Linked List Variations

Made Easy!

SIFAT TANVIR

LECTURER (PART-TIME)

PREVIEW

What is a Linked List?

A connection between nodes, where one node can lead to the Next and so on.

The first node is considered as the head and the last as the tail.

What is a Node?

An object consisting of one or several variables.
For example:

- A Value
- The reference to the next object/node
- The reference to the previous object/node

VARIATIONS

What Causes the Variations?

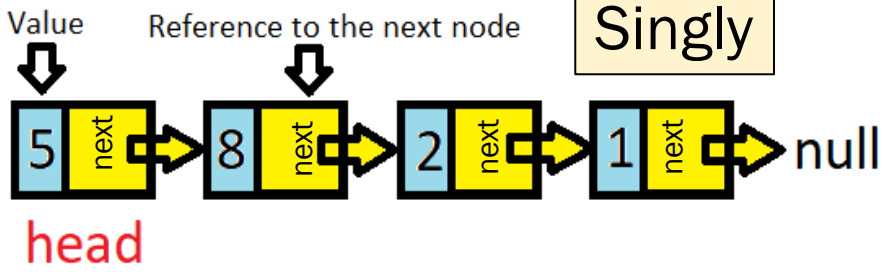
- How each node is connected to each other
- What value the variables of the head holds
- Which ways the linked list can be traversed

