

Data structures in python

LISTS

```
#creating list
a=[1,2,3,4,5,6,7,8,9]
print(a)

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

b=[1,2,3,'lohi']
print(b)

[1, 2, 3, 'lohi']
```

List operation

```
#accessing items
print(a[0])
print(b[3])

1
lohi

#modifying items
a[2]=7
a

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

#adding items
#append
a.append(4)
a

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

#insert
a.insert(9,8)
a

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

#removing items
#remove
a.remove(6)
a

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

```
#pop
a.pop(3)
a

[1, 2, 7, 5, 7, 8, 9, 8, 4]

#other operating system
x=[2,3,7,8,5,0,6]
len(x)

7
```

iterating through list

```
for i in a:
    print(a)

[3, 5, 8, 9, 1]
[3, 5, 8, 9, 1]
[3, 5, 8, 9, 1]
[3, 5, 8, 9, 1]
[3, 5, 8, 9, 1]
```

TUPLES

```
#creating tuple
spam=(10,30,29,50)
print(spam)

(10, 30, 29, 50)

#accessing items
spam[2]

29
```

DICTIONARIES

```
#creating dictionary
pan={'name':'Alice',
    'age':30,
    'marks':40
    }
pan

{'name': 'Alice', 'age': 30, 'marks': 40}

#accessing and modifying
#accessing
pan['age']
```

30

#modifying

```
pan['name']='bob'  
pan
```

```
{'name': 'bob', 'age': 30, 'marks': 40}
```

```
pan['grade']='A'  
pan
```

```
{'name': 'bob', 'marks': 40, 'brand': 'lol', 'grade': 'A'}
```

#adding

```
pan['brand']='lol'  
pan
```

```
{'name': 'bob', 'age': 30, 'marks': 40, 'brand': 'lol'}
```

#remove

```
del pan['age']  
pan
```

```
{'name': 'bob', 'marks': 40, 'brand': 'lol'}
```

#iterating through a dictionary

```
for key,value in pan.items():  
    print(key,value)
```

```
name bob  
marks 40  
brand lol
```

SETS

#creating set

```
num={1,2,3,4,5,6,}  
num
```

```
{1, 2, 3, 4, 5, 6}
```

#set operation

#add items

```
num.add(8)  
num
```

```
{1, 2, 3, 4, 5, 6, 8}
```

#remove

```
num.remove(4)  
num
```

```
{1, 2, 3, 5, 6, 8}
```

```
#union
```

```
a={1,2,3,4,5}
```

```
b={4,5,6,7,8}
```

```
a|b
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
#intersection
```

```
a&b
```

```
{4, 5}
```

```
#difference
```

```
a-b
```

```
{1, 2, 3}
```

merge of two list

```
a=[1,3,5,6,8,7]
```

```
b=[3,4,6,8,9,0]
```

```
c=a+b
```

```
c
```

```
[1, 3, 5, 6, 8, 7, 3, 4, 6, 8, 9, 0]
```

dictionary operations

```
student={'name':'john','age':45,'marks':86}
```

```
print('name:',student['name'])
```

```
student['marks']=90
```

```
print('marks:',student['marks'])
```

```
name: john
```

```
marks: 90
```

find the max and min in list

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

```
print('maximum:',max(num))
```

```
print('minimum:',min(num))
```

```
maximum: 9
```

```
minimum: 1
```

count frequency of elements in list

```

num=[10,20,30,40,50,40,60]
frequency={}
for i in num:
    frequency[i]=frequency.get(i,0)+1
print(frequency)

{10: 1, 20: 1, 30: 1, 40: 2, 50: 1, 60: 1}

```

sort a list of tuples by second element

```

tuples=[(1,'apple'),(2,'banana'),(3,'orange')]
sortedtuples=sorted(tuples,key=lambda x:x[1])
print('sorted tuples:',sortedtuples)

sorted tuples: [(1, 'apple'), (2, 'banana'), (3, 'orange')]

```

reverse a list

```

num=[1,2,4,6,8]
num.sort()
num

[1, 2, 4, 6, 8]

num.reverse()
num

[8, 6, 4, 2, 1]

```

Palindrome number

```

num=int(input('enter the number'))
reversenum=0
temp=num
while temp>0:
    digit=temp%10
    reversenum=reversenum*10+digit
    temp=temp//10
if num==reversenum:
    print(f'{num} is palindrome numbers')
else:
    print(f'{num} is not palindrome numbers')

enter the number121
121 is palindrome numbers

class Solution(object):
    def isPalindrome(self, x):
        if x < 0 or (x % 10 == 0 and x != 0):
            return False

```

```

        reverse = 0
        while x > reverse:
            reverse = reverse * 10 + x % 10
            x //= 10
        return x == reverse or x == reverse // 10
solution = Solution()
print(solution.isPalindrome(121))
print(solution.isPalindrome(-121))
print(solution.isPalindrome(10))
print(solution.isPalindrome(0))

```

```

True
False
False
True

```

```

class Solution(object):
    def isPalindrome(self, x):
        if x < 0 or (x % 10 == 0 and x != 0):
            return False
        reverse = 0
        while x > reverse:
            reverse = reverse * 10 + x % 10
            x //= 10
        return x == reverse or x == reverse // 10

```

```

text='hello'
text[3]

```

```

{"type": "string"}

```

```

copy=ram
copy=[1,2,3]
copy[0]=67
copy

```

```

[67, 2, 3]

```

```

import pprint
data={'name': 'lohi', 'age': 20}
pprint.pprint(data)

```

```

{'age': 20, 'name': 'lohi'}

```