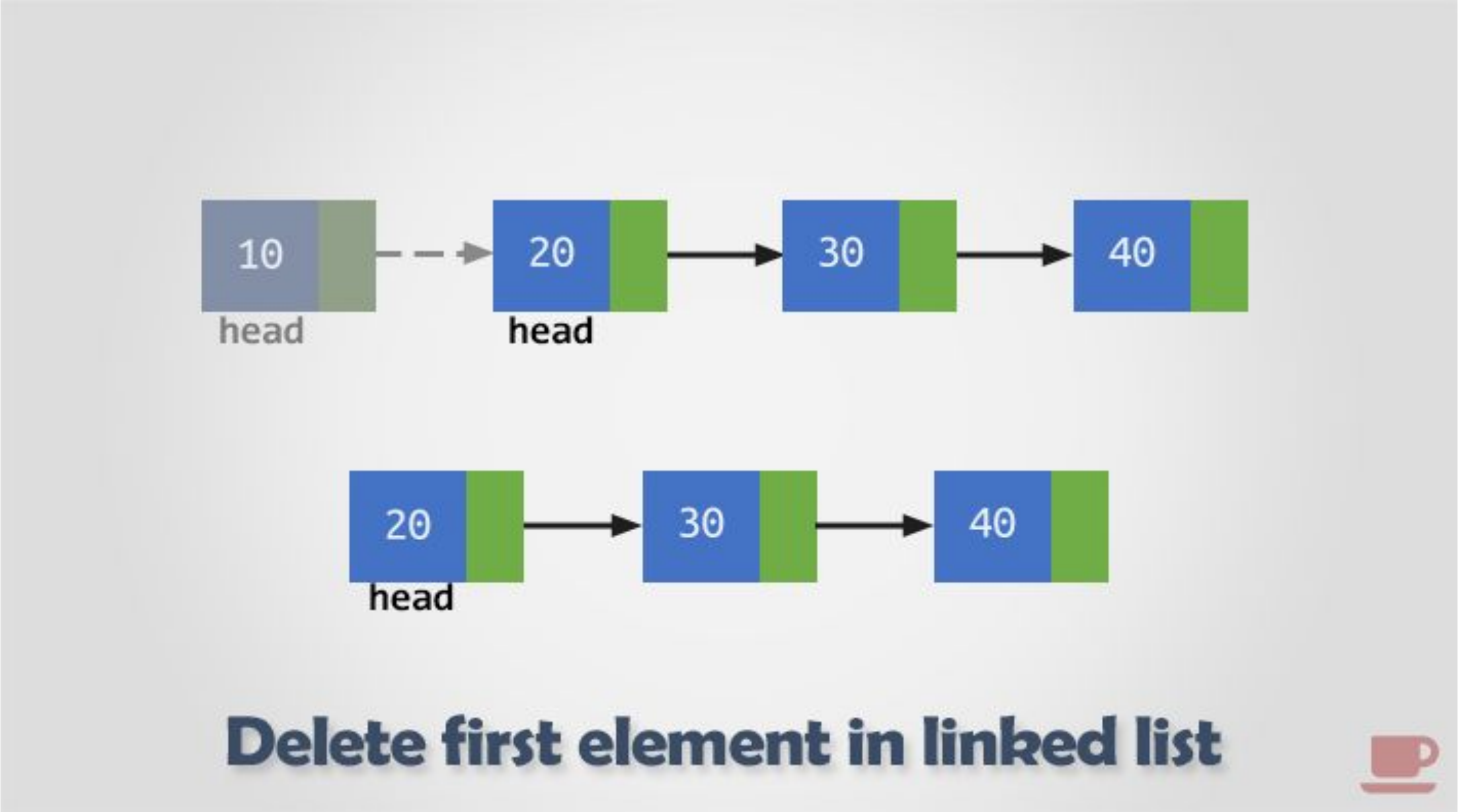
Deletion in a Single Linked List

- There are three possible positions where we can Deleter a new node in a linked list –
  - **Deletion at beginning**
  - **Deletion at end**
  - **Deletion from given position**

- Deleting new node in linked list is a more than one step activity.

Deletion in Single Linked List (from beginning)

- **Deletion from beginning**



Delete first element in linked list

Deletion in Single Linked List (from beginning)

```
# A single node of a singly linked list
class Node:
def __init__(self, data):
  self.data = data
  self.next = None
```

```
# A Linked List class with a single
  head node
class LinkedList:
 def __init__(self):
  self.head = None
```

```
# create method for the linked list
 def create(self, data):
  newNode = Node(data)
  if(self.head):
   current = self.head
   while(current.next):
    current = current.next
   current.next = newNode
  else:
   self.head = newNode
```

Deletion in Single Linked List (from beginning)

```
#Delete first node of the list
 def del_beg(self):
  if(self.head == None):
    print("Underflow-Link List is
  empty")


  else:
   temp = self.head
   self.head = self.head.next
   print("the deleted element is",
  temp.data)
   temp = None
```

