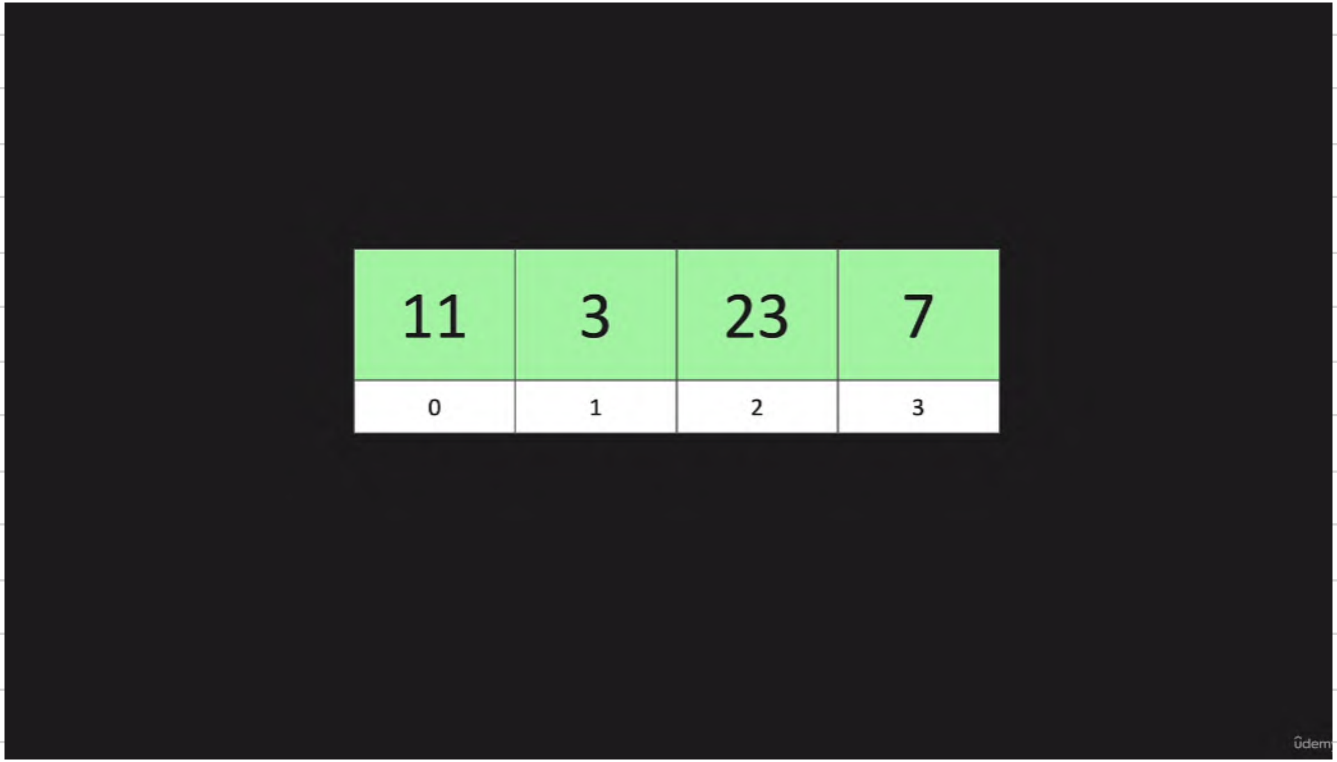
Struktur Data

Meet 04

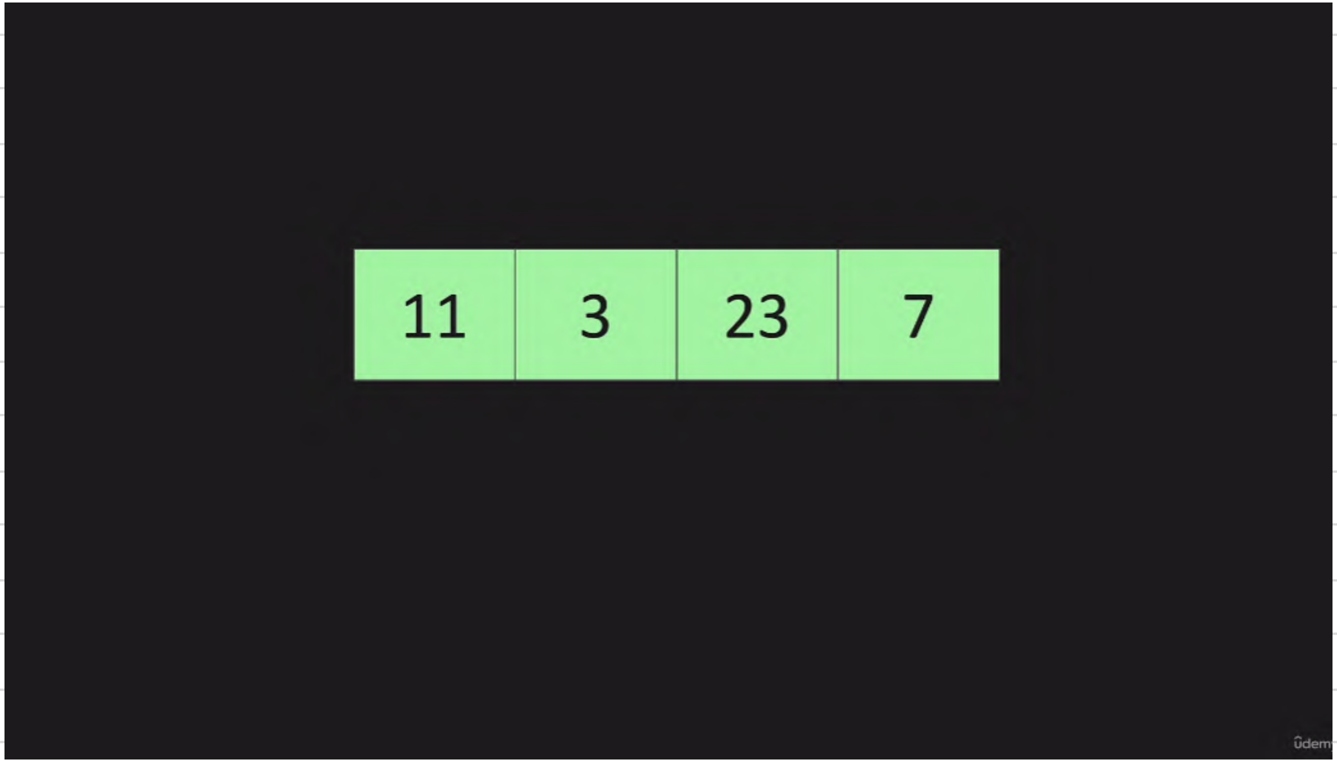
01

Linked List Intro

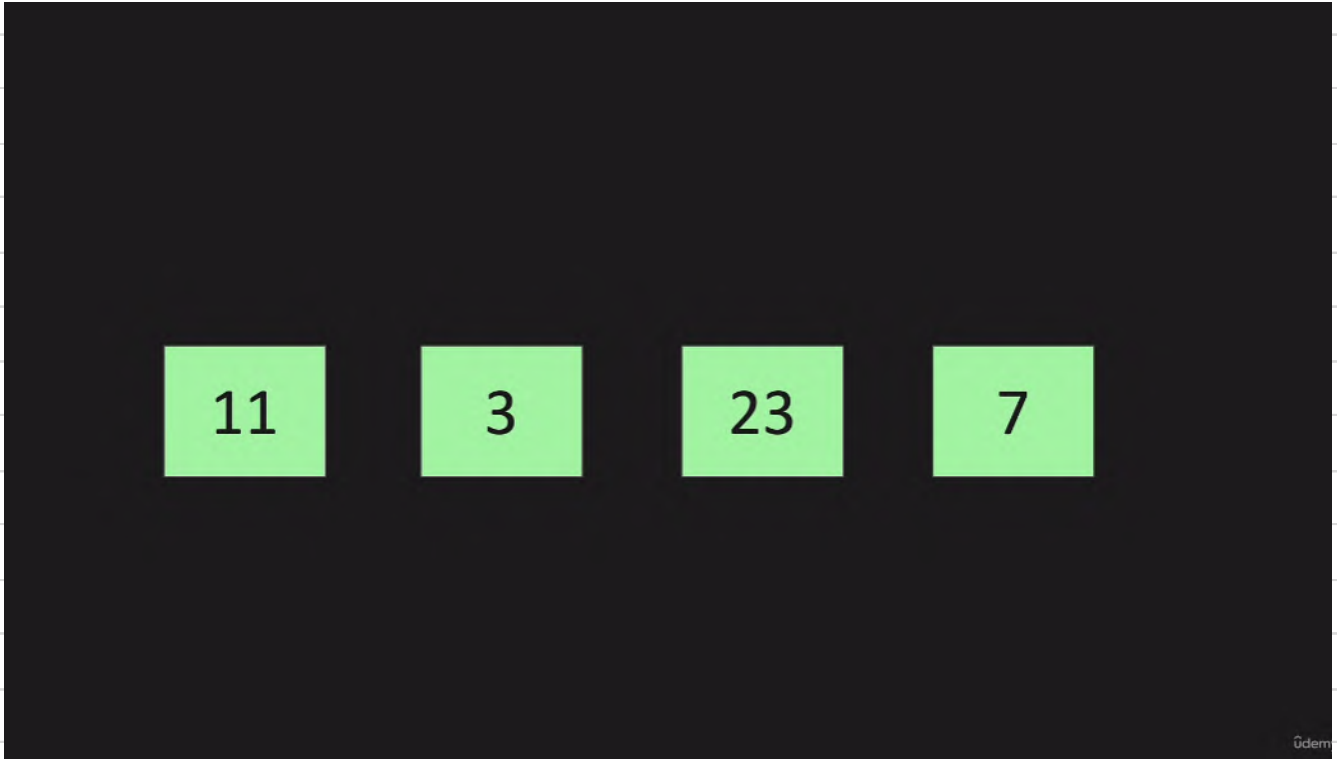
List VS Linked List



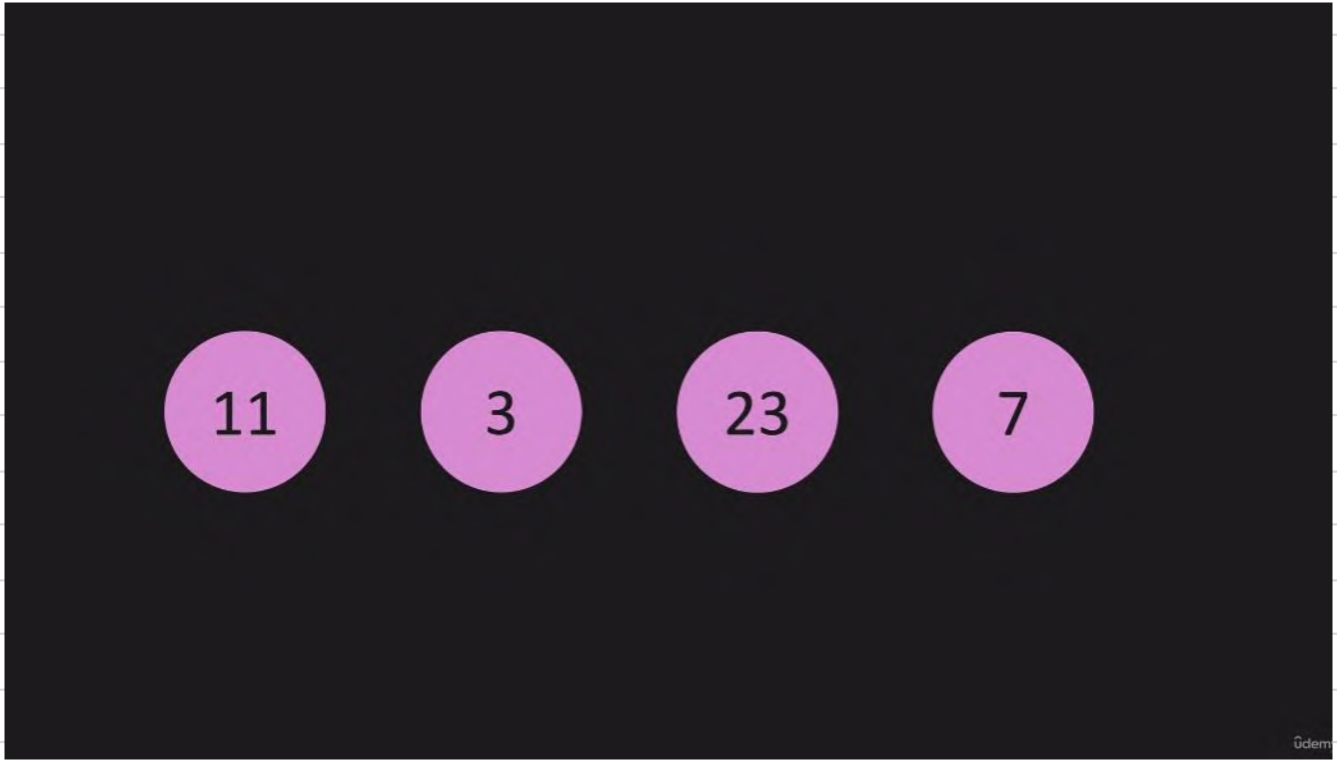
List VS Linked List



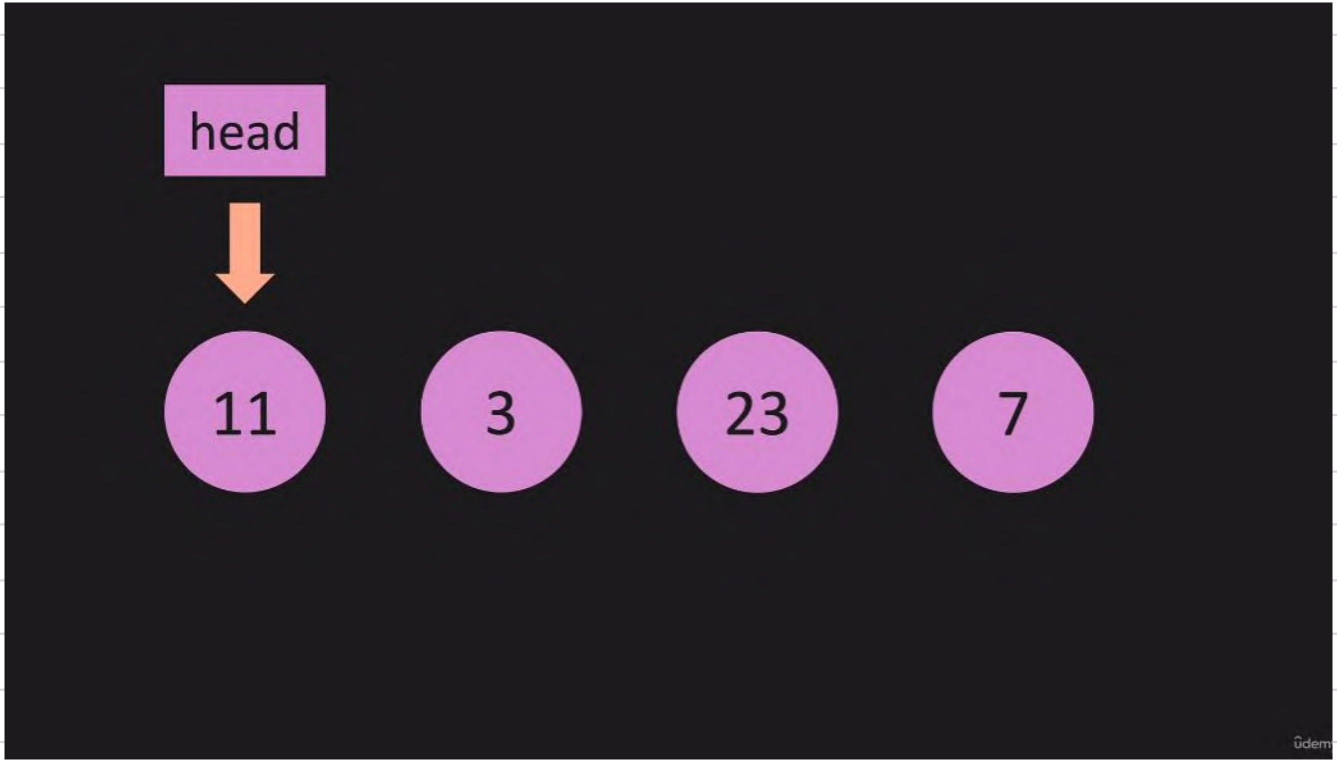
List VS Linked List



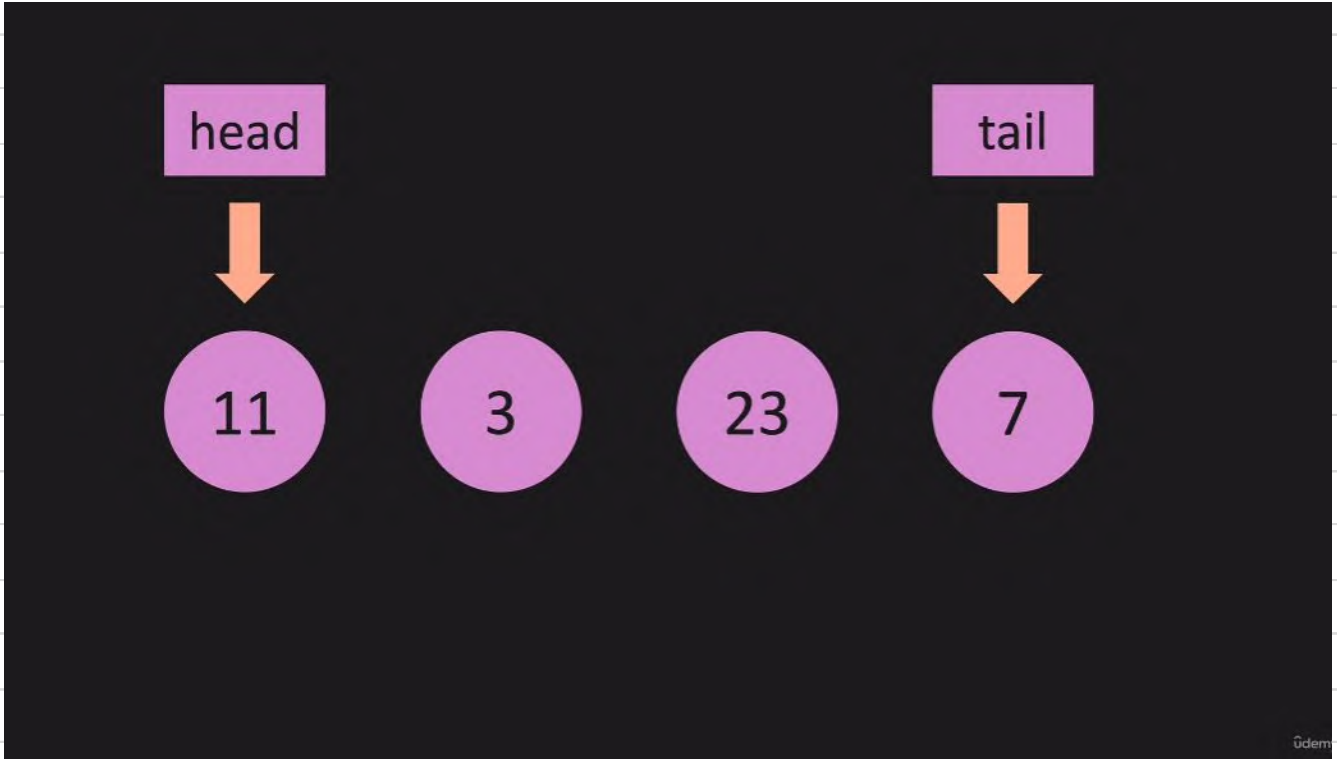
List VS Linked List



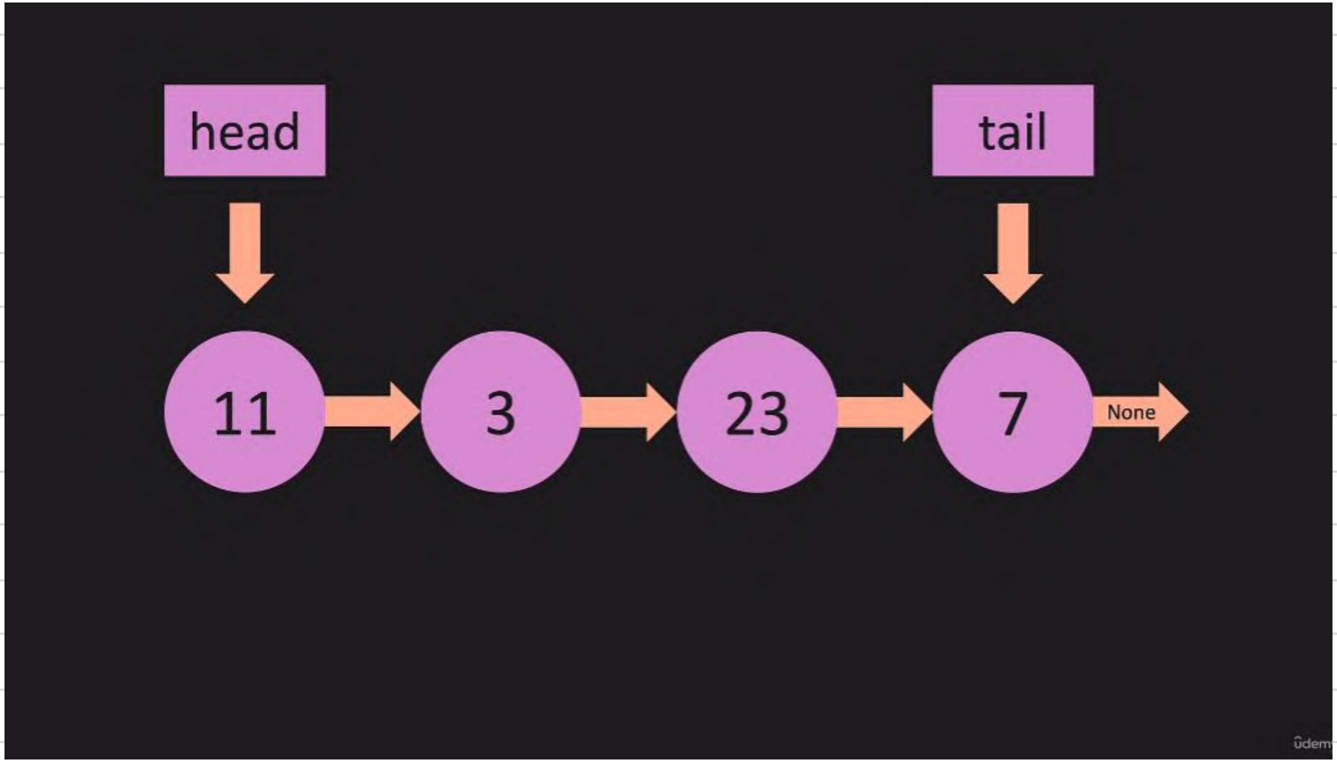
List VS Linked List



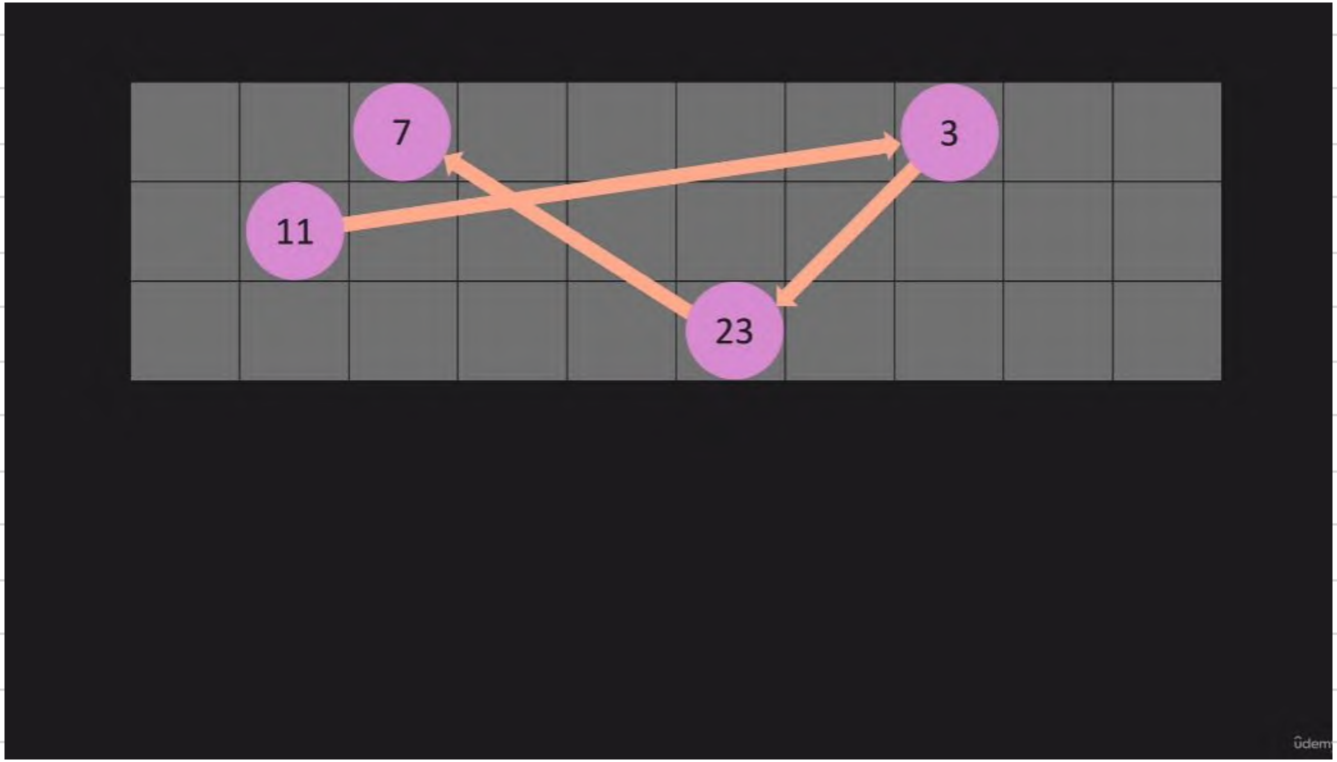
List VS Linked List



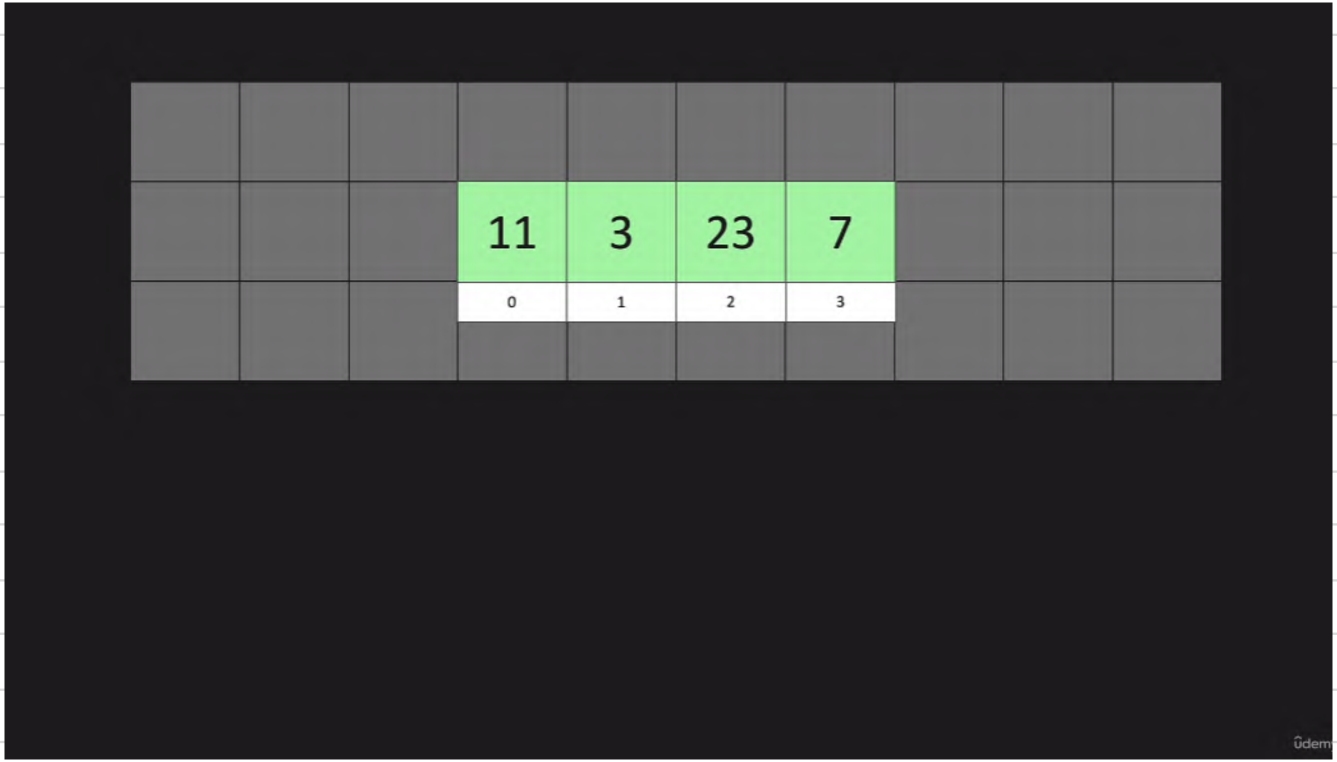
List VS Linked List



List VS Linked List



List VS Linked List

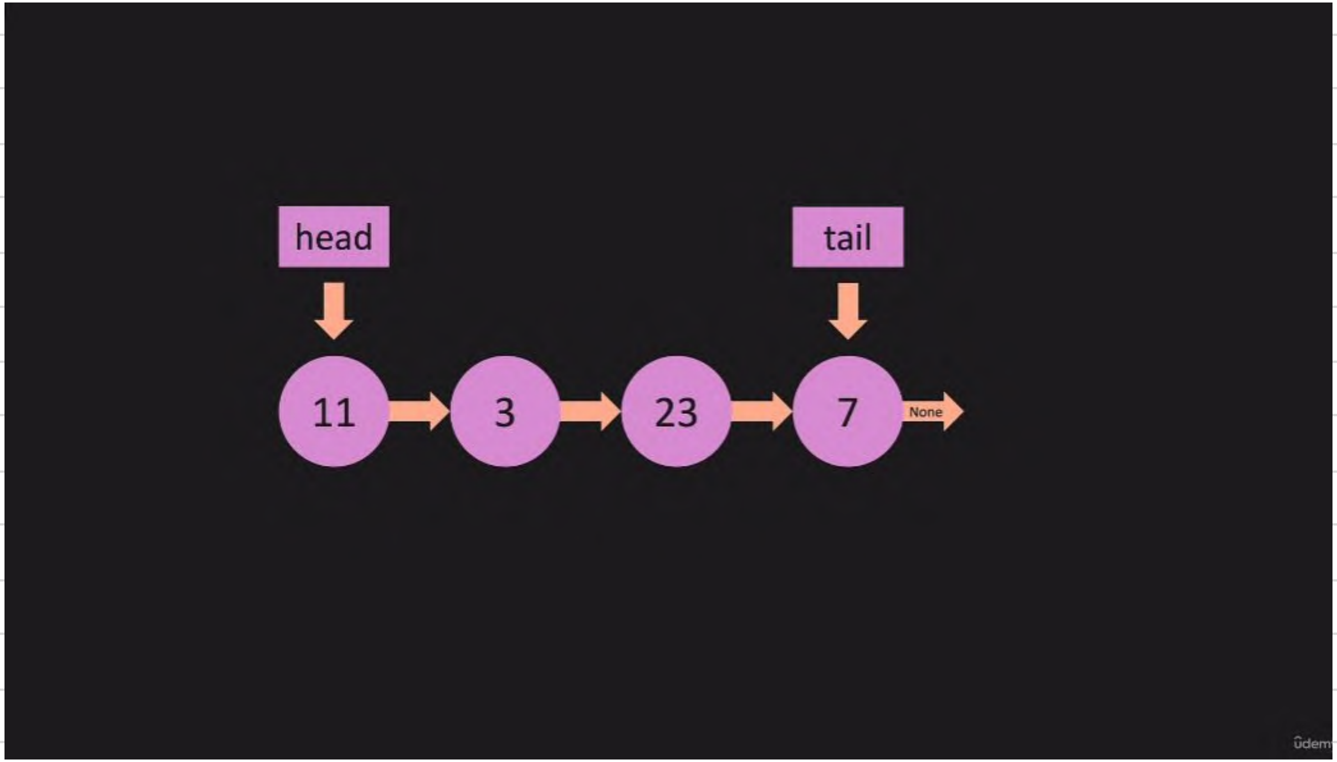


02

Linked List Big O

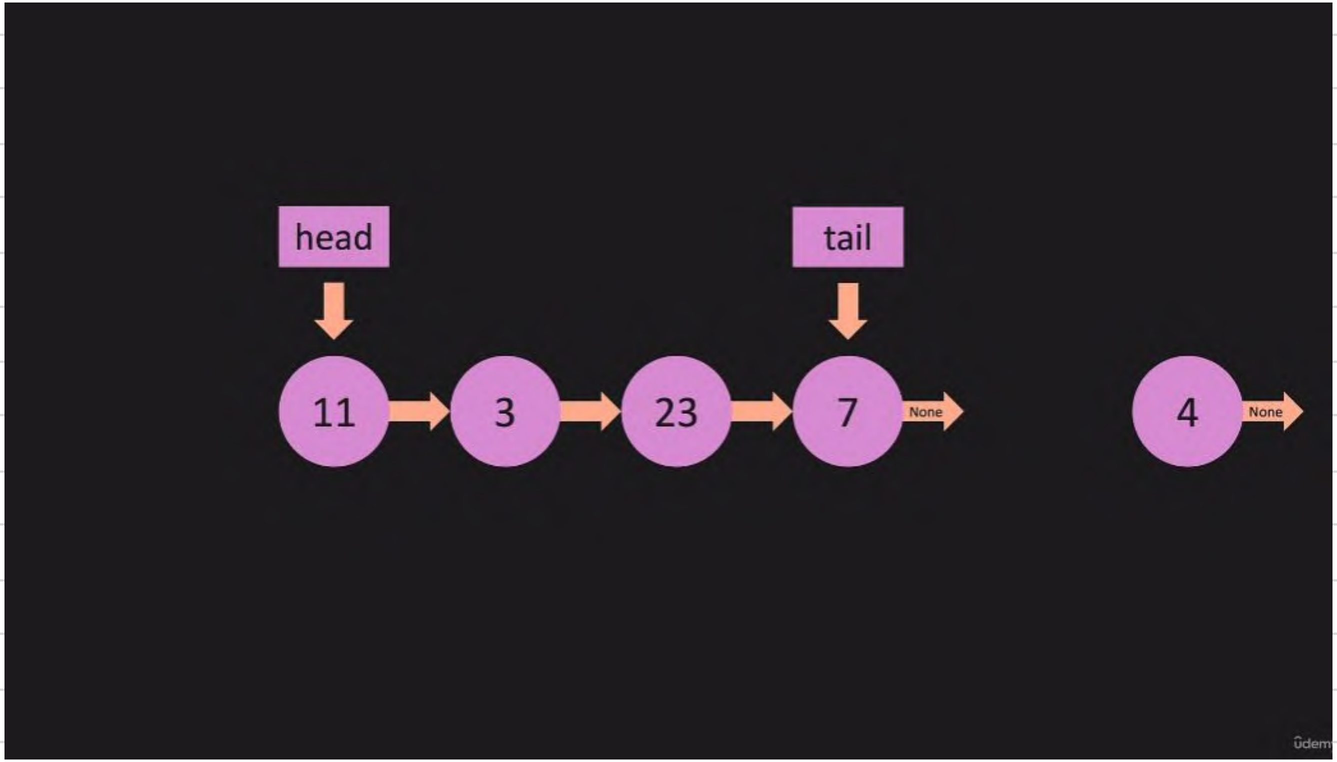
Diketahui suatu Linked List...

