AI WORKSHOP

PYTHON

What are the basic Data Structures in Python?

LISTS

It is a data type that stores a collection of data by storing several values in a single variable.

They are mutable(i.e. We can make any change in list),ordered,can be nested,allows duplicate elements.

Lists are used to store data that needs tampering regularly, i.e. adding,changing or deleting elements from the list

Elements in lists can be accessed easily using indexing.

What do I do if I want to make sure the data is unchanged?

TUPLE

A Tuple is a collection of Python objects separated by commas. In some ways, a tuple is similar to a list in terms of indexing, nested objects.

However unlike lists tuples are immutable i.e. We cannot add,change or delete data from the tuple.

We can however convert a tuple to a list, modify the list and convert it back to tuple.

Tuples can be used to store data which is not meant to be changed, as tuples prevent any accidental deletion or insertion of elements in the tuple.

Sets

A Set is an unordered collection data type that is iterable, mutable, and has no duplicate elements.

Sets are unordered thus we cannot access the elements using indexing or a key.However you can iterate over elements.

Sets elements cannot be changed but you can delete and add elements into the set as long as there are no duplicates in the set.

Python's set class represents the mathematical notion of a set.

Why do we use sets?

DICTIONARIES

Dictionary is an ordered collection of data values, used to store data values like a map, which, unlike other Data Types that hold only a single value as an element, Dictionary holds key:value pair to make it more optimized

The keys of dictionaries must be immutable ,ex: Int,str,float,boolean….

Dictionaries are mutable as long as keys are not duplicated.

Dictionaries can be indexed using the keys to obtain the respective values.

Why do we use dictionaries?

FUNCTIONS

How does functions help?

FUNCTIONS : MOTIVATION

- Use same piece of code to run multiple times
- Easy to write, read etc

FUNCTIONS

```
def  myfunction():
        #statements
```

ARGUMENTS

```
def  add(a,b):
        sum=a+b
        return sum
```

FUNCTION CALL

```
myfunction()

add(1,2)
```

Functions inside a Function

```
def outerFunction(text):
    text = text

    def innerFunction():
        print(text)

    innerFunction()
```

Can we access innerFunction by any chance ?

RECURSION

A complicated function can be split down into smaller sub-problems utilizing recursion.

```
def fact(n):
        if n == 0:
            return 1
        else:
            return   n * fact(n-1)
```

CLASSES

Why do we need Classes ?

CLASSES & OBJECTS

CAR
OBJECT

ATTRIBUTES

- NAME
- COLOR
- PRICE

METHODS

- ACCELERATE
- CHANGE GEAR
- APPLY BRAKE

CLASSES : MOTIVATION

- Faster and easier to execute
- Reusable code
- Easy to maintain , modify , debug

CLASSES

Blueprint for creating objects.

```
class Car:
    def __init__(self,name,color,price):
        self.name = name
        self.color = color
        self.price =price

    def brake(self):
        self.speed=0
        print("Car Stopped")
```

__init__()

- Assign values to object properties
- The function is called automatically every time the class is being used to create a new object.

```
class Car:
  def __init__(self,name,color,price):
    self.name = name
    self.color = color
    self.price =price

  def brake(self):
    self.speed=0
    print("Car Stopped")
```

self Parameter

- Refers to the current object of the class.
- Used to access variables that belongs to the class.
- Has to be the first parameter of any function in the class.

```
class Car:
  def __init__(self,name,color,price):
    self.name = name
    self.color = color
    self.price =price

  def brake(self):
    self.speed=0
    print("Car Stopped")
```

OBJECT (INSTANCE)

Create object	car1 = Car (“ABC” , “WHITE” , 1000)
Access attributes	print (car1.color)
Access member functions	car1.brake()

Modify object Properties

car1.color = “Red”

Delete object Property

del car1.color

Delete object

del car1

THANK YOU

RESOURCES

https://www.w3schools.com/python/