DATA STRUCTURES

Lists

```
#creating the listwith same data type
a=[1,2,3,4,5,6,7,78,9]
a
```

```
[1, 2, 3, 4, 5, 6, 7, 78, 9]
```

```
numbers=[1,2,3,4,5]
mixed=['hello',5,67]
#accesing items
print(numbers[4])
print(numbers[-3])
print(mixed[2])
print(mixed[-1])
#modifieng items
numbers[1]=23
numbers
mixed[0]='haiiii'
mixed
```

```
5
3
67
67
```

```
['haiiii', 5, 67]
```

Adding items

```
#append
numbers.append(4)
numbers
mixed.append('calling')
mixed
```

```
['haiiii', 5, 67, 'calling']
```

```
#insert
numbers.insert(3,45)
numbers
```

```
[1, 23, 3, 45, 4, 5, 4]
```

Removing items

```
#remove
numbers.remove(4)
numbers
```

```
[1, 23, 3, 45, 5, 4]
```

```
#pop
numbers.pop(0)
numbers
```

```
[23, 3, 45, 5, 4]
```

Other operations

```
#len
len(mixed)
```

```
4
```

```
#sort
numbers.sort()
numbers
```

```
[3, 4, 5, 23, 45]
```

```
#reverse
numbers.reverse()
numbers
```

```
[45, 23, 5, 4, 3]
```

```
#reverse
mixed.reverse()
mixed
```

```
['calling', 67, 5, 'haiiii']
```

Iterating through a list

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

```
45
23
5
4
3
```

TUPLES

```
x=(1,4,67,7)
```

```
#accesing items in a tuple
x=(67,89,90)
print(x[0])
```

```
67
```

Dictionaries

```
man={"name":'pintu',"age":16,"gender":'male'}
man
```

```
{'name': 'pintu', 'age': 16, 'gender': 'male'}
```

Accessing and modifying items

```
#accessing
print(man["name"])
print(man["gender"])
```

```
pintu
male
```

```
#modifying
man["name"]="achhu"
man['name']
```

```
{"type":"string"}
```

```
man
```

```
{'name': 'achhu', 'age': 16, 'gender': 'male'}
```

```
#adding
man["category:"]="dog"
man
```

```
{'name': 'achhu', 'age': 16, 'gender': 'male', 'category:': 'dog'}
```

```
#removing
del man["age"]
man
```

```
{'name': 'achhu', 'gender': 'male'}
```

Iterating through a dictionary

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

```
name achhu
gender male
```

SETS

```
y={2,30,25,10,45,67,88}
```

Set operations

```
#adding items
y.add(40)
y
```

```
{2, 10, 25, 30, 40, 45, 67, 88}
```

```
#removing items
y.remove(30)
y
```

```
{2, 10, 25, 40, 45, 67, 88}
```

Set operations

```
z={2,36,40,98,45}
```

```
#union
y|z
```

```
{2, 10, 25, 36, 40, 45, 67, 88, 98}
```

```
#intersection
y&z
```

```
{2, 40, 45}
```

```
#difference
z-y
```

```
{36, 98}
```

```
#differenec
y-z
```

```
{10, 25, 67, 88}
```

Manipulating strings

```
fruits=['apple','banana','orange']
fruits.append('mango')
fruits.remove('banana')
fruits
```

```
['apple', 'orange', 'mango']
```

Creating a dictionary

```
book={'title':'the python','author':'john','year':2021}
print(book['title'])
print(book['author'])
print(book['year'])
```

```
the python
john
2021
```

Working with sets

```
rani={1,2,3,4,5}
raja={1,4,6,7}
rani|raja
```

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

Merge two lists

```
list1=[1,2,3,4]
list2=[5,6,7,8]
merged_list=list1+list2
merged_list
```

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

Dictionary operations

```
student={'name':"john",'age':21,'marks':89}
print("name:",student["name"])
student['marks']=90
print('updated marks;',student['marks'])
```

```
name: john
updated marks; 90
```

Find the maximum and minimum in a list

```
numbers=[10,20,30,40,50]
print("maximum:",max(numbers))
print("minimum:",min(numbers))
```

```
maximum: 50
minimum: 10
```

count frequency of eliments in a list

```
numbers=[1,2,3,3,4,4,5,6,7,5]
frequency={}
for num in numbers:
  frequency[num]=frequency.get(num,0)+1
print('frequency of elements:',frequency)
```

```
frequency of elements: {1: 1, 2: 1, 3: 2, 4: 2, 5: 2, 6: 1, 7: 1}
```

sort a list of tuples by the second element

```
tuples=[('a',3),('b',1),('c',2)]
sorted_tuples=sorted(tuples,key=lambda x:x[1])
print('sorted tuples:',sorted_tuples)
```

```
sorted tuples: [('b', 1), ('c', 2), ('a', 3)]
```

Palindrome number

```
number=input('emter a number:')
if number==number[::-1]:
  print(f"{number} is palindrome number")
else:
  print(f"{number} is not palindrome number")
```

```
emter a number:545
545 is palindrome number
```

```
number=int(input('enter a number:'))
reverse_number=0
temp=number
while temp>0:
  digit=temp%10
  reverse_number=reverse_number*10+digit
  temp//=10
if number==reverse_number:
  print(f"{number} is palindrome number")
else:
    print(f"{number} is not palindrome number")
```

```
enter a number:121
121 is palindrome number
```

```
x=int(input('enter a number;'))
if str(x)==str(x)[::-1]:
 print("true")
else:
 print("false")
```

```
enter a number;232
true
```

```
class Solution(object):
    def isPalindrome(self, x):
        """
        :type x: int
        :rtype: bool
        """
        if x < 0 or (x % 10 == 0 and x != 0):
            return False

        reversed_half = 0
        while x > reversed_half:

            reversed_half = reversed_half * 10 + x % 10
            x //= 10


        return x == reversed_half or x == reversed_half // 10


solution = Solution()
print(solution.isPalindrome(121))
print(solution.isPalindrome(-121))
print(solution.isPalindrome(10))
print(solution.isPalindrome(0))
```

```
True
False
False
True
```

Magic 8 ball program

```
import random
responses = [
"It is certain",
"Reply hazy, try again",
"Don’t count on it",
"Yes, definitely",
"Ask again later",
"My reply is no",
"Outlook not so good",
"Signs point to yes"
]
print("Ask the Magic 8 Ball a question: ")
input()
print(random.choice(responses))
```

```
Ask the Magic 8 Ball a question:
6
Don’t count on it
```

```
#using copy
original = [1, 2, 3]
copy = original
copy[0] = 99
print(original)
```

```
[99, 2, 3]
```

pretty print

```
import pprint
data = {"name": "Alice", "subjects": ["Math", "Science"], "grades":
{"Math": "A",
"Science": "B"}}
pprint.pprint(data)
```

```
{'grades': {'Math': 'A', 'Science': 'B'},
 'name': 'Alice',
 'subjects': ['Math', 'Science']}
```

Representing a phone book using dictionaries.

```
phone_book = {
"John": "555-1234",
"Alice": "555-5678",
"Bob": "555-8765"
}
print(phone_book["Alice"])
```

```
555-5678
```