Singly
Doubly

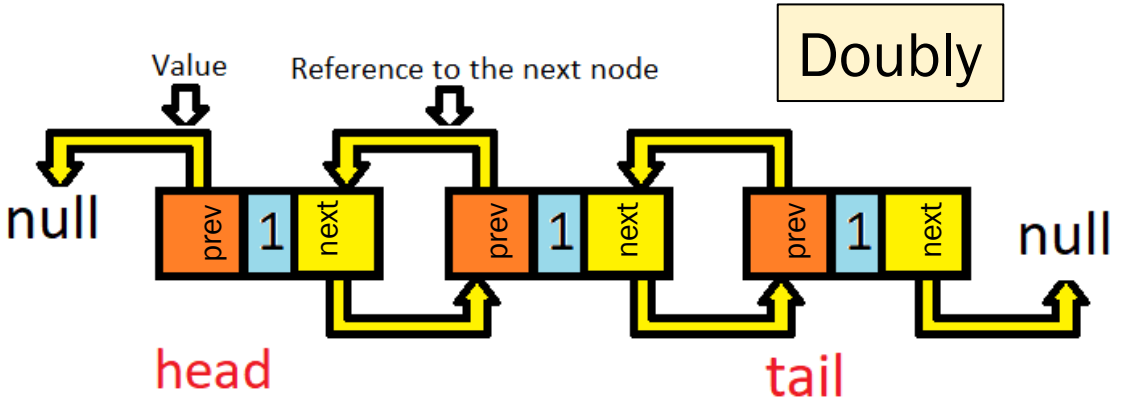
Linear
Circular

Dummy-Headed
Non Dummy-Headed

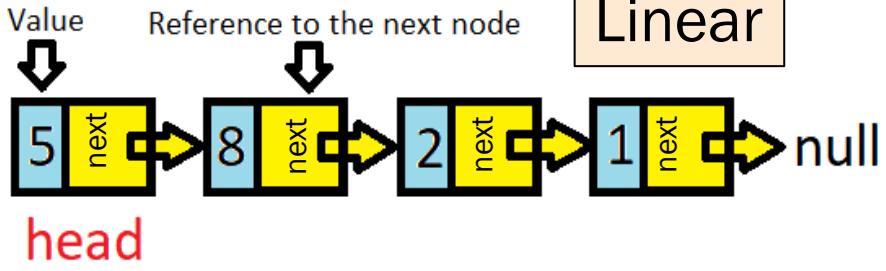
Singly



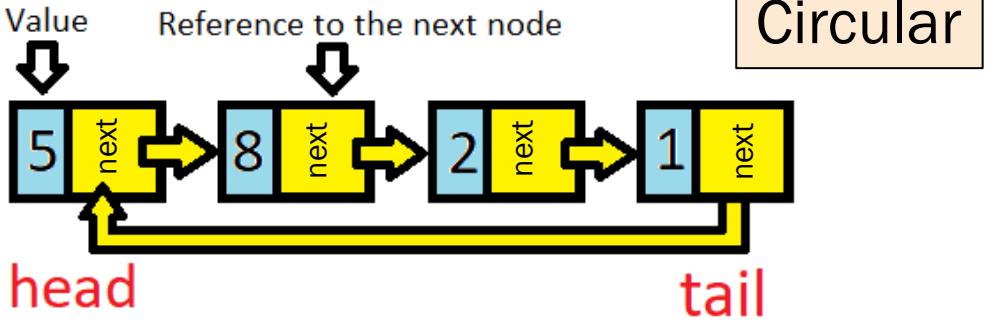
Doubly



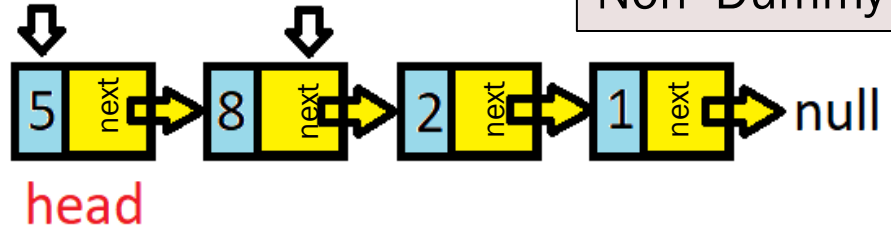
Linear



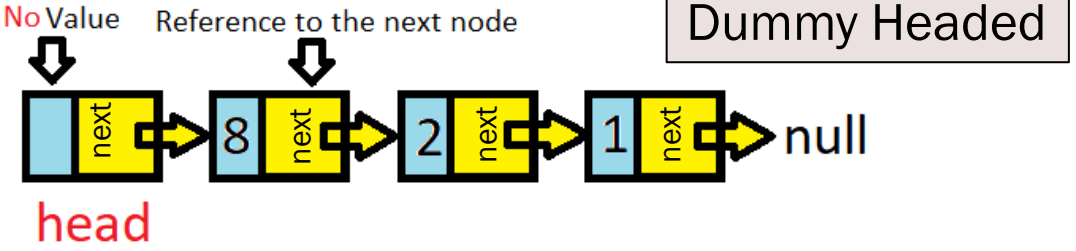
Circular



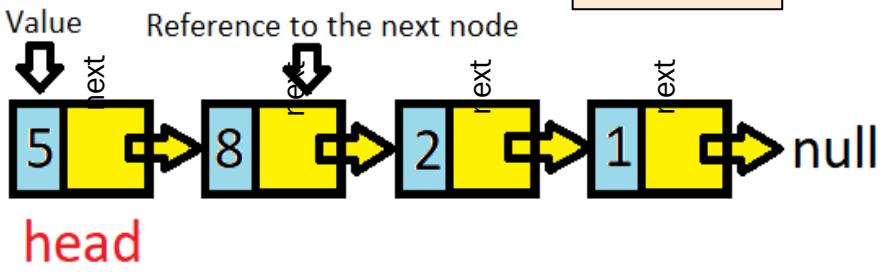
Non- Dummy Headed



Dummy Headed



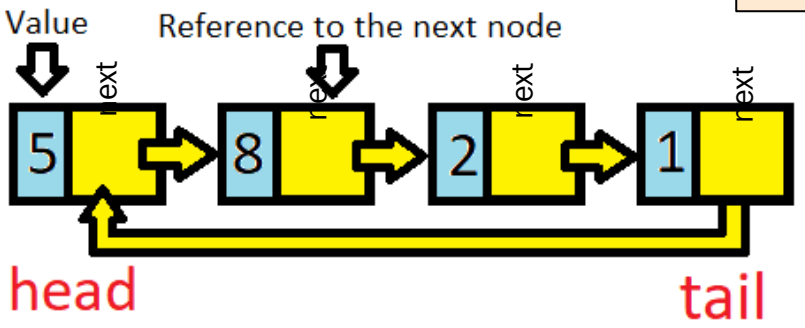
Linear



Printing a Linked List

```
def printList( self, head ):
        while(head != null):
                print(head.value)
                head= head.next
```