Add & Remove Node di Belakang



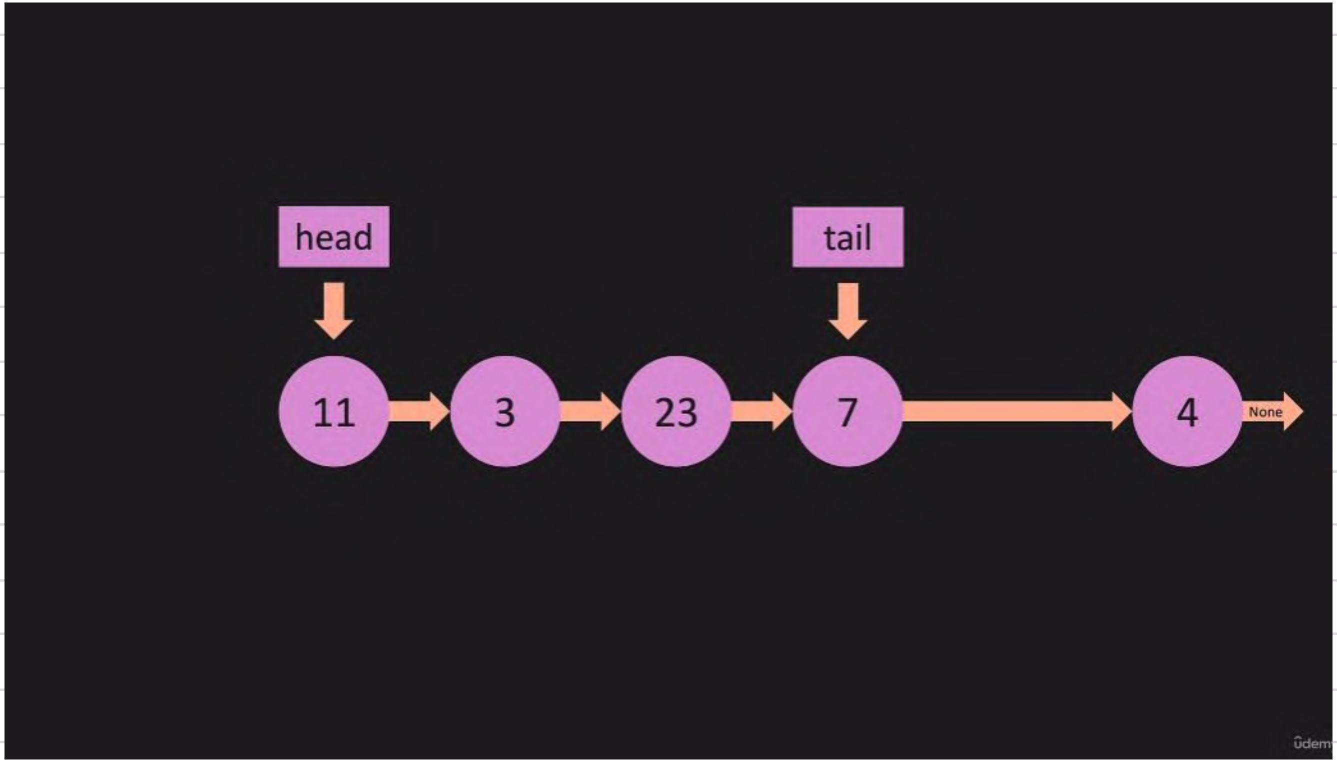
Add & Remove Node di Belakang

Bagaimana jika kita ingin menambahkan node “4” di belakang linked list



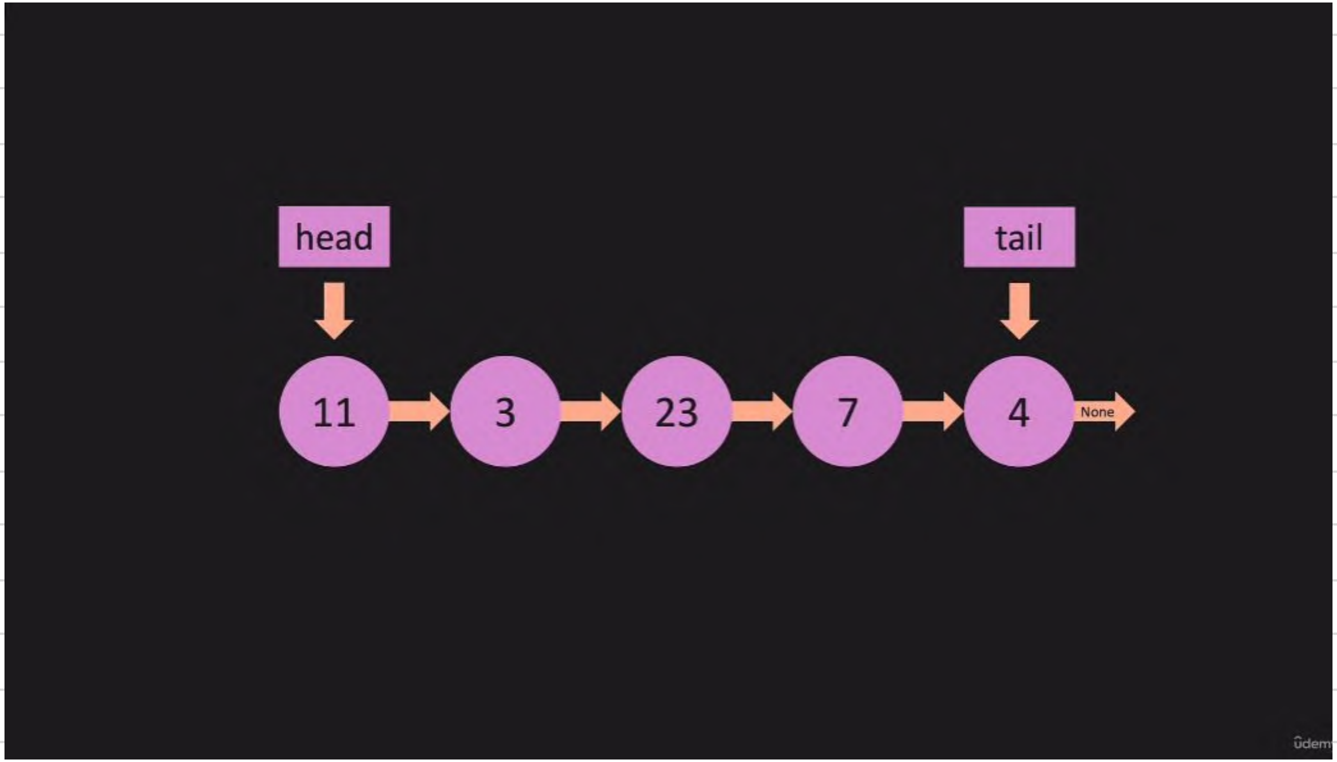
Add & Remove Node di Belakang

Bagaimana jika kita ingin menambahkan node “4” di belakang linked list



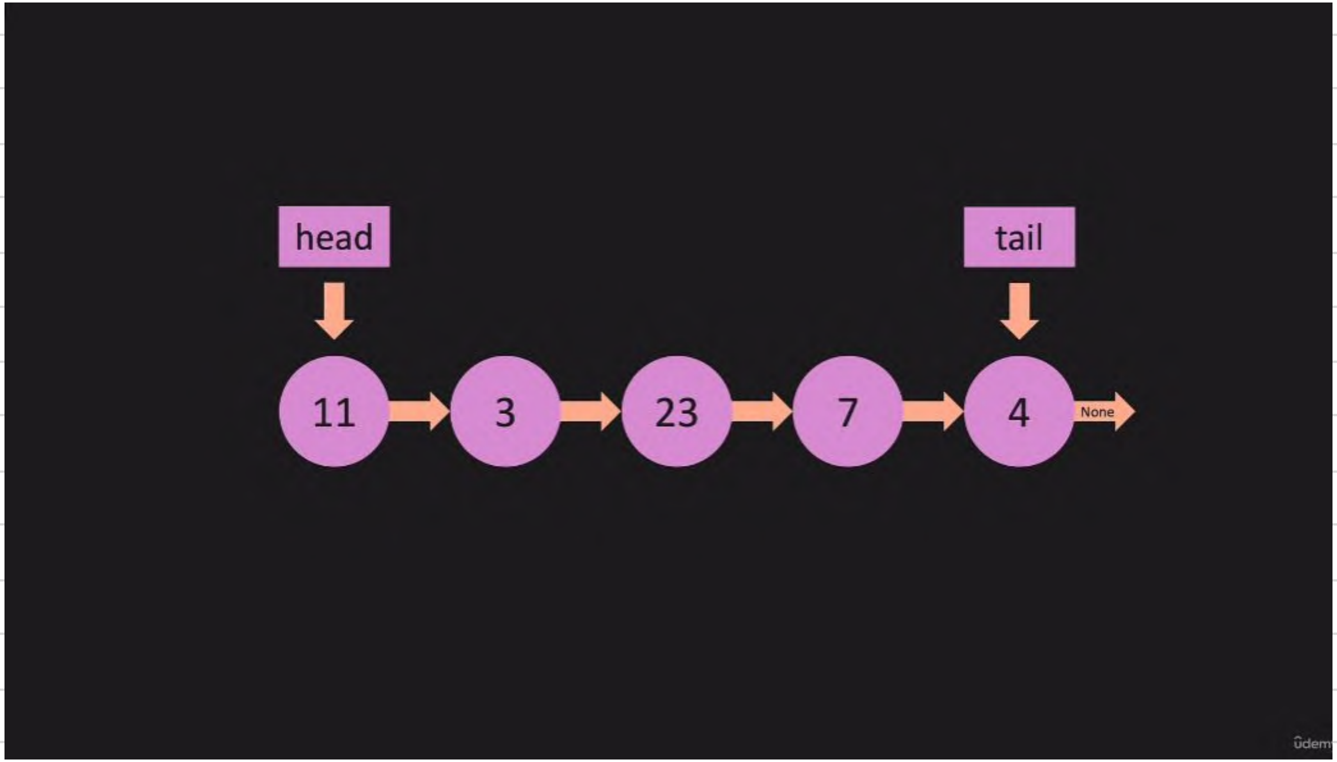
Add & Remove Node di Belakang

Bagaimana jika kita ingin menambahkan node “4” di belakang linked list

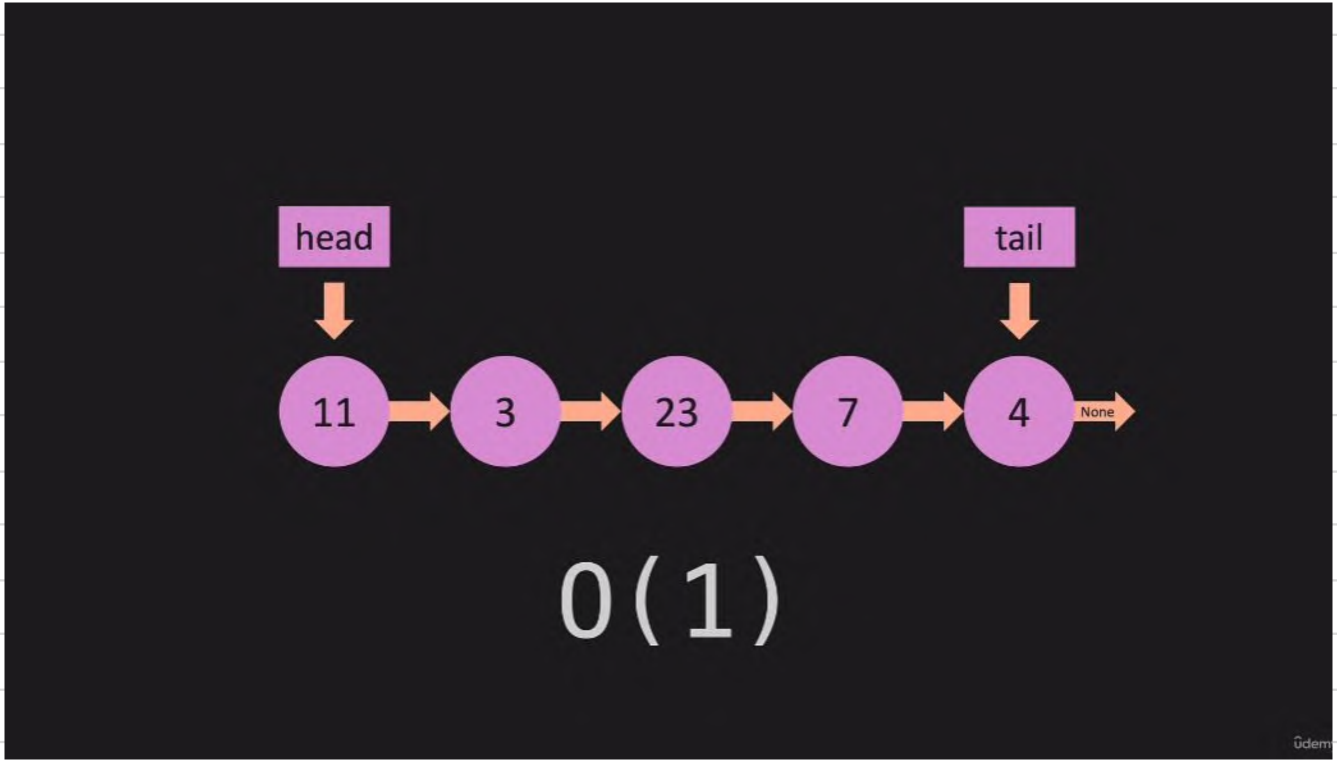


Add & Remove Node di Belakang

Apa notasi Big O untuk
Add Last ... ?

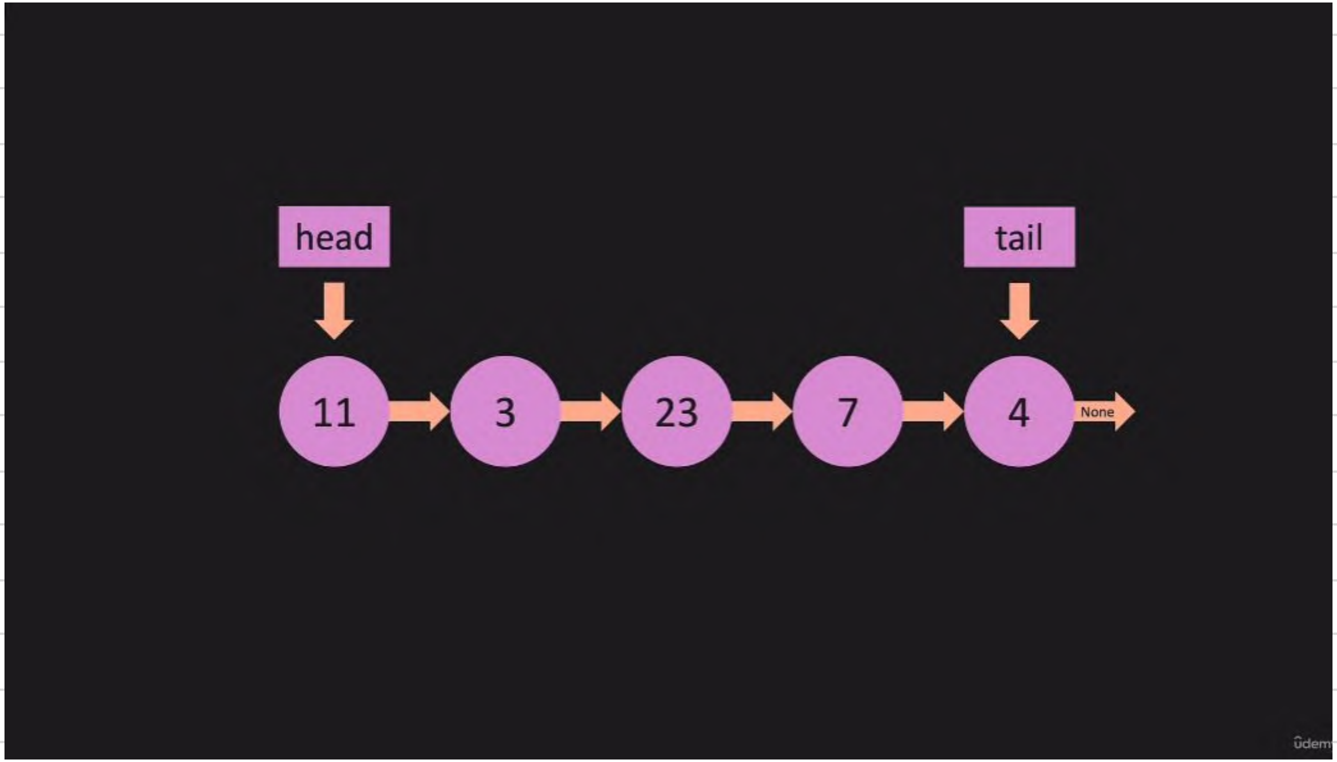


Add & Remove Node di Belakang



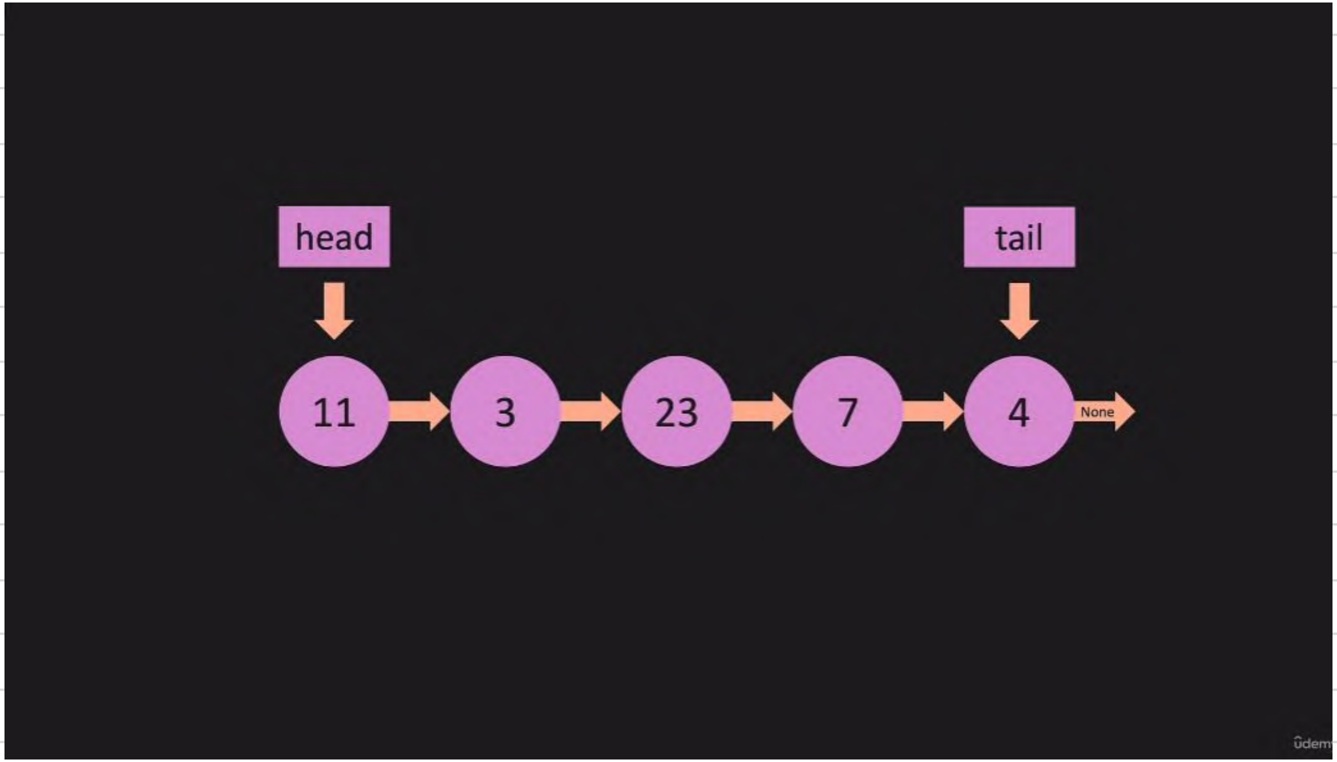
Diketahui suatu Linked List...

Add & Remove Node di Belakang



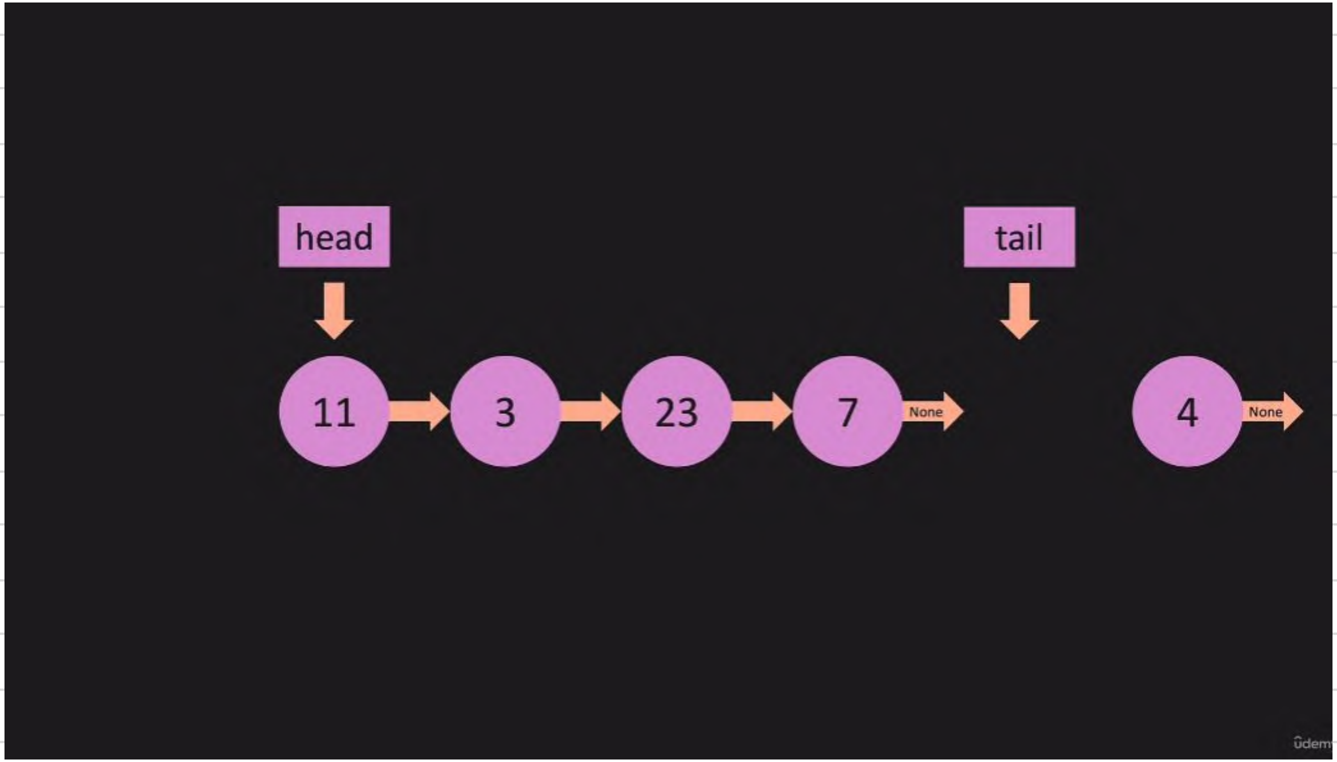
Add & Remove Node di Belakang

Bagaimana jika kita ingin menghapus node “4” dari linked list



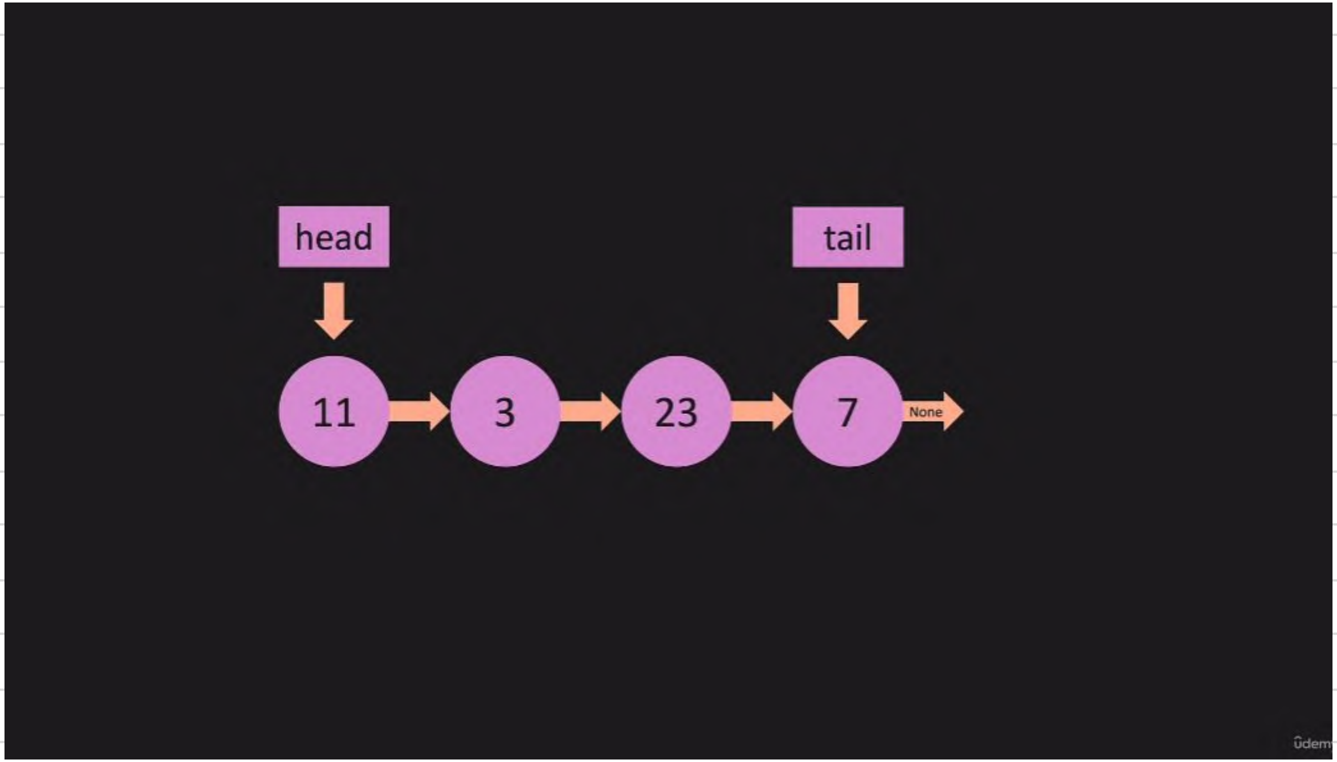
Add & Remove Node di Belakang

Bagaimana jika kita ingin menghapus node “4” dari linked list



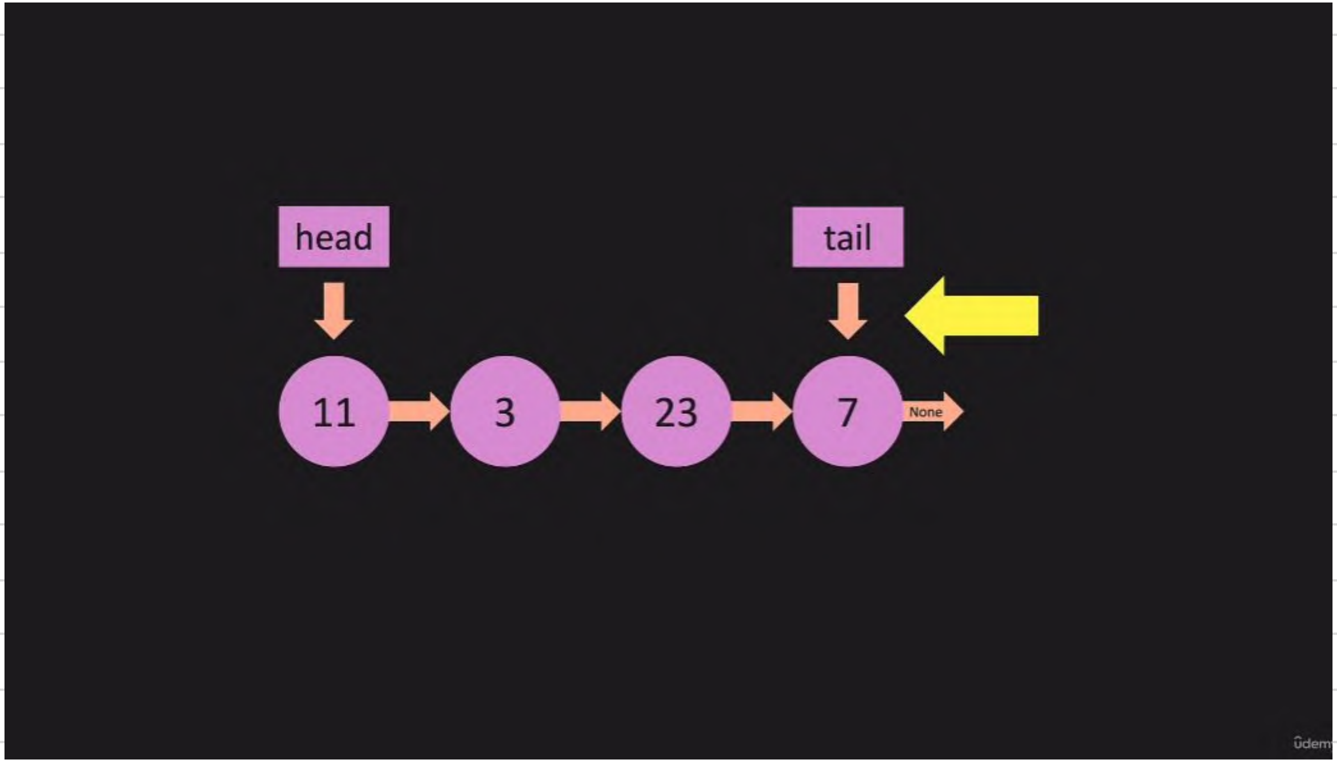
Add & Remove Node di Belakang

Bagaimana jika kita ingin menghapus node “4” dari linked list



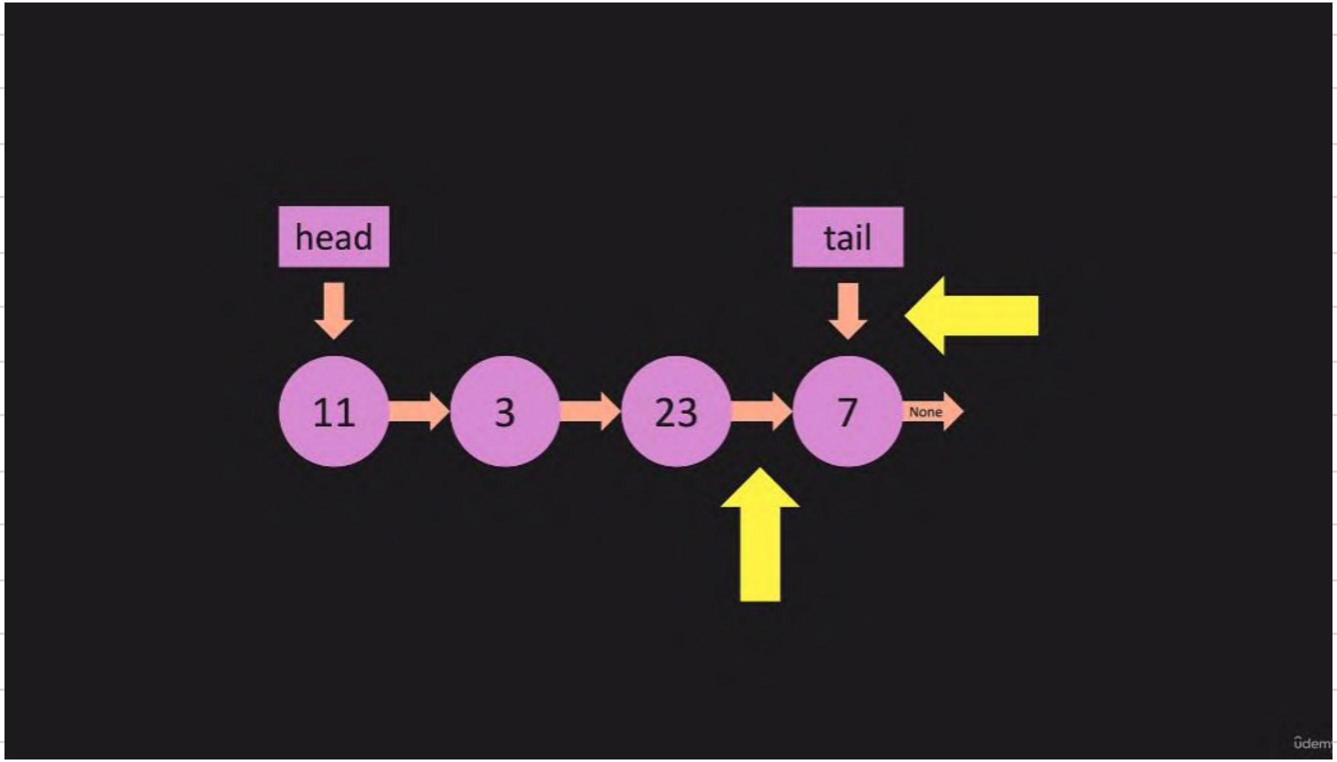
Add & Remove Node di Belakang

Bagaimana mengarahkan "tail" ke situ?



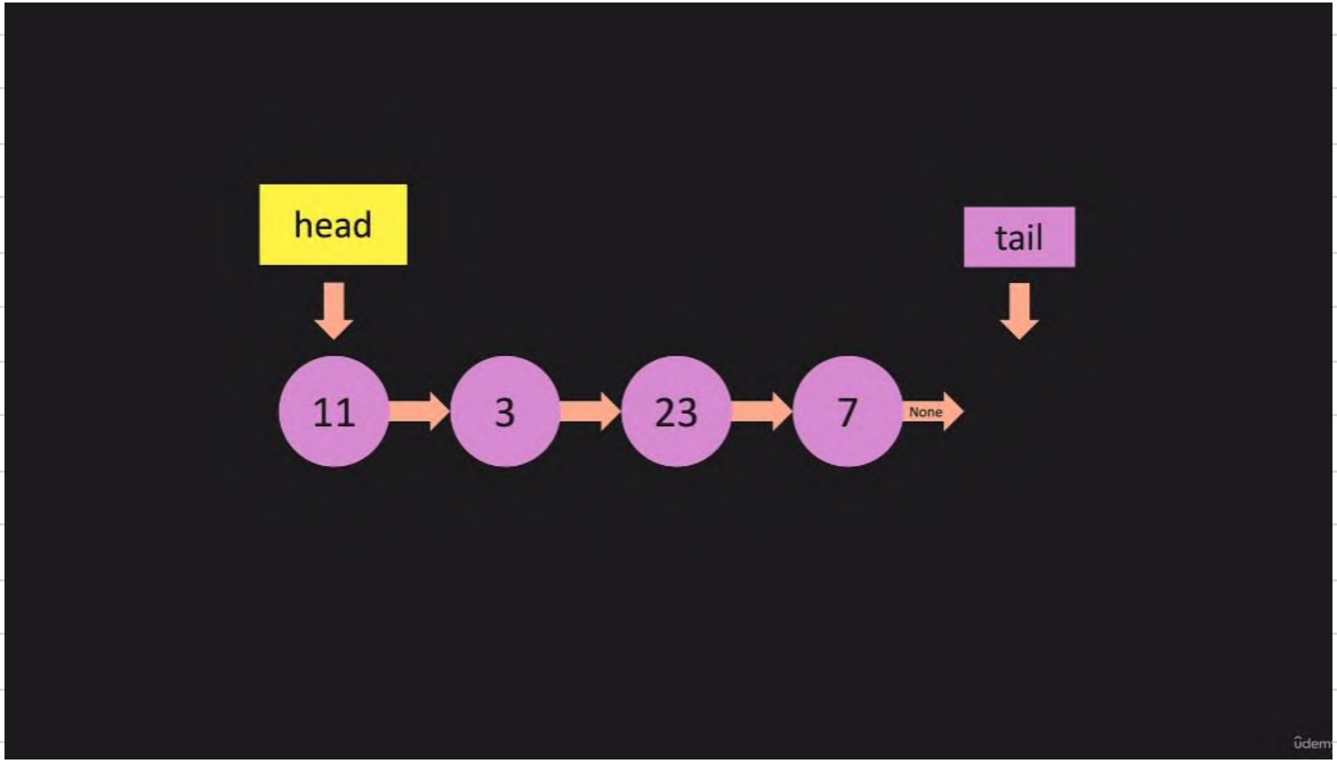
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



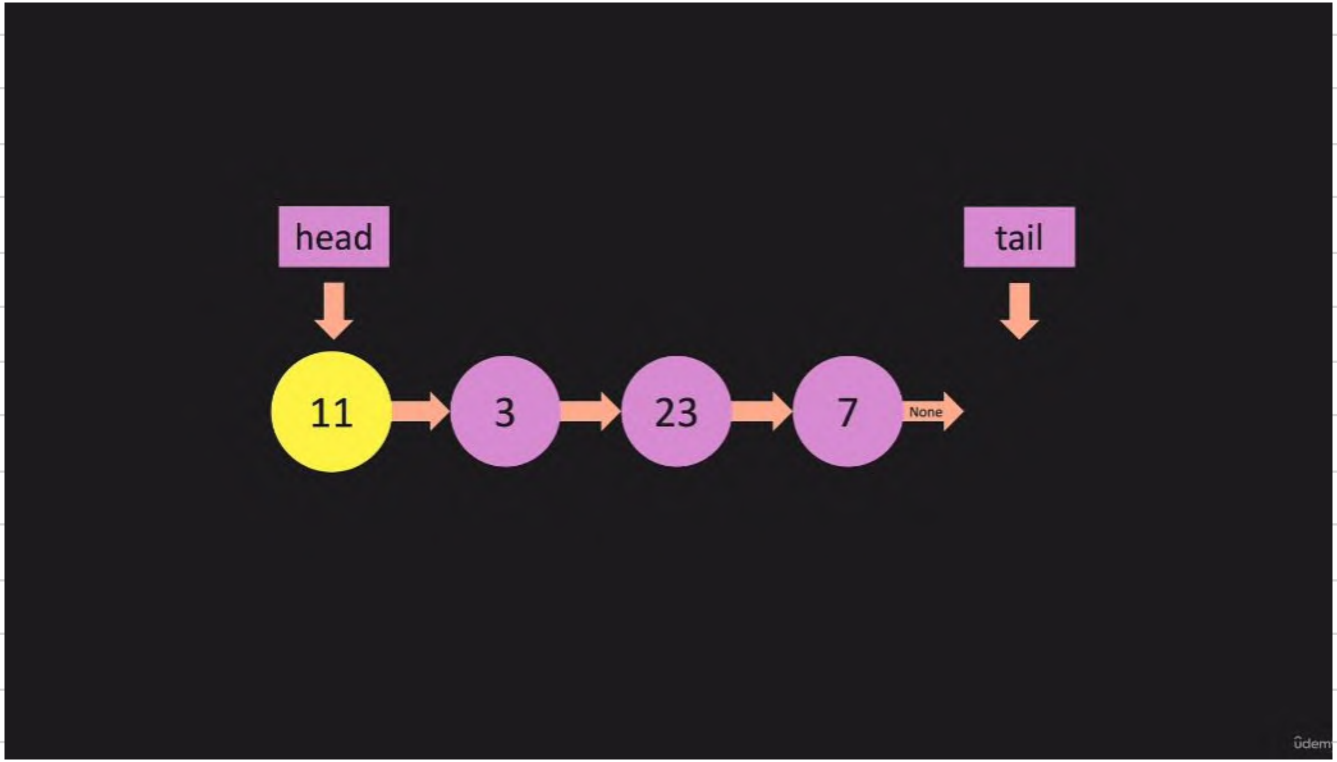
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



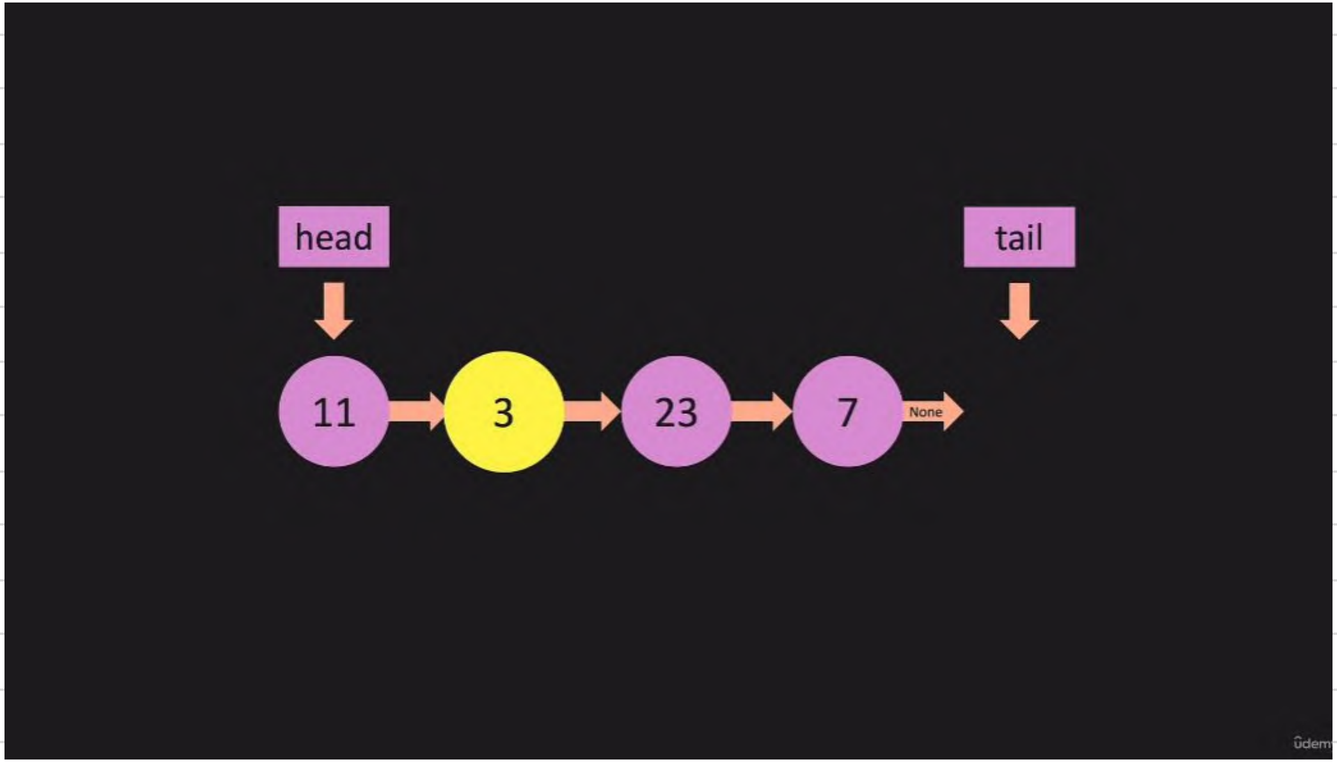
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



