

Data Science & AI & NIC - Param

Python-For Data Science

Linked List

Lecture No.- 01

By- Pankaj Sharma Sir



Recap of Previous Lecture



Topic

Object-Oriented Programming Part -05



Topics to be Covered



Topic

Linked List Part-01





Topic : Linked List

10, 20, 30, 40, 50



10

316

20

412

40

516

50

625

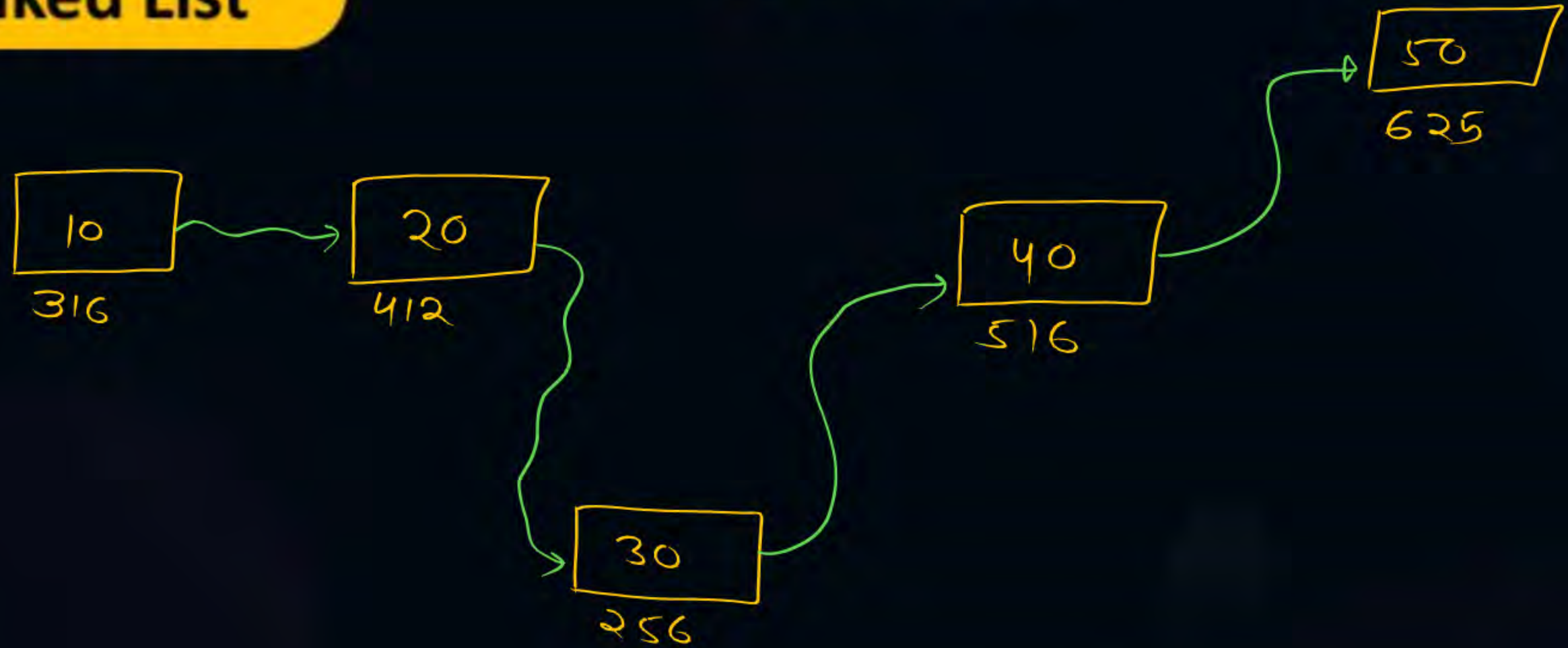
30

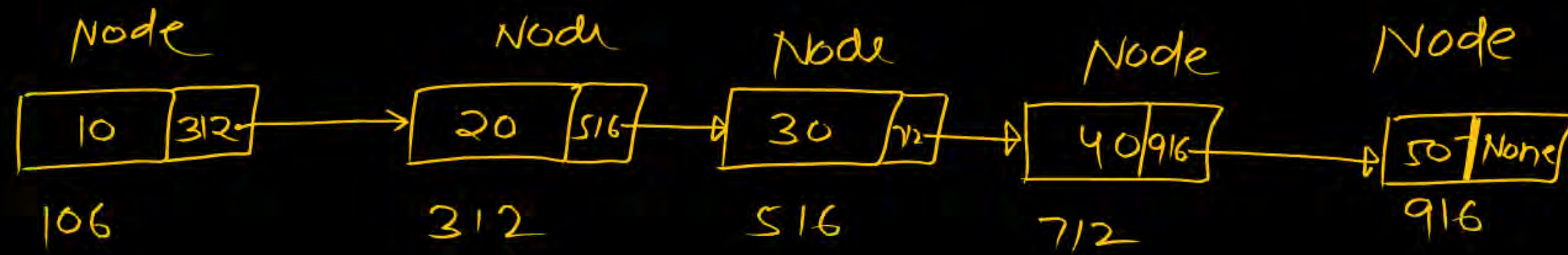
256



Topic : Linked List

10, 20, 30, 40, 50



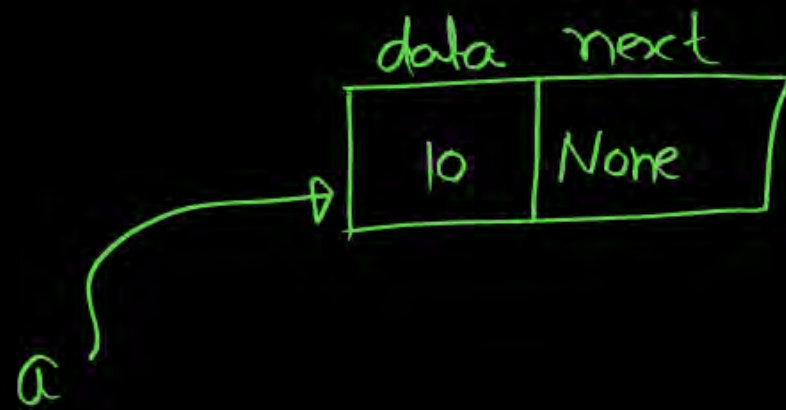


A collection of elements called Nodes, where each node consist of 2 parts

a) data

b) reference of next node in linked list

implement



class Node :

