**#Linked-List Python Implementation**

**class** Node:

1. **def** \_\_init\_\_(self, data, next\_node=None):
2. self.data = data
3. self.next\_node = next\_node
5. **def** get\_next(self):
6. **return** self.next\_node
8. **def** set\_next(self, next\_node):
9. self.next\_node = next\_node
11. **def** get\_data(self):
12. **return** self.data
14. **def** set\_data(self, data):
15. self.data = data
17. **class** LinkedList:
18. **def** \_\_init\_\_(self, root=None):
19. self.root = root
20. **if** root != None:
21. self.size = 1
22. **else**:
23. self.size = 0
25. **def** get\_size(self):
26. **return** self.size
28. **def** add(self, data):
29. new\_node = Node(data, self.root)
30. self.root = new\_node
31. self.size += 1
33. **def** remove(self, find\_data):
34. current\_node = self.root
35. prev\_node = None
37. **while** current\_node != None:
38. **if** current\_node.get\_data() == find\_data:
39. **if** prev\_node != None:
40. prev\_node.set\_next(current\_node.get\_next())
41. **else**:
42. self.root = current\_node.get\_next()
43. self.size -= 1
44. **return** True
45. **else**:
46. prev\_node = current\_node
47. current\_node = current\_node.get\_next()
48. **return** False
50. **def** find(self, find\_data):
51. current\_node = self.root
52. **while** current\_node != None:
53. **if** current\_node.get\_data() == find\_data:
54. **return** find\_data
55. **elif** current\_node.get\_next() == None:
56. **return** False
57. **else**:
58. current\_node = current\_node.get\_next()