**Question 1**

i. List:

A list is an ordered collection of items in Python. Lists are used to store multiple items in a single variable. Lists are created using square brackets [], with the items separated by commas. For example:

Example:

*fruits = ['apple', 'banana', 'cherry']*

Lists can contain items of different data types, including numbers, strings, and other objects. Lists are mutable, meaning you can change, add, and remove elements after the list is created.

ii. Files:

Files are used to permanently store data. Files can be of various types, such as text files, binary files, and so on. Python has a built-in open() function to open and work with files. Once a file is opened, you can perform operations like reading, writing, and appending data to the file. For example:

Example:

*file = open("example.txt", "r")*

*content = file.read()*

*file.close()*

iii. Dictionaries:

Dictionaries are used to store data in key-value pairs. They are unordered collections of items, where each item has a key and a corresponding value. Dictionaries are created using curly braces {}, with the key-value pairs separated by commas. For example:

Example:

*person = {*

*"name": "John Doe",*

*"age": 30,*

*"city": "New York"*

*}*

Dictionaries are useful for storing and retrieving data efficiently, as you can access the values using their corresponding keys.

**Question 2**

i. Concatenation:

Concatenation is the process of joining two or more strings together. In Python, you can use the + operator to concatenate strings. For example:

Example:

*first\_name = "John"*

*last\_name = "Doe"*

*full\_name = first\_name + " " + last\_name*

*print(full\_name) # Output: John Doe*

ii. Repetition:

Repetition is the process of creating a new string or list by repeating a specific element multiple times. In Python, you can use the \* operator to repeat strings or lists. For example:

Example 1:

*greeting = "Hello "*

*repeated\_greeting = greeting \* 3*

*print(repeated\_greeting) # Output: Hello Hello Hello*

Example 2:

*numbers = [1, 2]*

*repeated\_numbers = numbers \* 3*

*print(repeated\_numbers) # Output: [1, 2, 1, 2, 1, 2]*

iii. Deletion:

Deletion is the process of removing an element from a list or a key-value pair from a dictionary. In Python, you can use the del keyword to delete elements. For example:

Example 1:

*fruits = ['apple', 'banana', 'cherry']*

*del fruits[1]*

*print(fruits) # Output: ['apple', 'cherry']*

Example 2 :

*person = {*

*"name": "John Doe",*

*"age": 30,*

*"city": "New York"*

*}*

*del person["age"]*

*print(person) # Output: {'name': 'John Doe', 'city': 'New York'}*

In the list example, we used del fruits to remove the element at index 1 (which is 'banana'). In the dictionary example, we used del person["age"] to remove the key-value pair with the key "age".