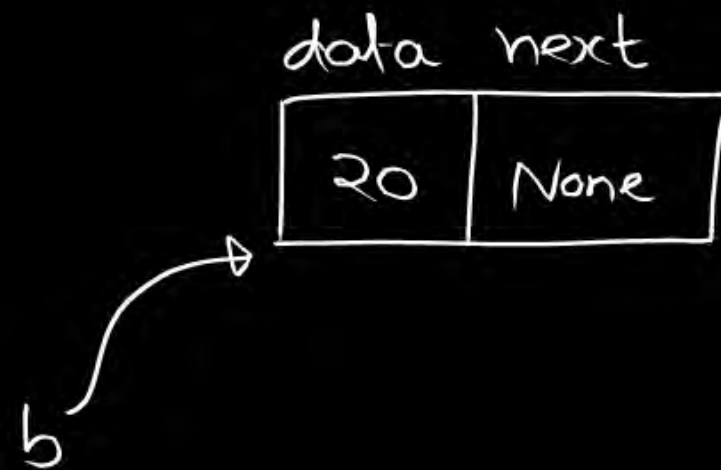
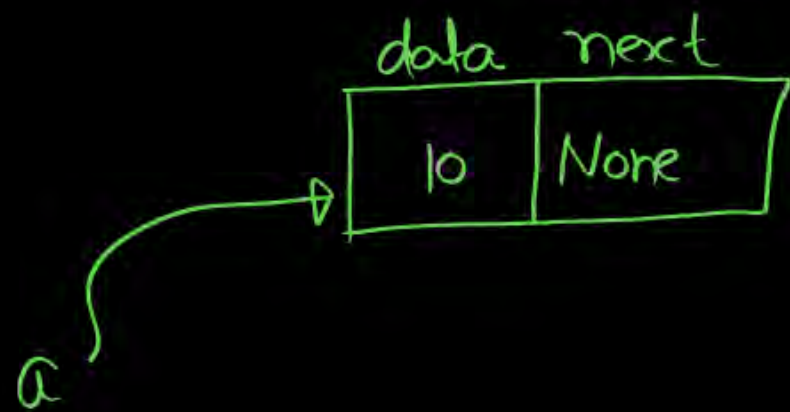
def __init__(self, data):

self.data = data

self.next = None

a = Node(10)

implement



class Node :

def __init__(self, data):

