**INBUILT DATA STRUCTURES**

**Queue:**

Queue is a FIFO data structure. To add the elements use put method, and to retrieve use get method.

**Code:**

import queue  
  
qu = queue.Queue()  
qu.put("Apple - iPhone")  
qu.put("Samsung - ZFold")  
qu.put("Samsung - Flip")  
qu.put("Samsung - S24")  
qu.put("Huawai - MX4")  
qu.put("Oppo - A700")  
  
*#BEFORE RETRIVAL*print("Size of the Queue before Removal", qu.qsize())  
  
*# GET THE ELEMENTS*while not qu.empty():  
 *# IT WILL REMOVE THE ELEMENT ONCE WE USED THE GET METHOD* print(qu.get())  
  
*#AFTER RETRIVAL*print("Size of the Queue after Removal", qu.qsize())

import queue  
  
qu = queue.Queue()  
qu.put("Apple - iPhone")  
qu.put("Samsung - ZFold")  
qu.put("Samsung - Flip")  
qu.put("Samsung - S24")  
qu.put("Huawai - MX4")  
qu.put("Oppo - A700")  
  
*#PRINT ALL THE ELEMENTS IN THE QUEUE*print(qu.queue)  
print("=====================================")  
  
*# TO CLEAR THE QUEUE*qu.queue.clear()  
  
*#BEFORE RETRIVAL*print("Size of the Queue before Removal", qu.qsize())  
print("=====================================")  
*# GET THE ELEMENTS*while not qu.empty():  
 *# IT WILL REMOVE THE ELEMENT ONCE WE USED THE GET METHOD* print(qu.get())  
print("=====================================")  
*#AFTER RETRIVAL*print("Size of the Queue after Removal", qu.qsize())  
print("=====================================")

**Output:**

Size of the Queue before Removal 6

Apple - iPhone

Samsung - ZFold

Samsung - Flip

Samsung - S24

Huawai - MX4

Oppo - A700

Size of the Queue after Removal 0

deque(['Apple - iPhone', 'Samsung - ZFold', 'Samsung - Flip', 'Samsung - S24', 'Huawai - MX4', 'Oppo - A700'])

=====================================

=====================================

Size of the Queue before Removal 0

=====================================

=====================================

Size of the Queue after Removal 0

=====================================

**LFQ:**

LIFO Queue is like a stack data structure. LIFO stands for Last In First Out. The get method is like a pop operation. The put method is like a push operation.

**Code:**

import queue  
  
lq = queue.LifoQueue()  
lq.put("C# .Net")  
lq.put("ASP.Net")  
lq.put(".Net Core")  
lq.put("ASP MVC")  
lq.put("Java")  
lq.put("Python")  
lq.put("Django")  
  
print(lq.queue)  
print()  
  
while not lq.empty():  
 print(lq.get())  
  
print()  
print("Size after GET Method", lq.qsize())

**Output:**

['C# .Net', 'ASP.Net', '.Net Core', 'ASP MVC', 'Java', 'Python', 'Django']

Django

Python

Java

ASP MVC

.Net Core

ASP.Net

C# .Net

Size after GET Method 0