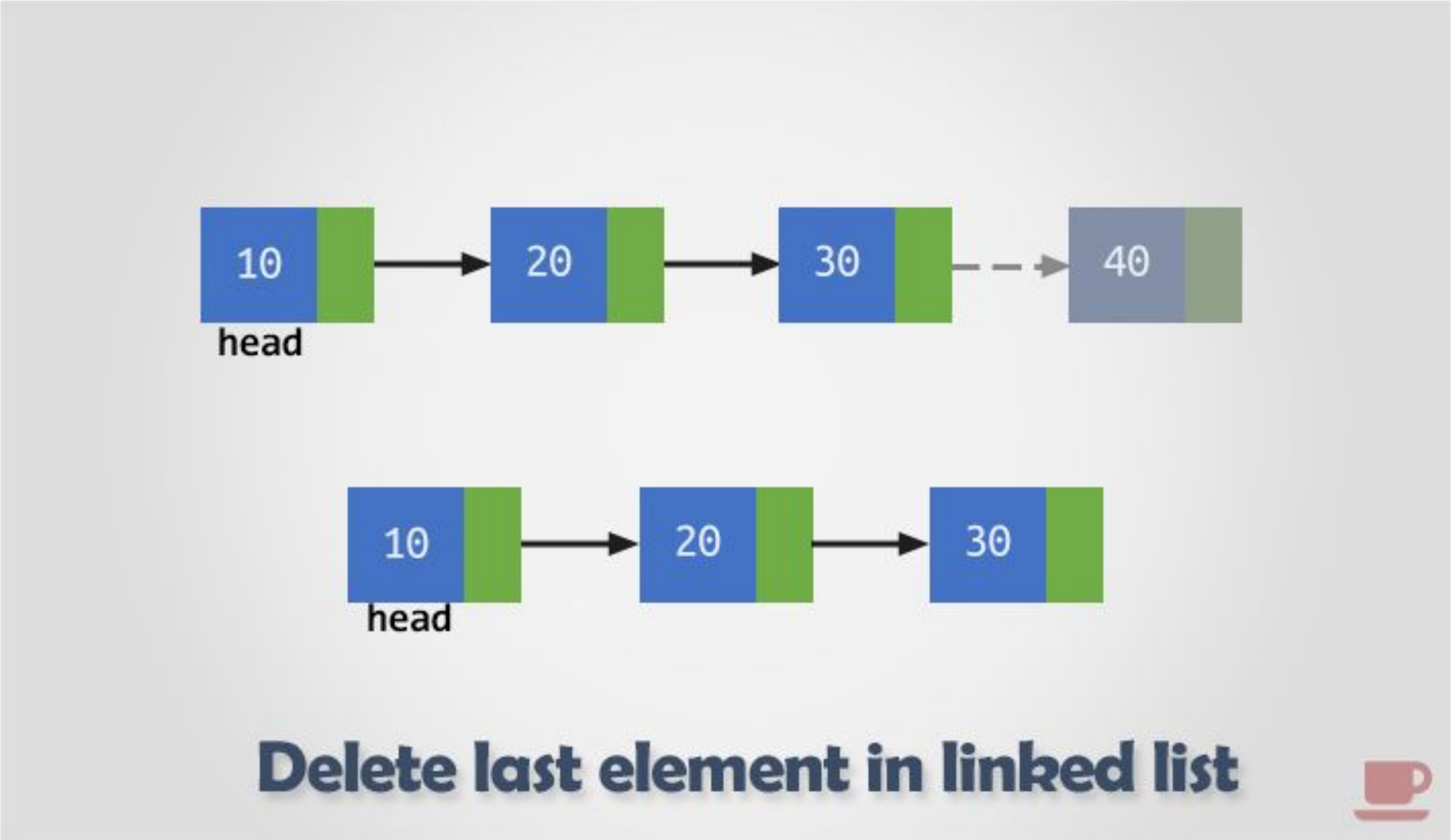
```
# print method for the linked list
 def printLL(self):
  current = self.head
  if(current!=None):
    print("The List Contains:",end="\n")
    while(current):
      print(current.data)
      current = current.next
  else:
    print("List is Empty.")
```

Deletion in Single Linked List (from beginning)

```
# Singly Linked List with deletion and print methods
LL = LinkedList()
LL.create(3)
LL.create(4)
LL.create(5)
LL.printLL()
LL.del_beg()
LL.printLL()
```

Deletion in Single Linked List (from end)

- **Deletion from end**



Deletion in Single Linked List (from end)

```
# A single node of a singly linked list
class Node:
def __init__(self, data):
  self.data = data
  self.next = None
```

```
# A Linked List class with a single head node
class LinkedList:
 def __init__(self):
  self.head = None
```

```
# create method for the linked list
 def create(self, data):
  newNode = Node(data)
  if(self.head):
   current = self.head
   while(current.next):
    current = current.next
   current.next = newNode
  else:
   self.head = newNode
```

Deletion in Single Linked List (from end)

```
#Delete last node of the list
def del_end(self):
 if(self.head == None):
   print("Underflow-Link List is empty")


 else:
  temp = self.head
  while(temp.next!=None):
    prev=temp
    temp=temp.next
  prev.next=None
  print("The deleted element is", temp.data)
  temp = None
```

```
# print method for the linked list
def printLL(self):
 current = self.head
 if(current!=None):
   print("The List Contains:",end="\n")
   while(current):
     print(current.data)
     current = current.next
 else:
   print("List is Empty.")
```

Deletion in Single Linked List (from end)

```
# Singly Linked List with deletion and print methods
LL = LinkedList()
LL.create(3)
LL.create(4)
LL.create(5)
LL.printLL()
LL.del_end()
LL.printLL()
```