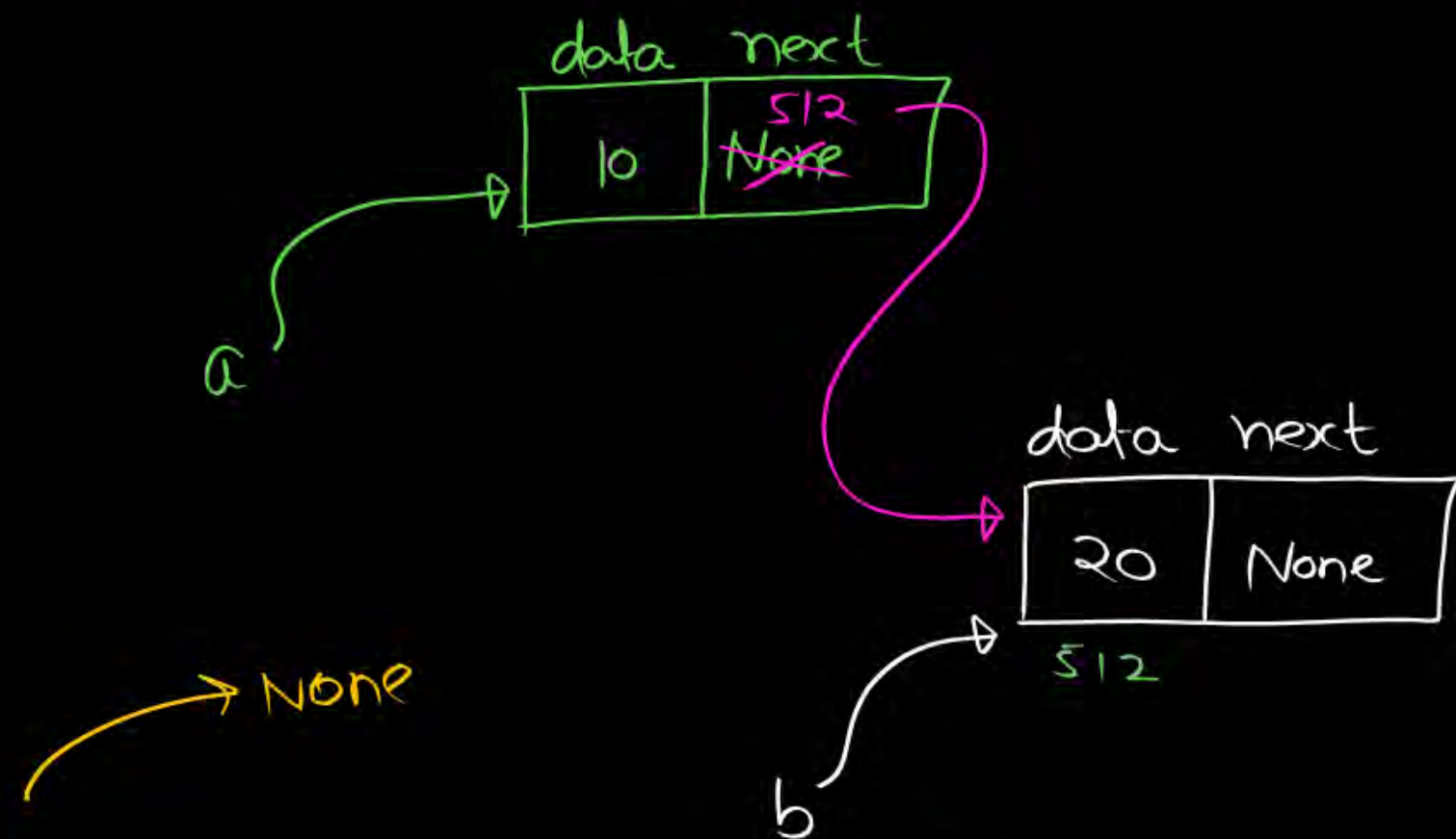
self.data = data

self.next = None

a = Node(10)

b = Node(20)

implement



a.next.next → None

a.next.next.data ⇒ ud ke last Padygi

class Node :

def __init__(self, data):

self.data = data

self.next = None

a = Node(10)

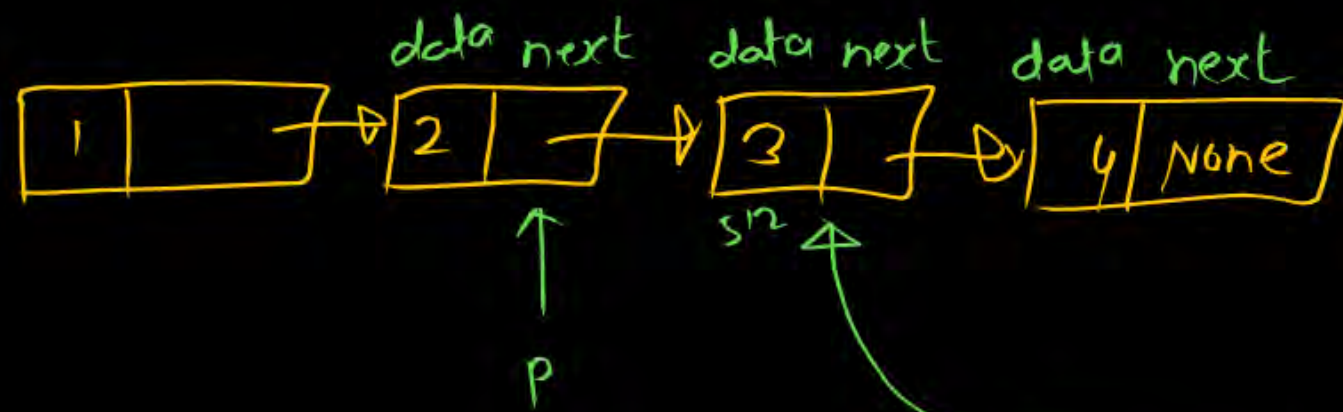
b = Node(20)

