PYTHON

1. What is init keyword ?

* In Python, \_\_init\_\_ is a special method known as the constructor. It is automatically called when a new instance (object) of a class is created.
* The \_\_init\_\_ method allows you to initialize the attributes (variables) of an object.

Eg: class Employee

def \_\_init\_\_(self,emp\_name,emp\_age):

self.emp\_name = emp\_name

self.emp\_age = emp\_age

def pint(self):

print(f”Employee name : {self.emp\_name}”)

print(f”Employee name : {self.emp\_age}”)

#---creating instance of Employee class

obj = Employee(“Sam”,30)

#--accessing method and attribute

obj.print()

1. What is self keyword ?

* The self keyword is used to represent an instance (object) of the given class.
* It allows access to the attributes and methods of each object in python. This allows each object to have its own attributes and methods.
* Self is always pointing to Current Object.

Eg : class Car:  
 def \_\_init\_\_(self, brand, model):  
 self.brand = brand  
 self.model = model  
  
 def info(self):  
 print(f"Car brand is {self.brand}. Model is {self.model} ")  
  
 def make\_sound(self):  
 print("Meow")  
obj = Car("Maruthi suzuki", "Swift")  
obj.info()

1. What is lambda functon?

* Lambda functions are similar to user-defined functions but without a name. They are commonly referred to as anonymous functions.
* Lambda functions are efficient whenever you want to create a function that will only contain simple expressions i.e., expressions that are usually a single line of a statement
* A lambda function can take any number of arguments, but can only have one expression.
* lambda argument(s) : expression

Eg: sum of 3 numbers:

x = lambda a, b, c : a + b + c  
print(x(5, 6, 2))

Find even numbers fom the list:

list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

print(list(filter(lambda x: x % 2 == 0, list1))

1. Difference between lambda and normal function?

Lambda Function:

* Interpretation might be tricky
* The limited operation can be performed using lambda functions
* No need of using the return statement
* Execution time of the program is fast for the same operation
* Defined using the keyword **lambda** and does not compulsorily hold a function name in the local namespace

Normal Function:

* Easy to interpret
* Can consists of any number of execution statements inside the function definition
* To return an object from the function, return should be explicitly defined
* Execution time is relatively slower for the same operation performed using lambda functions
* Defined using the keyword **def** and holds a function name in the local namespace

1. What are generators?

* A Generator is a function that returns an iterator using the Yield keyword
* If the body of a def contains yield, the function automatically becomes a Python generator function.
* def function\_name():  
   yield statement

Eg: Fibonacci series using generator:

def fibonacci(limit):  
 a,b = 0,1  
 while a < limit:  
 yield a  
 a,b = b,a + b  
x=fibonacci(5)  
# Iterating over the generator object using for loop.  
for i in fibonacci(5):  
 print(i)

1. Python is compiled or interpreted language? what does it mean?

The python program is first compiled and then interpreted. The compilation part is hidden. The compilation part is performed by Python Interpreter and done first when we execute our code and this will generate byte code and internally this byte code gets converted by the python virtual machine (p.v.m) according to the underlying platform (machine + operating system).

1. What is the difference between list and tuples in Python?

List :

* Lists are Mutable
* Iteration is time-consuming
* List is better for performing operations such as insertion and deletion
* Lists consume more memory
* Lists have several built-in methods

Tuple:

* Tuples are Immutable
* Iteration is comparatively faster
* Tuple data type is appropriate for accessing the elements
* Tuple consume less memory as compared to the list
* Tuple doesn’t have many built in methods

1. What is the difference between list and set in Python?

List:

* List can have duplicate values
* List is ordered
* Can change the items in a list and can add to the list

Set

* A set cannot have duplicate values. All values must be unique.
* A set is unordered
* Can't change the items of a set, but can add to the set and remove from it.

1. When to use dictionary?

When you want a mapping from keys to values, use a dictionary.

Eg: when you want a telephone book which maps names to phone numbers: {'John Smith' : '555-1212'}).

Note the keys in a dict are unordered. (If you iterate through a dict (telephone book), the keys (names) may show up in any order).

1. What are decorators?

A decorator is a design pattern that allows a user to add new functionality to an existing object without modifying its structure. Decorators are usually called before the definition of a function you want to decorate

* It allows programmers to modify the behaviour of a function or class.
* Decorators allow us to wrap another function in order to extend the behaviour of the wrapped function, without permanently modifying it.

1. What are Iterators?

* An iterator is an object that contains a countable number of values.
* Iterator can traverse through all the values.
* It implements the iterator protocol, which consist of the methods \_\_iter\_\_() and \_\_next\_\_().
* \_\_iter\_\_() : Called to initialize the iterator. It must return an iterator object.
* \_\_next\_\_() : Called to iterate over the iterator. It must return the next value in the data stream.

1. What is slicing?

Slicing is a feature that enables accessing parts of sequences like strings, tuples, and lists. You can also use them to modify or delete the items of mutable sequences such as lists. Slices can also be applied on third-party objects like NumPy arrays, as well as Pandas series and data frames.

1. What is mutable and immutable?

Mutable is the ability of objects to change their values. These are often the objects that store a collection of data.

Immutable is the value of an object cannot be changed over time. Once created, the value of these objects is permanent.