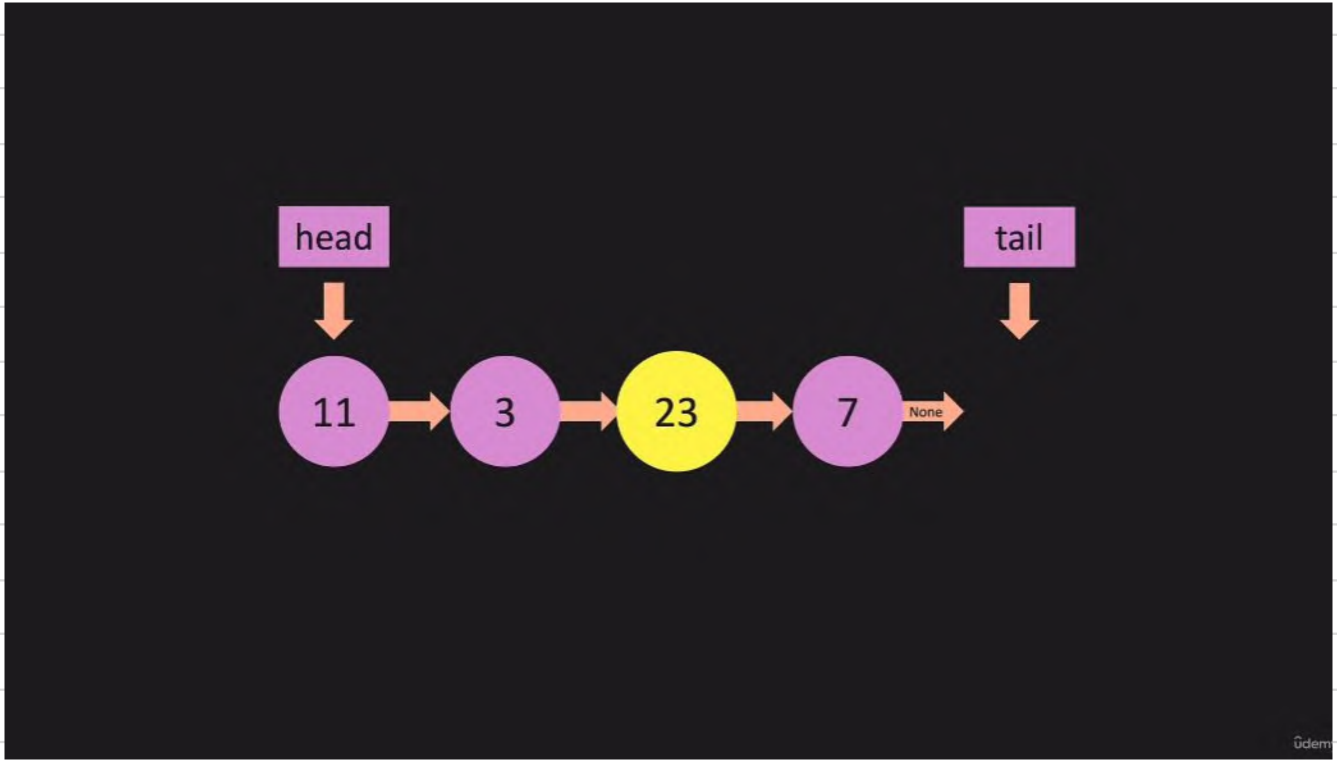
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



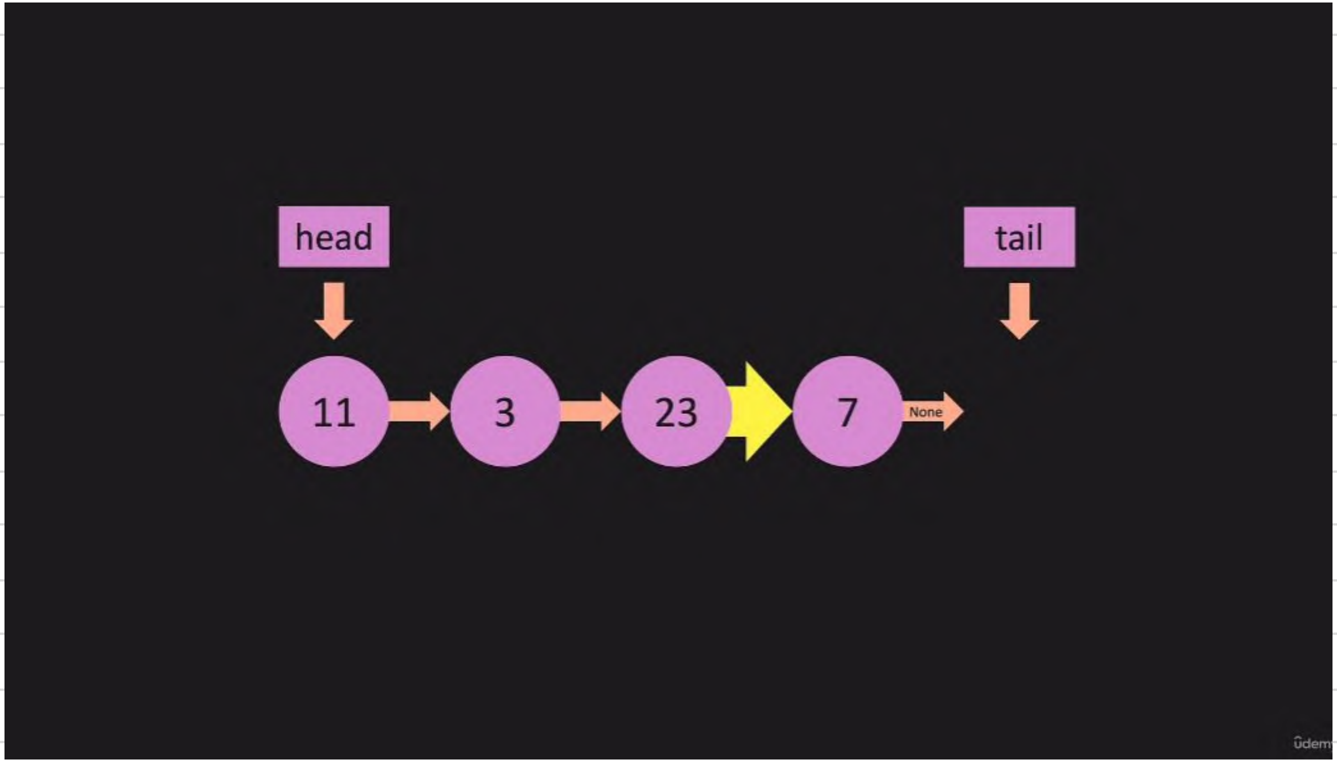
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



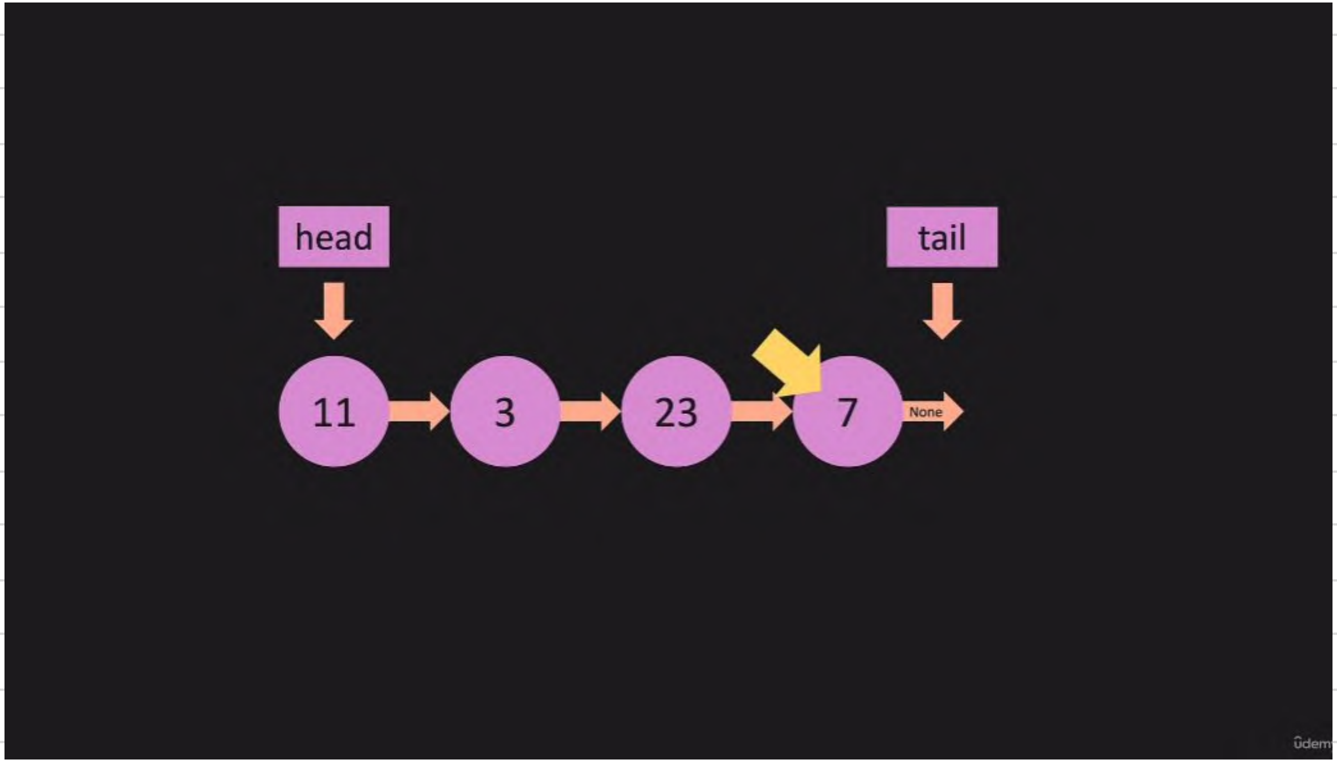
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



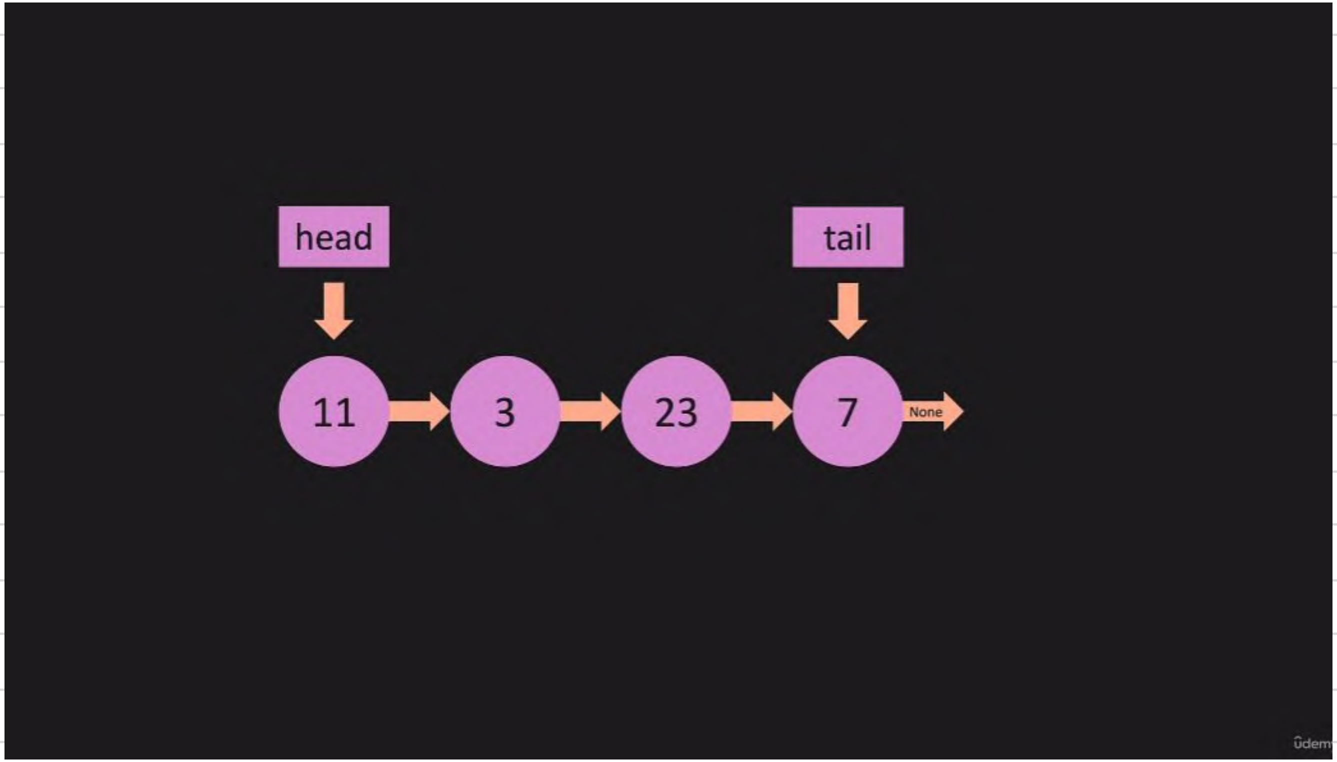
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



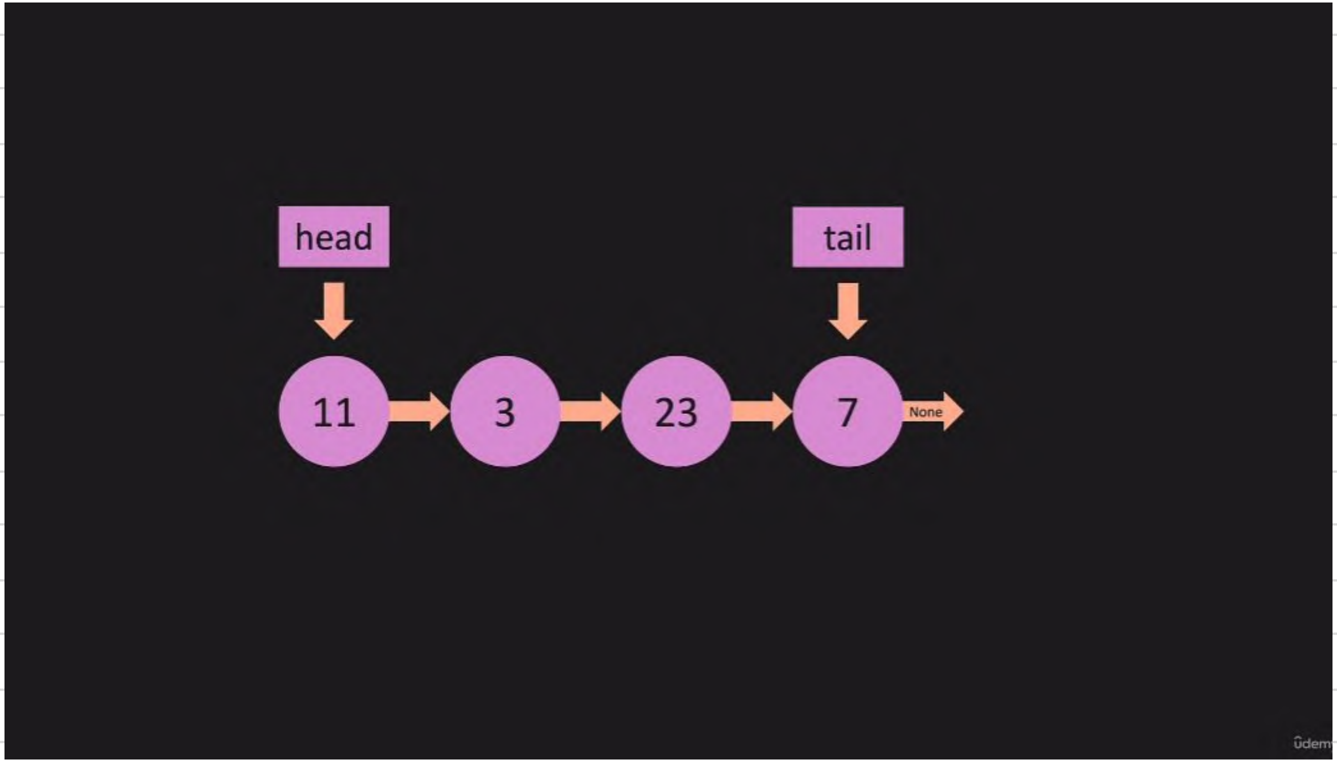
Add & Remove Node di Belakang

Bagaimana mengarahkan “tail” ke situ?



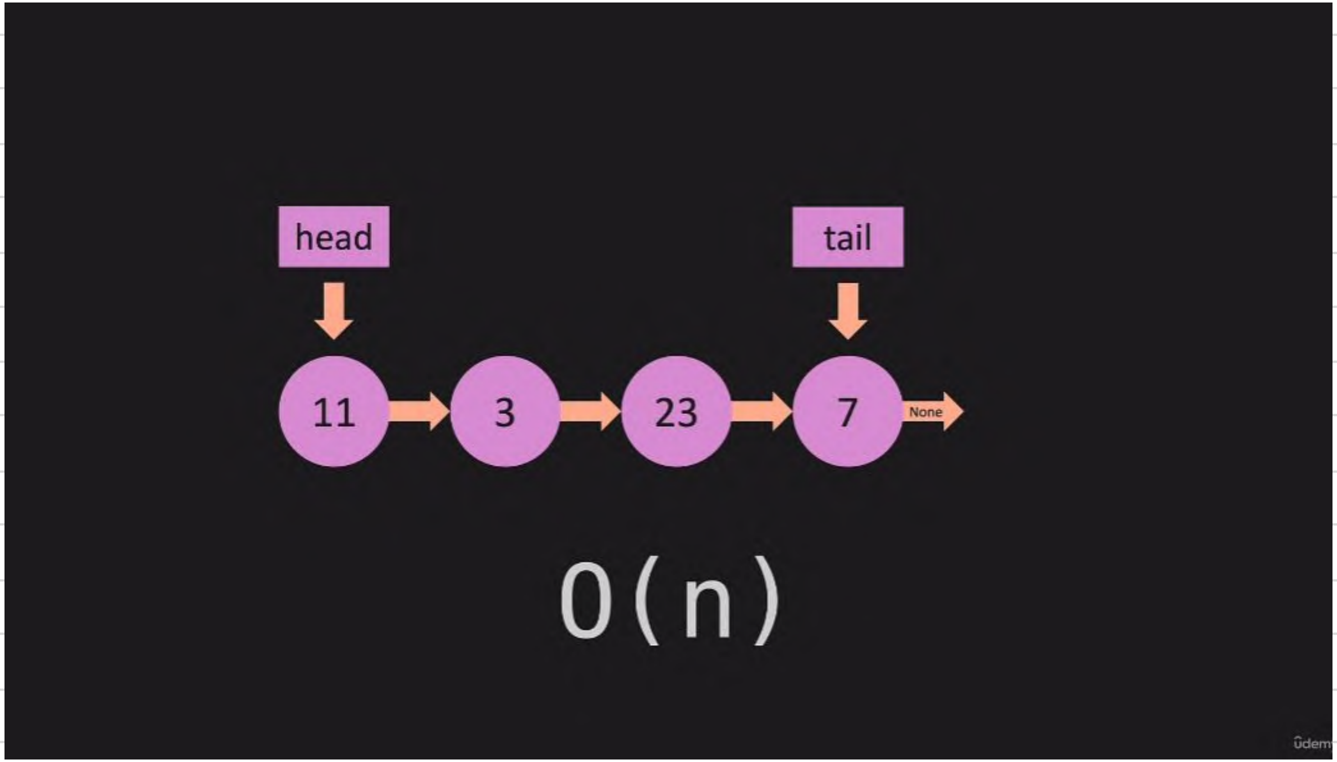
Add & Remove Node di Belakang

Apa notasi Big O untuk Remove Last … ?



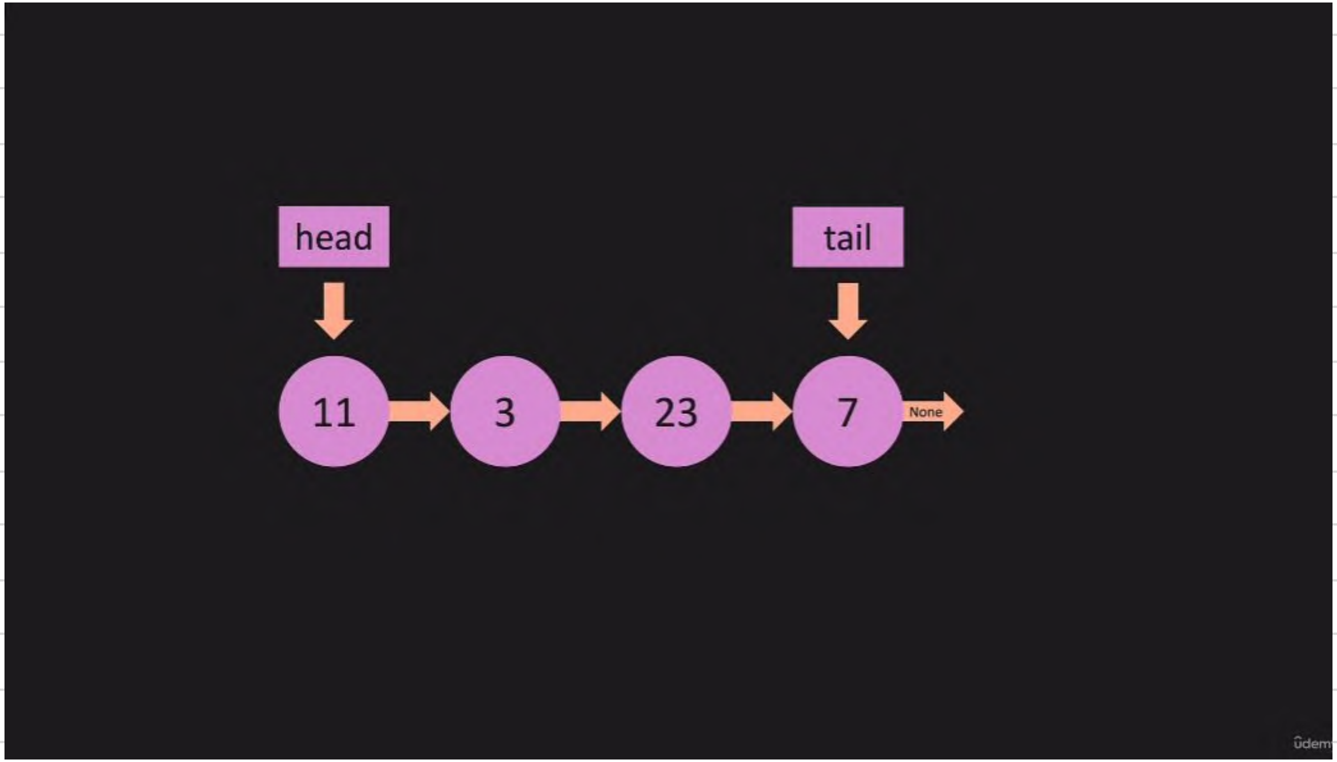
Add & Remove Node di Belakang

Apa notasi Big O untuk
Remove Last … ?



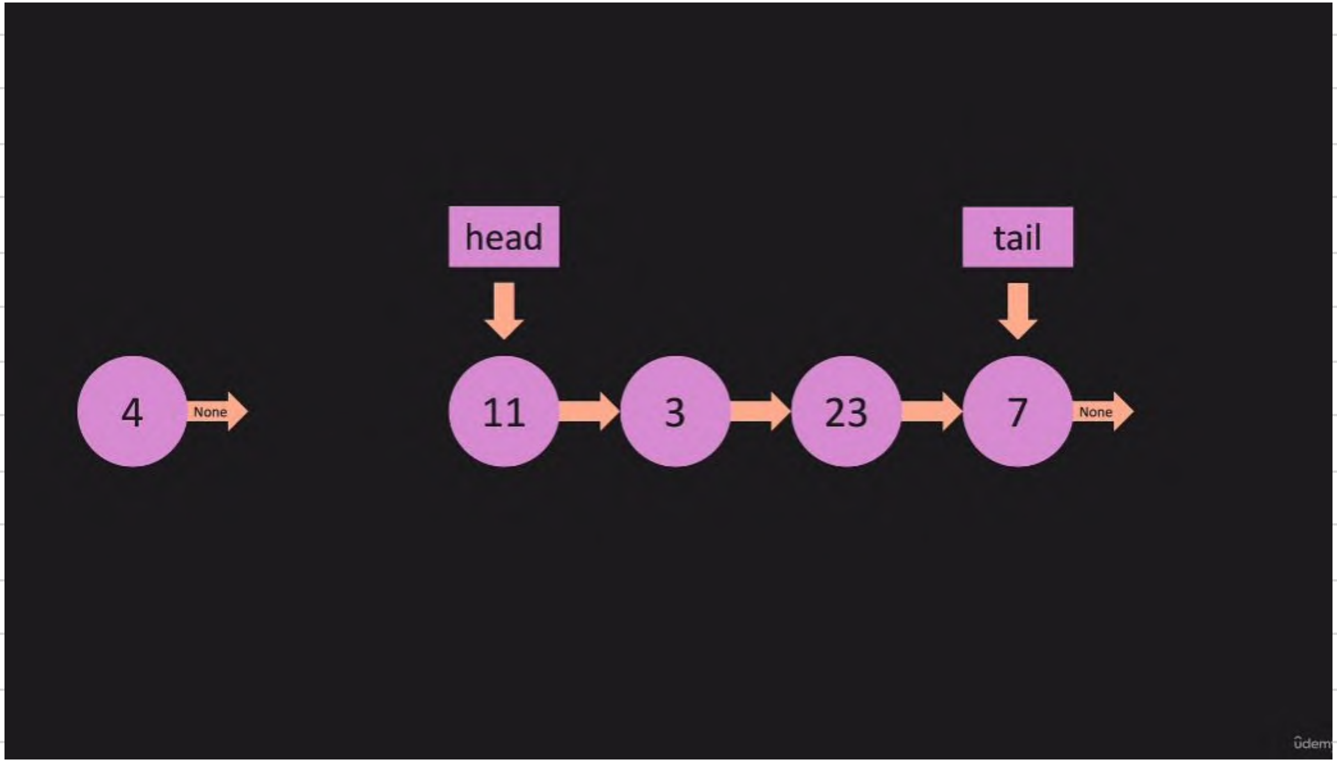
Diketahui suatu Linked List...

Add & Remove Node di Depan



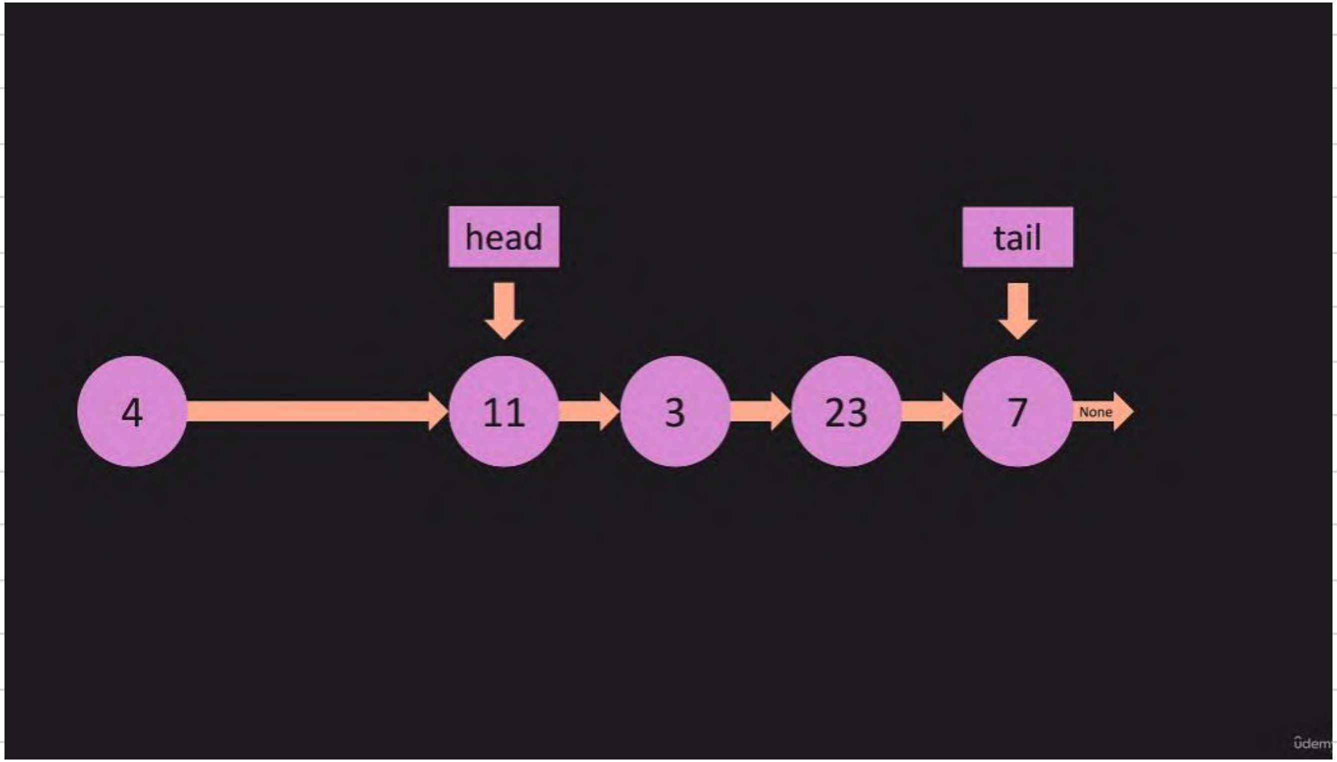
Add & Remove Node di Depan

Bagaimana jika kita ingin menambahkan node “4” di depan linked list



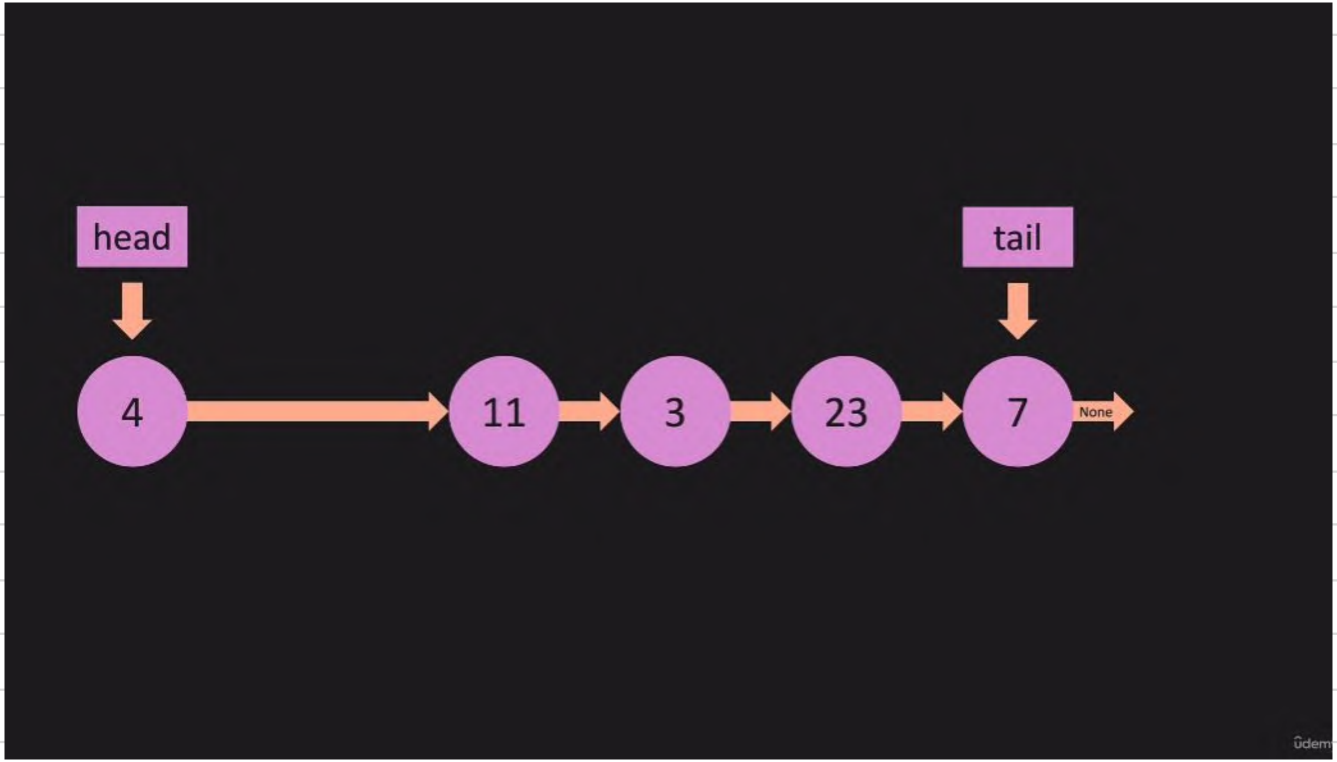
Add & Remove Node di Depan

Bagaimana jika kita ingin menambahkan node “4” di depan linked list



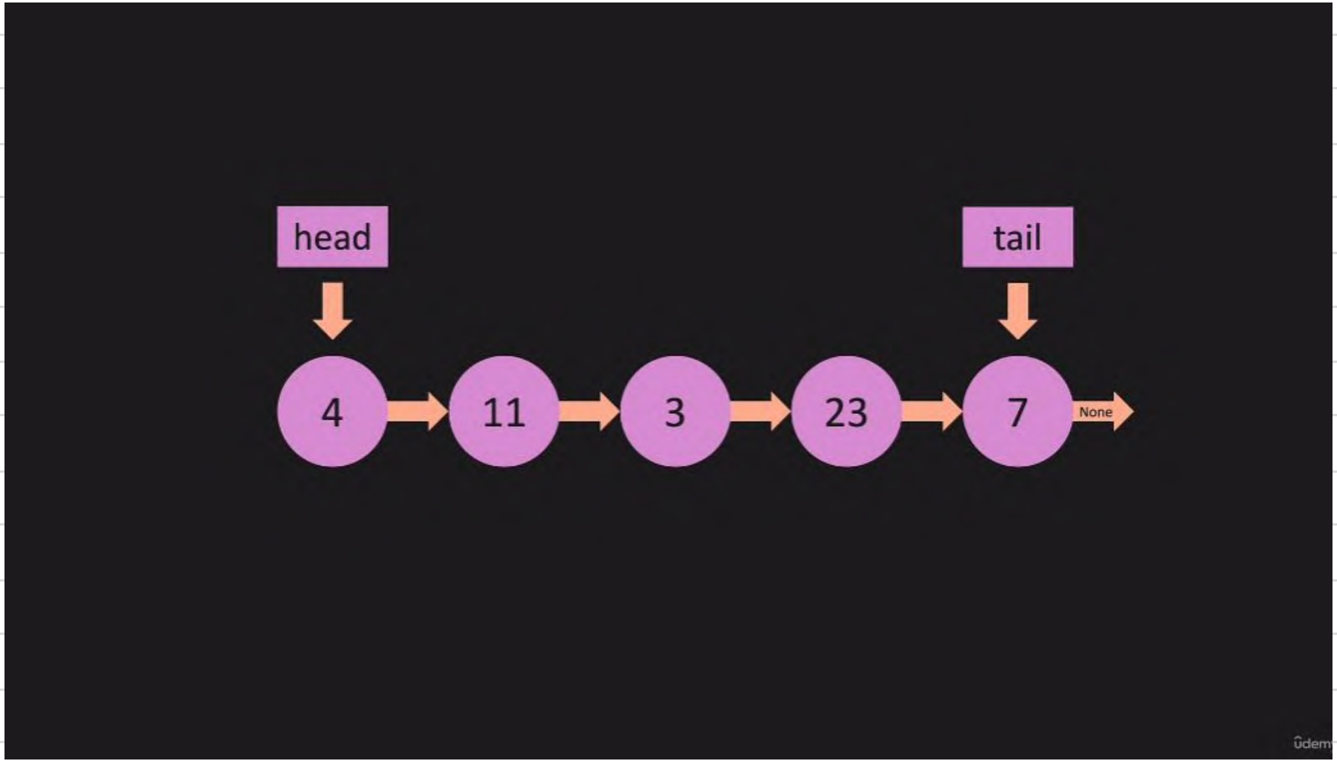
Add & Remove Node di Depan

Bagaimana jika kita ingin menambahkan node “4” di depan linked list



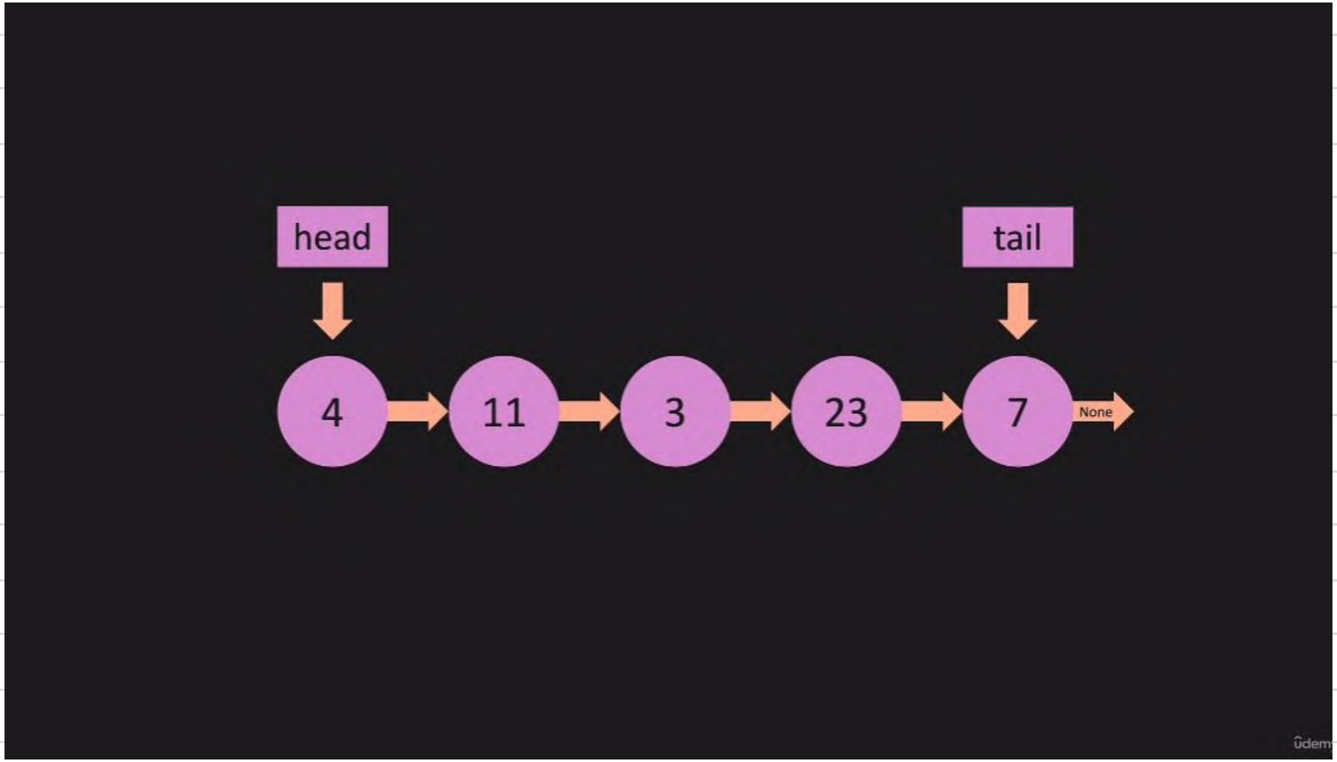
Add & Remove Node di Depan

Bagaimana jika kita ingin menambahkan node "4" di depan linked list



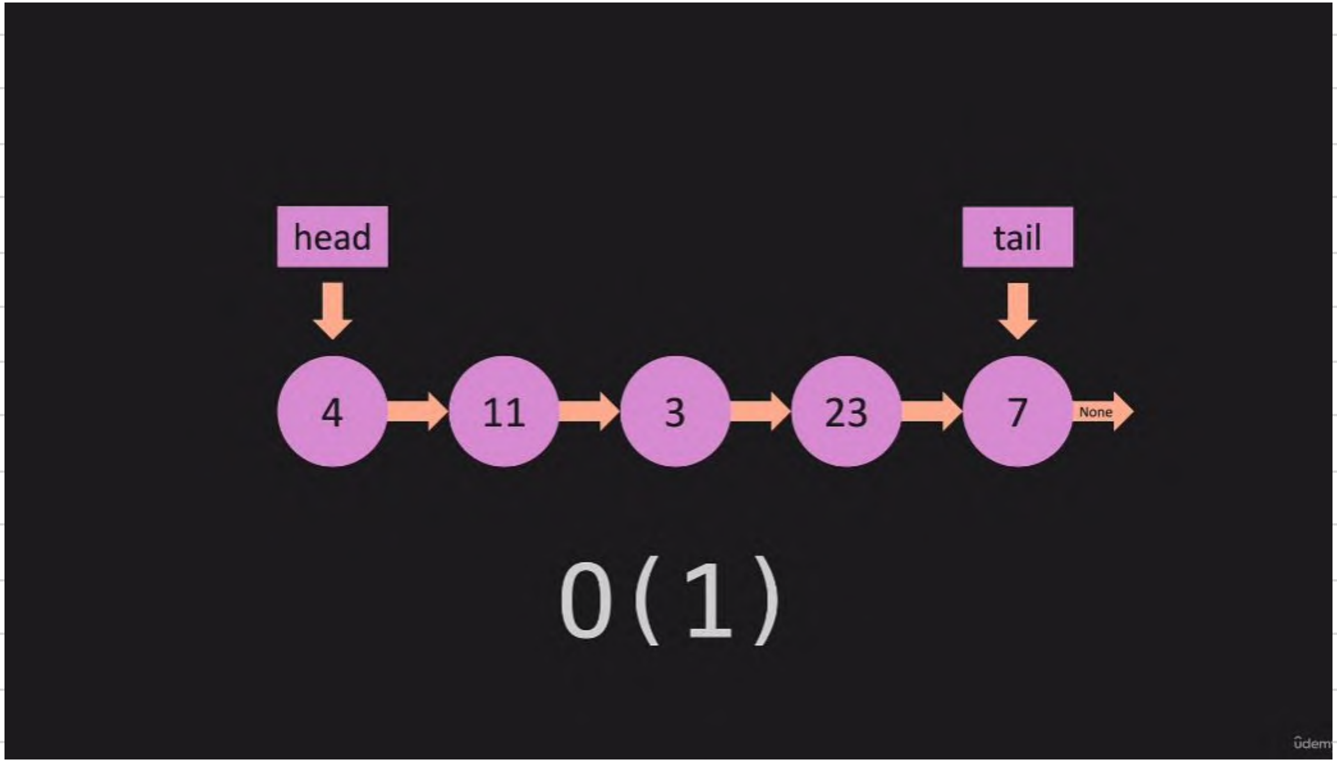
Add & Remove Node di Depan

Apa notasi Big O untuk
Add First ... ?