a.next = b

L.L \Rightarrow

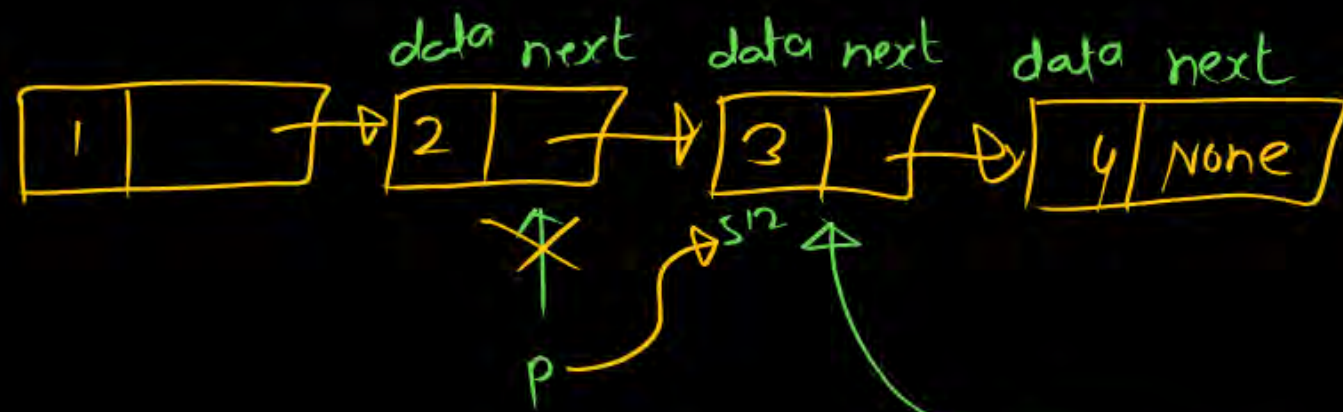
input \Rightarrow "1 2 3 4 5"
String





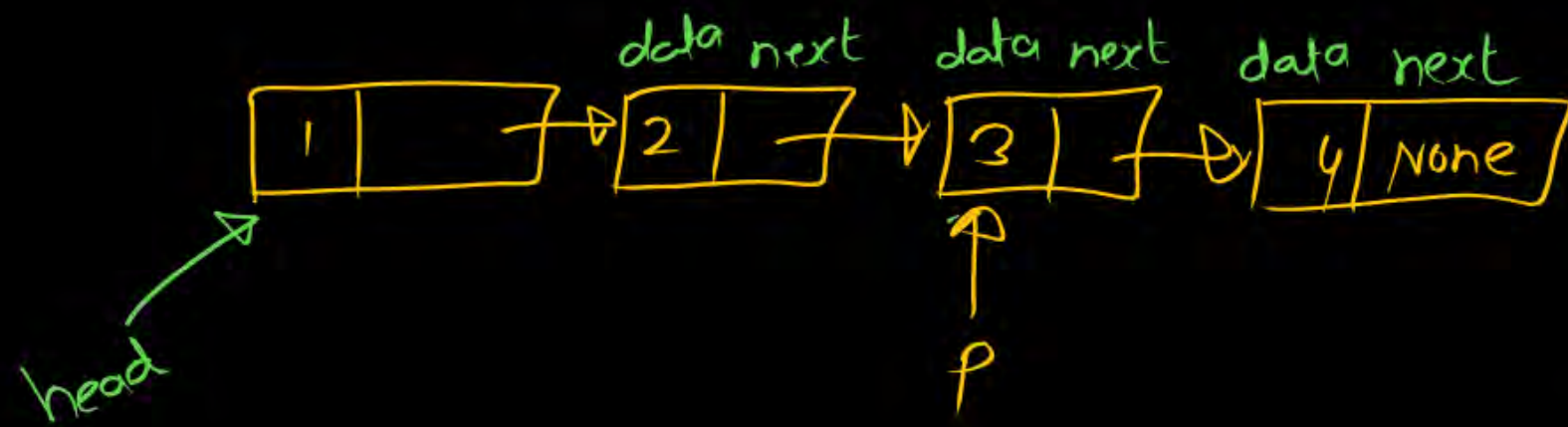
ref. of 3rd node

$P = P.\text{next}$
s12



ref. of 3rd node

$P = P.\text{next}$
S'2



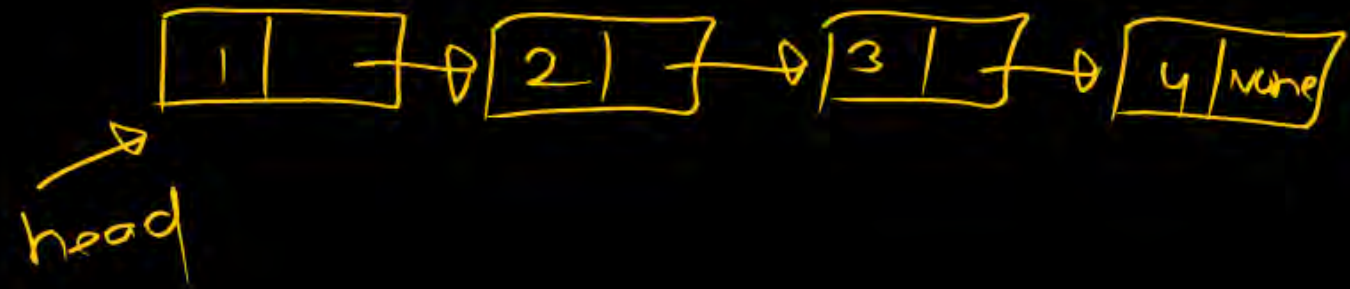
head : reference of 1st node in
linked list

head = None

⇒ LL is empty

"1 2 3 4"

O/P:



return head

"1 2 3 4" \Rightarrow [1, 2, 3, 4]

l = [int(ele) for ele in input().split()]

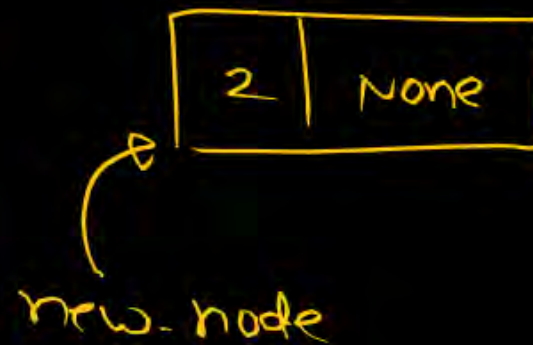
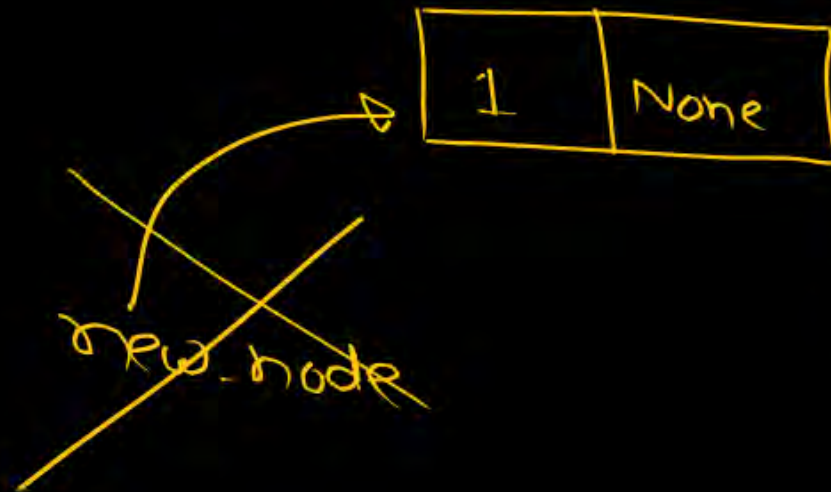
Initially head \rightarrow None

