Q1

List1.py

```
1  list1 = [1.0, 2.3, "hello"]
2  list2 = ["hi", 8.3, 9.6, "how"]
3
4  #print the list1 and list2
5  print("List1 Elements are:",list1)
6  print("List2 Elements are:",list2)
7  #Concatenate list1 and list2 and print the result
8  print("List after Concatenation:",list1+list2)
9
10
```

Q2

llows :

List2.py

```
1  list1 = ["hi", "hello", "Lists"]
2  # print every element in list1 using index
3  print(list1[0])
4  print(list1[1])
5  print(list1[2])
6  list1[2] = "Python"
7  # print list1
8  print(list1)
9  # add "Code is Life" to list1
10 list1.append("Code is Life")
11 # print list1
12 print(list1)
13 list1.extend([45, 67, 89])
14 # print list1
15 print(list1)
```

Q1

ıtable. We can

- [ ] A set is an ordered collection of unique items.
- [ ] myset = { }, creates an empty set.
- [x] A set is represented using curly braces **{ }** .
- [ ] Set allows duplicate elements.
- [ ] Set is a immutable data type.
- [ ] The elements of a set are mutable.

Q1

ıutable), while

ce

and a list into a

- [ ] Tuples are used to store similar type of data.
- [ ] tuple1 = (1.0) is correct way to create a tuple with single element.
- [x] tuples are immutable.
- [ ] Lists are immutable.
- [x] Converting a tuple into a list and list into tuple is possible.
- [ ] Lists are faster than tuples.

Q2

Tuple1.py

```python
mytuple = ("this", 10.0, "is", "float", 3.6)

# Print value at index 0
print(mytuple[0])

# Print value at index 1
print(mytuple[1])
# Print value at index -1
print(mytuple[-1])
# Print all the values from index 0
print(mytuple[0:])
# Print all the values except the last value
print(mytuple[:4])

print(mytuple[::-1])
```

lement

Q3

Tuple2.py

```python
mytuple = ("i", "love", "python")

#Write the missing code here
print("Given Tuple:",mytuple)
myList=list(mytuple)
print("After Converting Tuple into List:",myList)
myList[1]="practice"
print("List after changing element:",myList)
mytuple=tuple(myList)
print("After Converting List into Tuple:",mytuple)

```