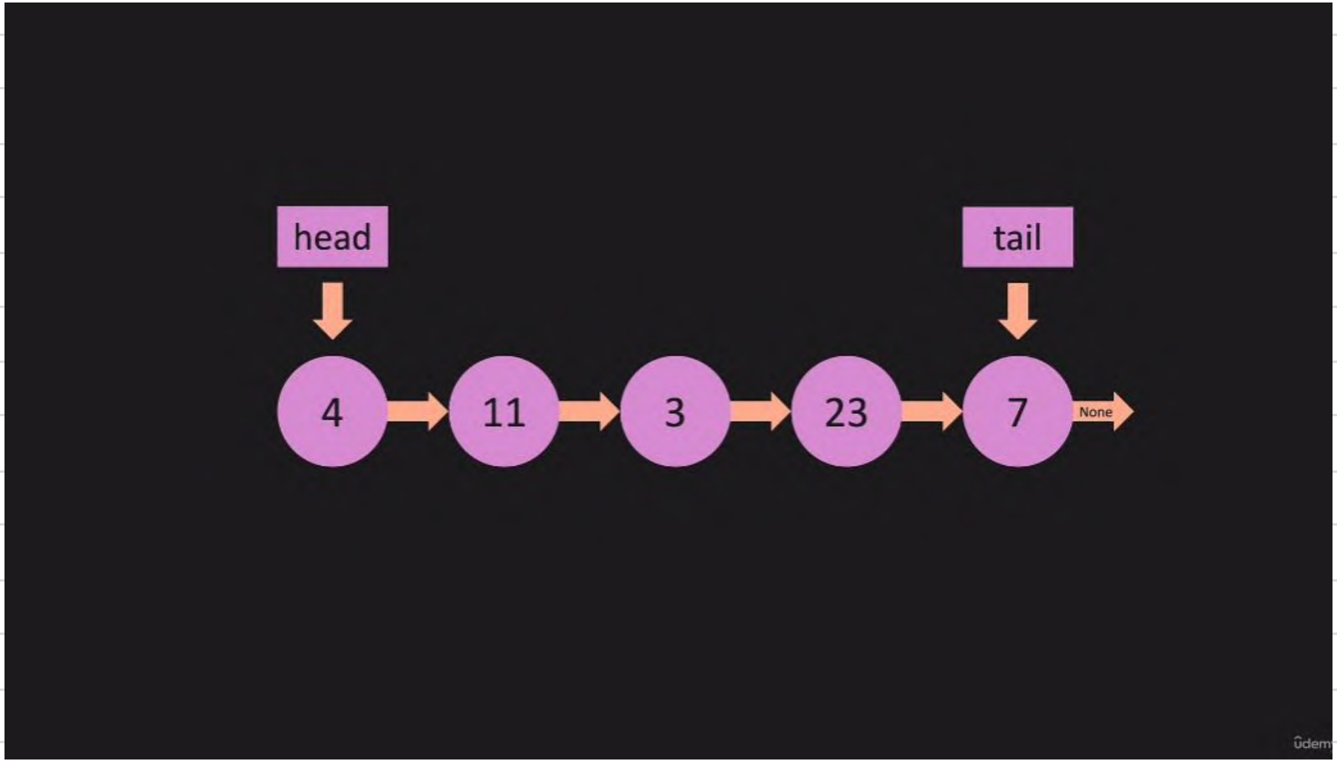
Add & Remove Node di Depan

Apa notasi Big O untuk
Add First … ?



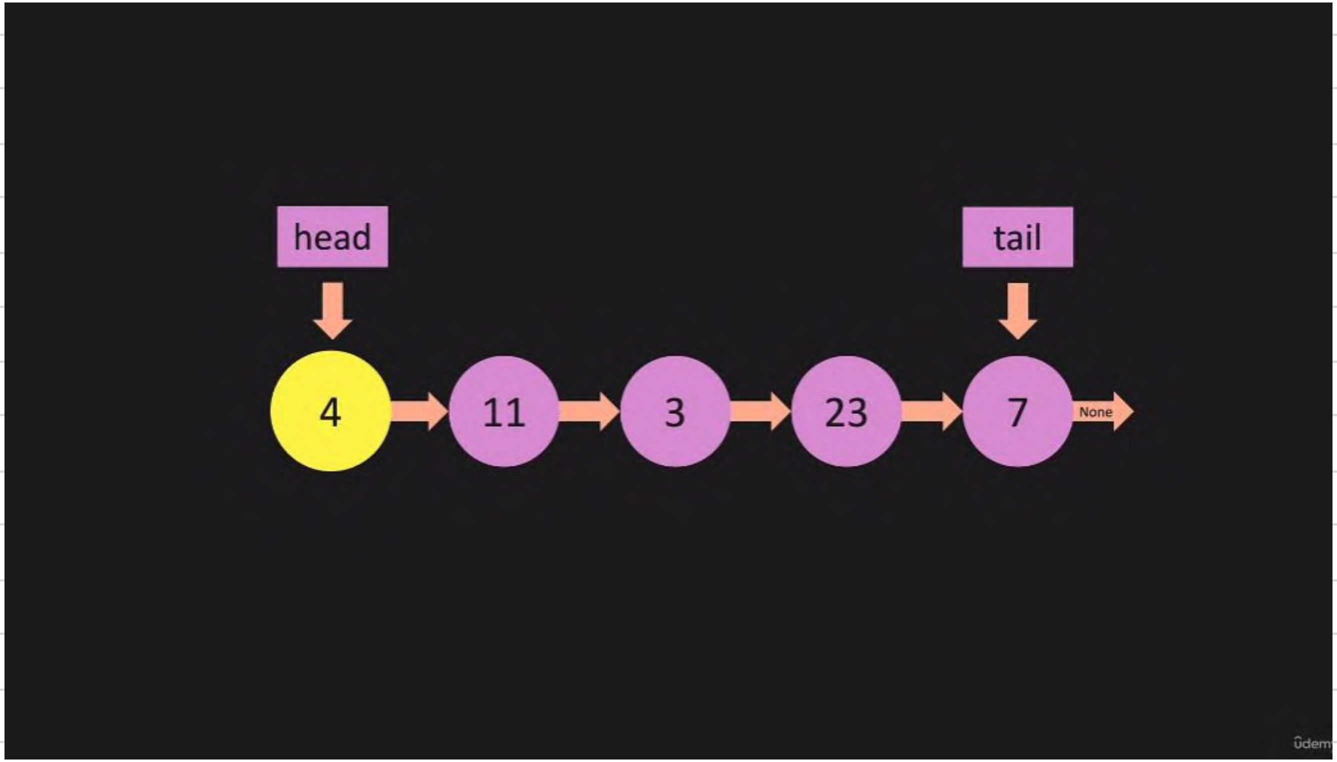
Diketahui suatu Linked List...

Add & Remove Node di Depan



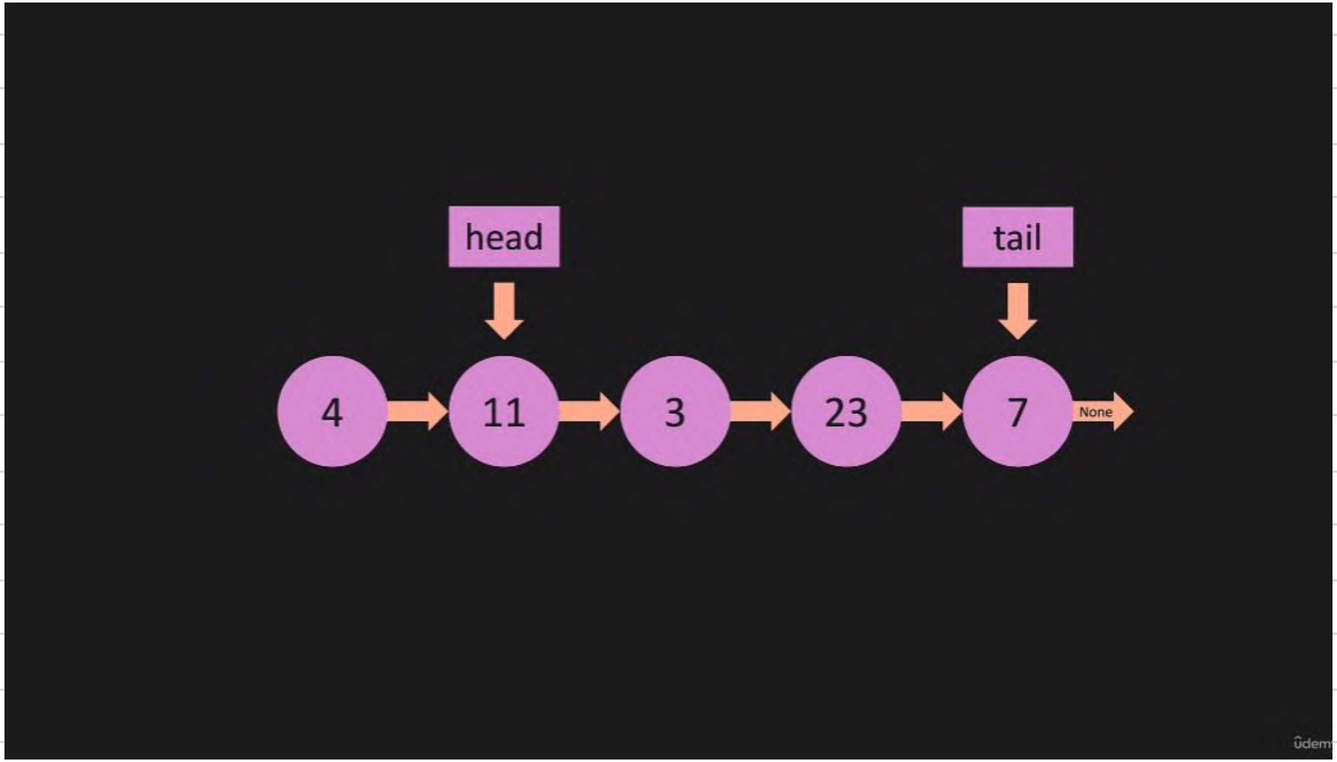
Add & Remove Node di Depan

Bagaimana jika kita ingin menghapus node “4” dari linked list



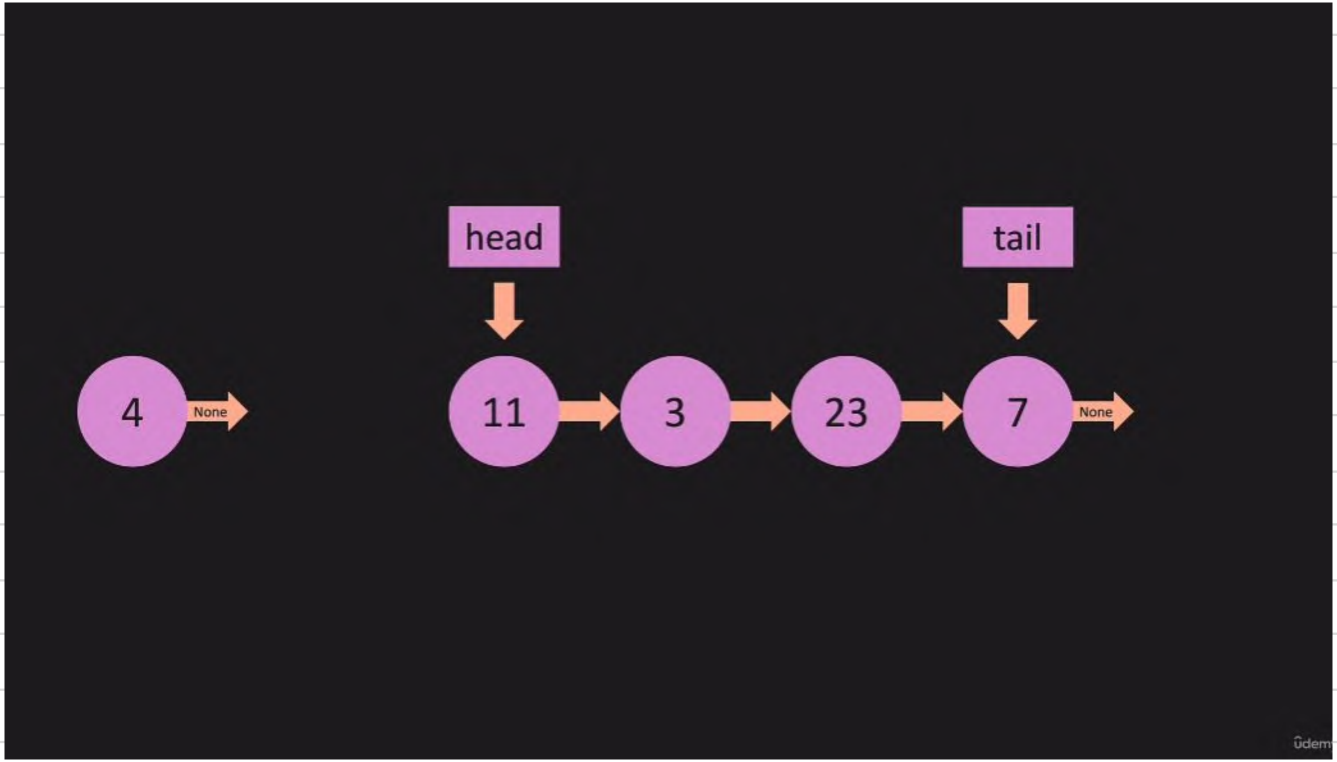
Add & Remove Node di Depan

Bagaimana jika kita ingin menghapus node “4” dari linked list



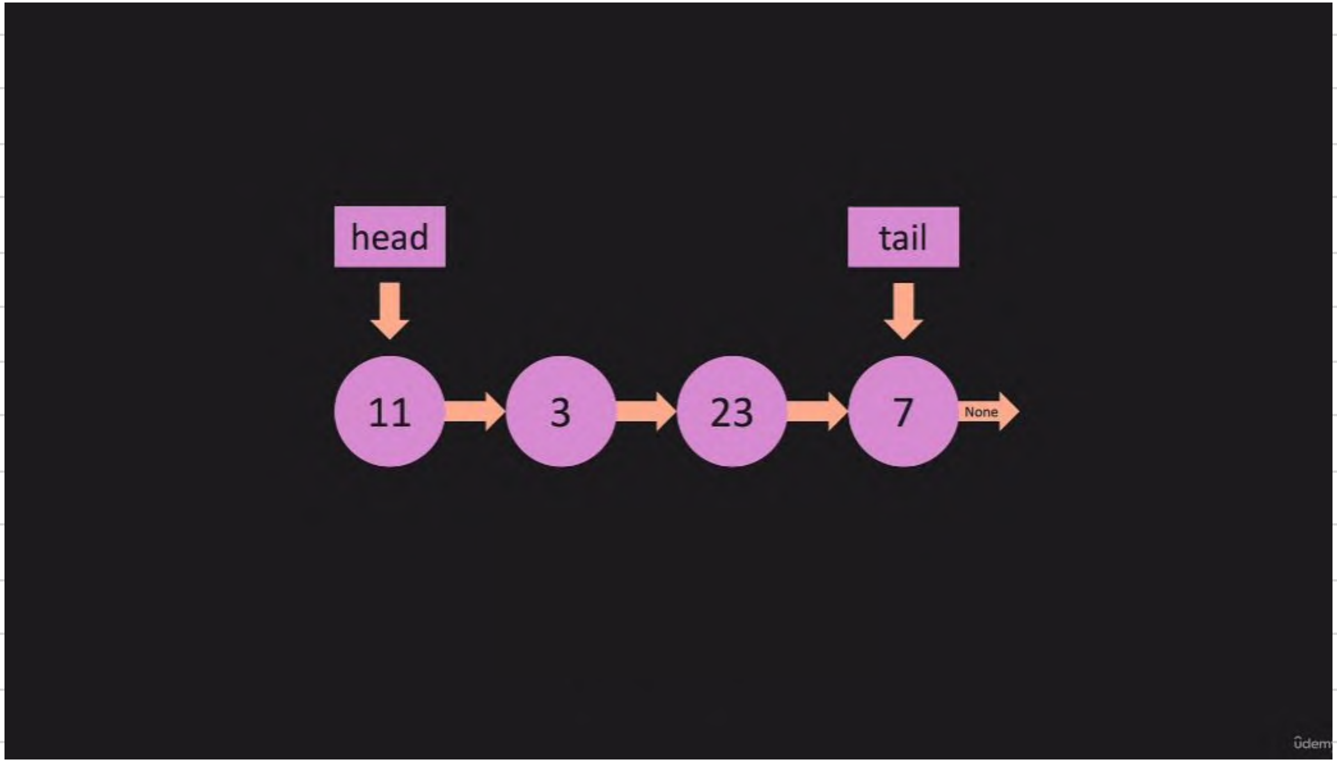
Add & Remove Node di Depan

Bagaimana jika kita ingin menghapus node "4" dari linked list



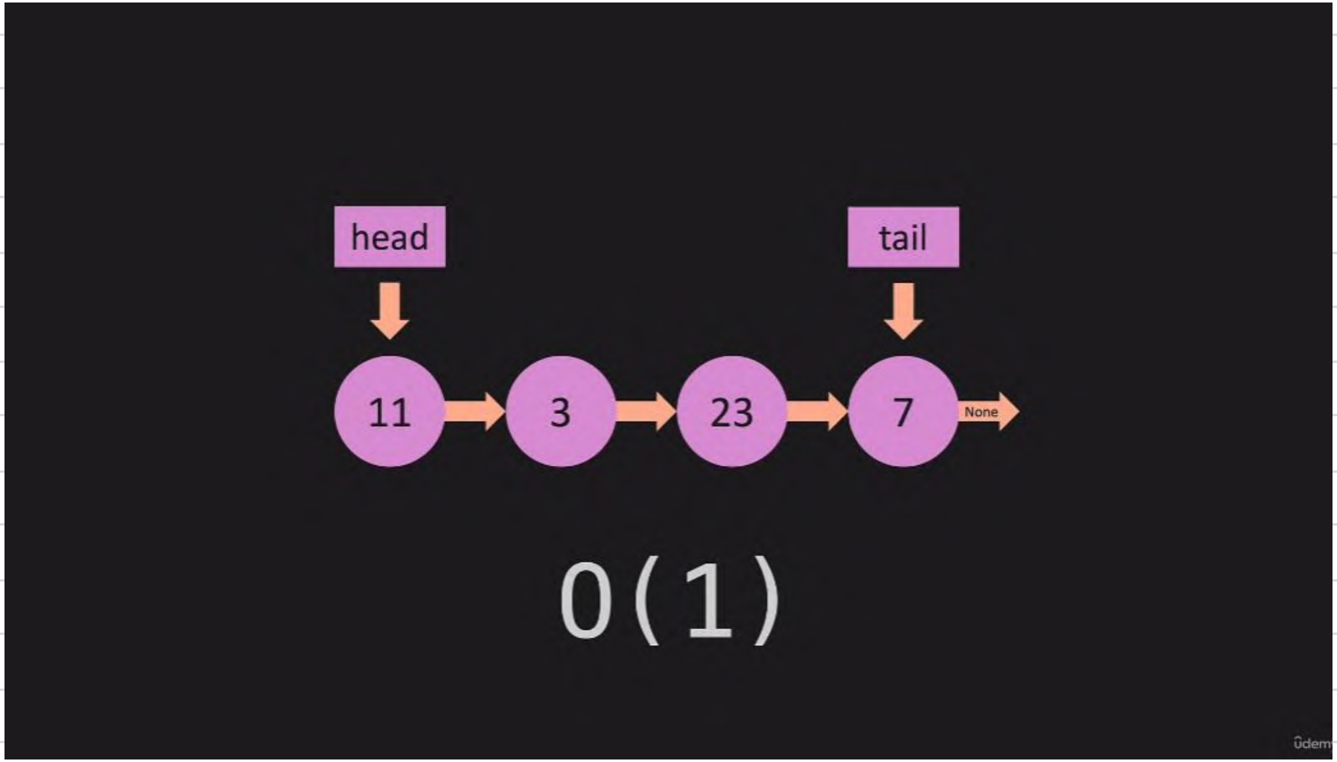
Add & Remove Node di Depan

Apa notasi Big O untuk
Remove First … ?



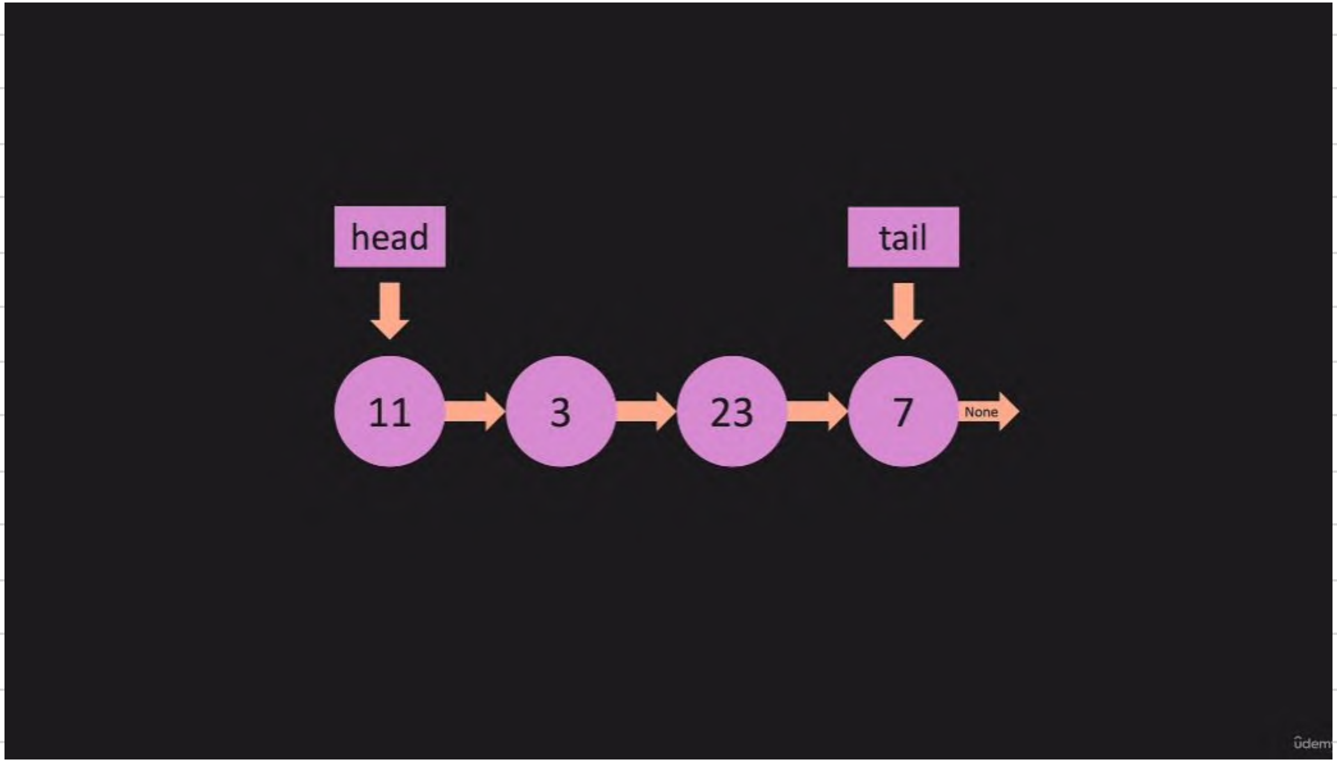
Add & Remove Node di Depan

Apa notasi Big O untuk Remove First … ?



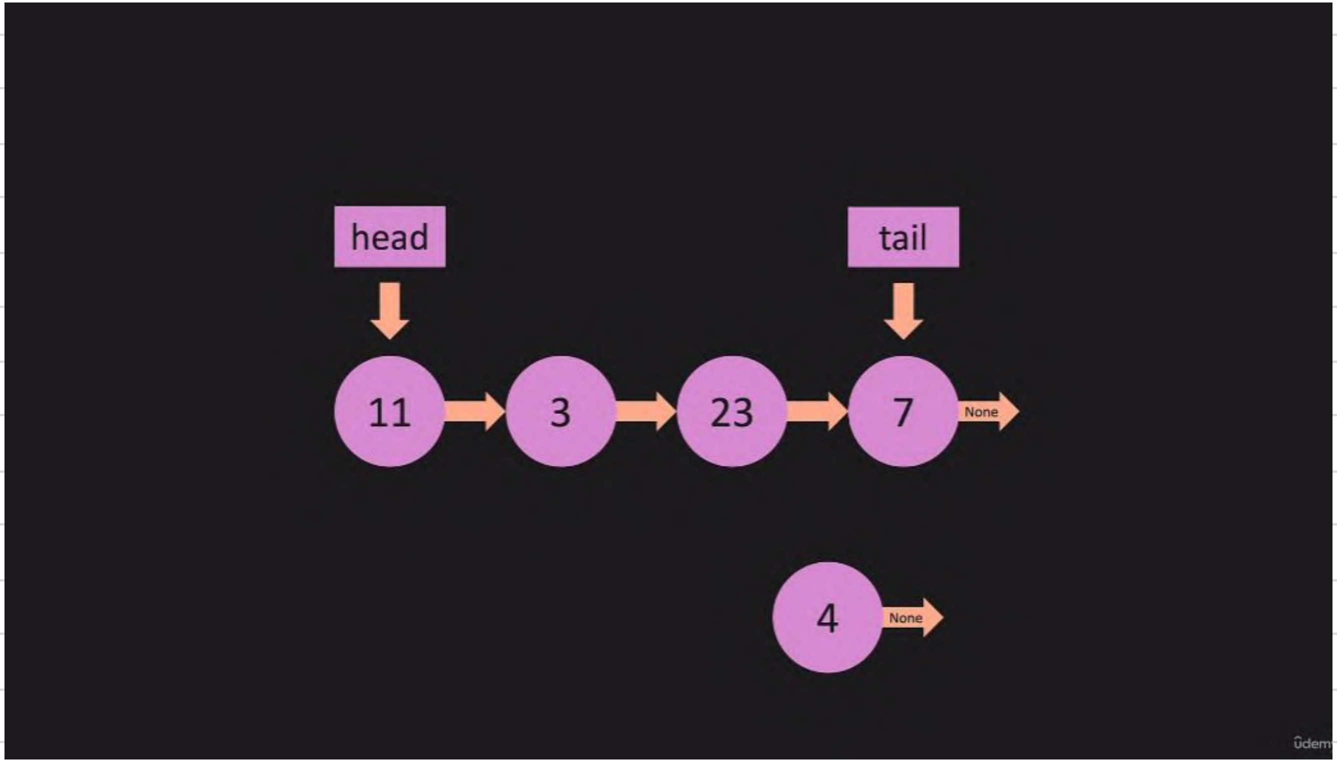
Diketahui suatu Linked List…

Add & Remove Node di Tengah



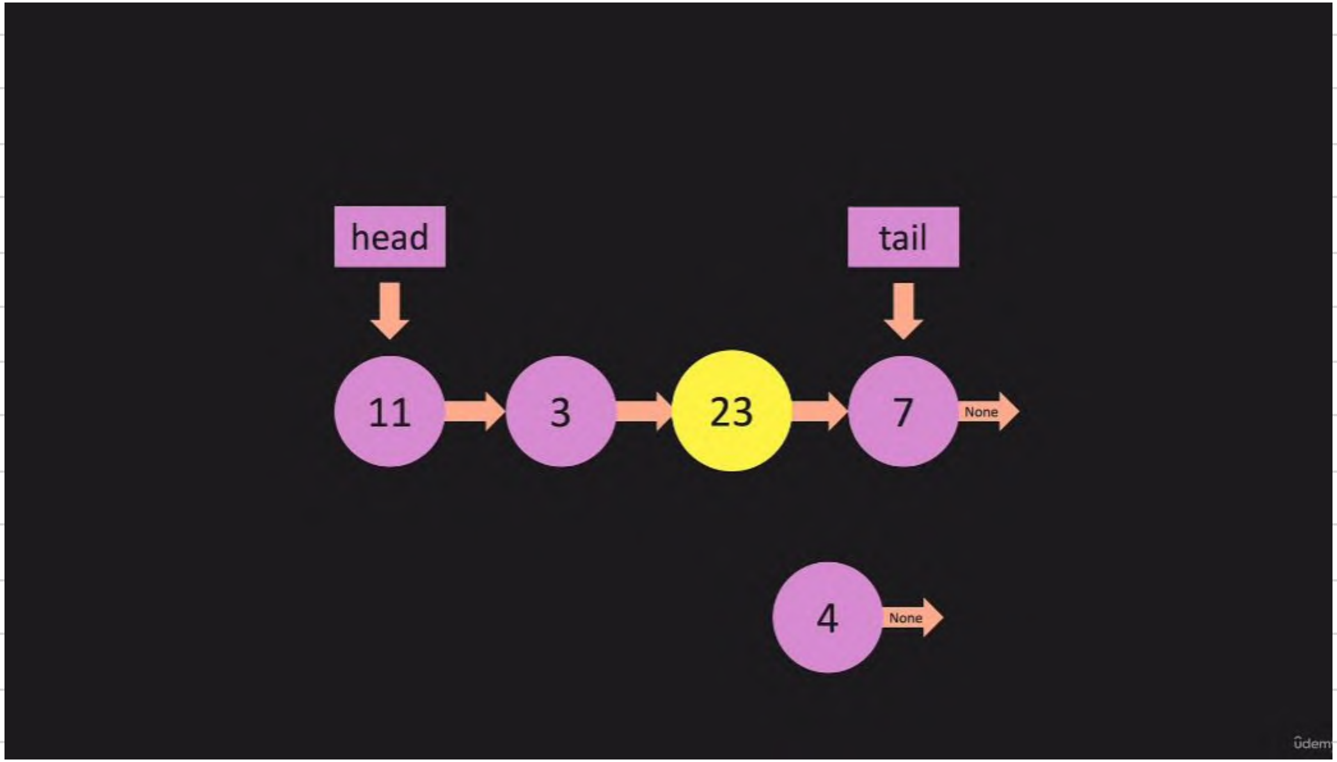
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di tengah linked list



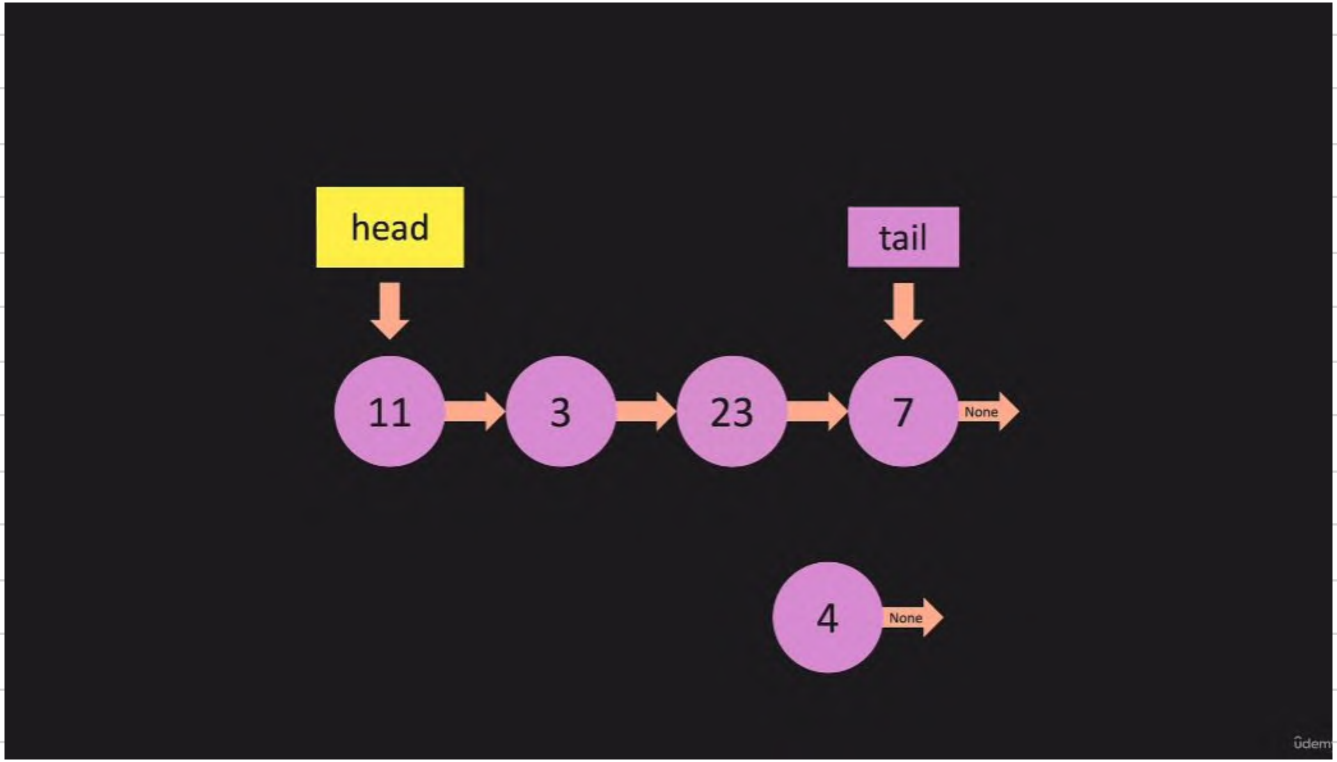
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



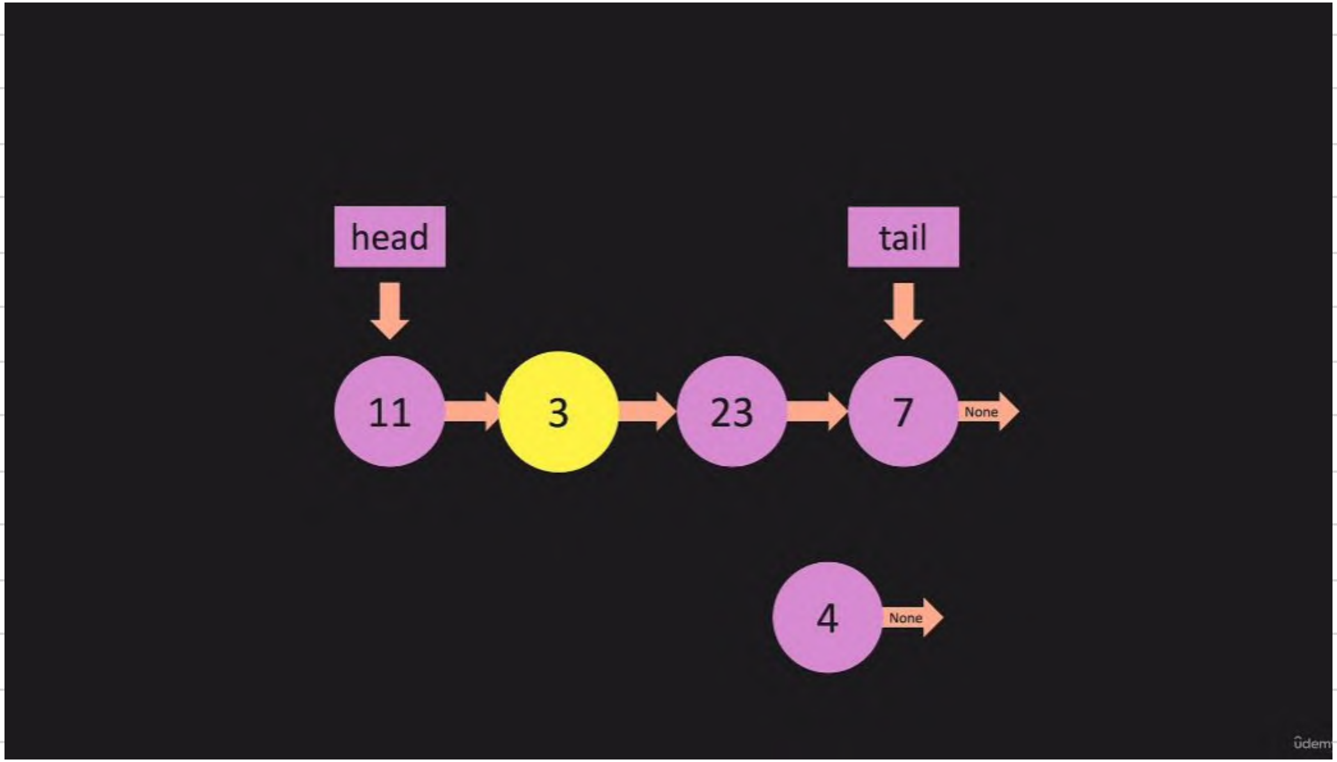
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