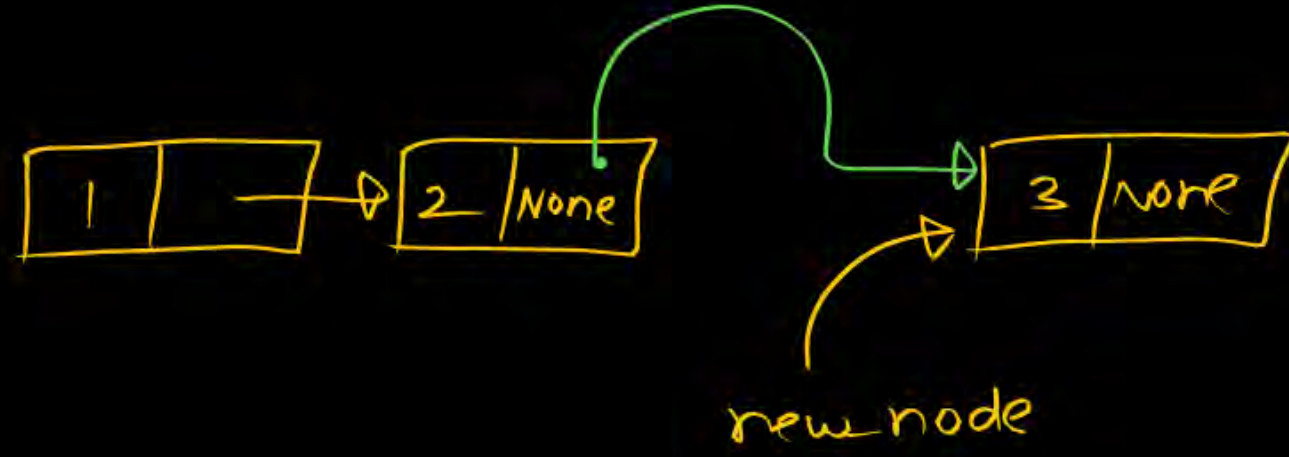
[1, 2, 3, 4]

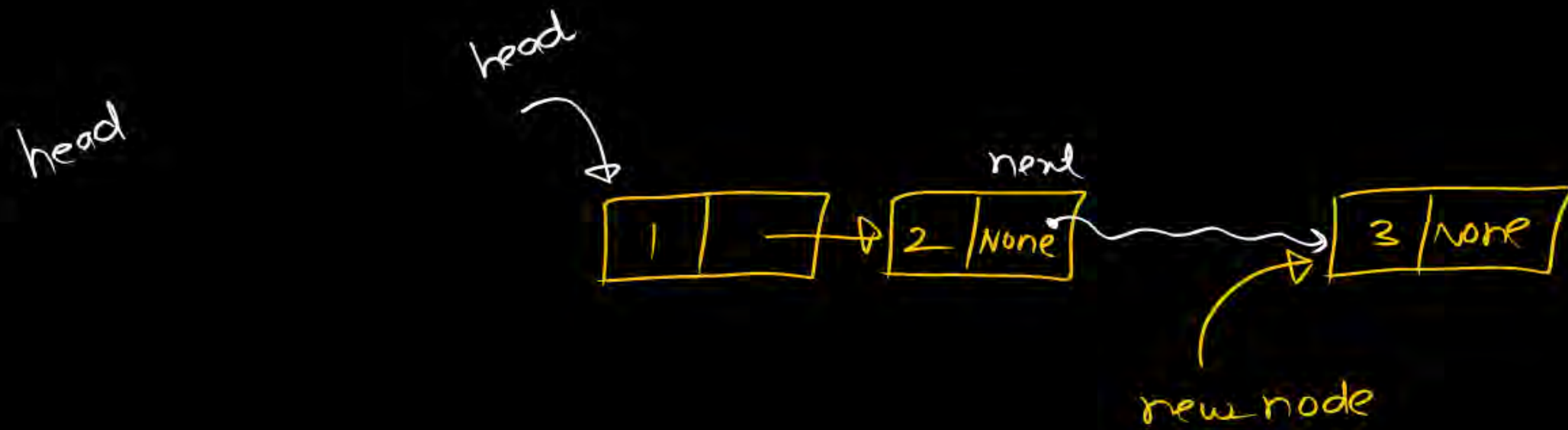
for i in range(len(l)):

This is incomplete code

new_node = Node(l[i])





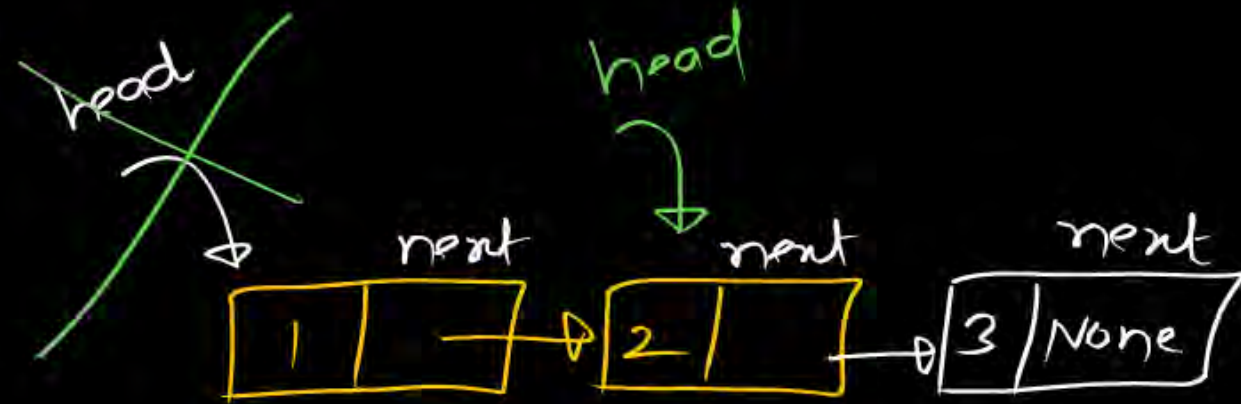


① Traverse till last node

How?

