OJT - Python Exercise

1. What are the differences between list and tuple? Which is faster and why?

Ans:

List :  list in Python is used to store various types of data. A list can be defined as a collection of values or items of different types and separated by comma within a Square Bracket [ ]

Tuple: tuple in Python is used to store various types of data. A tuple can be defined as a collection of values or items of different types and separated by comma within a parentheses ( )

|  |  |
| --- | --- |
| List | Tuple |
| Store in a Square Bracket [ ] | Store in a parentheses ( ) |
| Mutable in nature | Immutable in nature |
| Consume more space or memory | Consume less space or memory |
| Slow in nature | Fast in nature |

Which is faster and why?

Ans: Tuple is faster than list because Tuples are immutable data types which are stored in a single block of memory fixed size memory so it doesn’t require extra space to store new objects but list are mutable data types and are allocated in two blocks where the fixed one with all object information and a variable sized block for the data.

1. What is the lambda function in python? Explain with examples.

Ans:

A lambda function is an anonymous function (defined without a name) that can take any number of arguments but, unlike normal functions, evaluates and returns only one expression.

Keyword : lambda

Example:

z=(lambda x: x \* 10 if x > 10 else ( x \* 5 if x < 5 else x))

print(z(12))

output: 120

Map Example: use to map the task to every individual element

Lst=[1,2,3,4,5]

r=(list(map(lambda x : x\*2,lst)))

print(r)

output: [2,4,6,8,10]

reduce Example: reduce sequence to find consolidated result

Lst=[1,2,3,4,5]

r=(reduce(lambda x,y : x+y,lst))

print(r)

output: 15

Example:

r=(lambda x, y,z : x+y-z)

print(r(12,2,1))

output: 13

filter Example:

lst=[1,3,4,5,6]

print(list(filter(lambda x: x >4,lst )))

out put: [5, 6]

**Monkey Patching**

Monkey-patching is a term that refers to modifying a class or module at a run time. In simple words, a class or module's work can be changed at the runtime.

This allows us to modify its behavior at run time.

It is use to modify and update the some part of code and working of class/module at run time without changing the whole code .

Example:

class main\_func:

def og\_func(self):

print ("og\_func() is being called by developer")

def monkey\_f(self):

print ("monkey\_f() is being called by developer ")

# replacing address of "og\_func" with "monkey\_f"

main\_func.og\_func = monkey\_f

obj = main\_func ()

# calling function "og\_func" whose address got replaced with function "monkey\_f()"

obj.og\_func()

PEP (Python enhancement protocol):

* The main purpose of PEP is to enhance the readability and consistency of code.
* A PEP is a design document providing information to the Python community, or describing a new feature for Python or its processes or environment.

**Value Error:**

Value Error is a built-in Python exception that is raised when the type of a passed argument to a function is incorrect or when a function is called with an argument that has an invalid value