_list)
```

```
6
[[100, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Adding

[12]:
```
nested_list.append([10,11,12]) #append new inner list
nested_list[0].append(4)       #append element to an existing list
print(nested_list)
```

```
[[100, 2, 3, 4], [4, 5, 6], [7, 8, 9], [10, 11, 12]]
```

Removing

[13]:
```
nested_list.pop(1)         #remove an entire inner list
nested_list[1].remove(8)   #remove the element 8 from the second inner list
print(nested_list)
```

```
[[100, 2, 3, 4], [7, 9], [10, 11, 12]]
```

Iterating

[15]:
```
for row in nested_list:
    for element in row:
        print(element,end=" ")
    print()
```

```
100 2 3 4
7 9
10 11 12
```

list comprehensions

provide a concise way to create lists

new_list = [expression for item in iterable if condition]

[4]:
```
squares = [x**2 for x in range(10)]
evens = [x for x in range(10) if x % 2 == 0]
print(squares,evens,sep="\n")
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
[0, 2, 4, 6, 8]
```

Nested List Comprehension:

[5]:
```
matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
flat = [num for row in matrix for num in row]
print(flat)
```

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

Using Functions

[6]:
```
def square(x):
    return x**2
squared_numbers = [square(x) for x in range(10)]
print(squared_numbers)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

[ ]:
```
Practical Uses:Transforming Data,Filtering Data,Creating Complex Lists
```

[8]:
```
celsius = [0, 10, 20, 30]
fahrenheit = [(temp * 9/5) + 32 for temp in celsius]
print(fahrenheit)
numbers = [-5, -1, 0, 1, 5]
non_negative = [num for num in numbers if num >= 0]
print(non_negative)
pairs = [(x, y) for x in range(3) for y in range(3)]
print(pairs)
```

```
[32.0, 50.0, 68.0, 86.0]
[0, 1, 5]
[(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)]
```

[ ]: