

Build a python program utilising data structures,conditional statements and functions to solve a data-related task.

LIST

```
#Creating a list  
lst=('Name')
```

```
lst
```

```
'Name'
```

```
#list with different data types  
lst=['Name',True,70]
```

```
#Initialize a list with list function  
lst1=list()
```

```
lst1
```

```
[]
```

```
#Accessing a list  
#len function  
len(lst1)
```

```
0
```

Len function as in above,is use to find out the number of elements in a list

```
lst[2]
```

```
70
```

Using indexing to find out the sequence of an element in a list

Modifying Elements in a list

```
#Creating a new list  
lst2=['Maryam','Ahmad','Juwairiyya','Khadija','Umar']
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Khadija', 'Umar']
```

```
#Adding an element to the list  
lst2[3]='Zainab'
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Zainab', 'Umar']
```

When you add an element in place of another element like i did in the above cell,it takes it's place by removing the initial element

```
#Inserting an element in the list  
lst2.insert(5,'Hussain')
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Zainab', 'Umar', 'Hussain']
```

```
#Inserting an element using-Append  
lst2.append('Khadija')
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Zainab', 'Umar', 'Hussain', 'Khadija']
```

In inserting an element like in the above cell,the element is added without removing any of the elements

```
#Removing an element from a list  
#Using_Remove  
lst2.remove('Hussain')
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Zainab', 'Umar']
```

```
#Using pop_function  
pop_lst=lst2.pop()
```

```
lst2
```

```
['Maryam', 'Ahmad', 'Juwairiyya', 'Zainab']
```

```
pop_lst
```

```
    'Umar '
```

NUMBERS

```
#Numbers: Integers
```

```
Var1=70
```

```
Var1
```

```
    70
```

```
#To know which data type the number belongs to:
```

```
type(Var1)
```

```
    int
```

```
#Numbers: Floats
```

```
Var2=30.5
```

```
Var2
```

```
    30.5
```

```
#To know which data type the number belongs to:
```

```
type(Var2)
```

```
    float
```

```
#Trying an arithmetic expressions
```

```
Var1*2
```

```
    140
```

```
Var2+0.5
```

```
    31.0
```

Organising a list

```
#Creating a new list
```

```
lst3=['Fatima','Abdul','Amina','Bilyamin','Ibrahim','Binta']
```

```
print(lst3)
```

```
    ['Fatima', 'Abdul', 'Amina', 'Bilyamin', 'Ibrahim', 'Binta']
```

Sorting of a list

```
#Using the sort() method
#Copy the list
lst3_copy=lst3.copy()

lst3_copy

['Fatima', 'Abdul', 'Amina', 'Bilyamin', 'Ibrahim', 'Binta']

#Sorting
lst3_copy.sort()

lst3_copy

['Abdul', 'Amina', 'Bilyamin', 'Binta', 'Fatima', 'Ibrahim']
```

As you can see,the list above is sorted alphabetically and it's permanent

TUPLES

Tuples are immutable data structures,which makes them impossible to modify either by adding,inserting,removing,etc

```
#Creatung a tuple
tuple=(23,'maryam',56,'Hussain')

tuple2=('Amina','Hafiz',54,'taufeeq')

print(tuple)

(23, 'maryam', 56, 'Hussain')

print(tuple2)

('Amina', 'Hafiz', 54, 'taufeeq')
```

SETS

Sets are also immutable data structures that don't allow duplicate element and the order of the elements aren't specific or guaranteed

```
#Creating a set
set_1={2,8,4,3,6,2,8,5,9,1}
```

```
set_1

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

As you can see in the above cell,all duplicate elements are gone and the order of the elements is changed

```
#Creating a set with a list
set_lst=[2,8,4,3,6,2,8,5,9,1]
```

```
set_lst

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

```
#creating another set
set_2={5,7,'Ibrahim',4,5,'Aisha',4,4,5,'Aisha',7}
```

```
set_2

{4, 5, 7, 'Aisha', 'Ibrahim'}
```

```
#Accessing elements in a set
print('Aisha' in set_2)
```

```
True
```

I accessed the element 'Aisha' to find out whether it exist in the set

SET UNION

Set union is used to unite 2 or more sets as the name Implies

```
#Create 2 sets
set1=set_1
set2=set_2
```

```
#Uniting the 2 sets
set3 = set1.union(set2)
print(set3)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 'Aisha', 'Ibrahim'}
```

SET INTERSECTION

```
#Intersecting the 2 sets
set4 = set1.intersection(set2)
print(set4)
```

```
{4, 5}
```

DOCTIONARY

Dictionary is another data structure, it is mutable and is made up of key-value pairs, where a key and value are paired or connected by a colon(:) and each pair is separated from another by a comma(,)

```
#Creating a Dictionary
dict={'name':'Maryam','age':22,'job':'Nurse'}
```

```
#Printing the whole Dictionary
print(dict)
```

```
{'name': 'Maryam', 'age': 22, 'job': 'Nurse'}
```

```
#printing each pair individually
print(dict['name'])
```

```
Maryam
```

```
print(dict['age'])
```

```
22
```

```
print(dict['job'])
```

```
Nurse
```

```
#Adding a key-Value pair
dict['address']='Fatara'
```

```
dict
```

```
{'name': 'Maryam', 'age': 22, 'job': 'Nurse', 'address': 'Fatara'}
```

Double-click (or enter) to edit

```
#Modifying the dictionary
dict['address']='Dutse'
```

```
dict
```

```
{'name': 'Maryam', 'age': 22, 'job': 'Nurse', 'address': 'Dutse'}
```

By modifying the key-value pair,I am able to change a pair from one thing to another

```
#Removing a key-value pair
del dict['address']
```

```
dict
```

```
{'name': 'Maryam', 'age': 22, 'job': 'Nurse'}
```

When removing a key-value pair,you don't have to call the value,you just mention the key and the value will automatically disappear with the key

LOOPING THROUGH A DICTIONARY

```
#Write a for loop for dict
for key, value in dict.items():
    print(f'\nKey: {key}')
    print(f'Value: {value}')
```

```
Key: name
Value: Maryam
```

```
Key: age
Value: 22
```

```
Key: job
Value: Nurse
```

CONDITIONAL STATEMENTS

```
#Conditional statement
BMI = 30.0
if BMI < 30.0:
    print('Not obese')
elif BMI==30.0:
    print('Obese')
```

```
else:  
    print('dangerously obese')
```

Obese

```
BMI_2= 25.5  
if BMI_2 > 20.5:  
    print('healthy')  
elif BMI_2 ==25.5:  
    print('normal')  
else:  
    print('risk of obesity')
```

healthy

FUNCTIONS

```
#Creating a function  
greetings= 'Assalamu Alaikum '
```

```
print(greetings)
```

Assalamu Alaikum

```
#Defining a function  
def greet_people():  
    """greeting function"""  
    print('Assalamu Alaikum')
```

```
#Function call  
salute_people()
```

Assalam

```
#Another Example  
def greet(name):  
    print("Salam, " + name)
```

```
#Calling the function  
greet('Maryam')
```

Salam, Maryam