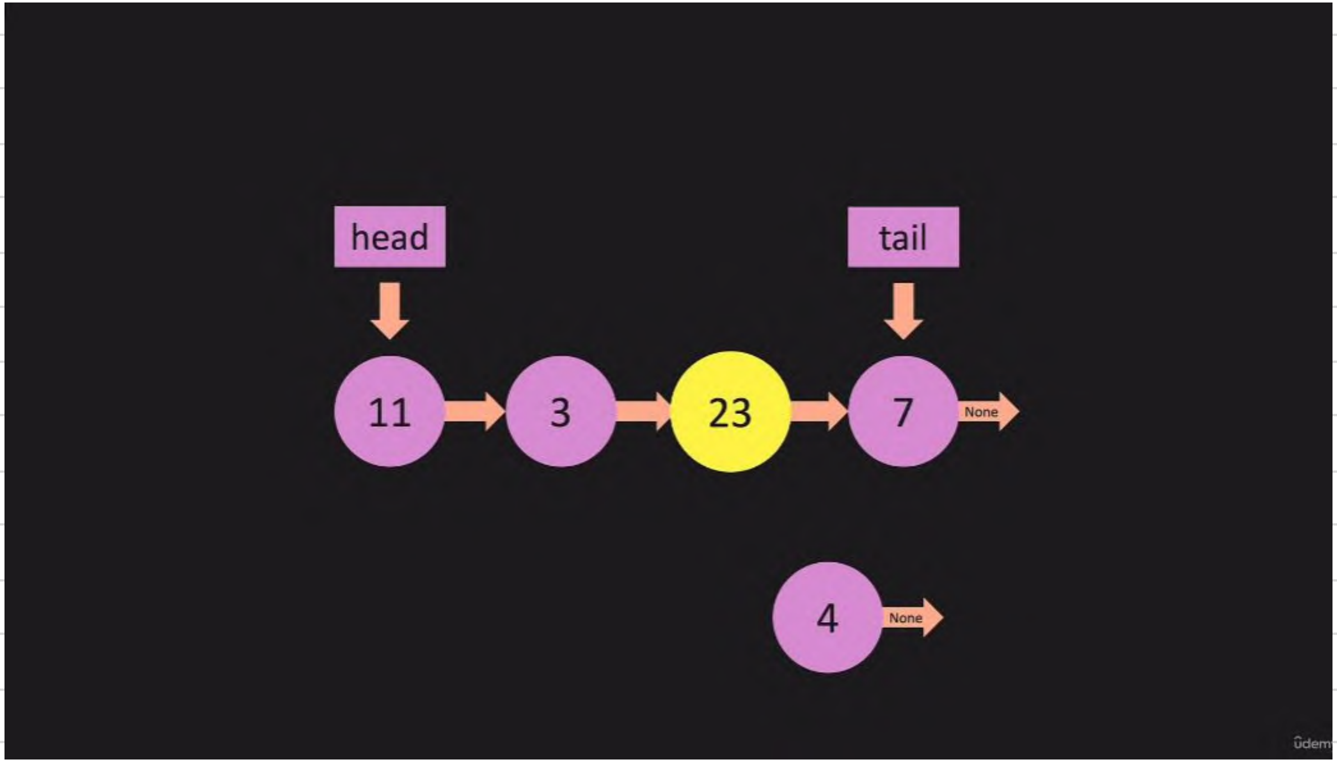
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



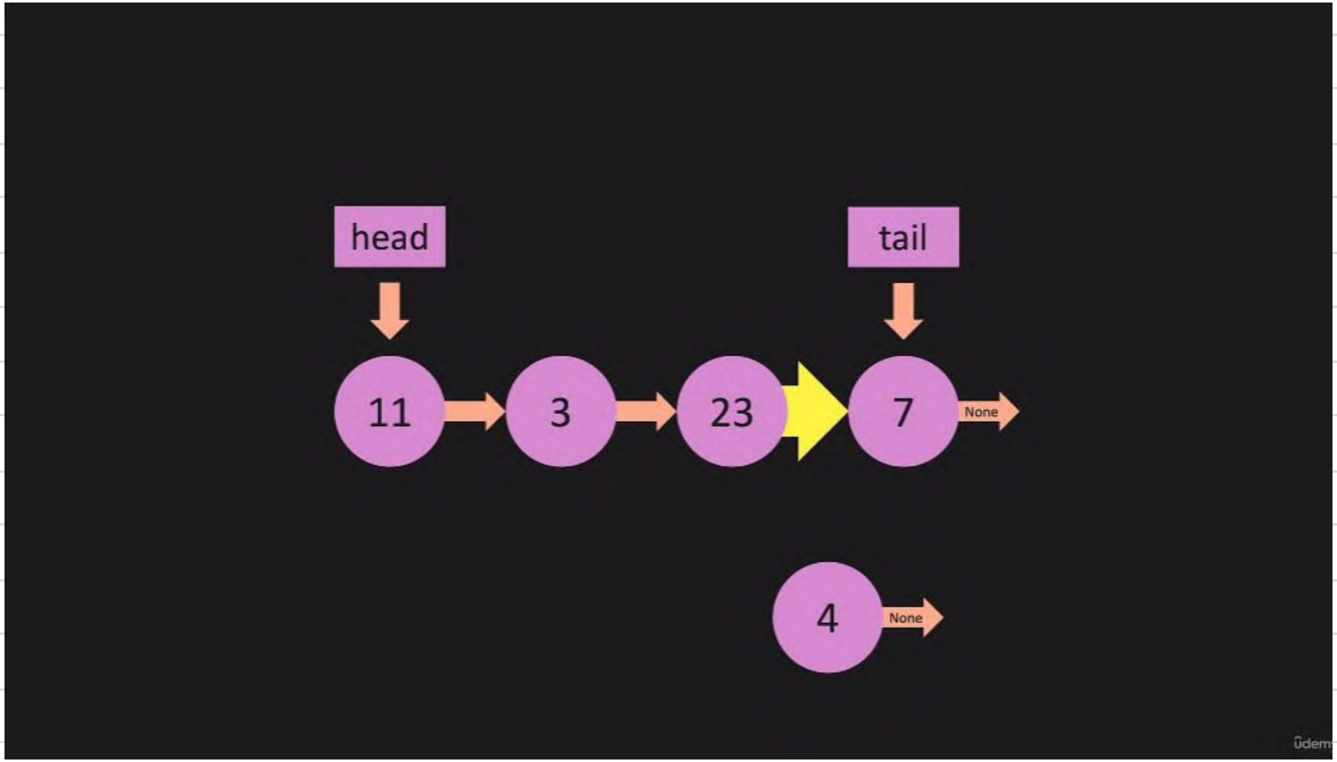
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



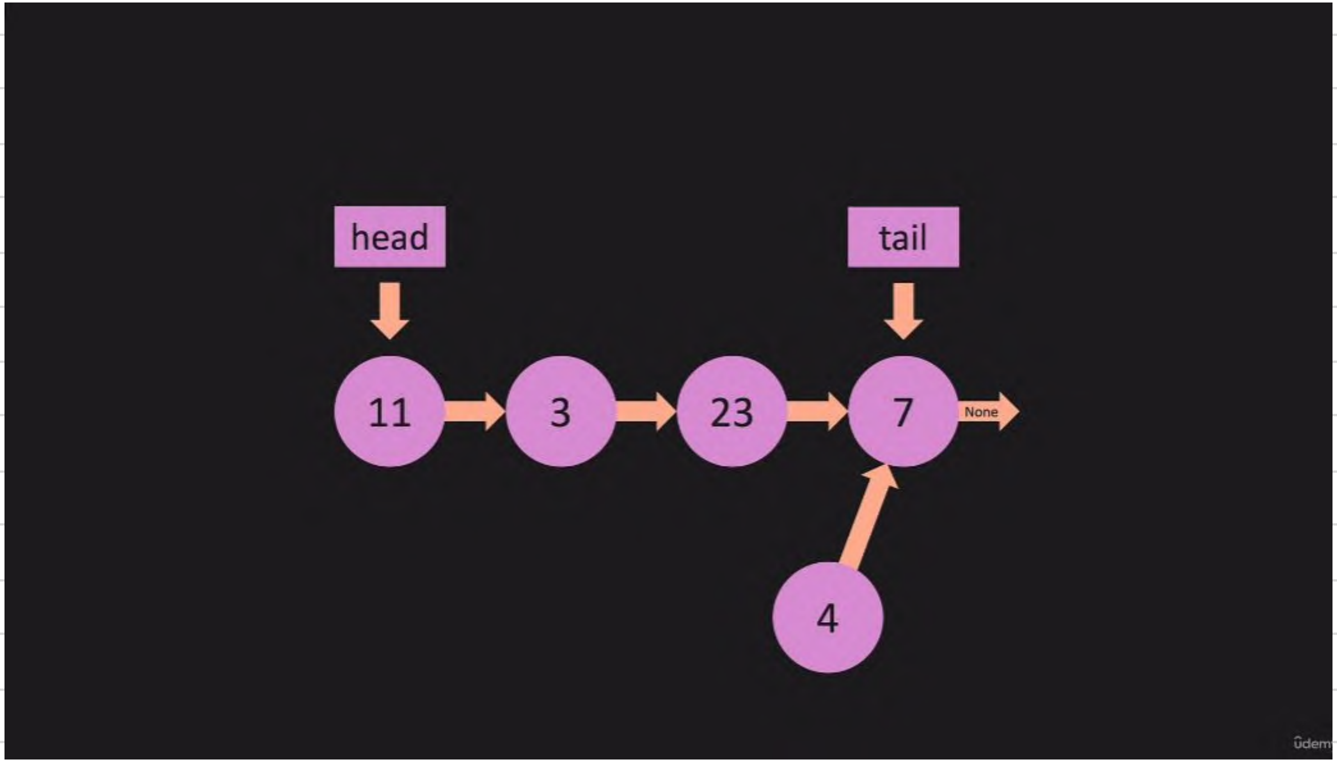
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



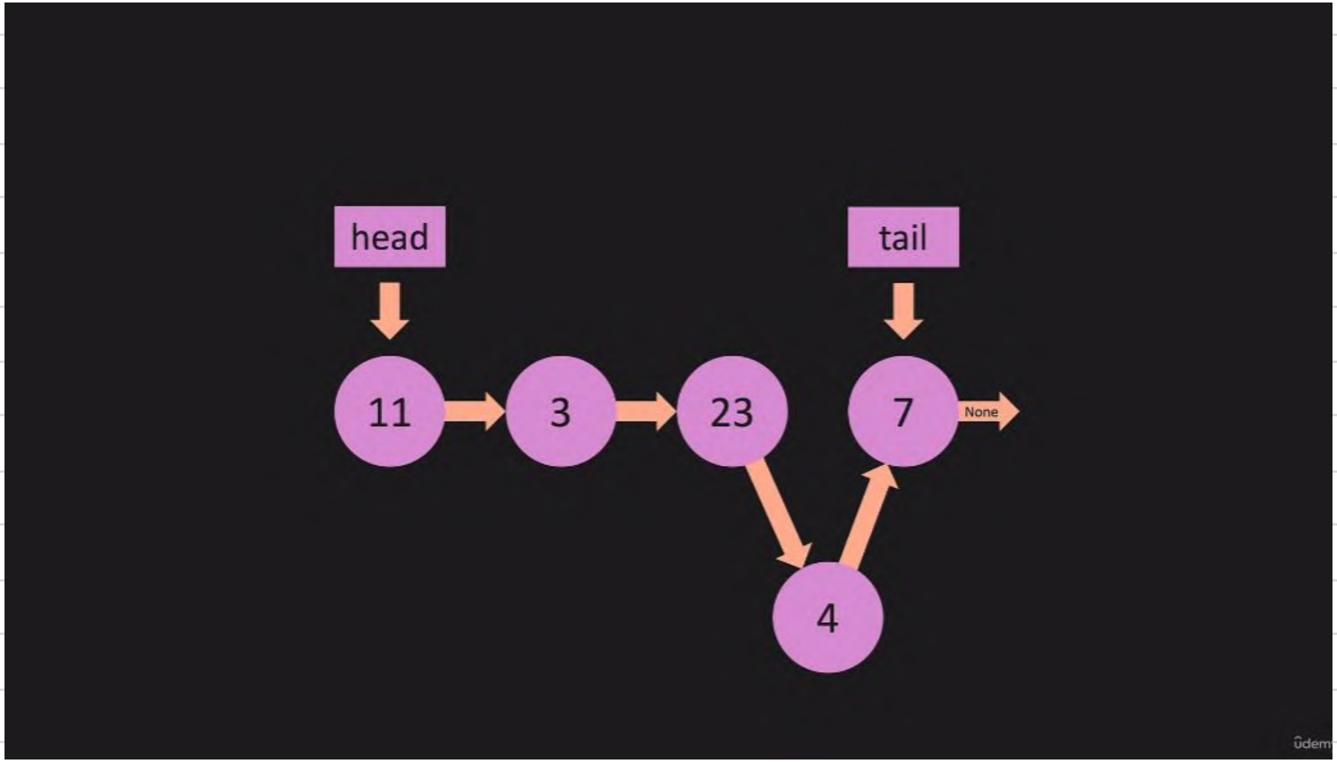
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



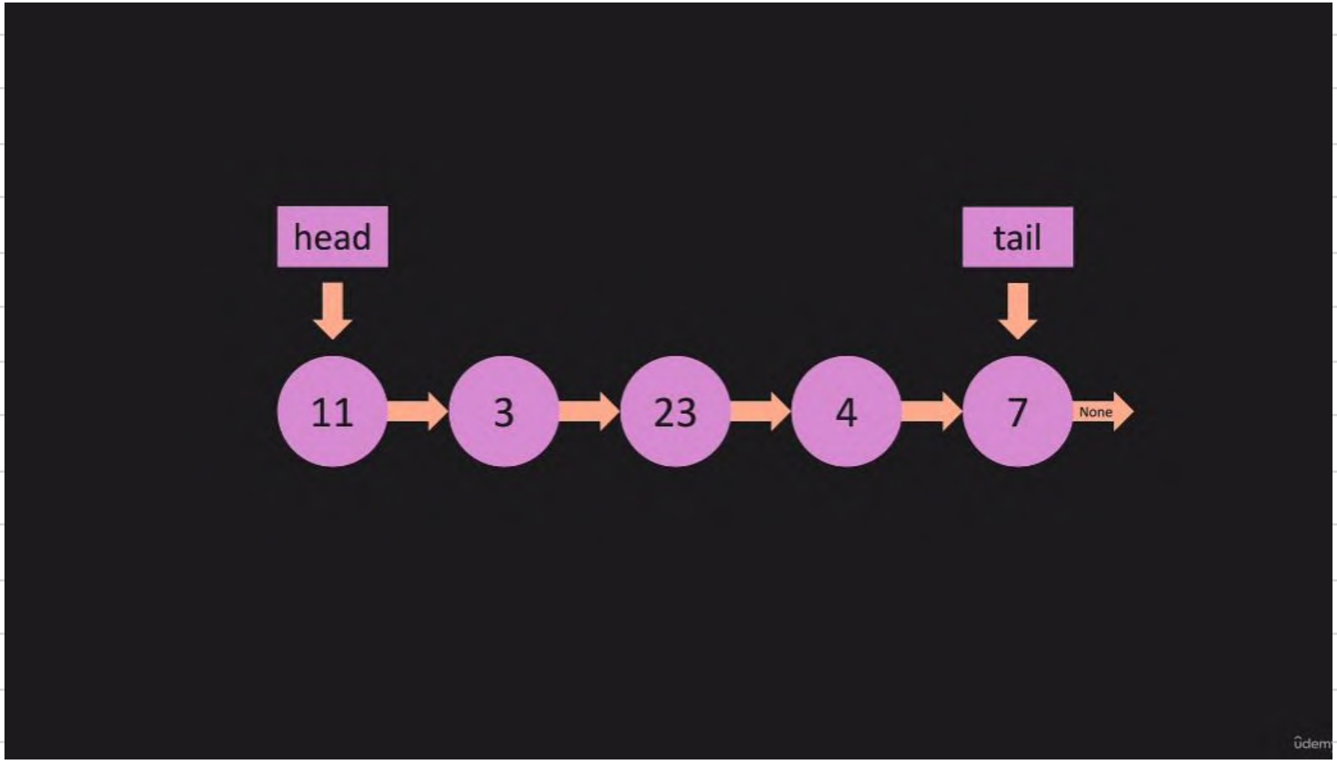
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



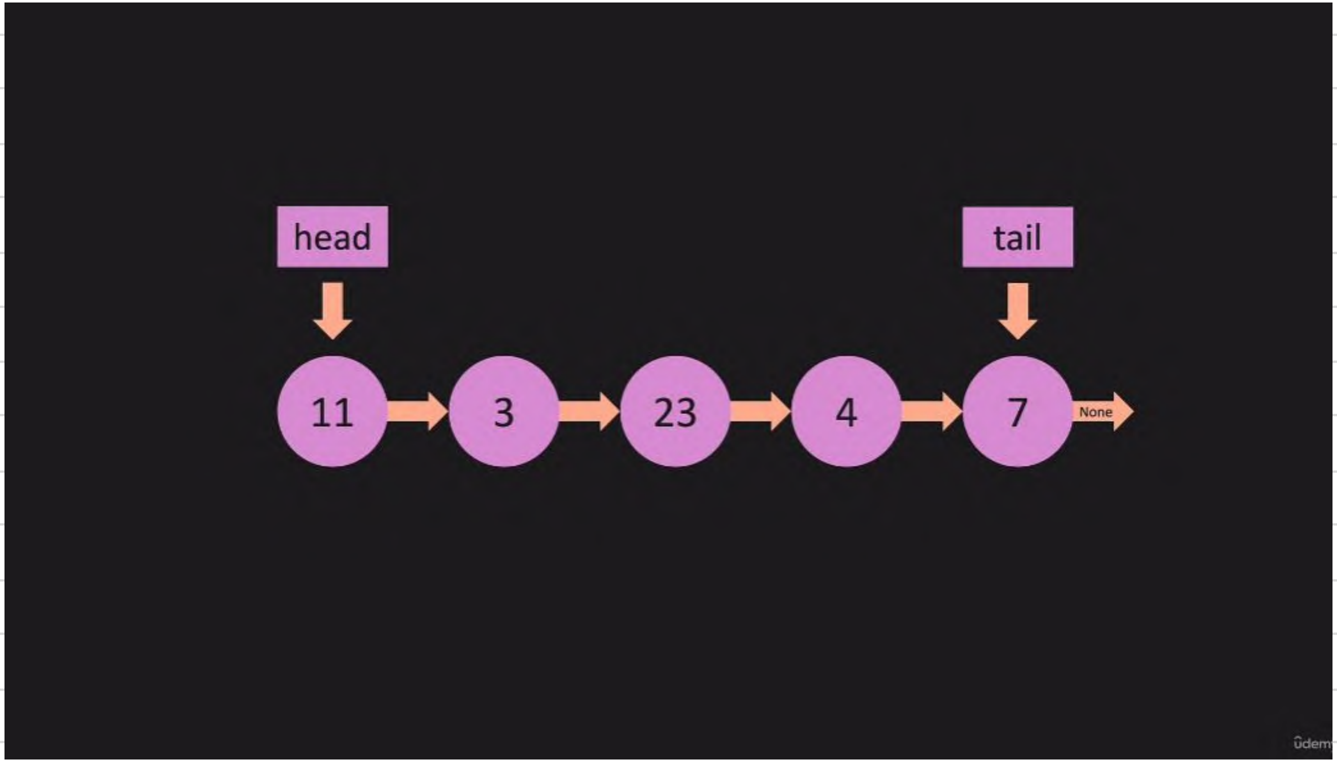
Add & Remove Node di Tengah

Bagaimana jika kita ingin menambahkan node “4” di setelah node “23”



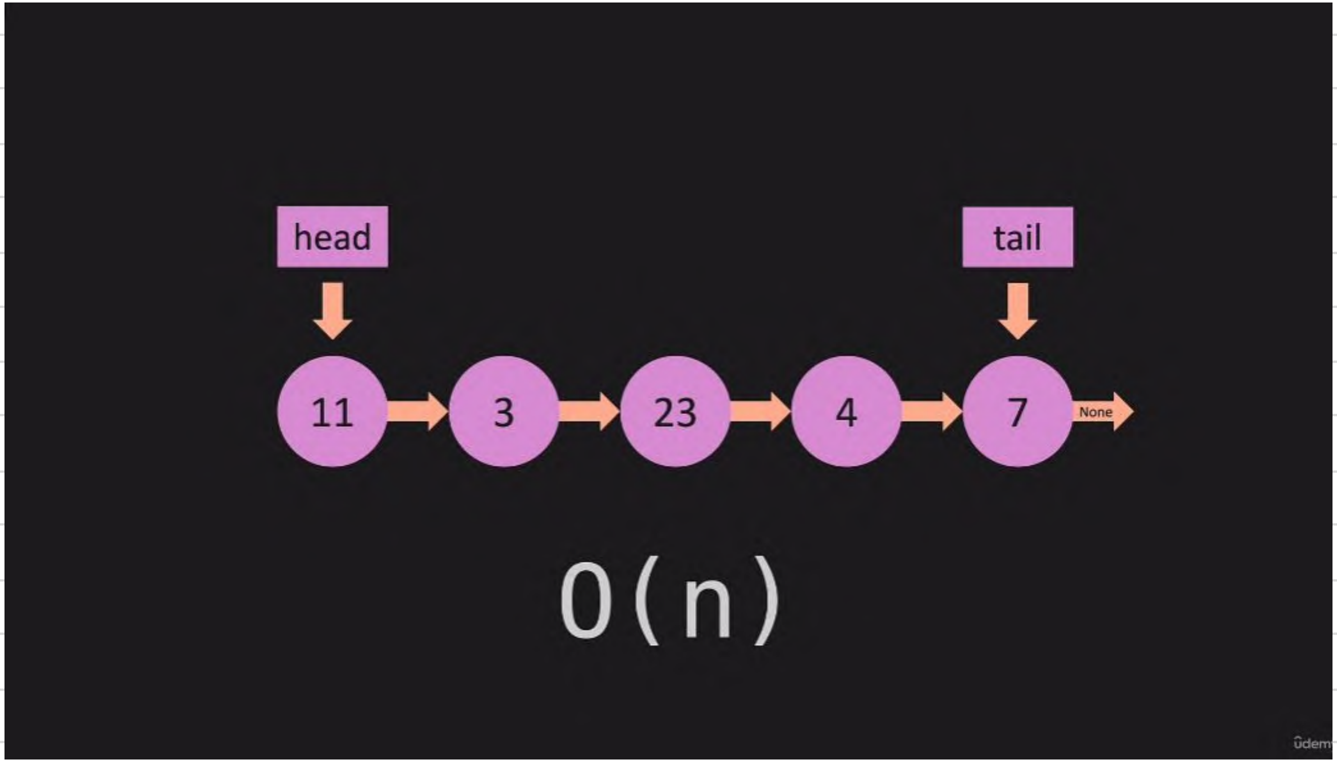
Add & Remove Node di Tengah

Apa notasi Big O untuk Insert Node … ?



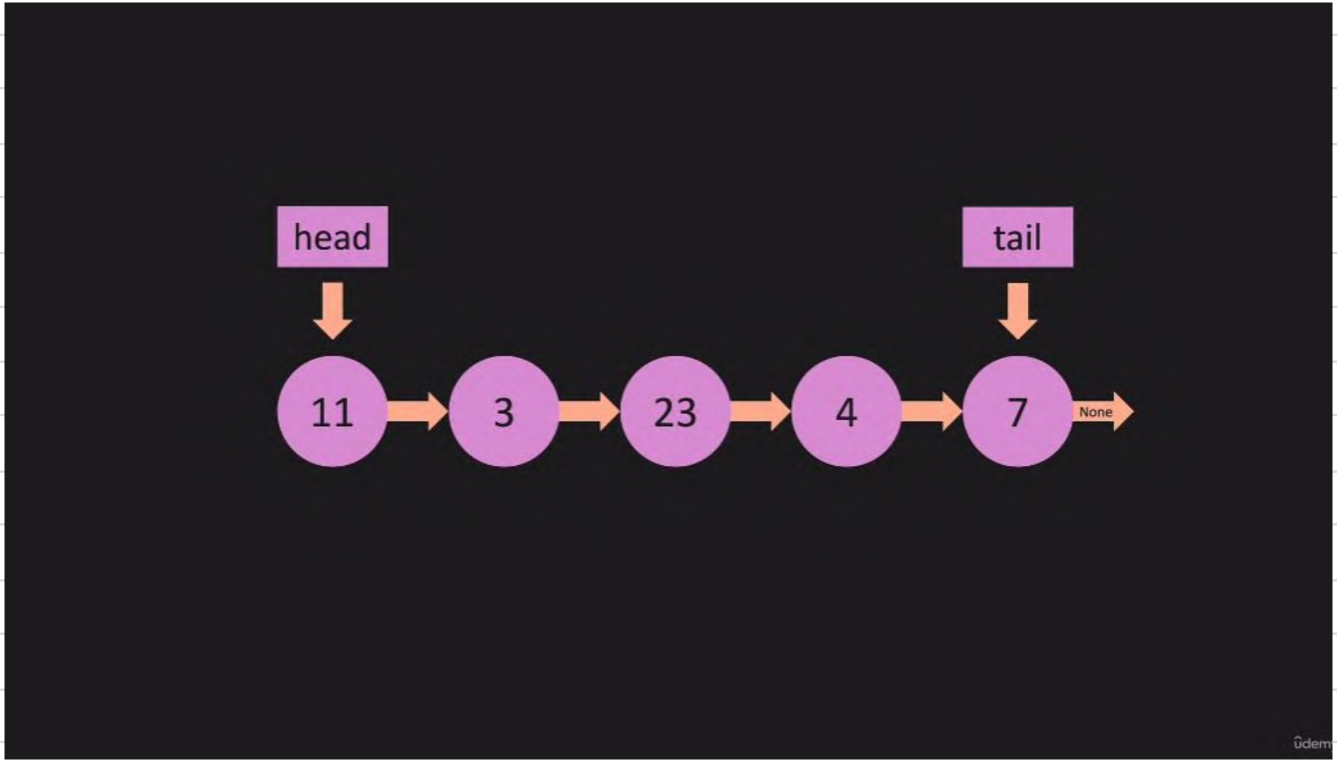
Add & Remove Node di Tengah

Apa notasi Big O untuk Insert Node ... ?



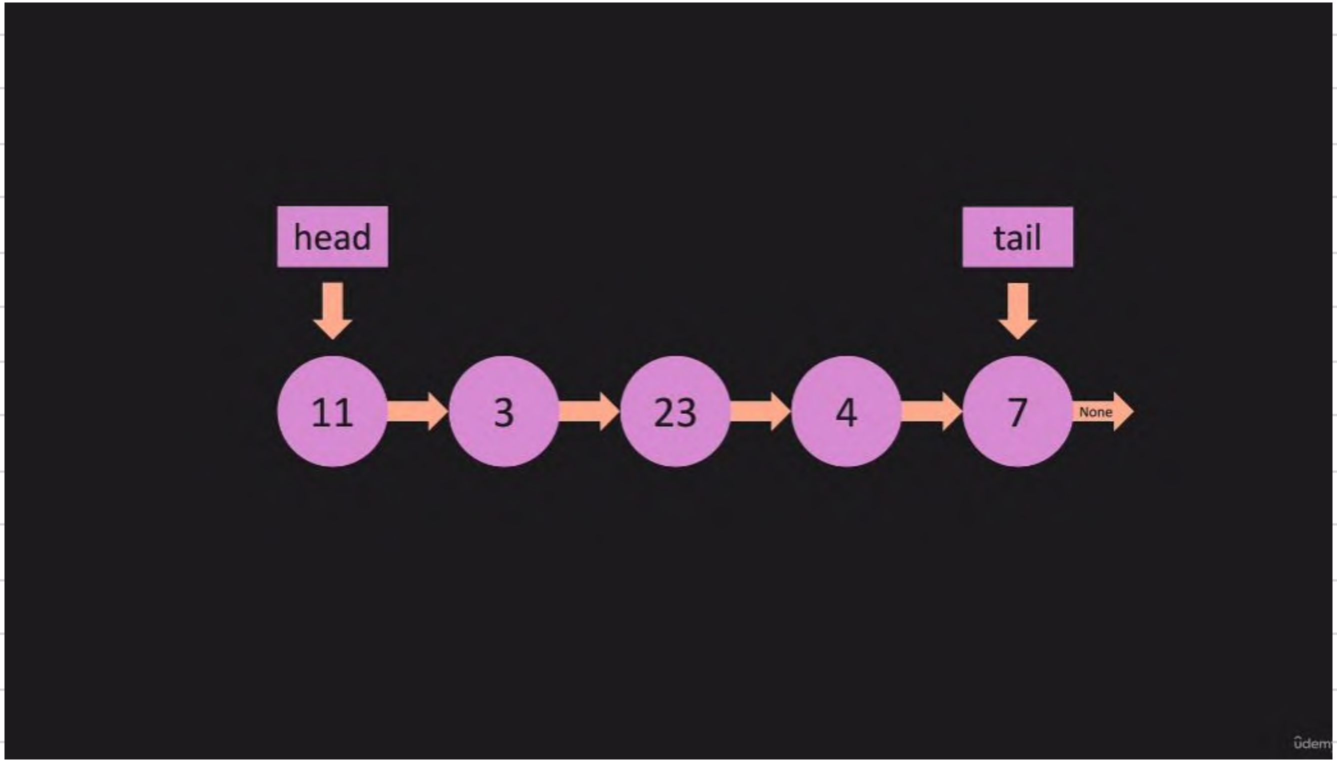
Diketahui suatu Linked List...

Add & Remove Node di Tengah



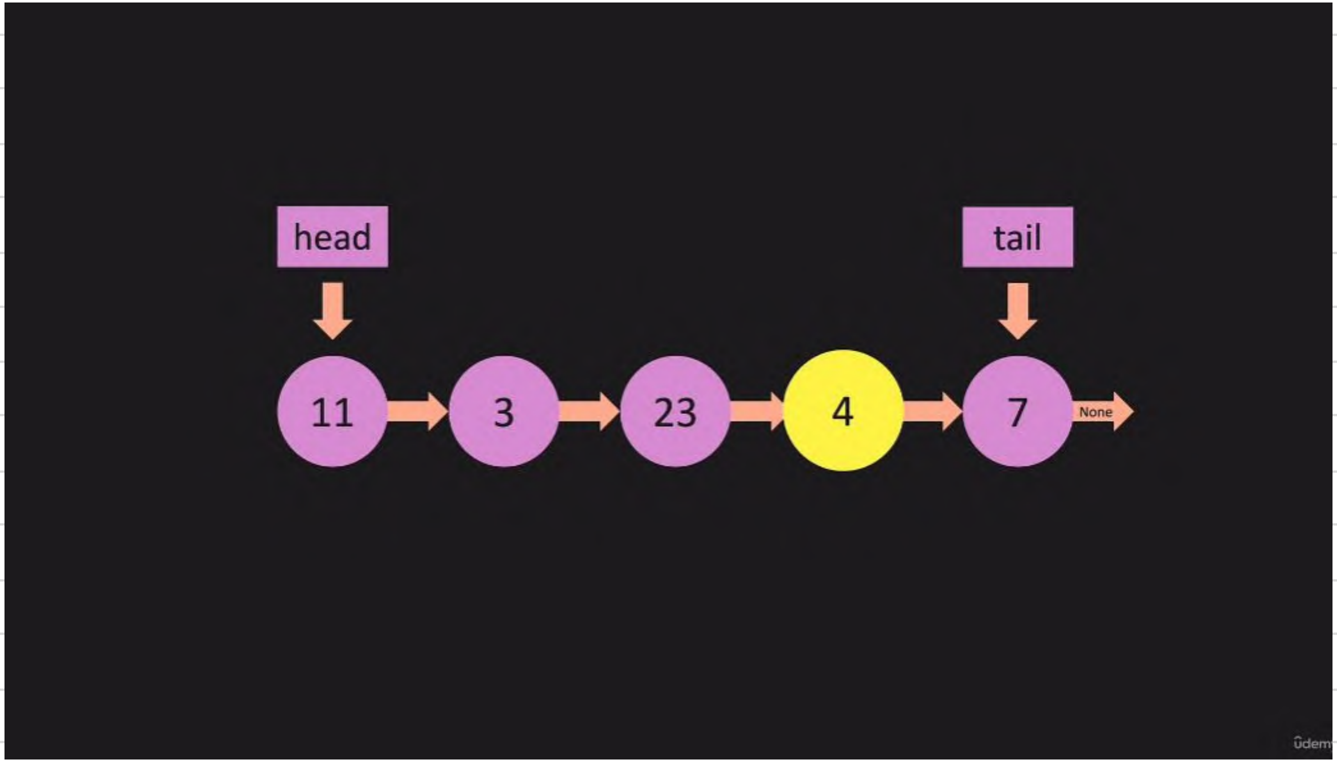
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



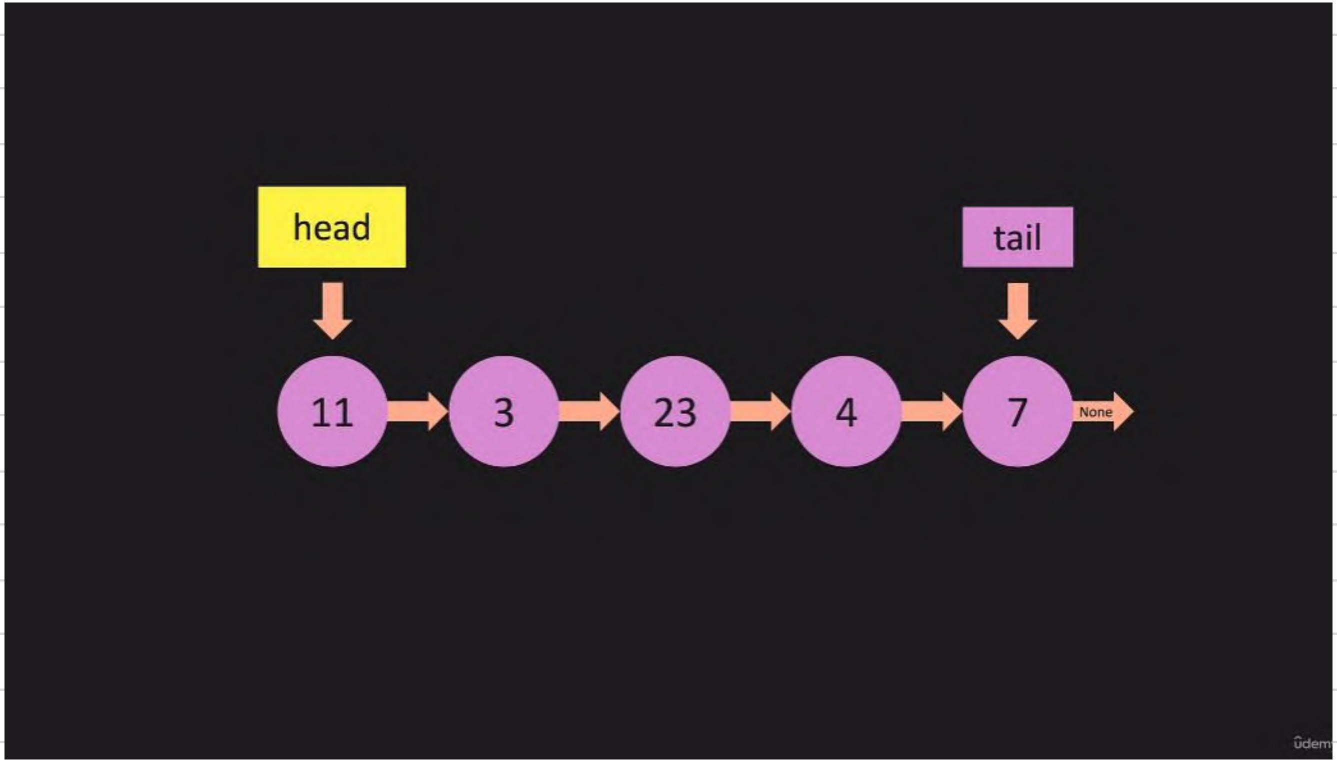
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



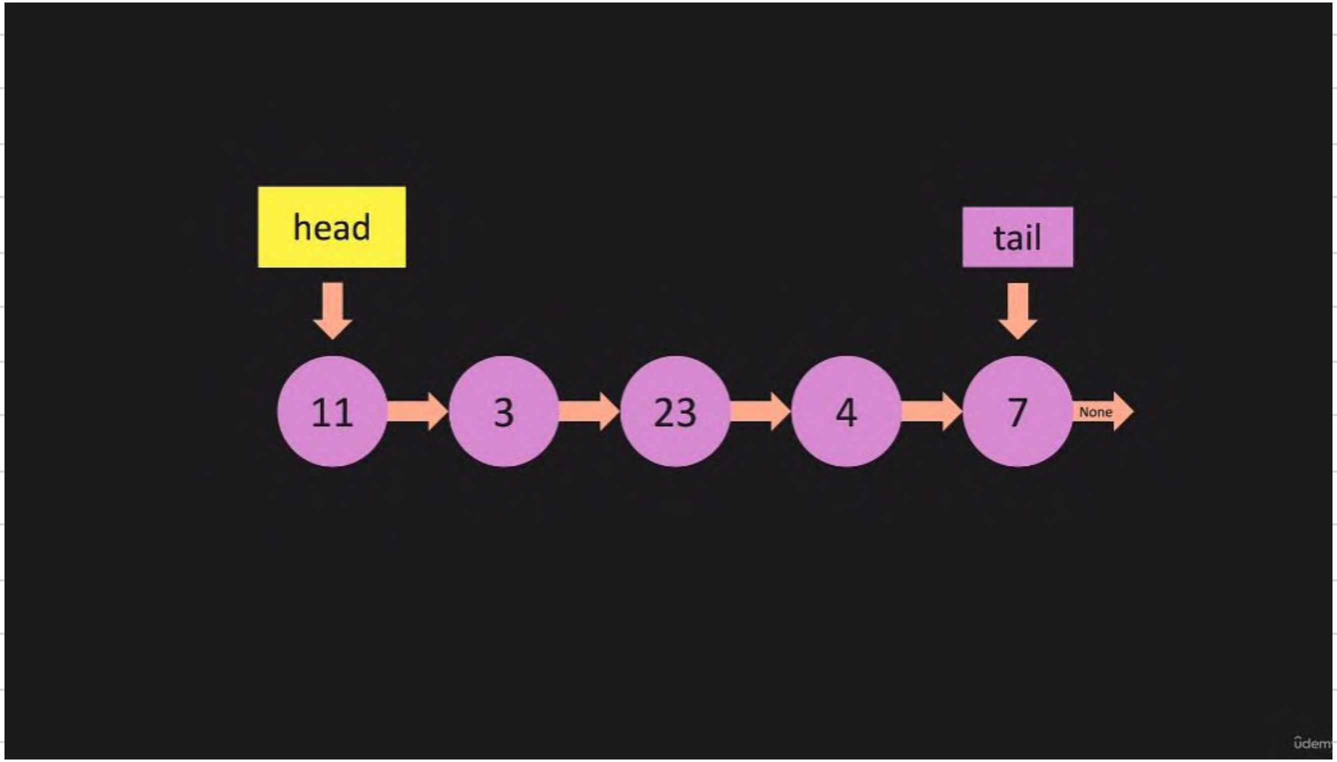
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