② last_node_ref.next = new_node

head



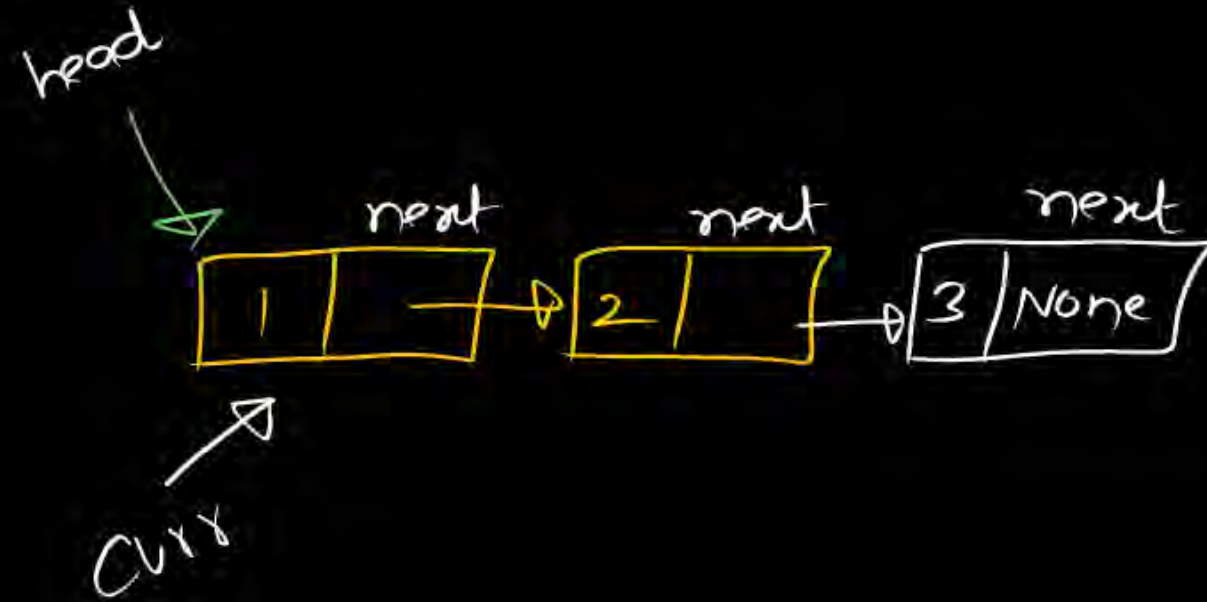
head = head.next

→ worst

सर्ज ऑन २ ऑन ३

① Traverse till last node
How?

head



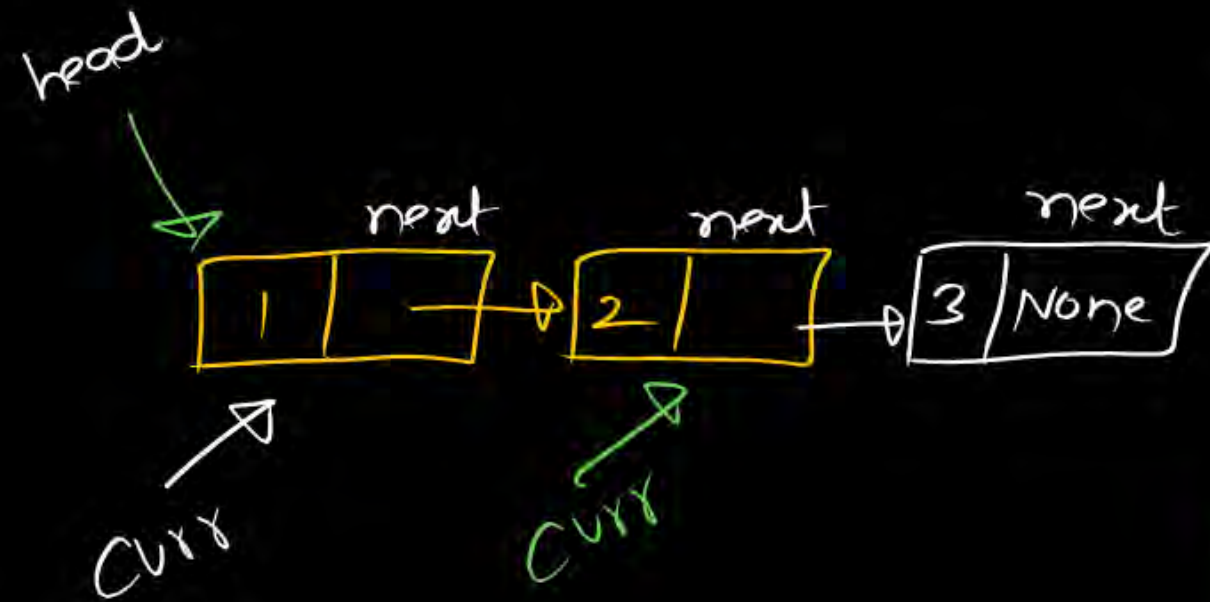
$curr \Rightarrow \text{last node}$



$curr = curr.next$

① Traverse till last node
How?

head



① Traverse till last node
How?

① $curr \Rightarrow \text{last node}$



$curr = curr.next$

② $curr \Rightarrow \text{last node}$



$curr = curr.next$

head

head



While curr.next is not
None :

① Traverse till last node
How?

