Data Strcture LAB 5

PREPARED BY:
AYOOB ABDULMUNEM
Ahmed Eskander Mezher



Lecture Outline: -

- 1- insert a node at the beginning of the liked list
- 2- compute the length of the linked list



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```

head = None



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```

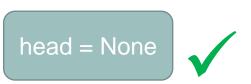
head = None

nod

4



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



head

nod

4



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```

head

nod

5

None



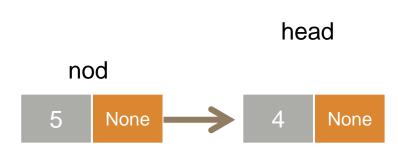
```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



5



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def insert_begin(self,data):
     nod=Node(data)
     if self.head==None:
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```





```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
                                                                head
  def insert_begin(self,data):
                                                                 nod
     nod=Node(data)
     if self.head==None:
                                        6
                                                               5
                                              None
                                                                                            None
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
                                                                head Y
  def insert_begin(self,data):
                                          nod
     nod=Node(data)
    if self.head==None:
                                        6
                                              None ____
                                                                                           None
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
                                                                head
  def insert_begin(self,data):
                                           nod
     nod=Node(data)
     if self.head==None:
                                         6
                                                                                            None
       self.head=nod
       return
     nod.next=self.head
     self.head=nod
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
```



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def length(self):
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```



```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def length(self):
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 0
     self.data = data
     self.next = next
class linklist:
                                                p
  def __init__(self,head=None):
     self.head=head
                                               head
  def length(self):
                                             6
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 0
     self.data = data
     self.next = next
class linklist:
                                                p
  def __init__(self,head=None):
     self.head=head
                                              head
  def length(self):
                                             6
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 0
     self.data = data
     self.next = next
class linklist:
                                                                     p
  def __init__(self,head=None):
     self.head=head
                                               head
  def length(self):
     p=self.head
                                             6
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```



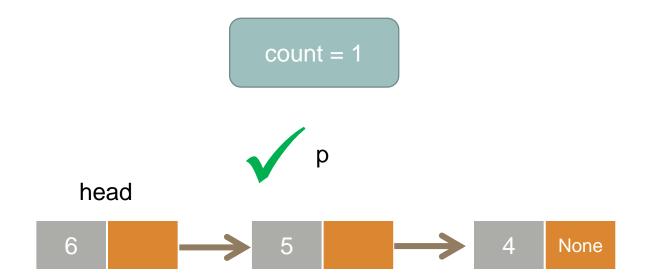


```
class Node:
  def __init__(self,data,next=None):
                                                                count = 1
     self.data = data
     self.next = next
class linklist:
                                                                     p
  def __init__(self,head=None):
     self.head=head
                                               head
  def length(self):
     p=self.head
                                             6
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```





```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def length(self):
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 1
     self.data = data
     self.next = next
class linklist:
                                                                     p
  def __init__(self,head=None):
     self.head=head
                                               head
  def length(self):
     p=self.head
                                             6
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```



```
class Node:
  def __init__(self,data,next=None):
                                                                count = 1
     self.data = data
     self.next = next
class linklist:
                                                                                          p
  def __init__(self,head=None):
     self.head=head
                                               head
  def length(self):
                                             6
                                                                                            None
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```



```
class Node:
  def __init__(self,data,next=None):
                                                                count = 2
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
                                              head
  def length(self):
     p=self.head
                                             6
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```







```
class Node:
  def __init__(self,data,next=None):
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
  def length(self):
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```

count = 2

p = None

head





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 2
     self.data = data
     self.next = next
class linklist:
  def __init__(self,head=None):
     self.head=head
                                              head
  def length(self):
                                             6
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```

p = None





```
class Node:
  def __init__(self,data,next=None):
                                                                count = 3
     self.data = data
     self.next = next
class linklist:
                                                                                             p = None
  def __init__(self,head=None):
     self.head=head
                                              head
  def length(self):
                                                                                            None
                                             6
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```

```
class Node:
  def __init__(self,data,next=None):
                                                                count = 3
     self.data = data
     self.next = next
class linklist:
                                                                                             p = None
  def __init__(self,head=None):
     self.head=head
                                              head
  def length(self):
                                                                                            None
                                             6
     p=self.head
     count=0
     while p!= None:
       p=p.next
       count +=1
     return count
LL=linklist()
LL.insert_begin(4)
LL.insert_begin(5)
LL.insert_begin(6)
print(LL.length())
```

Thank you