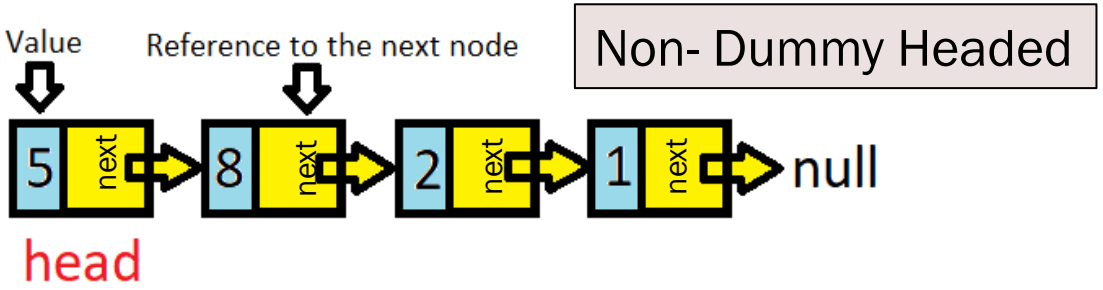
Circular



Printing a Linked List

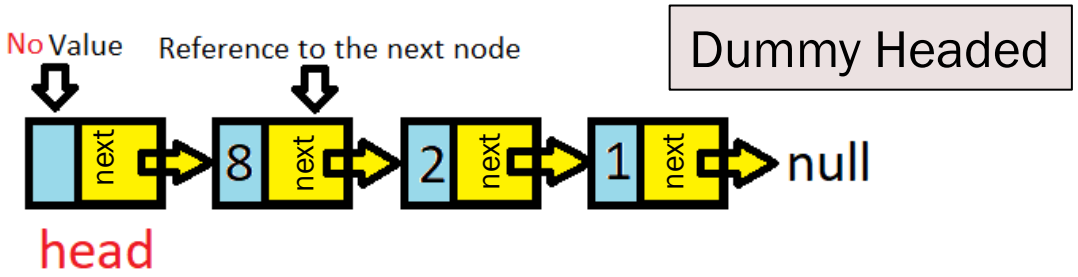
```
def printList(self, head ):
        if(head != null):
                print(head.value)
                n=head.next
                while(n != head):
                        print(n.value)
                        n= n.next
```

Any node can be the head
Convenient for Shifting and Rotating



Printing a Linked List

```
def printList(self, head ):
        while(head != null):
                print(head.value)
                head= head.next
```

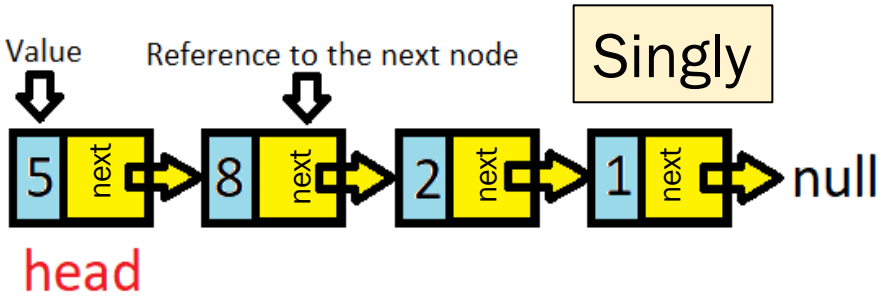


Printing a Linked List

```
def printList(self, head ):
        n= head.next
        while(n != null):
                print(n.value)
                n= n.next
```