① curr \Rightarrow last node



curr = curr.next

② curr \Rightarrow last node



curr = curr.next

③ curr \Rightarrow last node

Yes

~~if~~
LL is empty
→ Error

curr = None

None next

curr = head

while curr.next is not None:

curr = curr.next

will work
when atleast
1 node is
in LL

[1, 2, 3, 4]

```
l = [int(ele) for ele in input().split()]  
head = None
```

[1, 2, 3, 4]

```
for i in range(len(l)):
```

```
    new_node = Node(l[i])
```

```
    if head is None:
```

```
        head = new_node
```

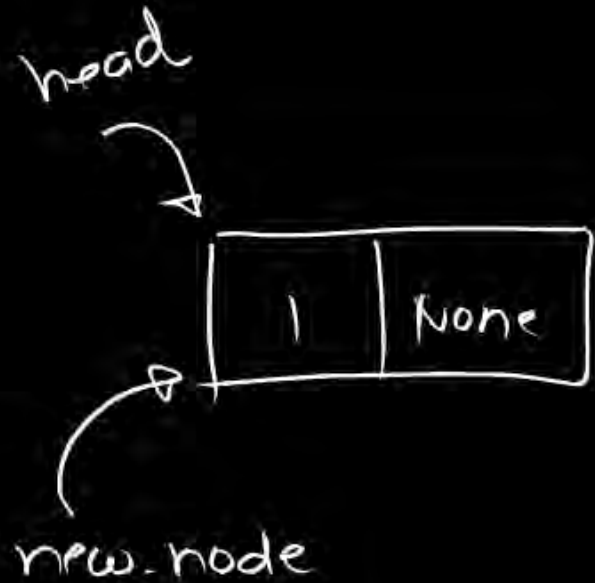
```
    else:
```

```
        curr = head
```

```
        while curr.next is not None:
```

```
            curr = curr.next
```

```
        curr.next = new_node
```



`l = [int(ele) for ele in input().split()]`

`head = None`

`[1, 2, 3, 4]`

`head → None`

`l = [0, 2, 3, 4]`

`i → 0`

`for i in range(len(l)):`

`new_node = Node(l[i])`

`if head is None:`

`head = new_node`

`else:`

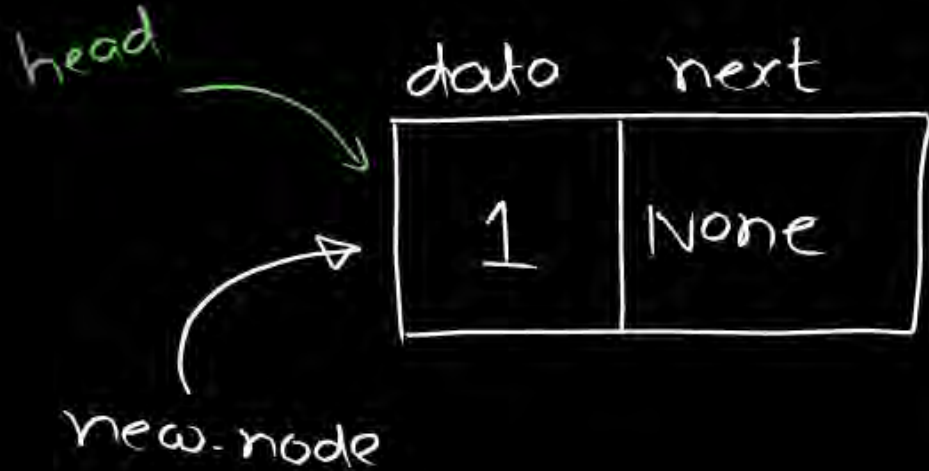
`curr = head`

`while curr.next is not None:`

`curr = curr.next`

`curr.next = new_node`

① `i = 0`



`True`

$l = [\text{int}(ele) \text{ for } ele \text{ in } \text{input().split()}]$

head = None

[1, 2, 3, 4]

head \rightarrow None

$l = [0, 2, 3, 4]$

$i \rightarrow 0$

for i in $\text{range}(\text{len}(l))$:

① $i = 0$

head

data	next
1	None

$\text{new_node} = \text{Node}(l[i])$
if head is None :
head = new_node
else :

false

curr = head

while curr.next is not None :

curr = curr.next

curr.next = new_node

② $i = 1$

data next

2	Node
---	------

1256

new_node

$l = [\text{int}(ele) \text{ for } ele \text{ in } \text{input().split()}]$

head = None

[1, 2, 3, 4]

head \rightarrow None

$l = [0, 2, 3, 4]$

$i \rightarrow 0$

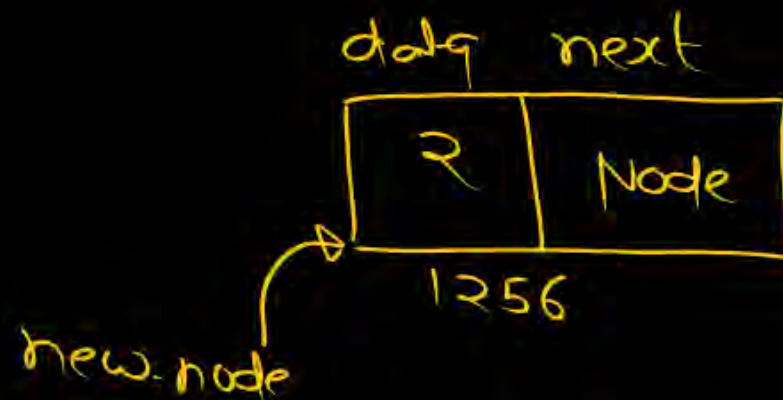
for i in $\text{range}(\text{len}(l))$:

① $i = 0$



new_node = Node($l[i]$)
if head is None:
head = new_node
else:

② $i = 1$



curr = head
while curr.next is not None:
curr = curr.next
curr.next = new_node ✓

$l = [\text{int}(ele) \text{ for } ele \text{ in } \text{input().split()}]$

head = None

[1, 2, 3, 4]

head \rightarrow None

$l = [0, 2, 3, 4]$

$i \rightarrow 0$

for i in $\text{range}(\text{len}(l))$:

new_node = Node($l[i]$)

if head is None:

head = new_node

else:

