```
In [80]:
              #Represents the node of list.
            1
            2
              class Node:
            3
                   def init (self,data):
                       self.data = data
            4
            5
                       self.next = None
            6
            7
              class CreateList:
                    #Declaring head and tail pointer as null.
            8
                   def __init__(self):
           9
                       self.head = Node(None)
           10
           11
                       self.tail = Node(None)
           12
                       self.head.next = self.tail
           13
                       self.tail.next = self.head
                       #self.dataadd = dataadd
           14
           15
                   def add(self,data):
           16
                       newNode = Node(data)
                       #Checks if the list is empty.
           17
           18
                       if self.head.data is None:
           19
                       #If list is empty, both head and tail would point to new node.
           20
           21
                           self.head = newNode
           22
                           self.tail = newNode
           23
                           newNode.next = self.head
           24
                       else:
           25
                           #tail will point to new node.
           26
                           self.tail.next = newNode
           27
                           #New node will become new tail.
           28
                           self.tail = newNode
           29
                           #Since, it is circular linked list tail will point to head.
           30
                           self.tail.next = self.head
           31
           32
           33
                   def search(self,element):
           34
                       current = self.head;
           35
                       i = 1;
           36
                       flag = False;
                       #Checks whether list is empty
           37
           38
                       if(self.head == None):
           39
                           print("List is empty");
           40
                       else:
           41
                           while(True):
           42
                               #Compares element to be found with each node present in the
           43
                               if(current.data == element):
                                   flag = True;
           44
           45
                                   break;
           46
                               current = current.next;
           47
                               i = i + 1;
           48
                               if(current == self.head):
           49
                                   break;
           50
                           if(flag):
           51
                               print("Element is present in the list at the position : " +
           52
                           else:
           53
                               print("Element is not present in the list");
           54
           55
           56
                 #Displays all the nodes in the list
```

```
57
         def display(self):
 58
             current = self.head
 59
             if self.head is None:
                 print("List is empty")
 60
 61
                 return
 62
             else:
                 print("\nNodes of the circular linked list: ")
 63
                 #Prints each node by incrementing pointer.
 64
 65
                 print(current.data)
                 while(current.next != self.head):
 66
                     current = current.next
 67
                     print(current.data)
 68
 69
 70
         #This function will add to the end of the list.
 71
         def addAtEnd(self,data):
 72
             newNode = Node(data);
 73
             #Checks if the list is empty.
 74
             if self.head.data is None:
 75
                 #If list is empty, both head and tail would point to new node.
 76
                 self.head = newNode;
 77
                 self.tail = newNode;
 78
                 newNode.next = self.head;
 79
             else:
 80
                 #tail will point to new node.
 81
                 self.tail.next = newNode;
 82
                 #New node will become new tail.
 83
                 self.tail = newNode;
                 #Since, it is circular linked list tail will point to head.
 84
 85
                 self.tail.next = self.head;
         def delete(self, value):
 86
             current = self.head
 87
 88
             prev = current.next
 89
             while prev.next != self.head:
 90
                 prev = prev.next
 91
             if current.data == value:
 92
                 prev.next = current.next
 93
 94
             current = current.next
 95
             prev = prev.next
 96
             while current.data != value and current != self.head:
 97
                 current = current.next
 98
                 prev = prev.next
 99
100
             if current.data == value:
101
                 prev.next = current.next
102
```

```
In [81]: cl = CreateList();
         i = True
         print("Enter The Data Below To Perform CircularLinkedList or Enter Exit to End")
         while i == i:
             a=input("Enter : ")
             if a.lower() == "exit":
                 i=False
                 break
             else:
                 cl.add(a)
         cl.display();
         Enter The Data Below To Perform CircularLinkedList or Enter Exit to End
         Enter : Awais
         Enter : Ajmal
         Enter : Arham
         Enter : Aqib
         Enter: Manohar
         Enter : Apple
         Enter: 3
         Enter: 2.0
         Enter: exit
         Nodes of the circular linked list:
         Awais
         Ajmal
         Arham
         Aqib
         Manohar
         Apple
         2.0
In [82]: | nsearch = input("Enter The Data To Find : ")
         cl.search(nsearch)
         Enter The Data To Find : Apple
```

Element is present in the list at the position : 6

localhost:8888/notebooks/Circular linked list Pyhton.ipynb

```
In [83]: print("Enter The Number to add in A List")
         class CircularLinkedList:
             addinalist = input("Enter : ")
             #Adding 1 to the list
             cl.addAtEnd(addinalist);
             print("\nAdded Successfully")
             cl.display();
         Enter The Number to add in A List
         Enter : Mazz
         Added Successfully
         Nodes of the circular linked list:
         Awais
         Ajmal
         Arham
         Aqib
         Manohar
         Apple
         3
         2.0
         Mazz
In [84]: | ndelete = input("Enter The Data To Delete From List : ")
         cl.delete(ndelete)
         Enter The Data To Delete From List: 3
In [85]: print("\nAfter Deletion Process Data:\n")
         cl.display();
         After Deletion Process Data:
         Nodes of the circular linked list:
         Awais
         Ajmal
         Arham
         Aaib
         Manohar
         Apple
         2.0
         Mazz
 In [ ]: |#Done
```