DATA STRUCTURES

creating list with same data type

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

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

creating list with different data type

```
b=[1,3.5,2,"hello"]
print(b)
```

```
[1, 3.5, 2, 'hello']
```

LIST OPERATIONS

Accesing items

```
print(a[0])
print(b[1])
print(a[3])
```

```
1
3.5
4
```

Modifying items

```
a[0]=10
print(a)
```

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

Adding items

```
a=[1,2,3,4,5]
a.append(10)
print(a)
```

```
[1, 2, 3, 4, 5, 10]
```

```
a=[1,2,3,4,5,6,7]
a.insert(3,10)
print(a)
```

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

Removing items

```
c=[1,2,3,4,5,6,7,8,9]
c.remove(6)
print(c)
```

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

```
r=[1,2,3,4,6,7,9,10]
r.pop(5)
print(r)
```

```
[1, 2, 3, 4, 6, 9, 10]
```

Other operations

```
a=[1,3,5,6,8,9]
print(len(a))
```

```
6
```

```
a=[3,5,4,6,2,7,9,8]
a.sort()
print(a)
a.reverse()
print(a)
```

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

Tuple

Creating a tuple

```
a=(1,2,3,4,6,5,7)
print(a)
```

```
(1, 2, 3, 4, 6, 5, 7)
```

Accesing items in a tuple

```
a[6]
```

```
7
```

Dictionaries

Creating a dictionaries

```
n={"name":"rakshi","age":18,"gender":"female"}
print(n)
```

```
{'name': 'rakshi', 'age': 18, 'gender': 'female'}

n["name"]="kitty"
n

{'name': 'kitty', 'age': 18, 'gender': 'female'}
```

Accessing and modifying items

accessing:

```
student={"name":"rakshi","age":18,"gender":"female"}
student

{'name': 'rakshi', 'age': 18, 'gender': 'female'}
```

modifying:

```
student["age"]=19
student

{'name': 'rakshi', 'age': 19, 'gender': 'female'}
```

Adding:

```
student["grade"]="A"
student

{'name': 'rakshi', 'age': 19, 'gender': 'female', 'grade': 'A'}
```

Removing:

```
del student["gender"]
student

{'name': 'rakshi', 'age': 19, 'grade': 'A'}
```

Iterating through a dictionary

```
print(student.keys())
print(student.values())
print(student.items())

dict_keys(['name', 'age', 'grade'])
dict_values(['rakshi', 19, 'A'])
dict_items([('name', 'rakshi'), ('age', 19), ('grade', 'A')])
```

SETS

```
numbers={1,2,3,4,5,6,7,8,9,10}
numbers
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

set operations

adding items

```
numbers.add(16)
numbers
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16}
```

removing items

```
numbers.remove(9)
numbers
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 10, 16}
```

SET OPERATIONS

```
a={1,2,3,4,5,11,24,56}
b={3,4,5,6,7,11,24,12}
print(a.union(b))
print(a.intersection(b))
print(a.difference(b))
```

```
{1, 2, 3, 4, 5, 6, 7, 11, 12, 24, 56}
{3, 4, 5, 11, 24}
{56, 1, 2}
```

Merge two lists

```
list1=[1,2,3,]
list2=[4,5,6]
merged_list=list1+list2
print("merged list:",merged_list)
```

```
merged list: [1, 2, 3, 4, 5, 6]
```

Maximum and minimum in a list

```
numbers=[10,20,30,40,50,60,70]
print("maximum:",max(numbers))
print("minimum:",min(numbers))
```

```
maximum: 70
minimum: 10
```

Count frequency of elements in a list

```
numbers=[1,2,2,3,3,3,4,4,4,4]
frequency={}
for num in numbers:
  if num in frequency:
    frequency[num]+=1
  else:
    frequency[num]=1
    print(" element frequency:", frequency)
```

```
 element frequency: {1: 1}
 element frequency: {1: 1, 2: 1}
 element frequency: {1: 1, 2: 2, 3: 1}
 element frequency: {1: 1, 2: 2, 3: 3, 4: 1}
```

sort a list of tuples by the second elements

```
tuples=[(1,'apple'),(2,'banana'),(3,'cherry')]
sorted_tuples=sorted(tuples,key=lambda X:X[1])
print("sorted tuples:", sorted_tuples)
```

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

PALINDROME NUMBERS

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

```
enter a number:353
353 palindrome
```

PALINDROME 2

```
number=input("enter a number:")
if number==number[::-1]:
  print("palindrome")
```

```
else:
  print("not palindrome")

enter a number:rakshi
not palindrome
```