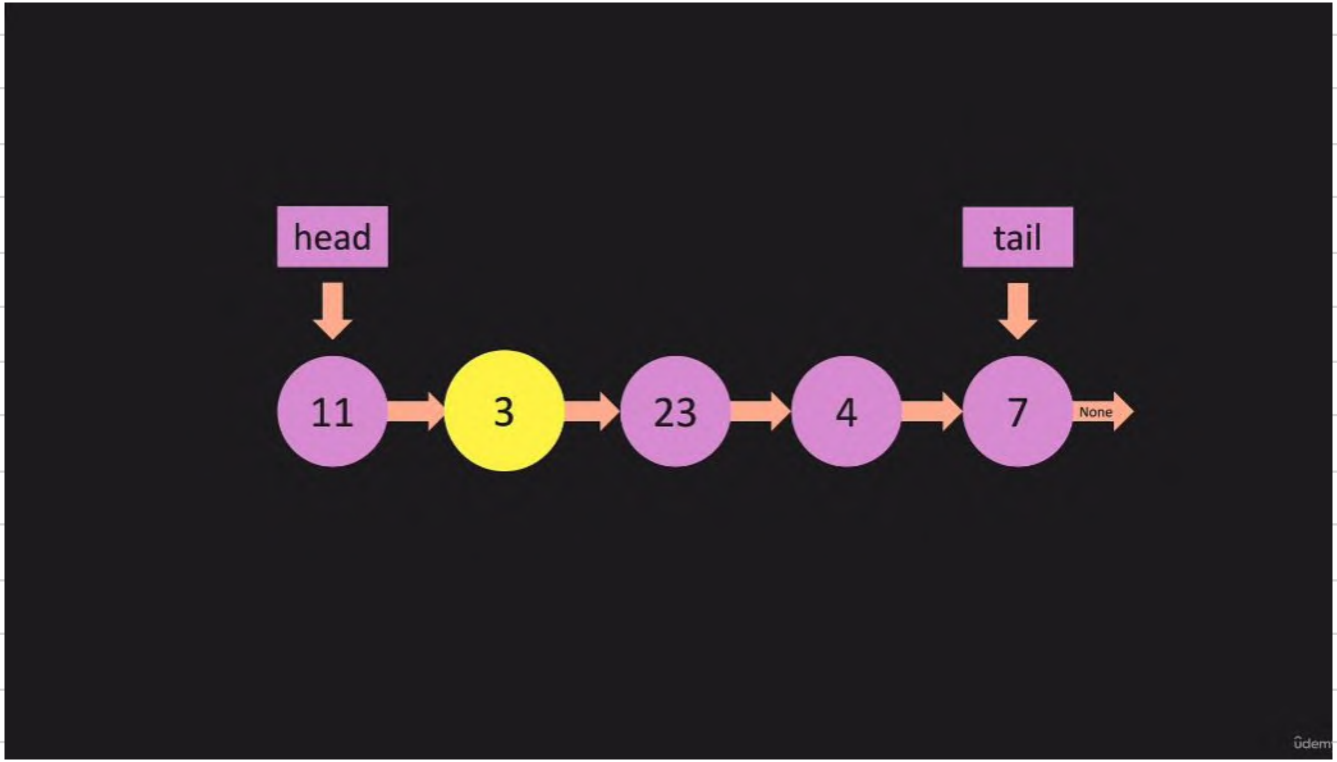
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



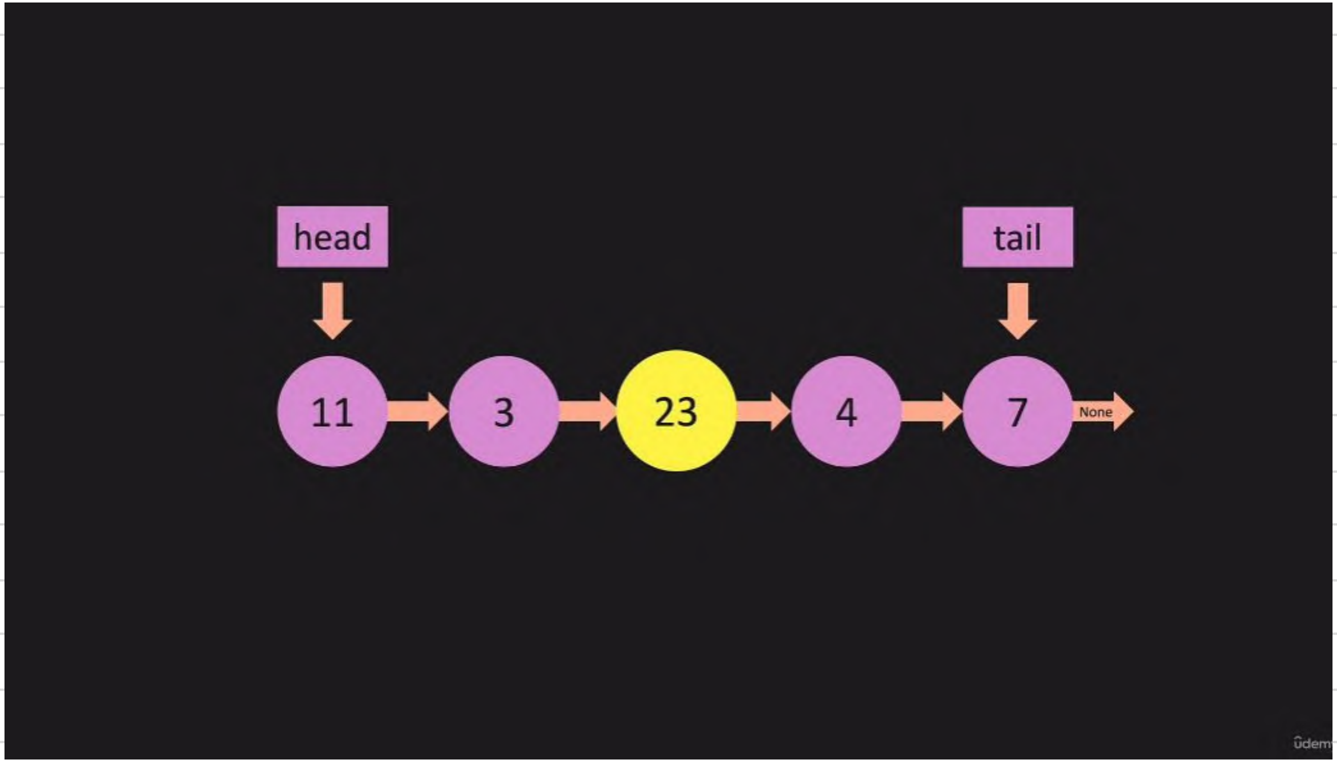
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



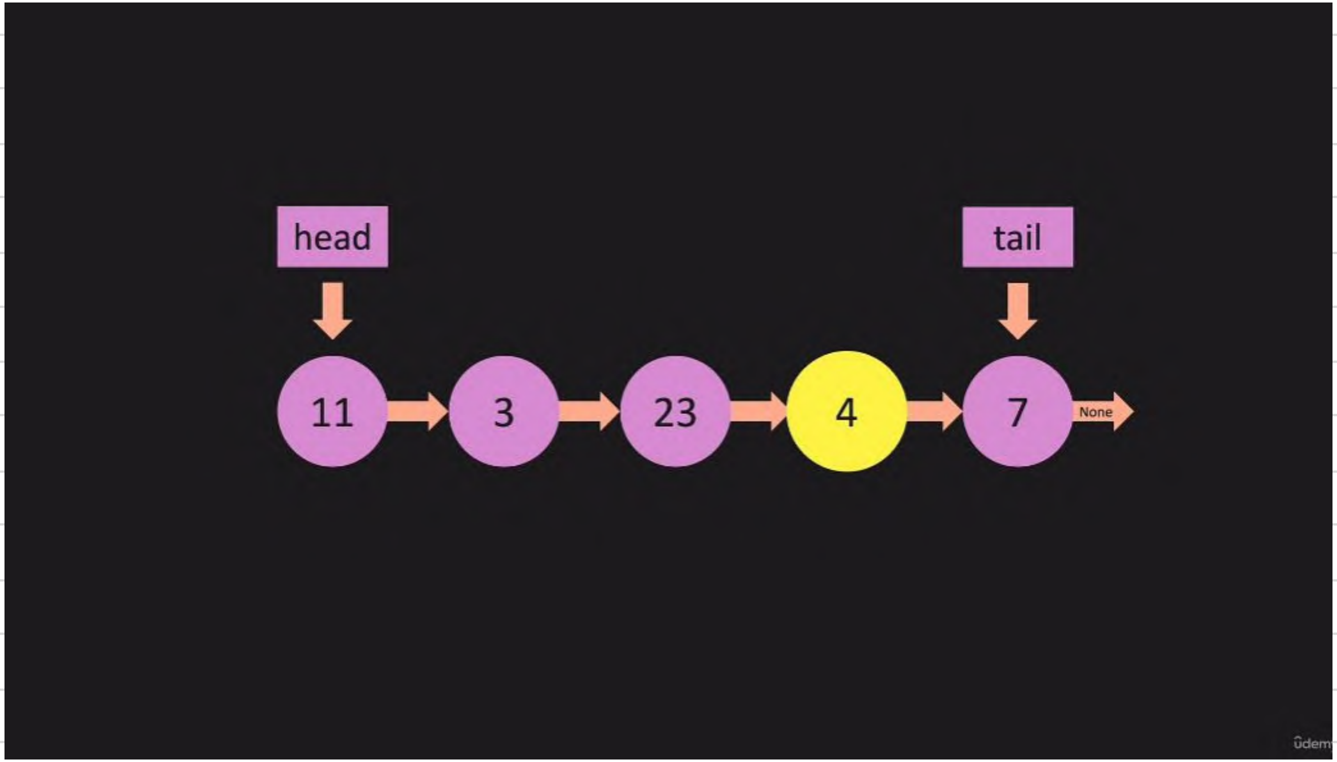
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



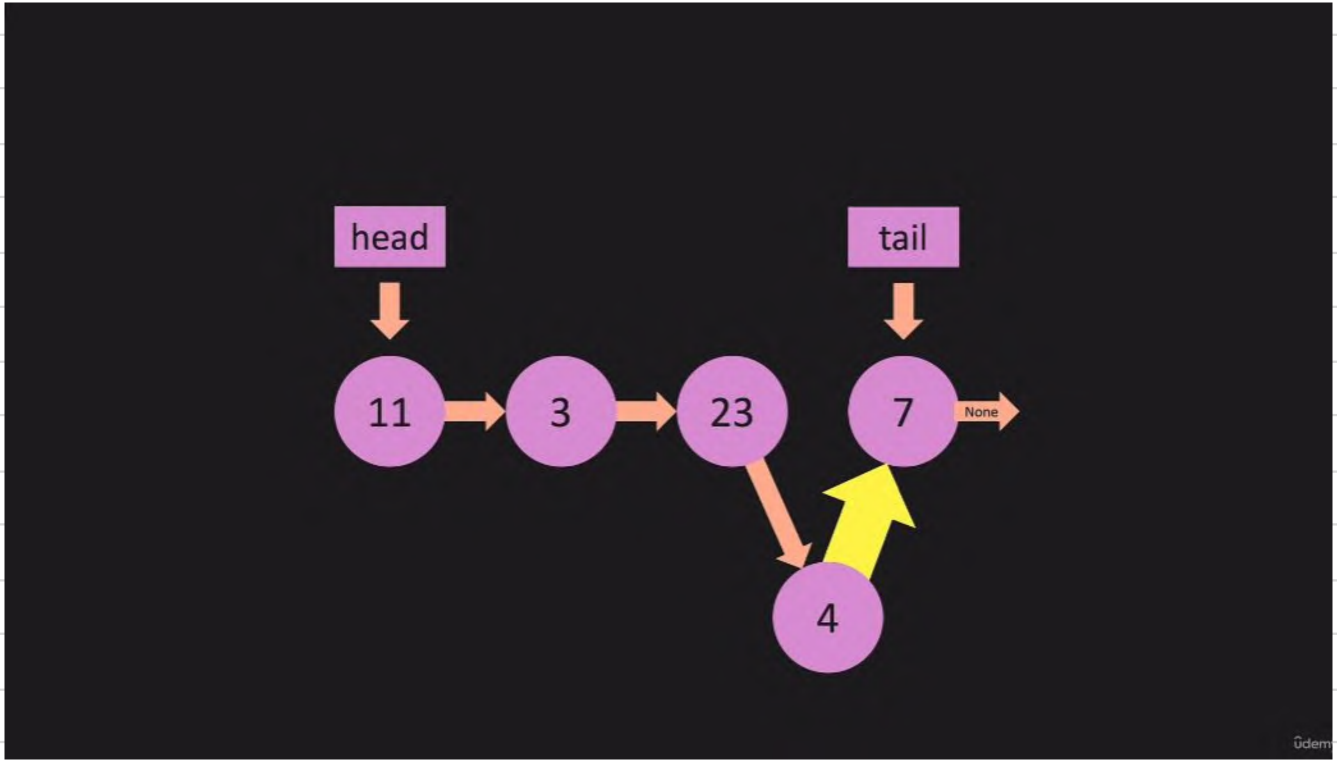
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



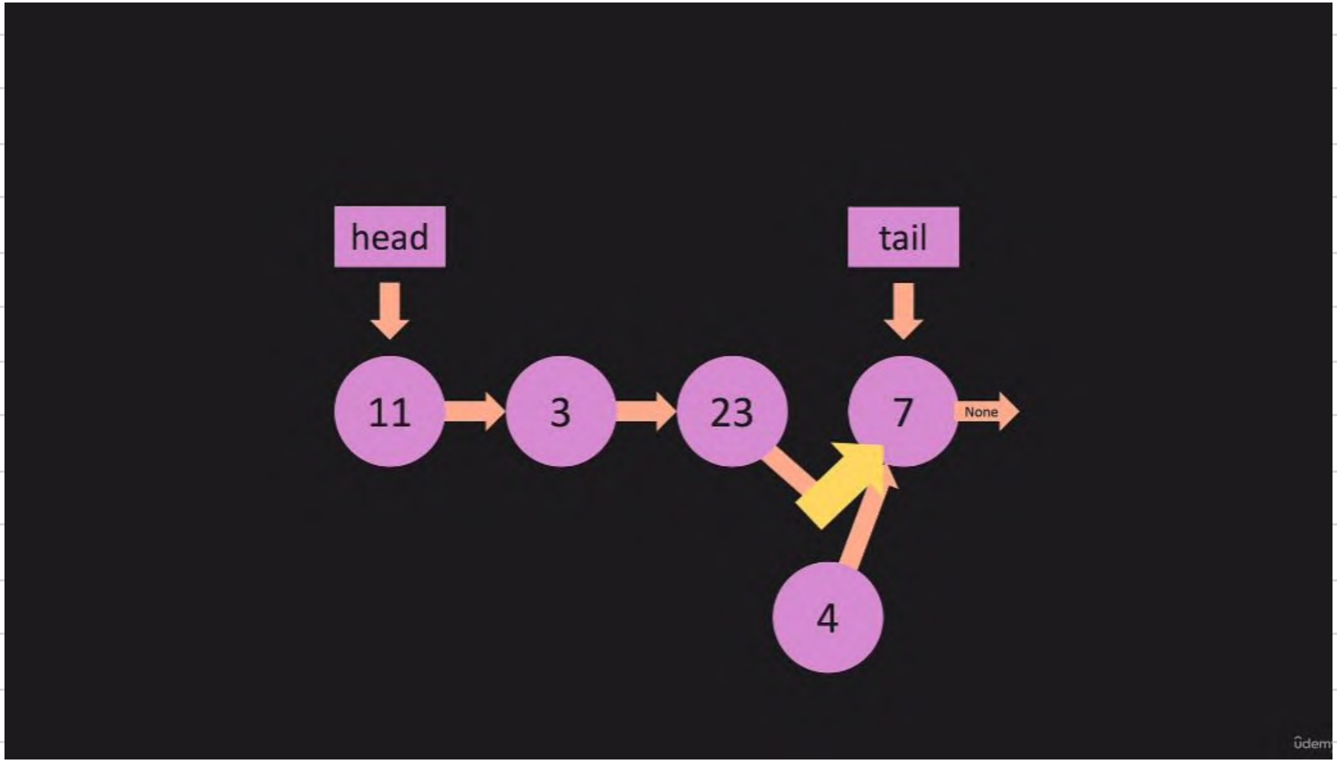
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



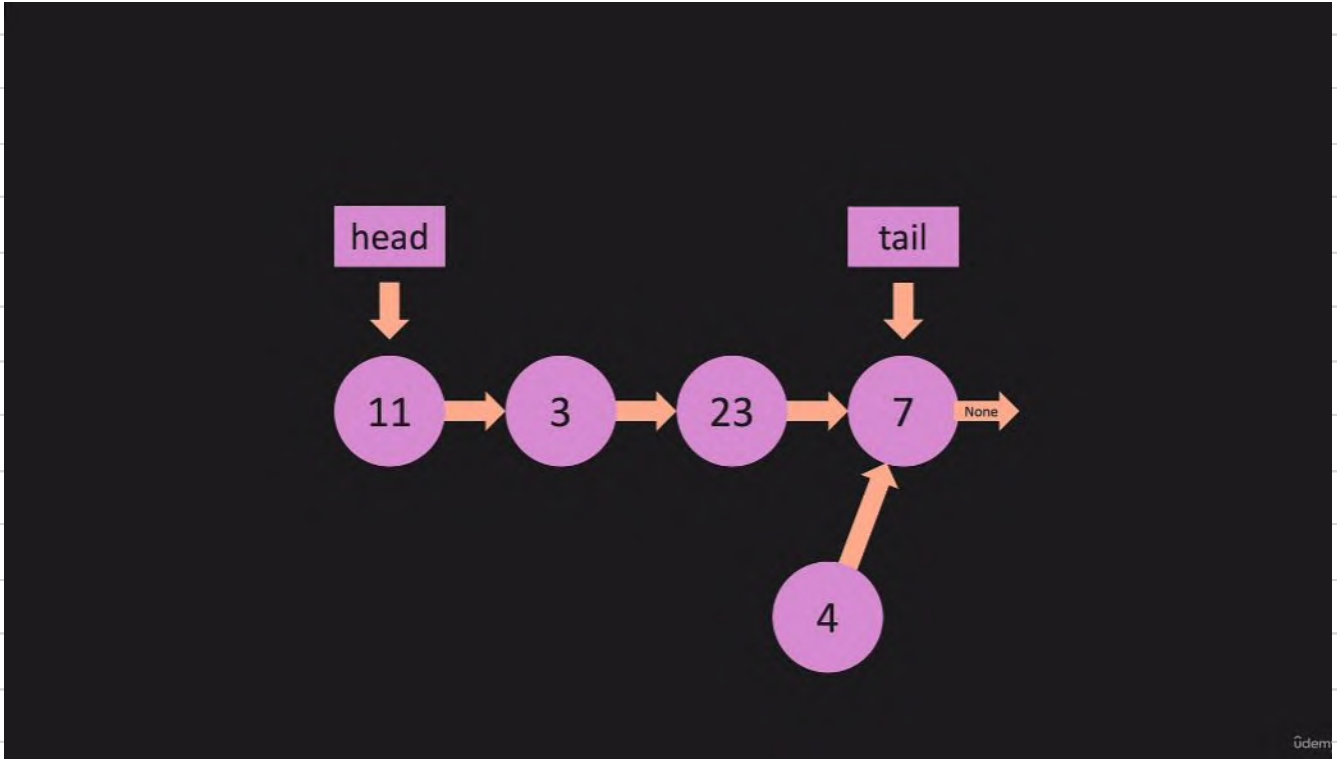
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



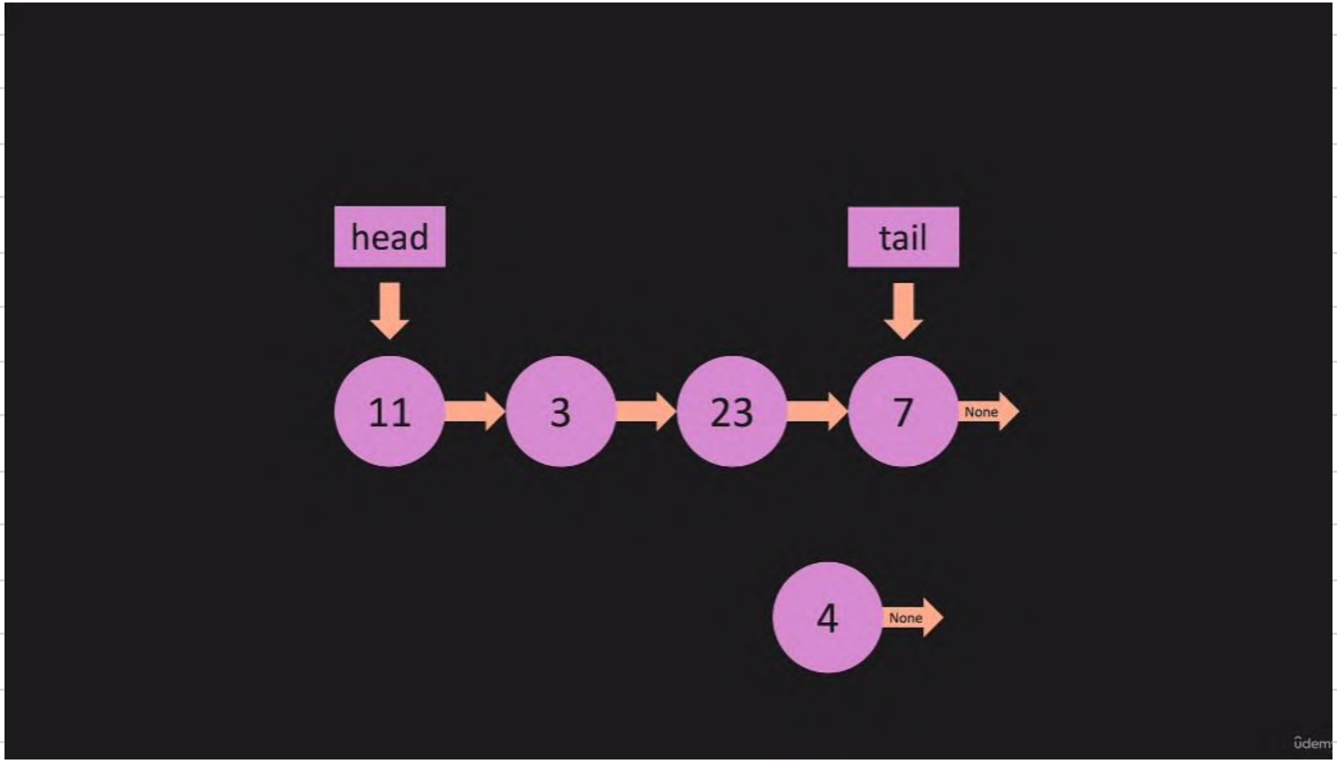
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



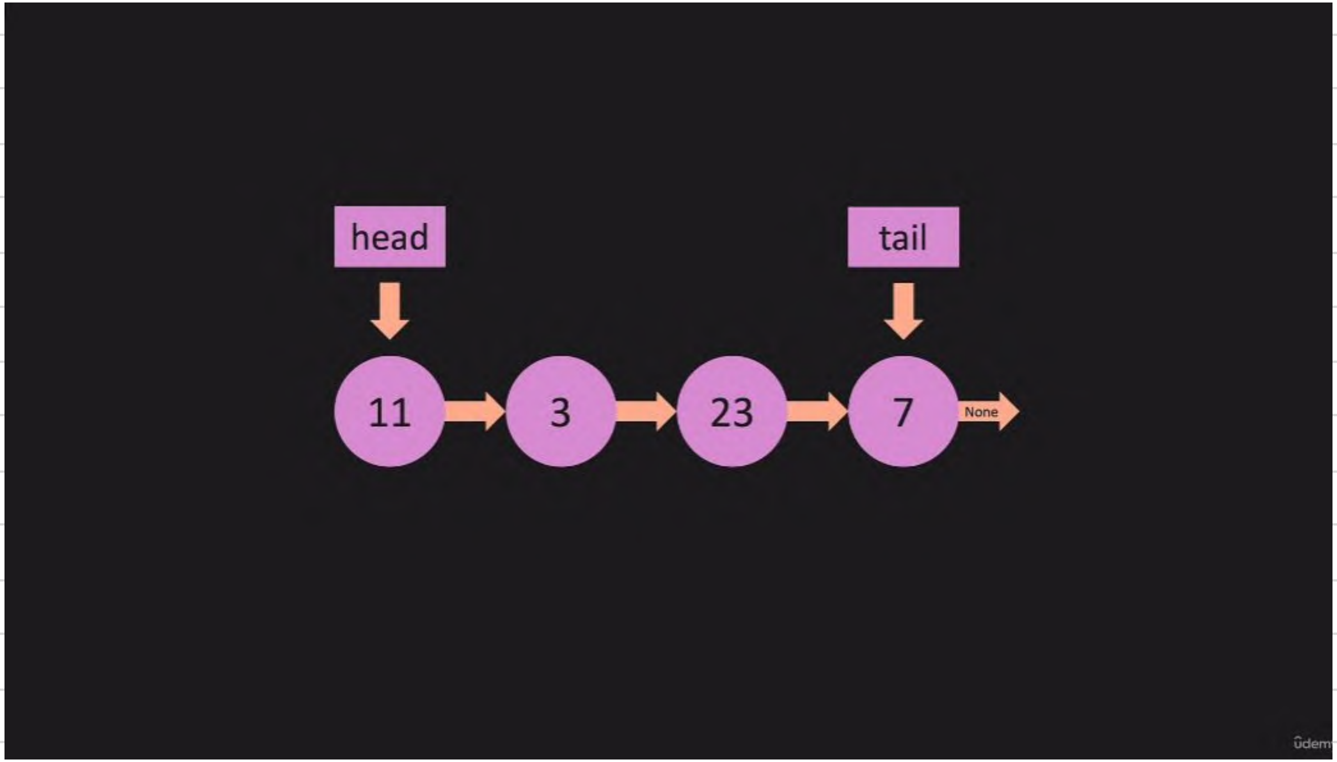
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node "4" dari linked list



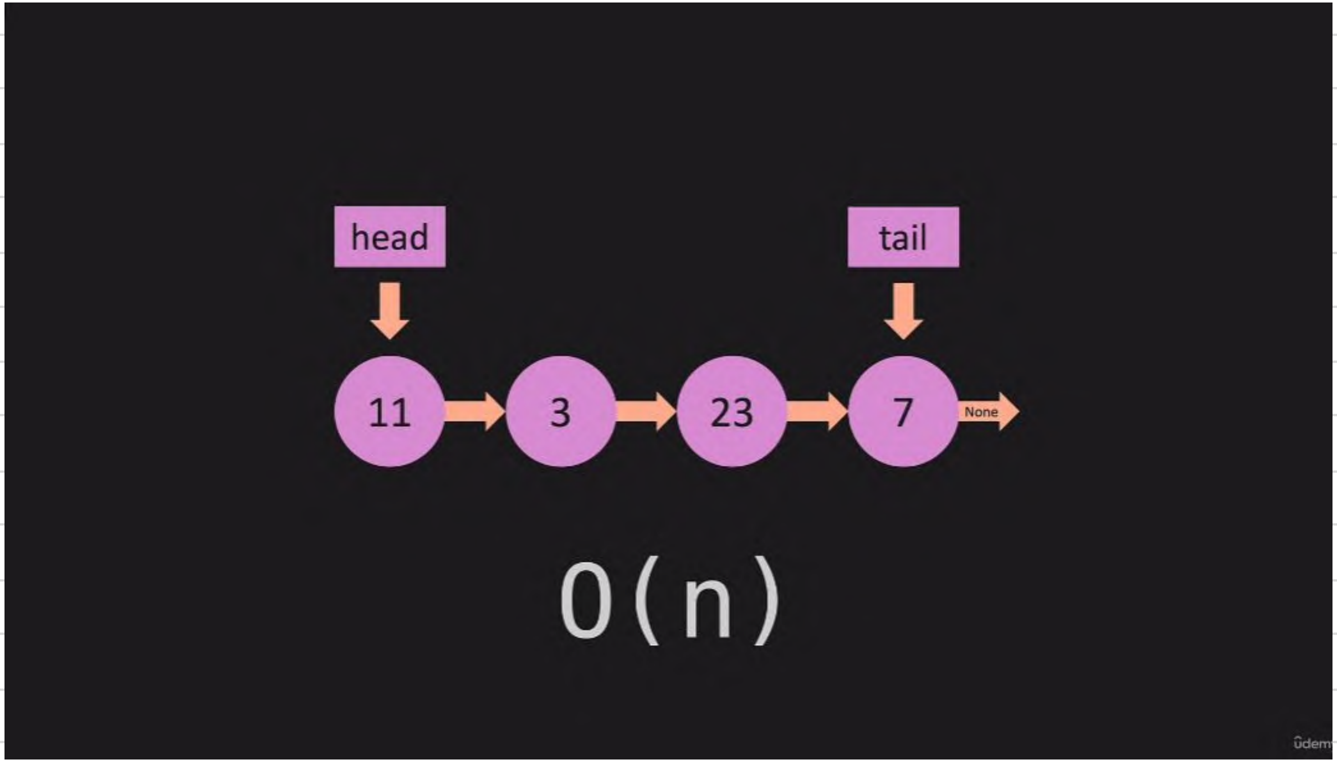
Add & Remove Node di Tengah

Bagaimana jika kita ingin menghapus node “4” dari linked list



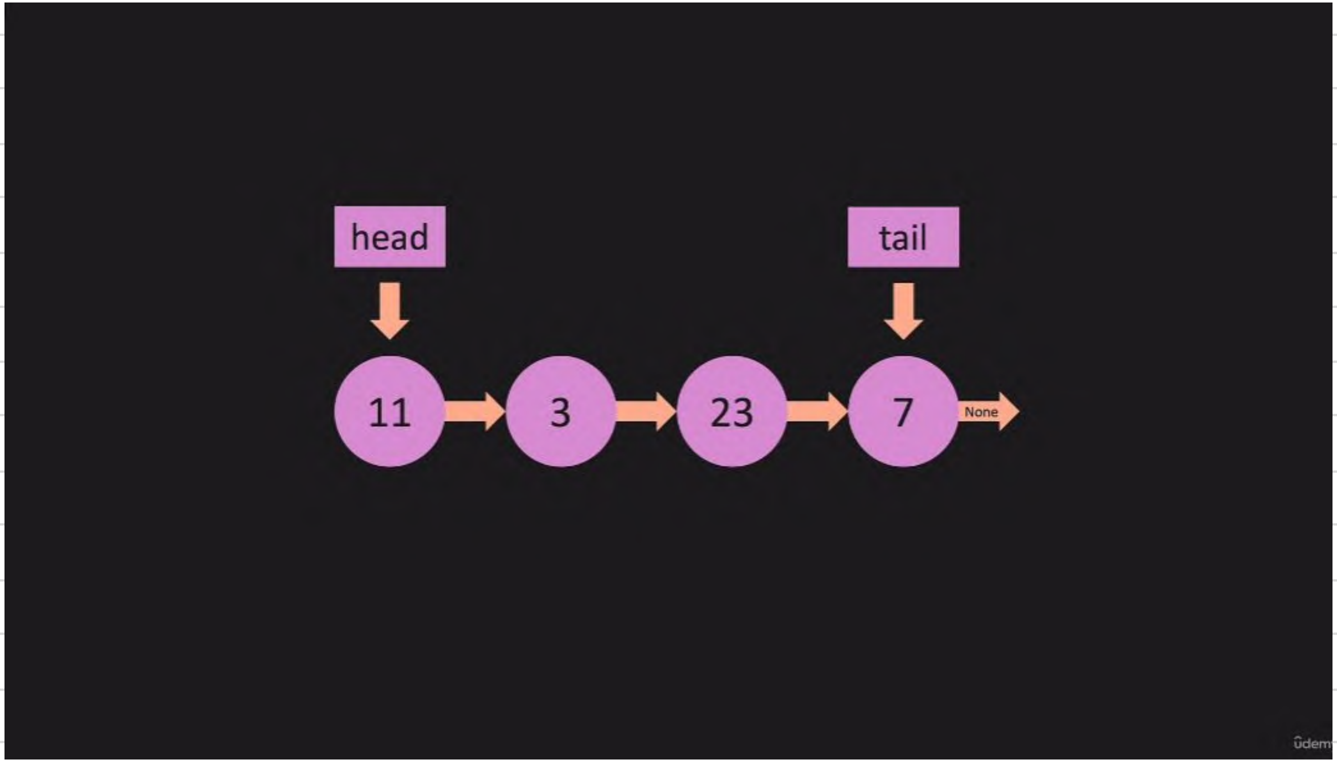
Add & Remove Node di Tengah

Apa notasi Big O untuk
Remove Node di Tengah... ?



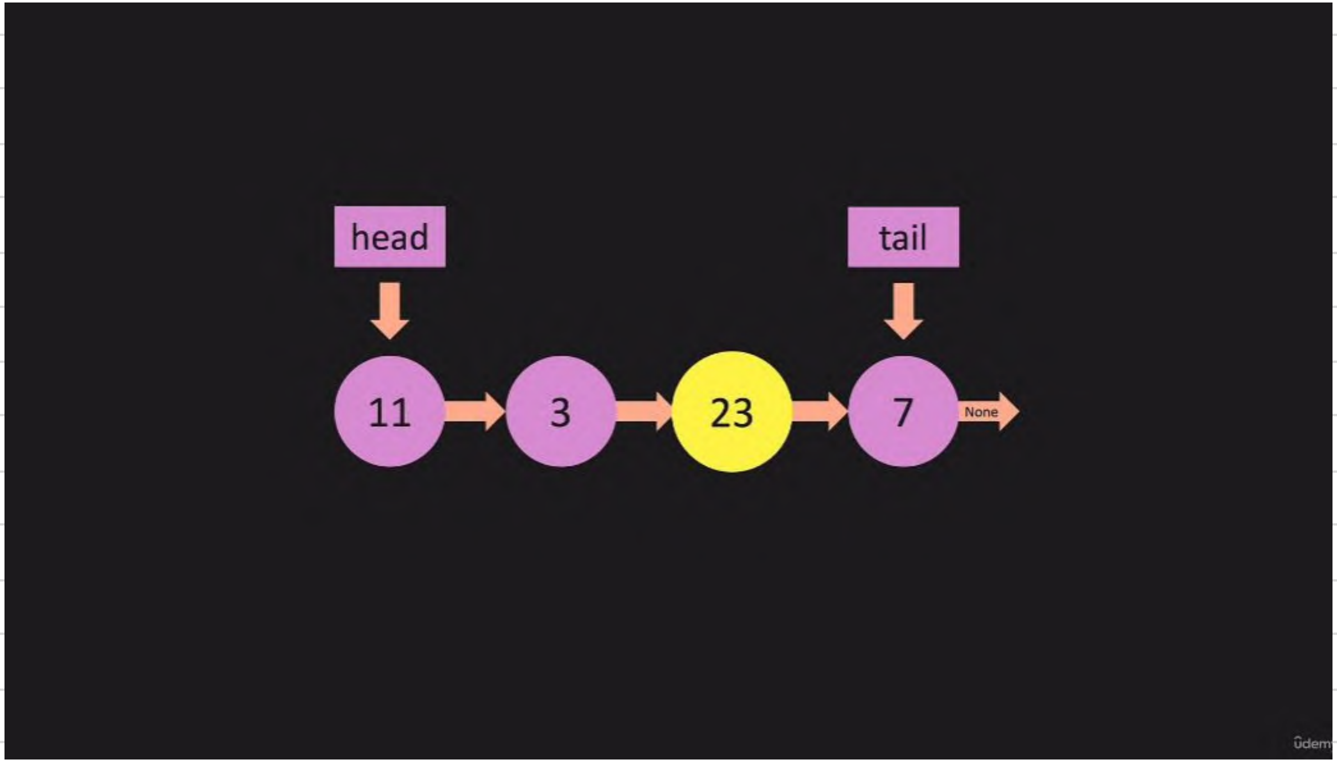
Diketahui suatu Linked List...