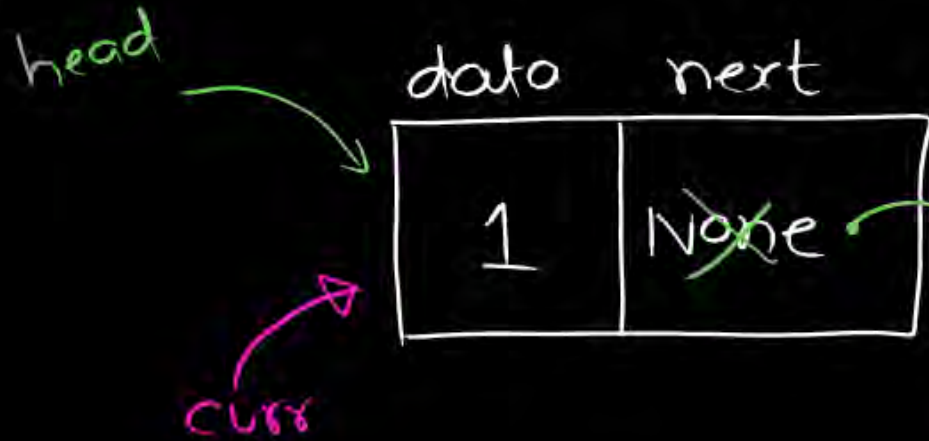
curr = head

while curr.next is not None:

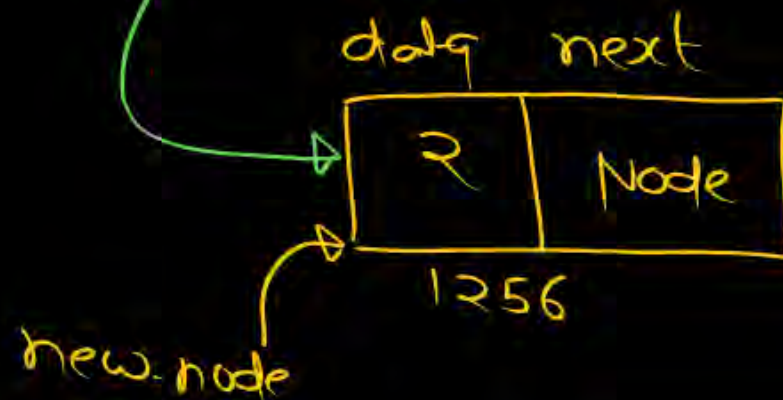
curr = curr.next

curr.next = new_node ✓

① $i=0$



② $i=1$



$l = [\text{int}(ele) \text{ for } ele \text{ in } \text{input().split()}]$

head = None

[1, 2, 3, 4]

head \rightarrow None

$l = [1, 2, 3, 4]$

$i \rightarrow 0$

for i in $\text{range}(\text{len}(l))$:

new_node = Node($l[i]$)

if head is None:

head = new_node

else:

curr = head

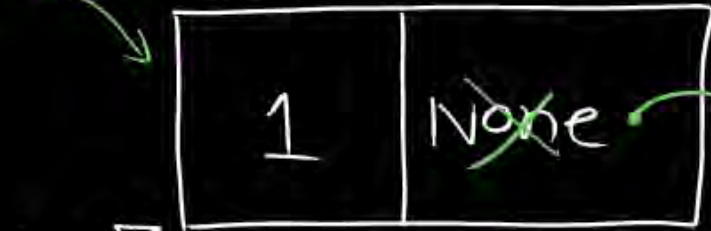
while curr.next is not None:

curr = curr.next

curr.next = new_node

① $i=0$

head



~~curr~~

curr

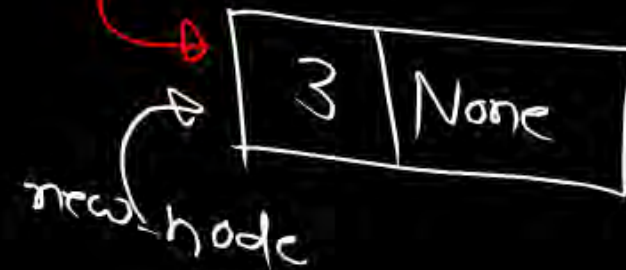
data next



1256

② $i=1$

③ $i=2$



```
In [5]: class Node:
        def __init__(self,data):
            self.data=data
            self.next=None
a=Node(10)
b=Node(20)
a.next=b
print(a)
print(b)
print(a.next)#same as b
print(a.next.data) #b node ka data ==>20
print(a.next.next)
print(a.next.next.data)
```

```
<__main__.Node object at 0x000002BF5B4CA710>
<__main__.Node object at 0x000002BF5B533590>
<__main__.Node object at 0x000002BF5B533590>
20
None
```

```
-----
AttributeError                                Traceback (most recent call last)
Cell In[5], line 13
     11 print(a.next.data) #b node ka data ==>20
     12 print(a.next.next)
--> 13 print(a.next.next.data)

AttributeError: 'NoneType' object has no attribute 'data'
```

```
In [7]: l=[int(ele) for ele in input().split()]
        head=None
        for i in range(len(l)):
            new_node=Node(l[i])
            if head is None:
                head=new_node
            else:
                curr=head
                while curr.next is not None:
                    curr=curr.next
                curr.next=new_node
```

```
1 2 3 4 5
```

```
In [8]: def takinginput():
        l=[int(ele) for ele in input().split()]
        head=None
        for i in range(len(l)):
            new_node=Node(l[i])
            if head is None:
                head=new_node
            else:
                curr=head
                while curr.next is not None:
                    curr=curr.next
                curr.next=new_node
        return head
```

```
In [13]: r=takinginput()
```

1 2 3 4

In [14]: head

Out[14]: <__main__.Node at 0x2bf5c18e690>

In []:

In [15]: r

Out[15]: <__main__.Node at 0x2bf5c1f0050>

In []:

THANK - YOU