Convenient for Adding or Deleting Nodes at the start of the Linked List

Printing of a Non-Dummy Headed and Dummy Headed Linked Lists

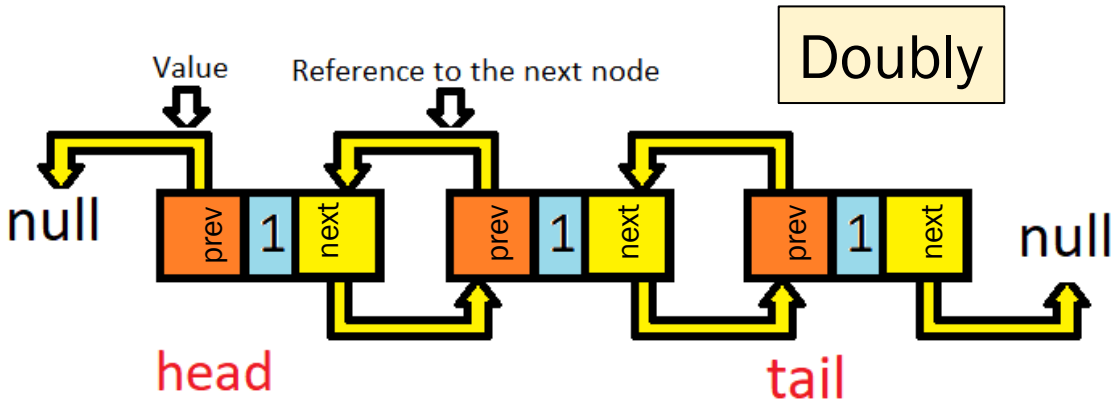


Printing a Linked List

```
def printList(self, head ):
        while(head != null):
                print(head.value)
                head= head.next
```

Printing a Linked List in Reverse Order

Not Possible using Loops!
Can be done using Recursion



Printing a Linked List

```
def printList(self, head ):
        while(head != null):
                print(head.value)
                head= head.next
```

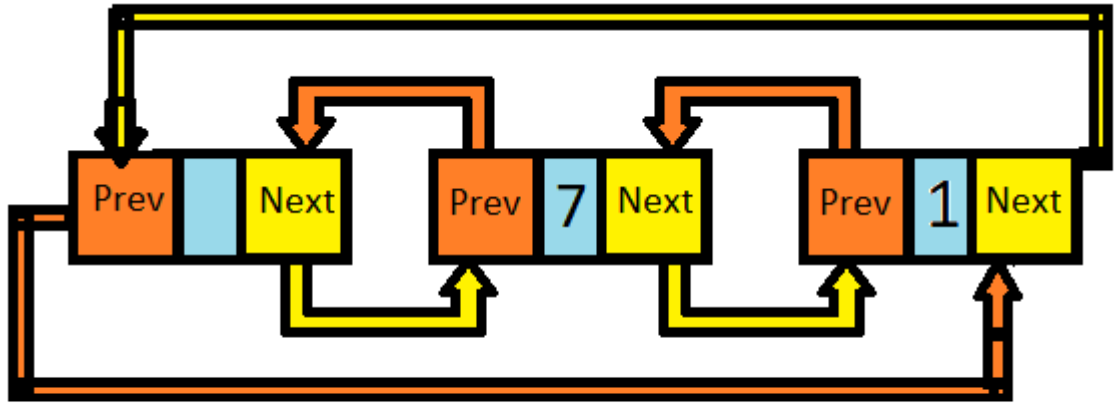
Printing a Linked List in Reverse Order

```
def printList(self, tail ):
        while(tail != null):
                print(tail.value)
                tail= tail.prev
```

Which Linked List Variant would be most convenient in terms of

- Shifting and Rotating
- Adding or Deleting Nodes at the start of the Linked List
- Reverse Printing

Dummy Headed, Doubly And Circular



Combination of Multiple Variants