Search Node



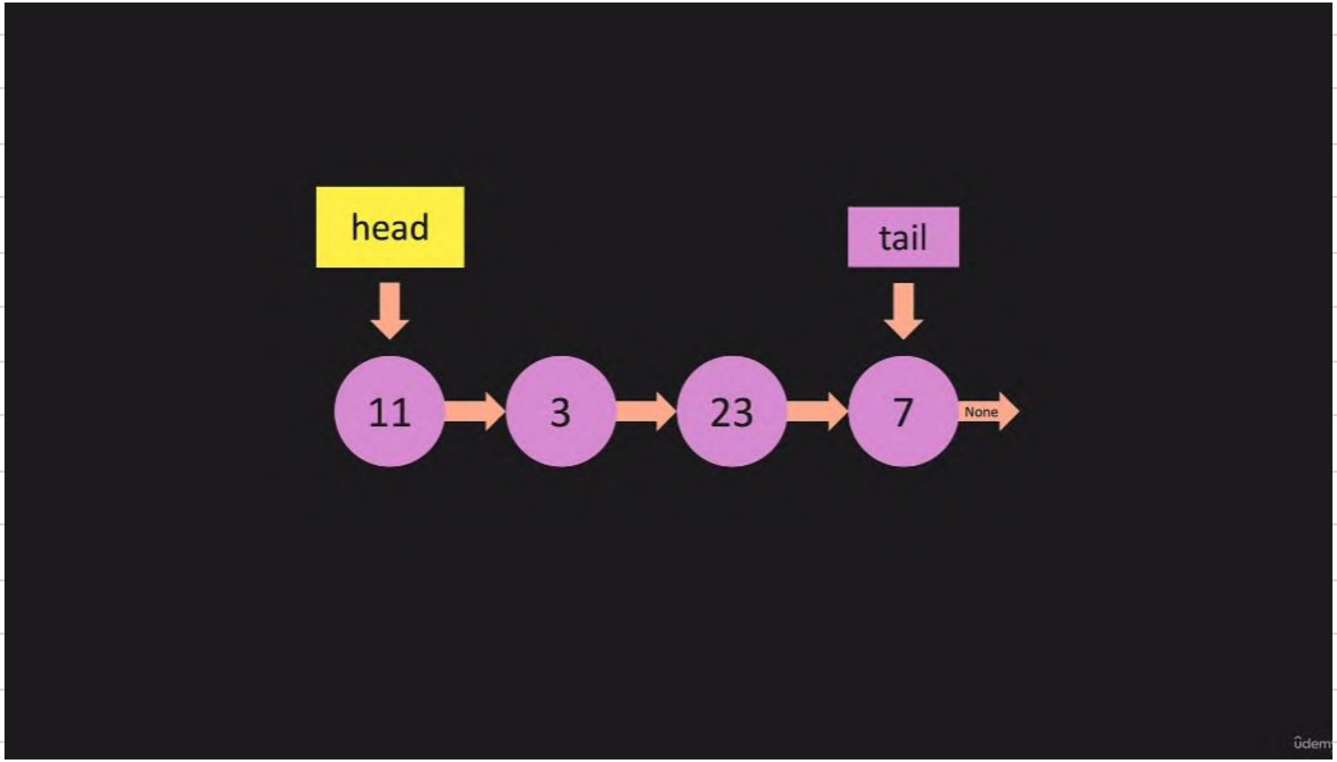
Search Node

Bagaimana jika kita ingin mencari node “23” di dalam linked list



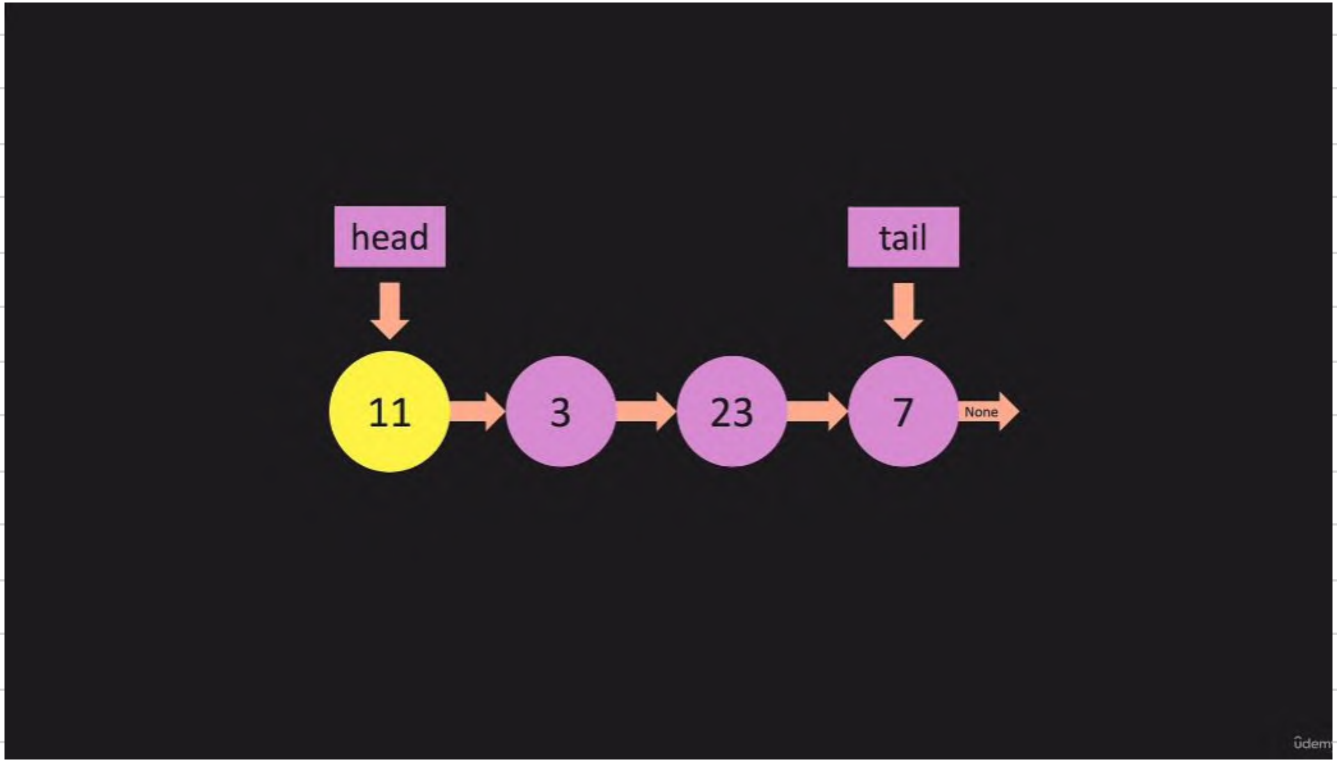
Search Node

Bagaimana jika kita ingin mencari node “23” di dalam linked list



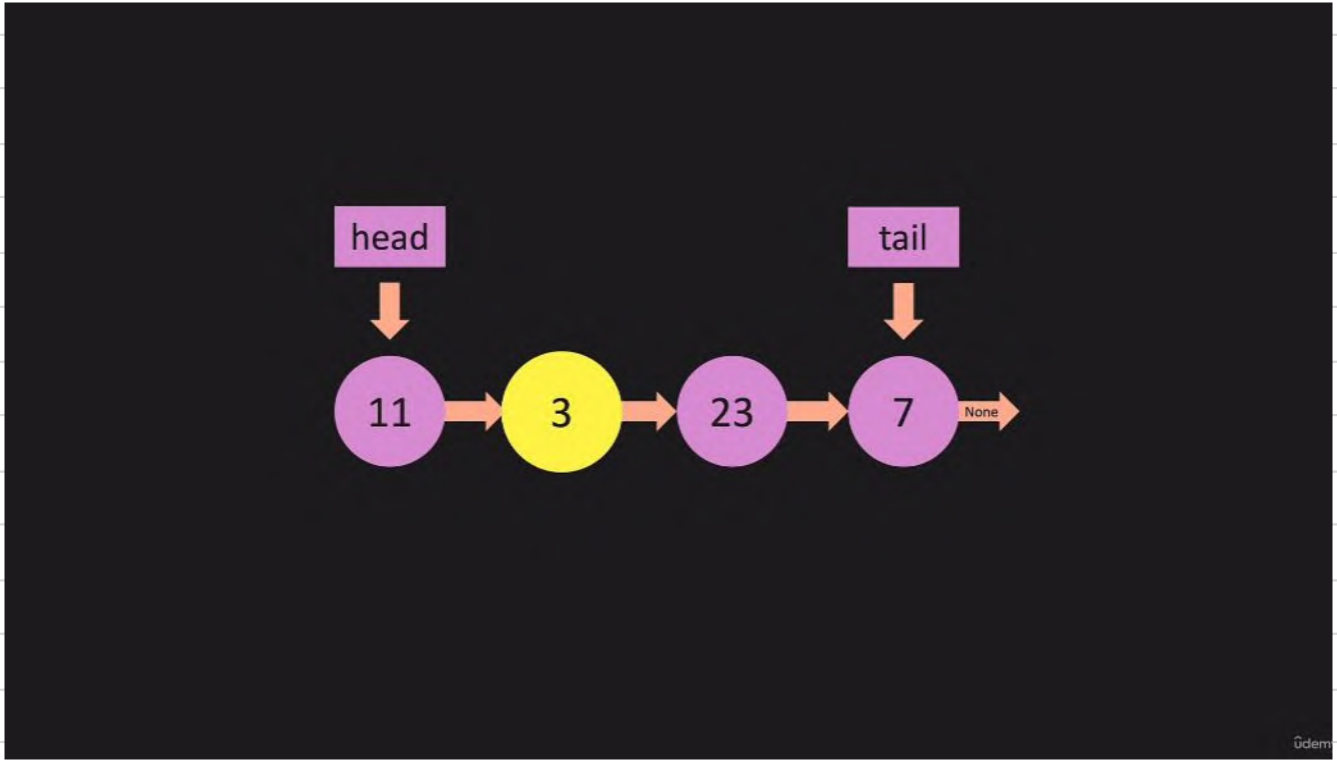
Search Node

Bagaimana jika kita ingin mencari node “23” di dalam linked list



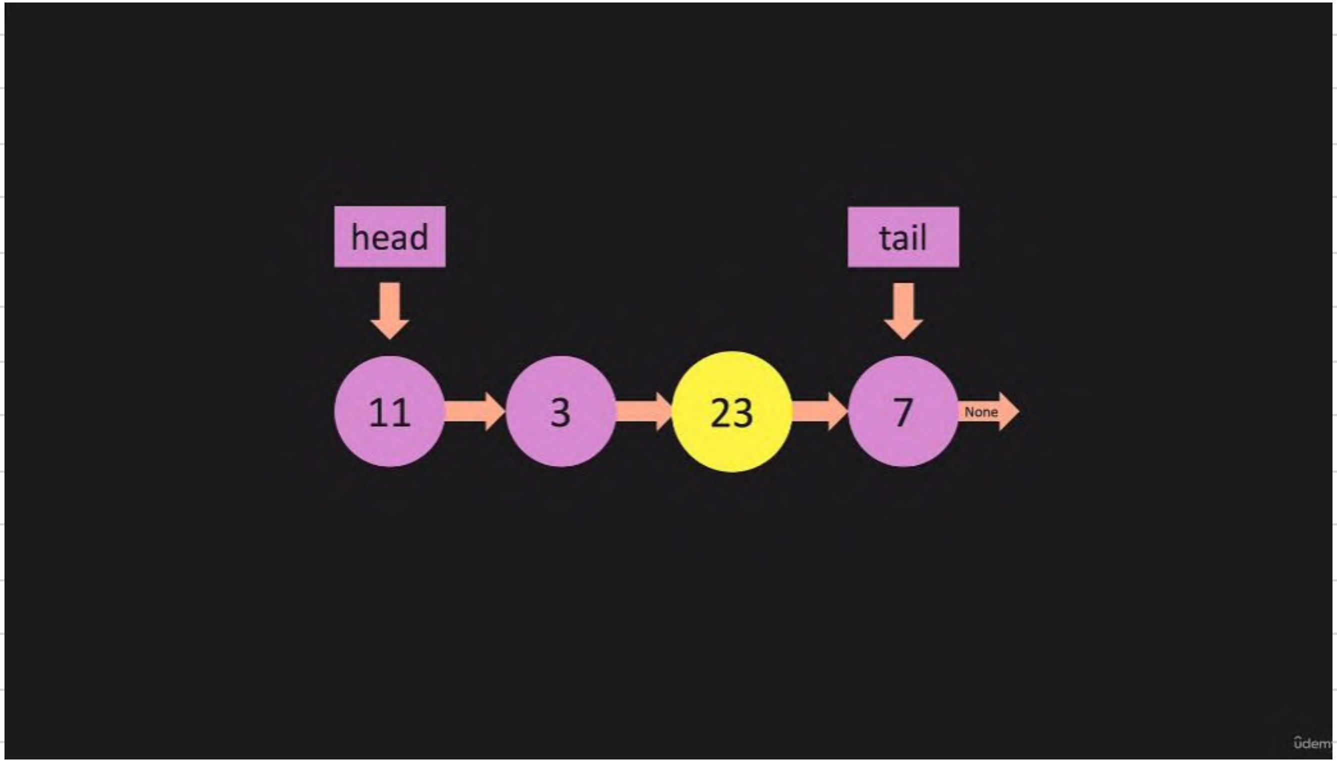
Search Node

Bagaimana jika kita ingin mencari node “23” di dalam linked list



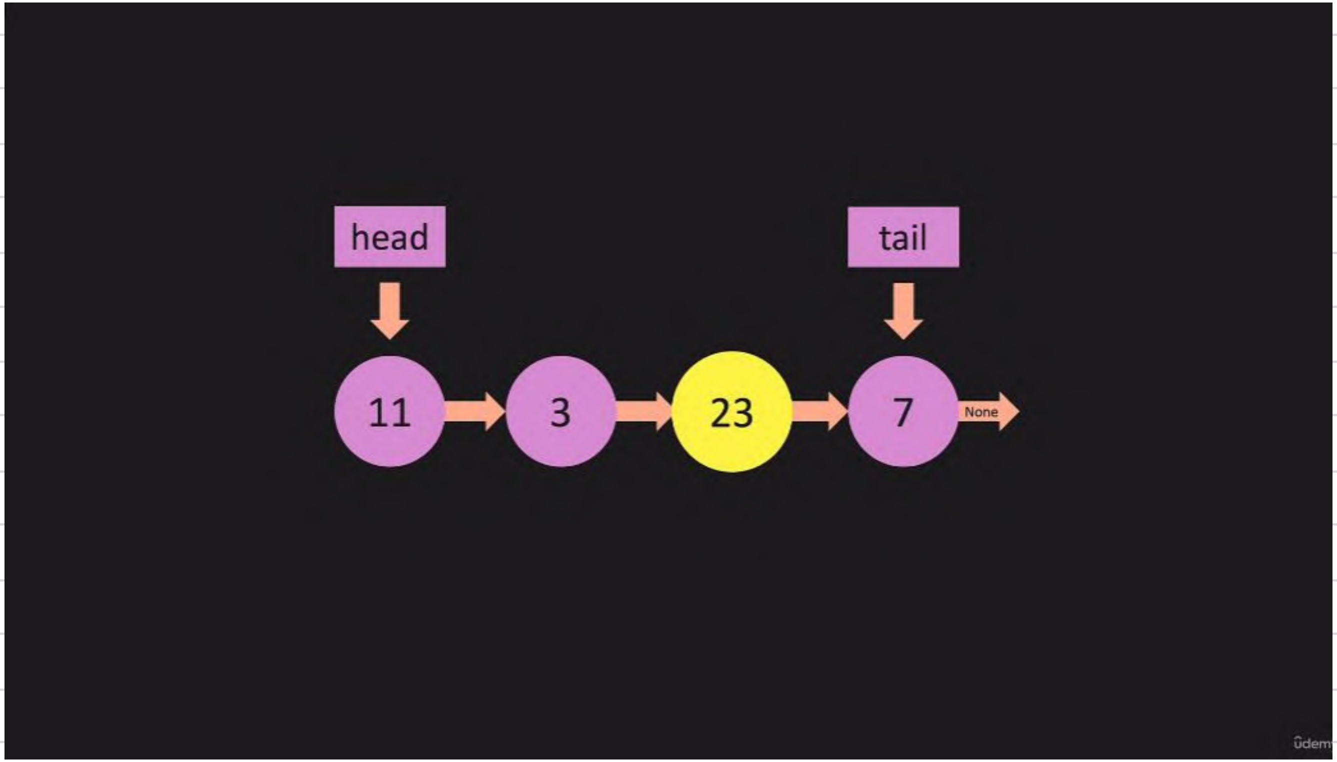
Search Node

Bagaimana jika kita ingin mencari node “23” di dalam linked list



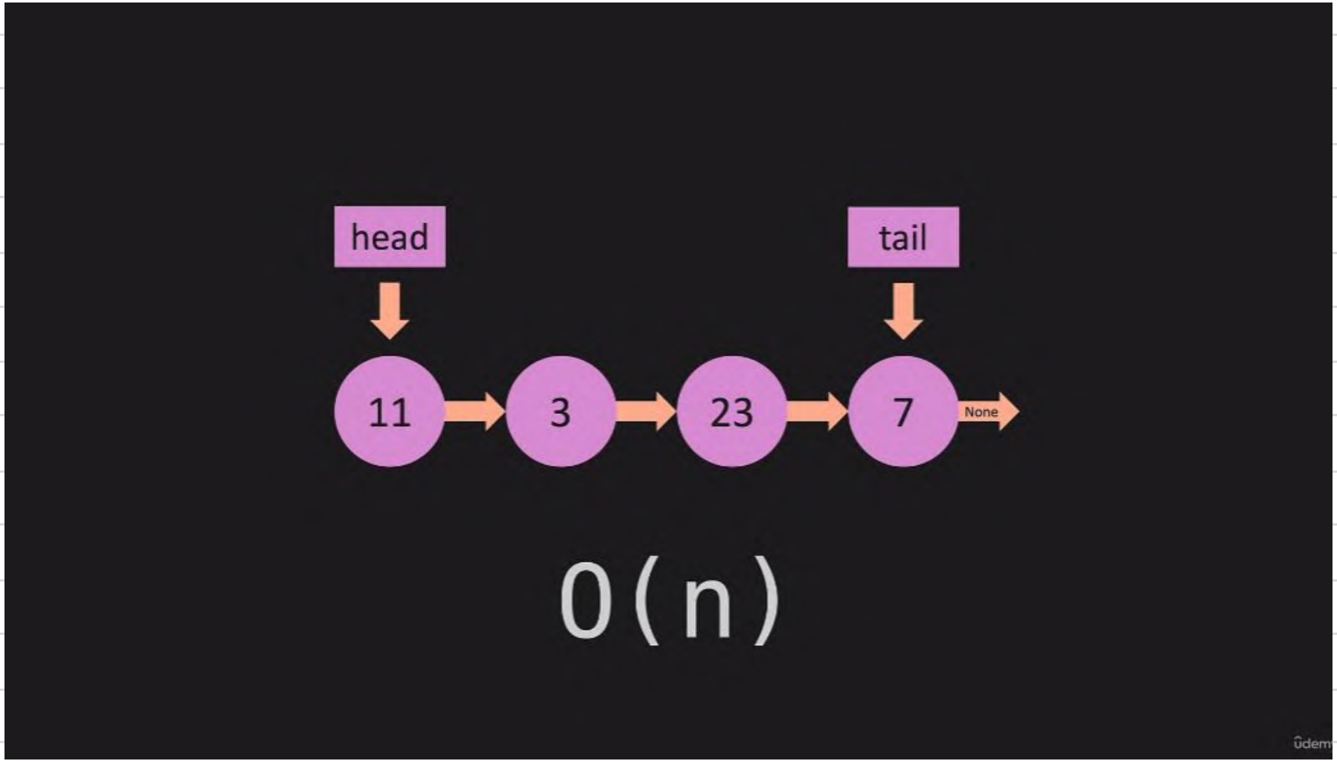
Search Node

Apa notasi Big O untuk
Mencari suatu node… ?

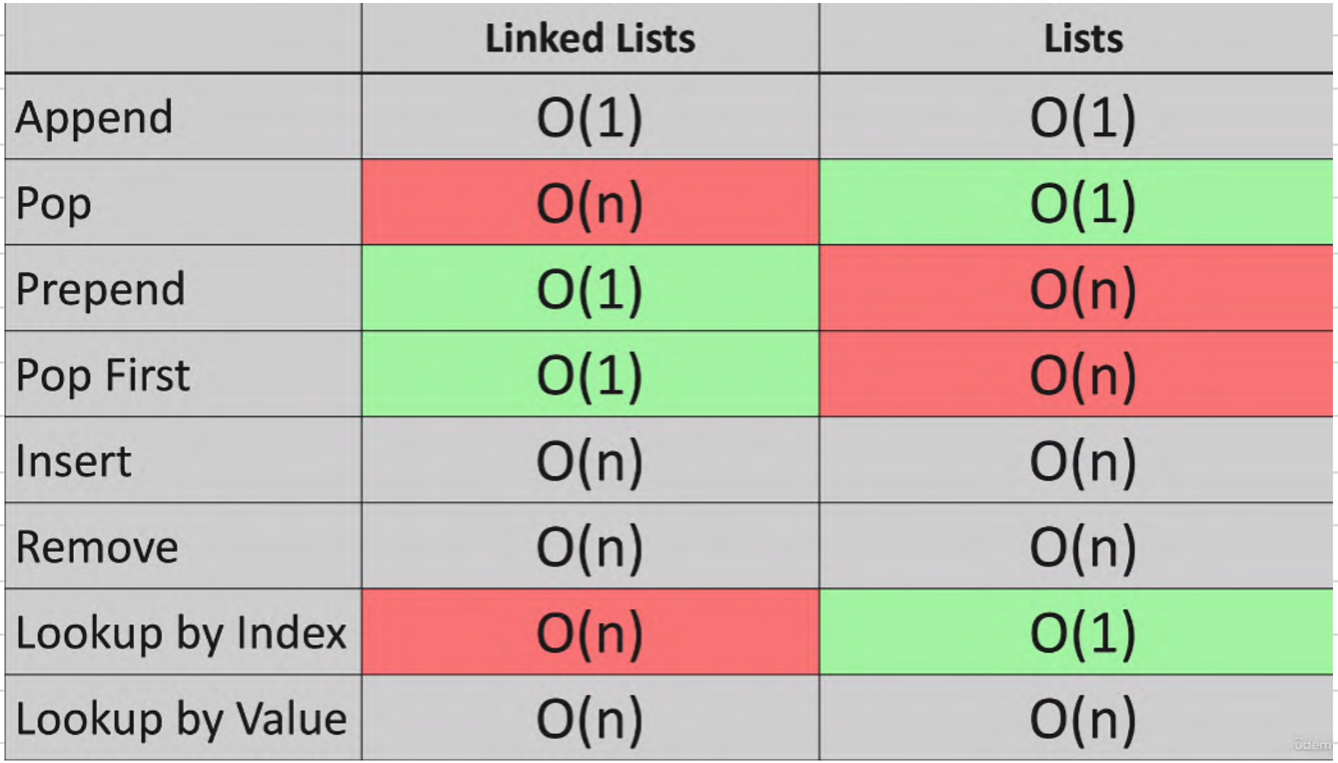


Search Node

Apa notasi Big O untuk
Mencari suatu node… ?

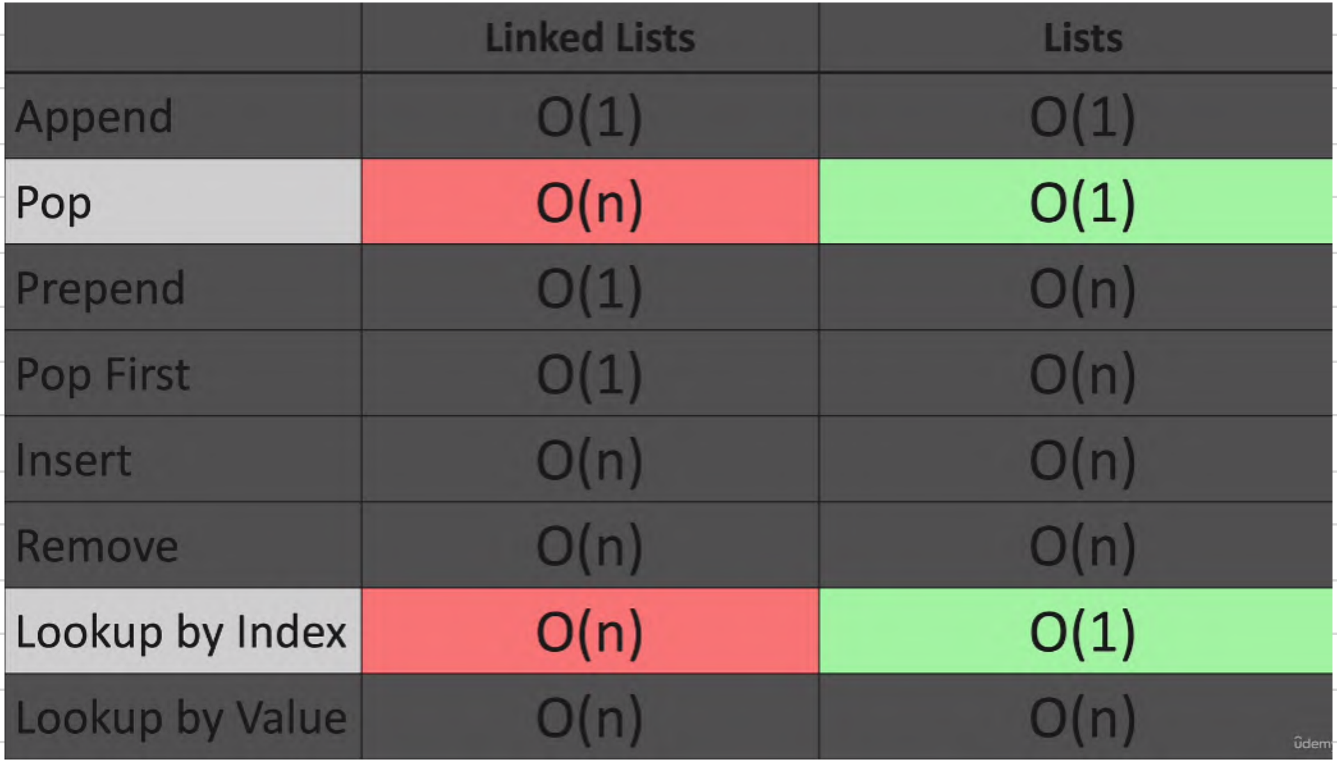


List VS Linked List



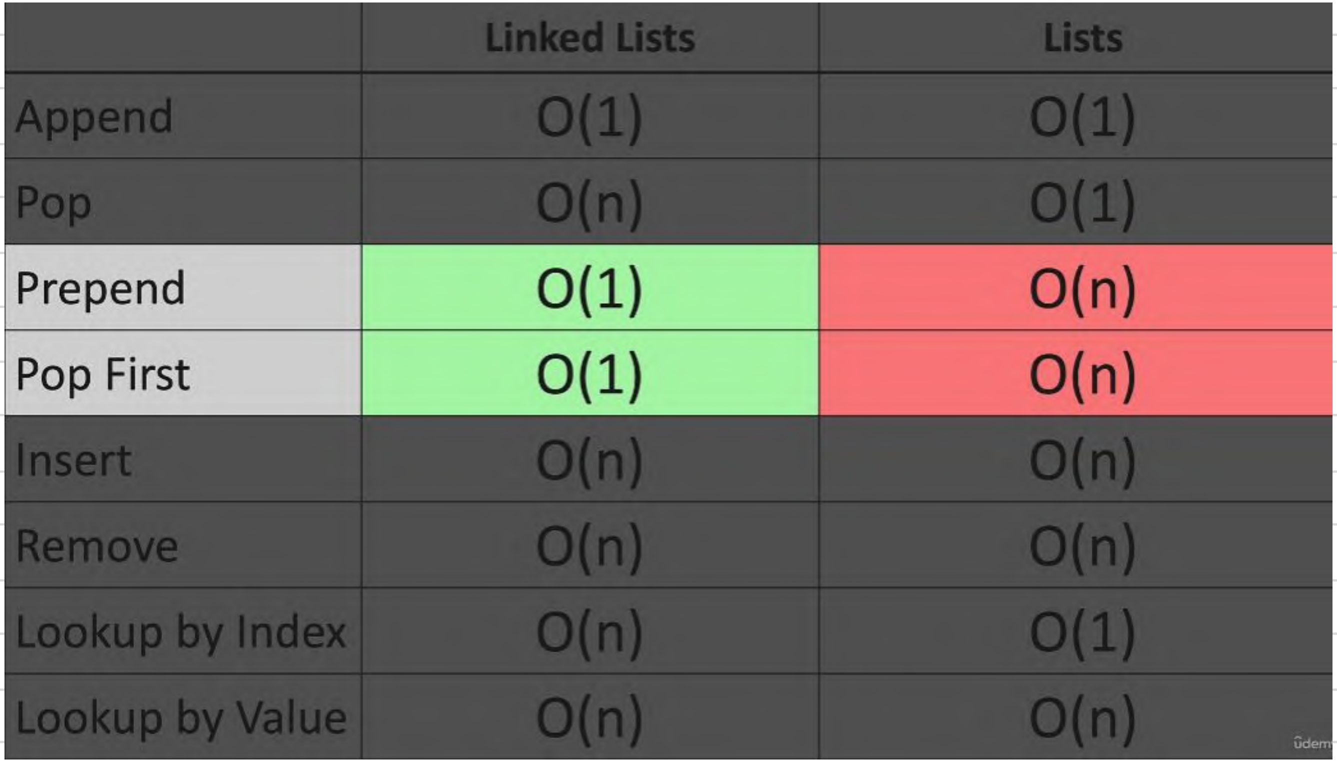
	Linked Lists	Lists
Append	O(1)	O(1)
Pop	O(n)	O(1)
Prepend	O(1)	O(n)
Pop First	O(1)	O(n)
Insert	O(n)	O(n)
Remove	O(n)	O(n)
Lookup by Index	O(n)	O(1)
Lookup by Value	O(n)	O(n)

List VS Linked List



	Linked Lists	Lists
Append	O(1)	O(1)
Pop	O(n)	O(1)
Prepend	O(1)	O(n)
Pop First	O(1)	O(n)
Insert	O(n)	O(n)
Remove	O(n)	O(n)
Lookup by Index	O(n)	O(1)
Lookup by Value	O(n)	O(n)

List VS Linked List



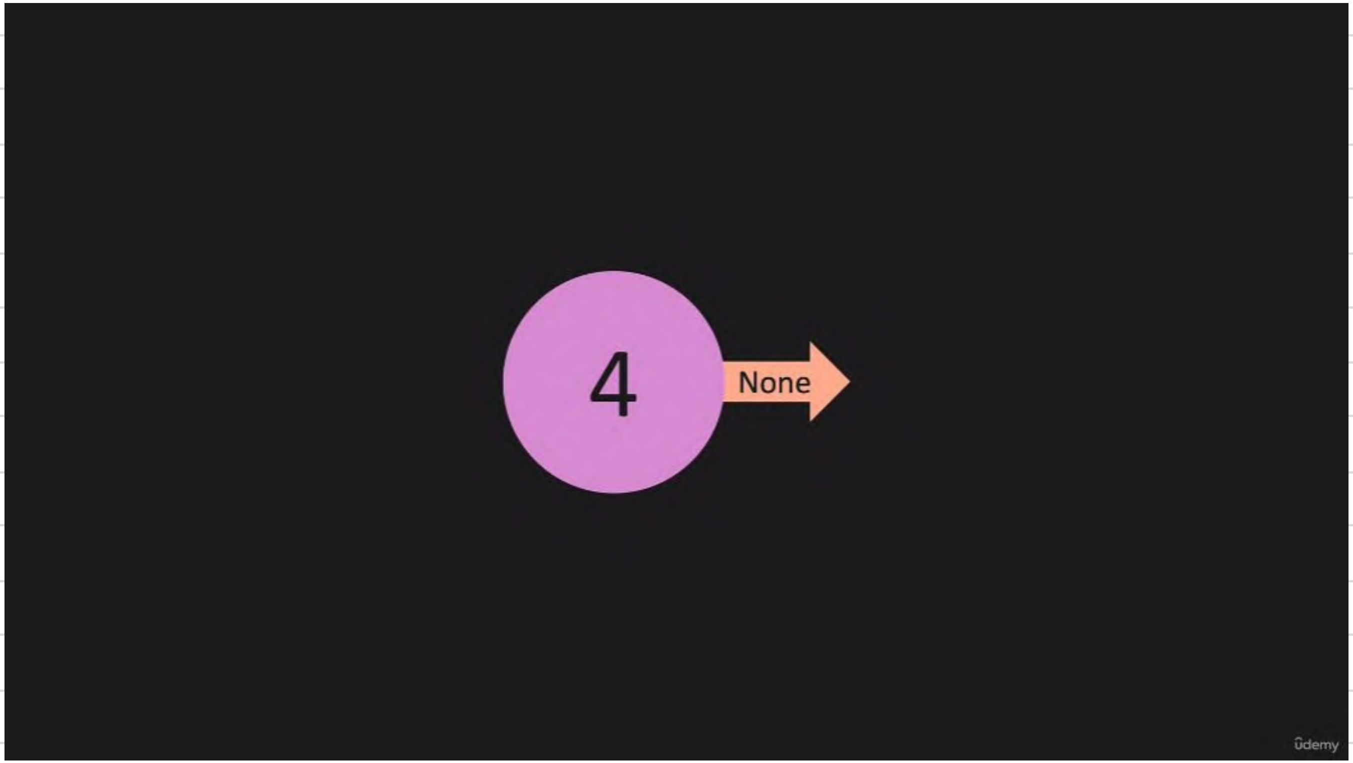
	Linked Lists	Lists
Append	O(1)	O(1)
Pop	O(n)	O(1)
Prepend	O(1)	O(n)
Pop First	O(1)	O(n)
Insert	O(n)	O(n)
Remove	O(n)	O(n)
Lookup by Index	O(n)	O(1)
Lookup by Value	O(n)	O(n)

03

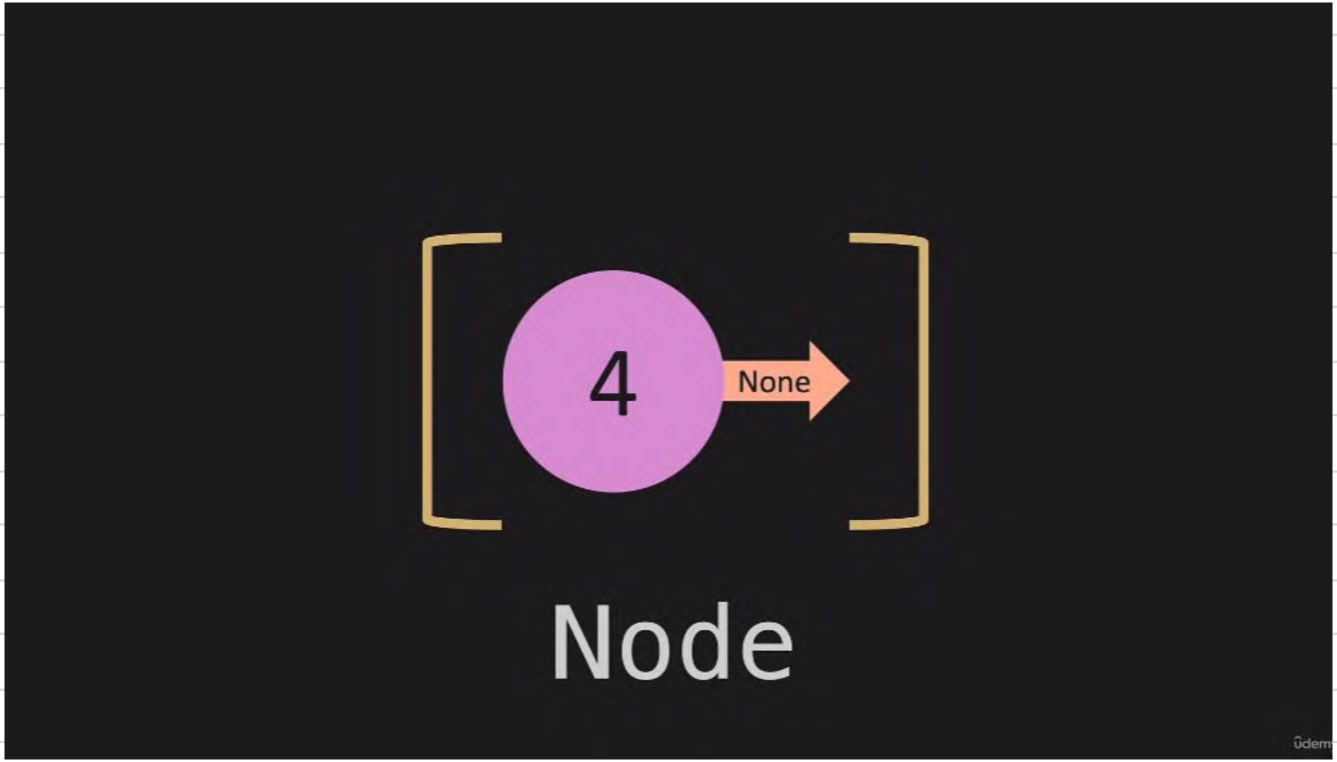
Linked List Under The Hood

Apa itu Node...?

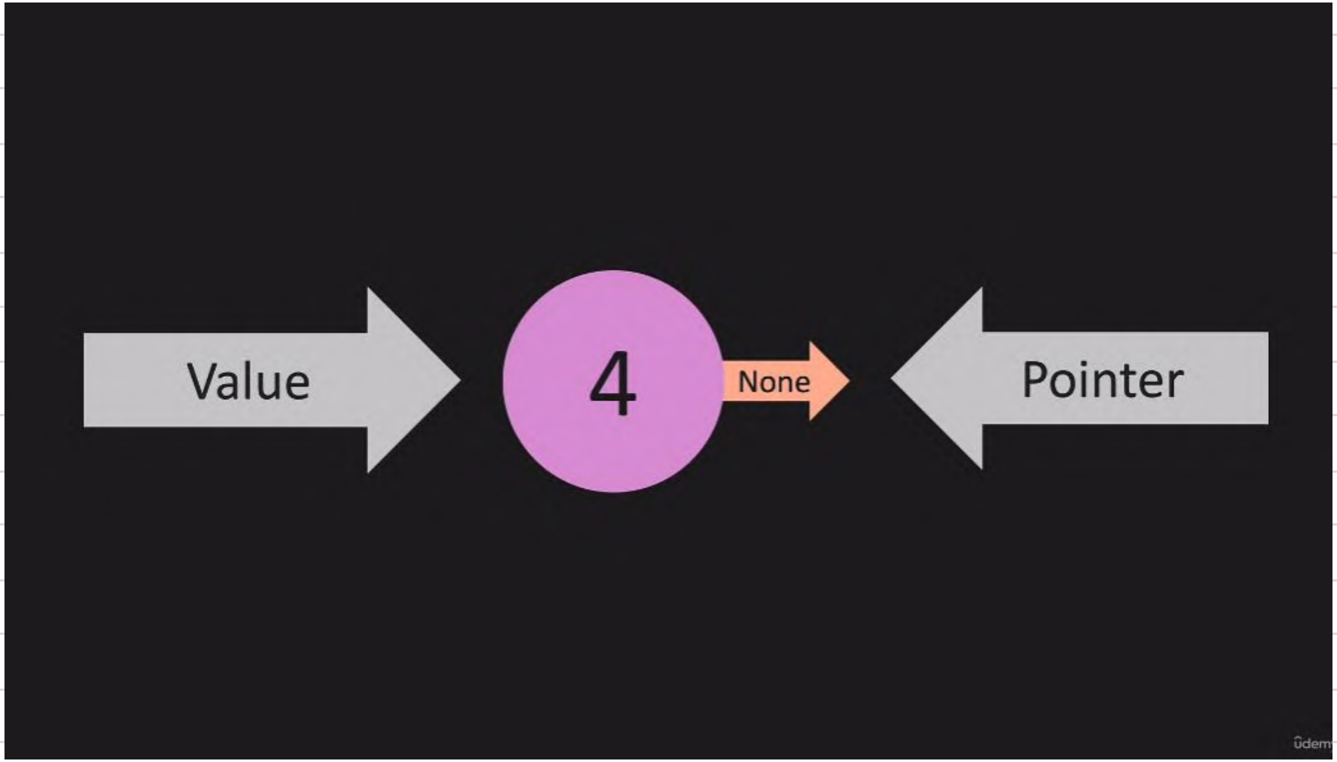
Apa itu Node...?



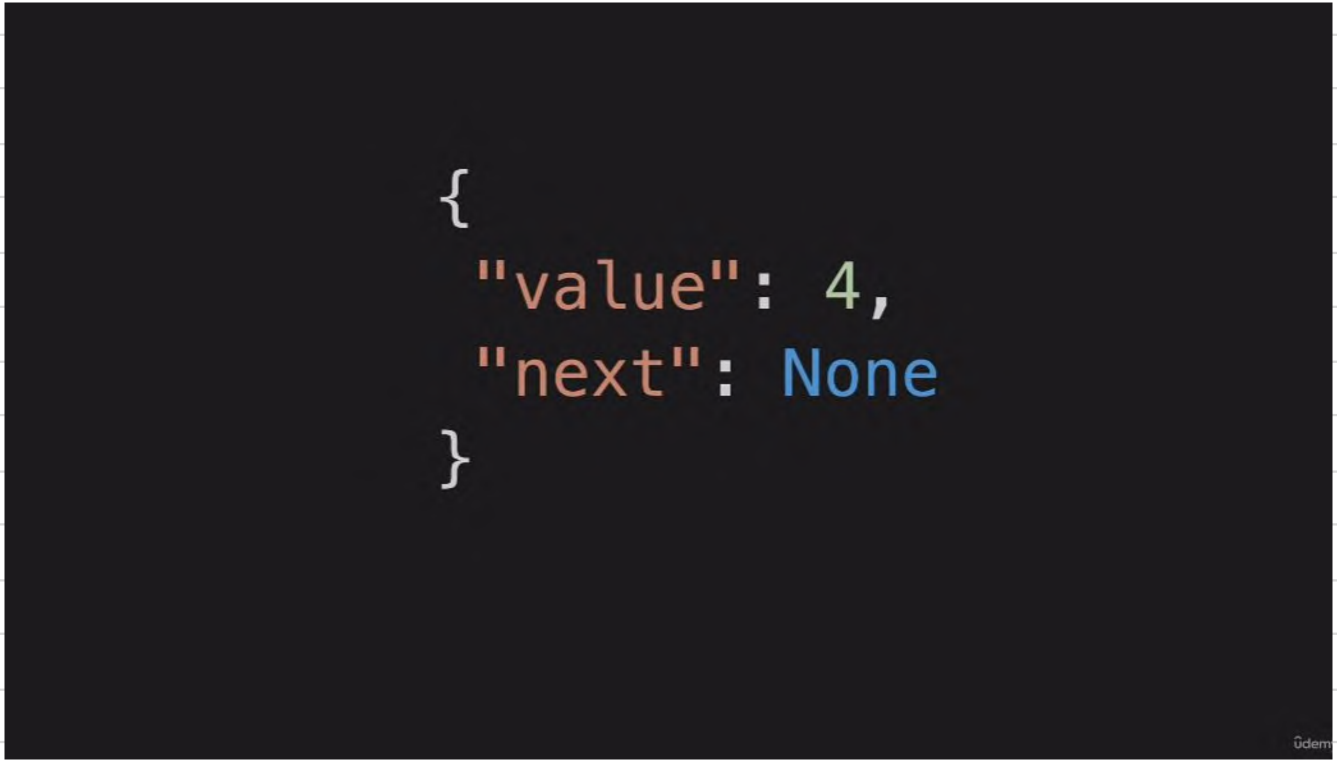
Apa itu Node...?



Apa itu Node...?

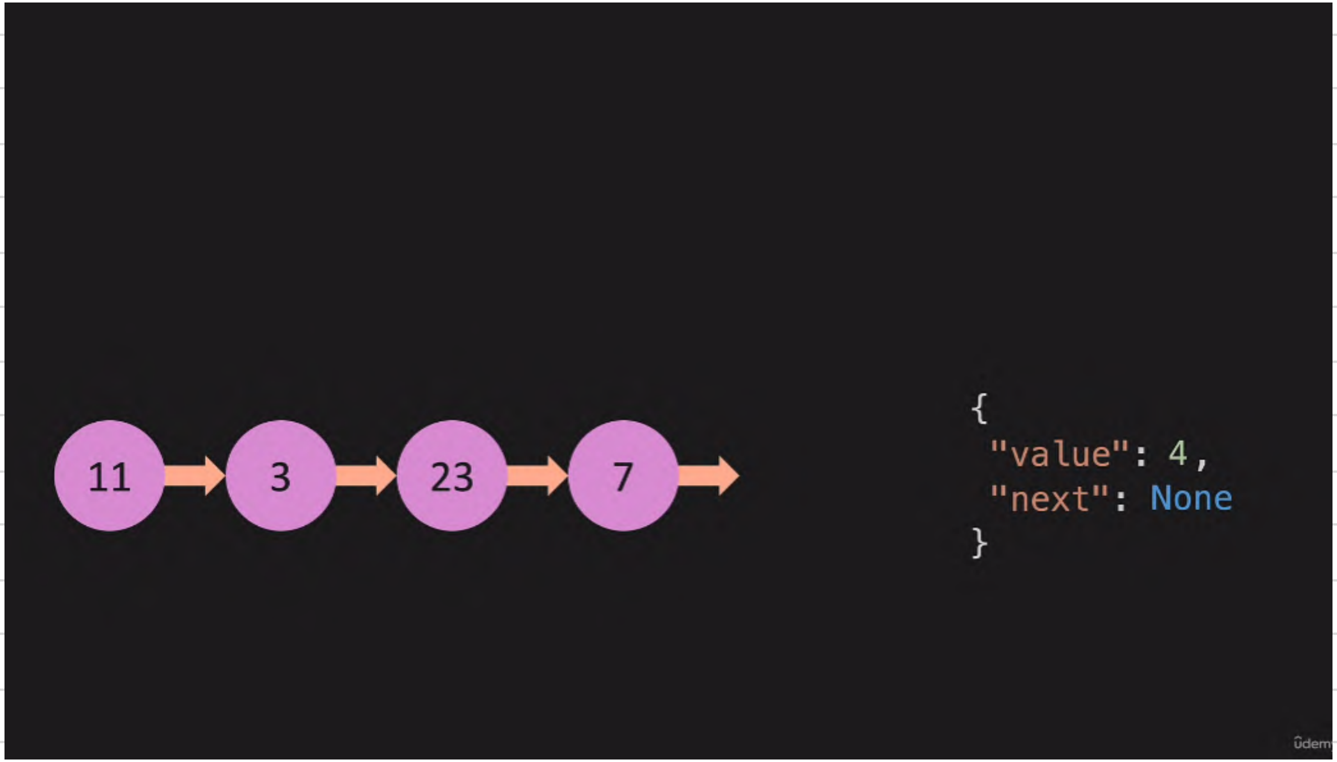


Apa itu Node...?



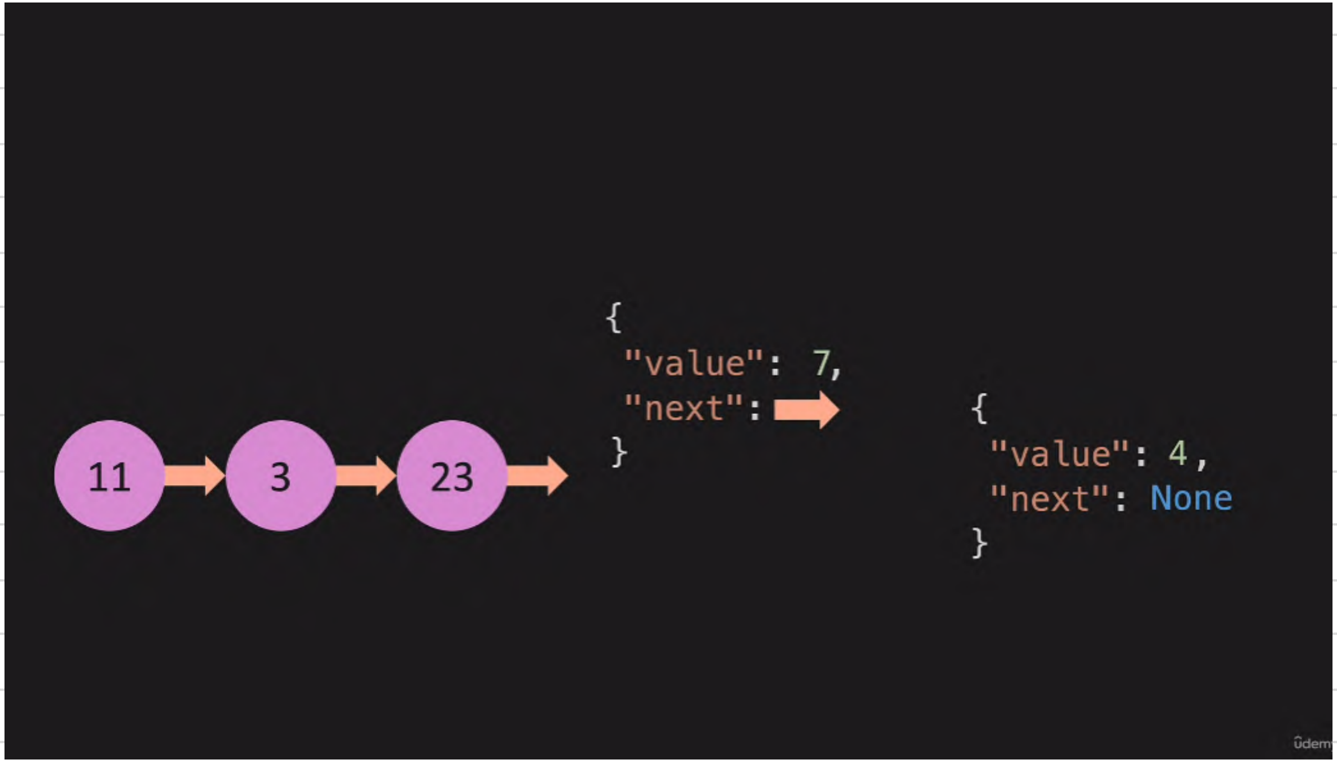
```
{
  "value": 4,
  "next": None
}
```

Apa itu Node...?



```
{
  "value": 4,
  "next": None
}
```

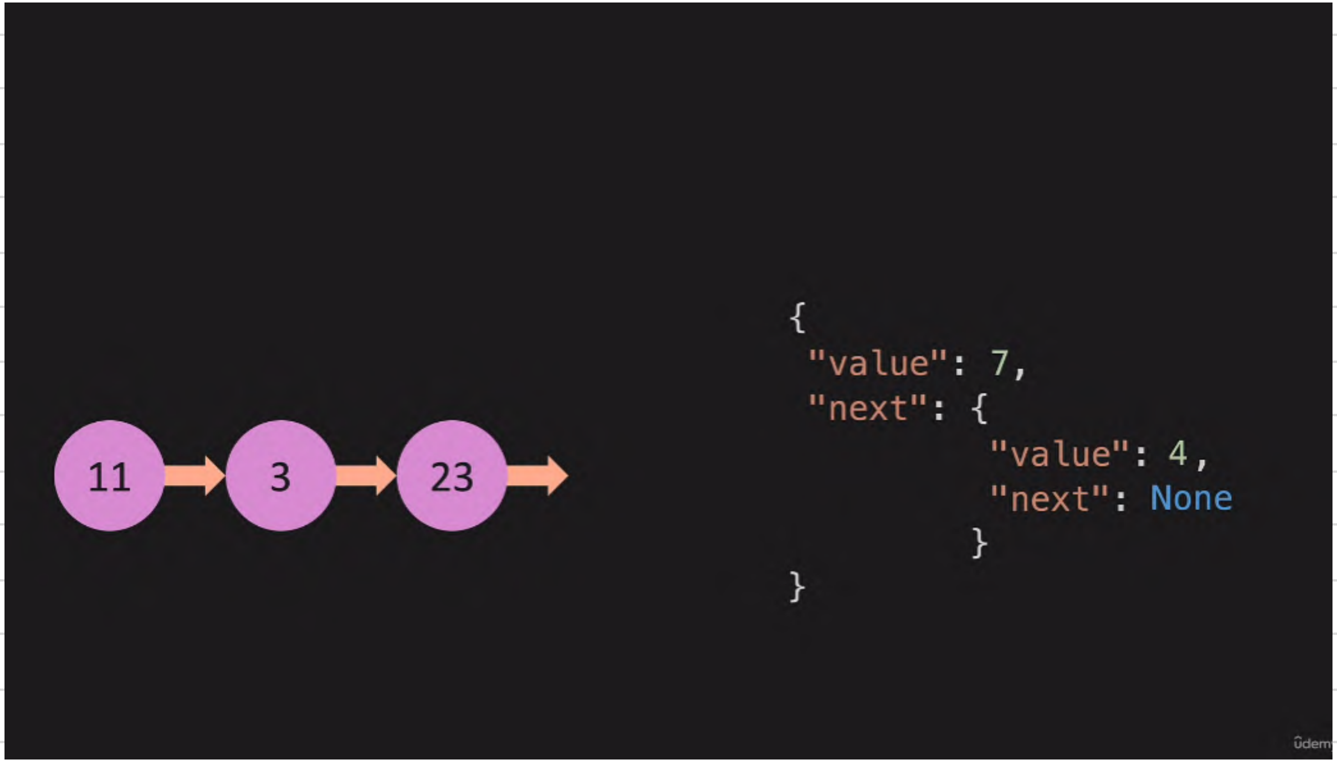
Apa itu Node...?



```
{
  "value": 7,
  "next": 
}
```

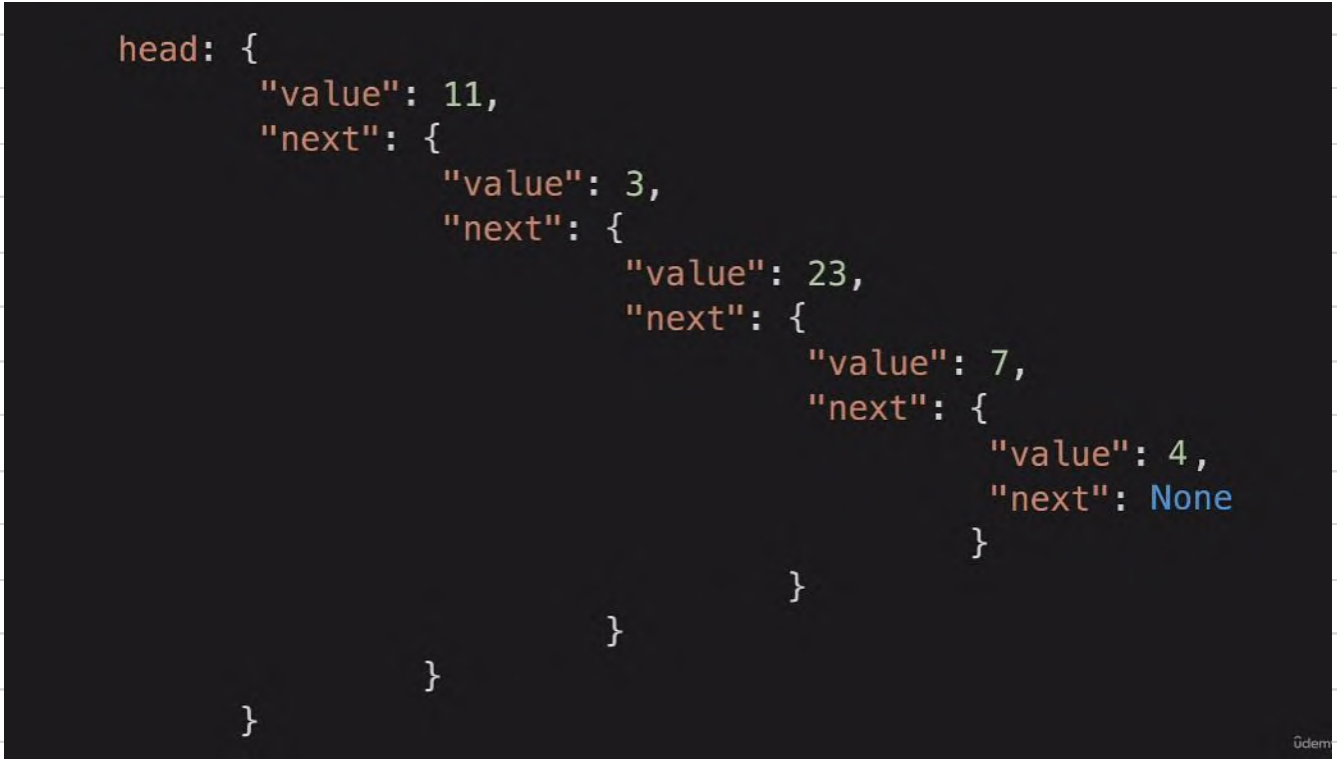
```
{
  "value": 4,
  "next": None
}
```

Apa itu Node...?



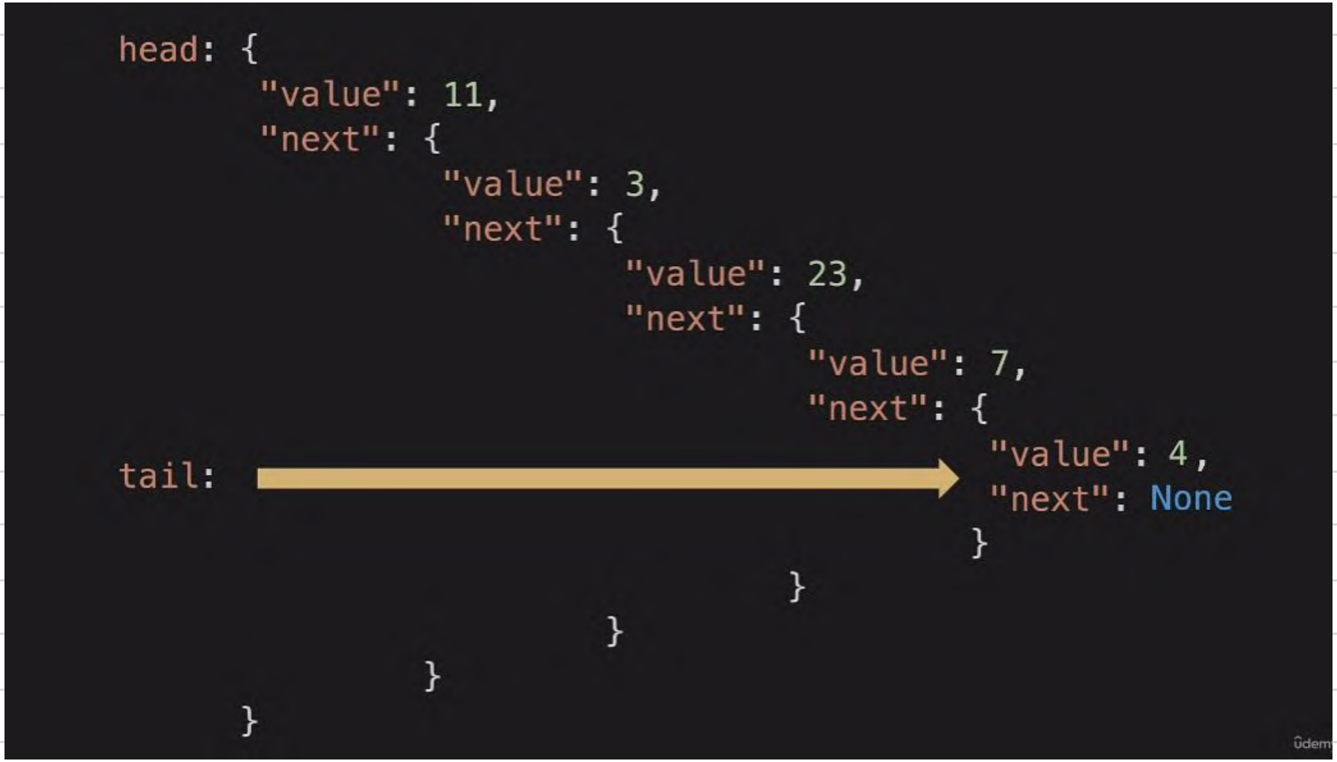
```
{
 "value": 7,
 "next": {
          "value": 4,
          "next": None
         }
}
```

Apa itu Node...?



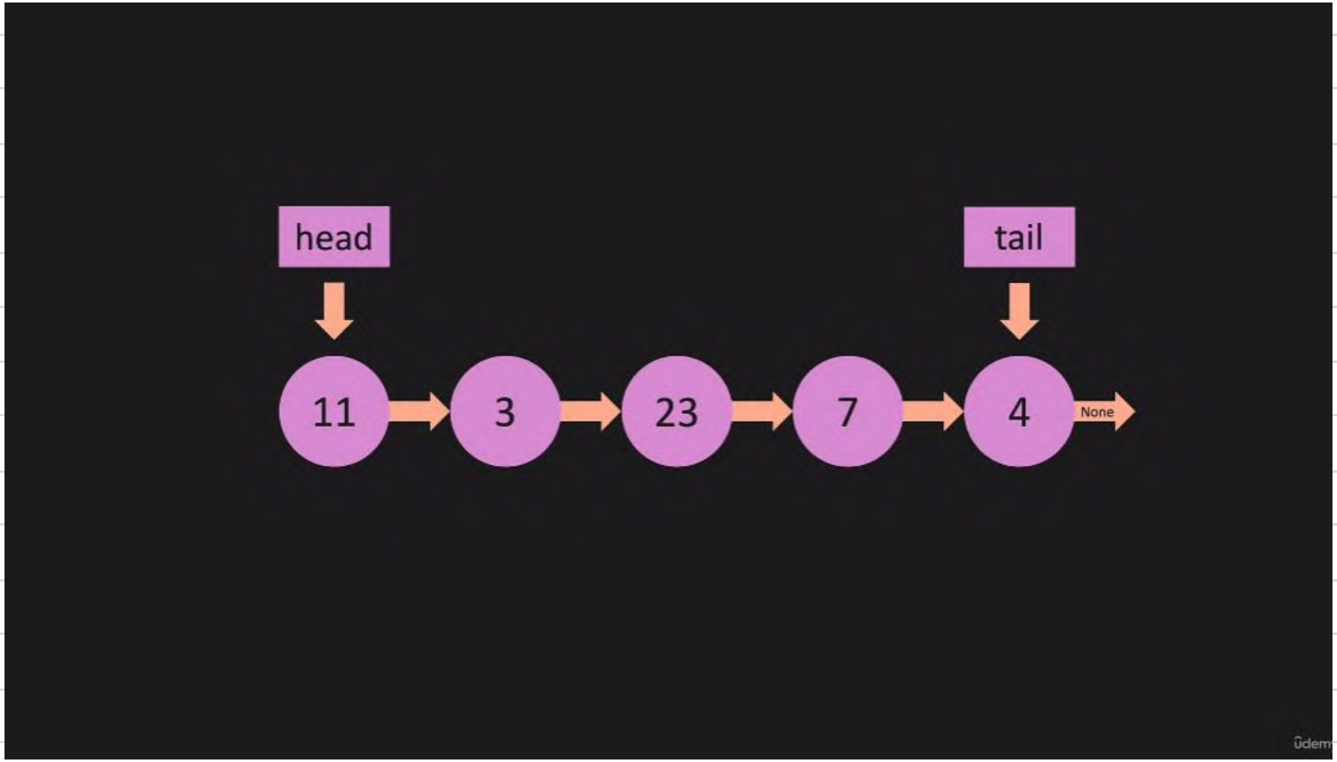
```
head: {
       "value": 11,
       "next": {
                "value": 3,
                "next": {
                         "value": 23,
                         "next": {
                                  "value": 7,
                                  "next": {
                                           "value": 4,
                                           "next": None
                                          }
                                 }
                        }
               }
      }
```

Apa itu Node...?

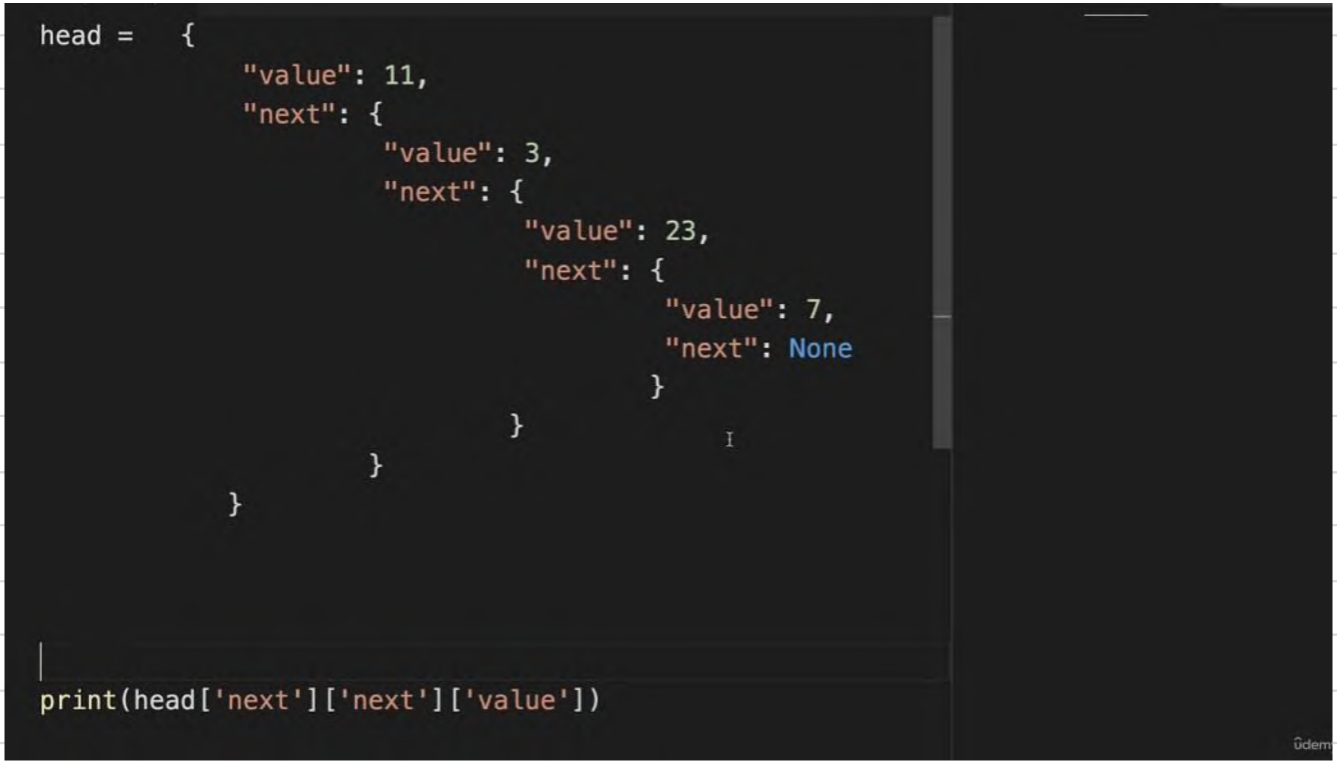


```
head: {
        "value": 11,
        "next": {
                  "value": 3,
                  "next": {
                            "value": 23,
                            "next": {
                                      "value": 7,
                                      "next": {
tail:                                           "value": 4,
                                                "next": None
                                               }
                                      }
                            }
                  }
        }
```

Apa itu Node...?

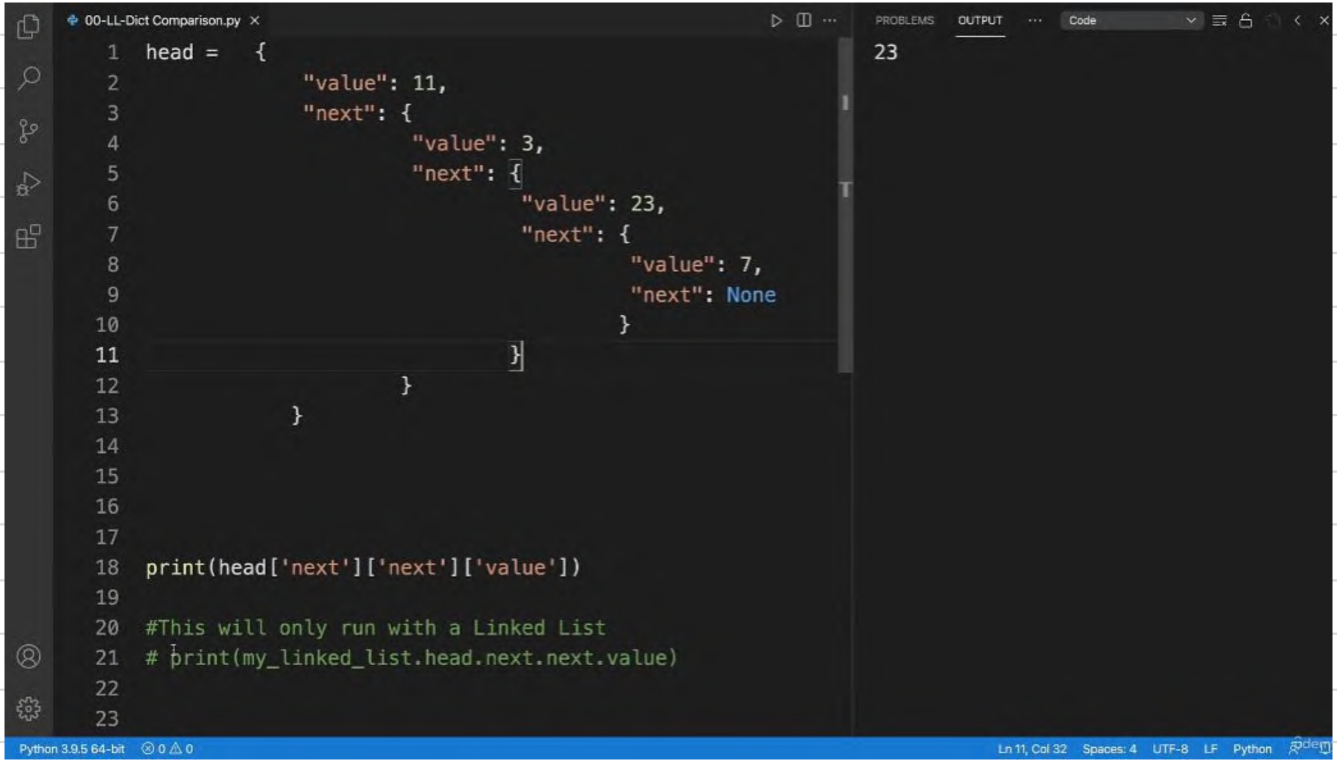


Apa itu Node...?



```
head =   {
               "value": 11,
               "next": {
                         "value": 3,
                         "next": {
                                   "value": 23,
                                   "next": {
                                             "value": 7,
                                             "next": None
                                            }
                                  }
                        }
              }


print(head['next']['next']['value'])
```

Apa itu Node...?



00-LL-Dict Comparison.py

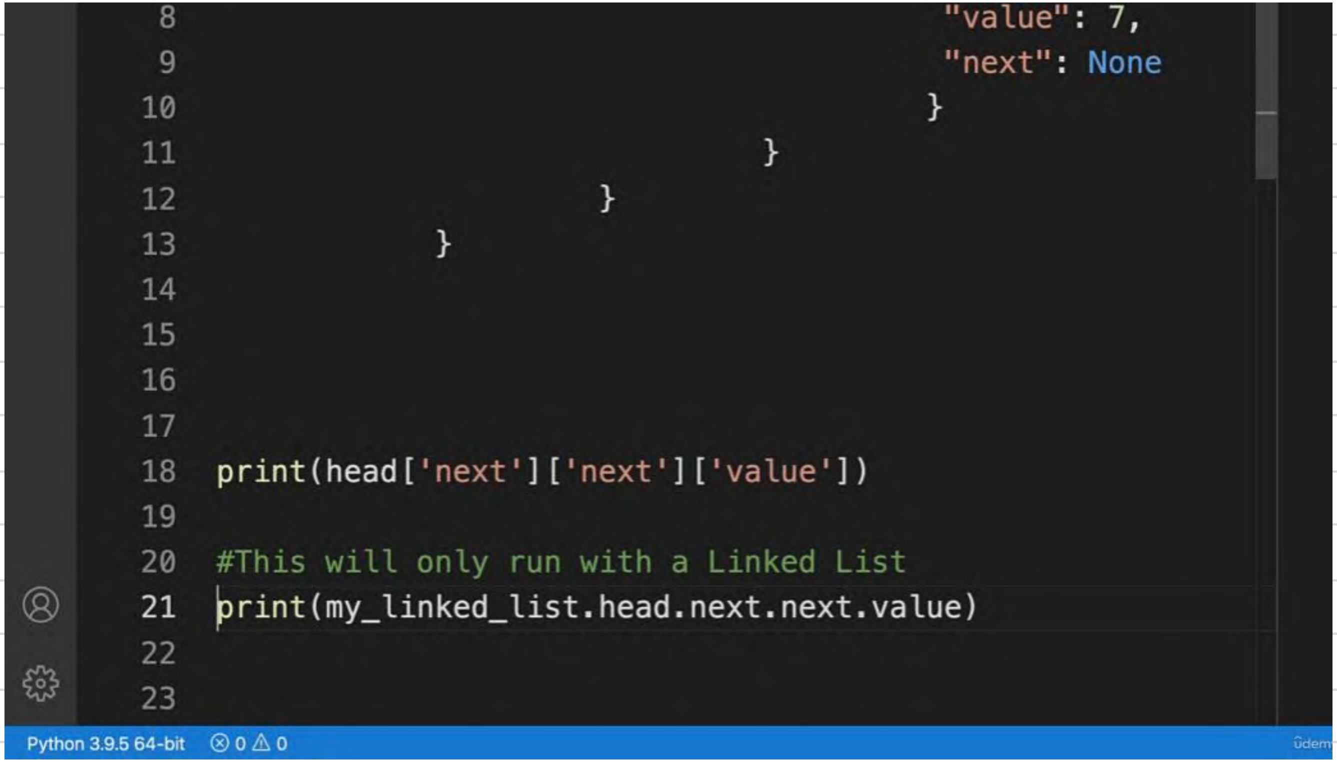
```python
head =   {
                "value": 11,
                "next": {
                            "value": 3,
                            "next": {
                                        "value": 23,
                                        "next": {
                                                    "value": 7,
                                                    "next": None
                                                }
                                    }
                        }
            }



print(head['next']['next']['value'])

#This will only run with a Linked List
# print(my_linked_list.head.next.next.value)
```

OUTPUT

```
23
```

Python 3.9.5 64-bit   Ln 11, Col 32   Spaces: 4   UTF-8   LF   Python

Syntax sangat mirip

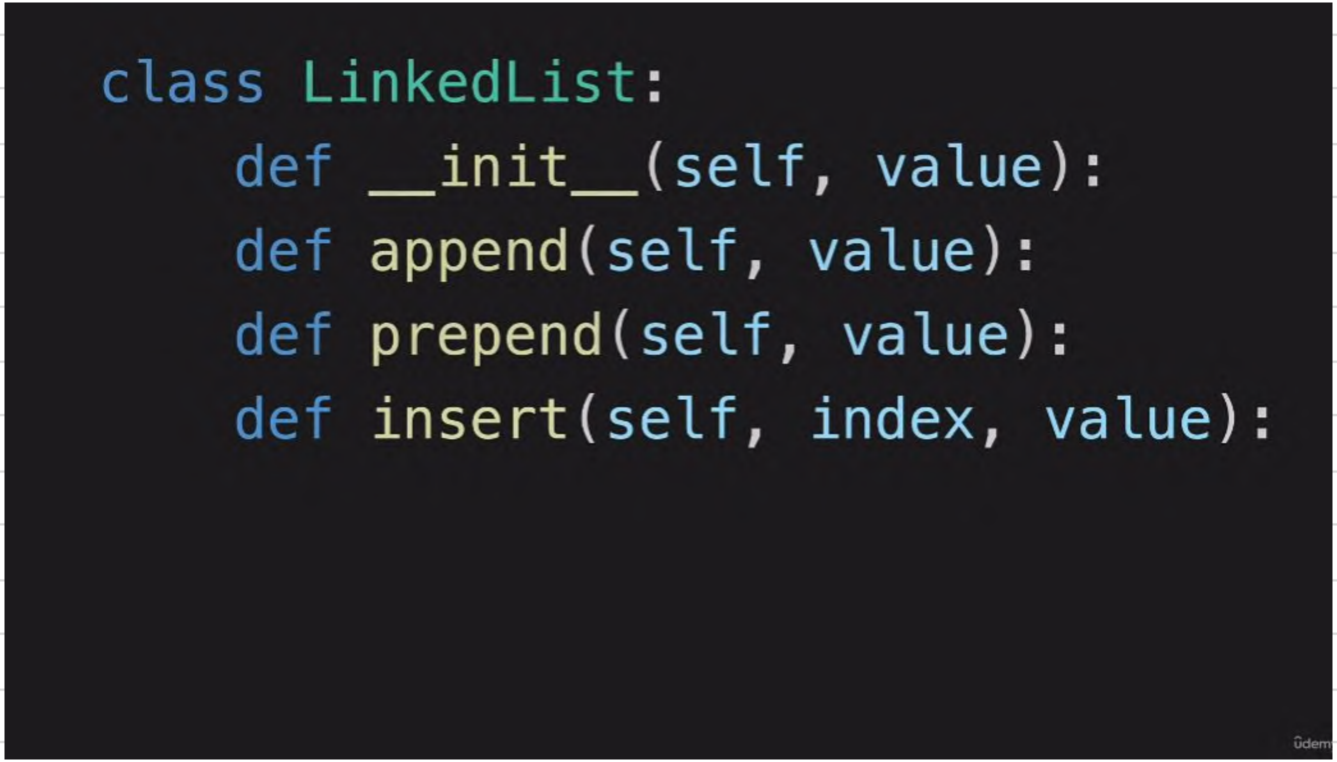


```
                                                  "value": 7,
                                                  "next": None
                                                 }
                                        }
                               }
                      }



print(head['next']['next']['value'])

#This will only run with a Linked List
print(my_linked_list.head.next.next.value)
```

04

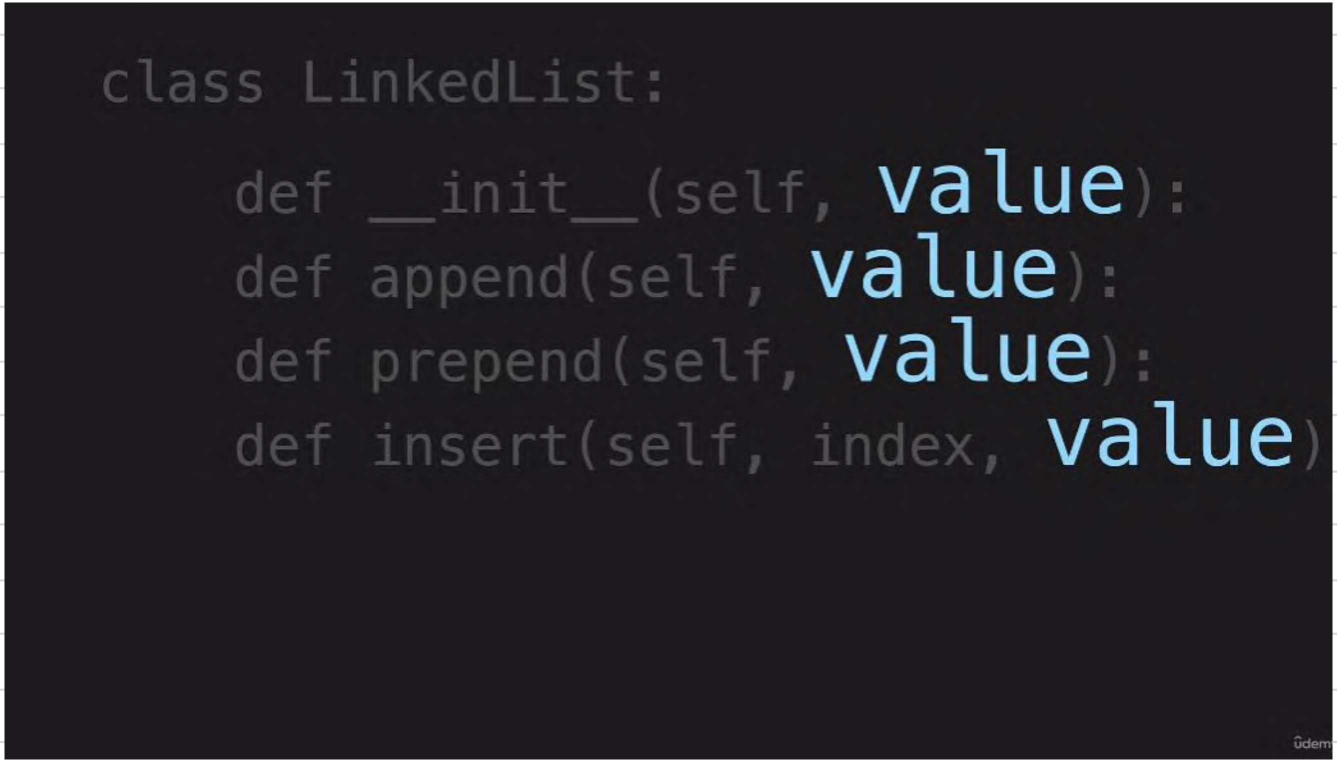
Linked List Constructor

Linked List: Class



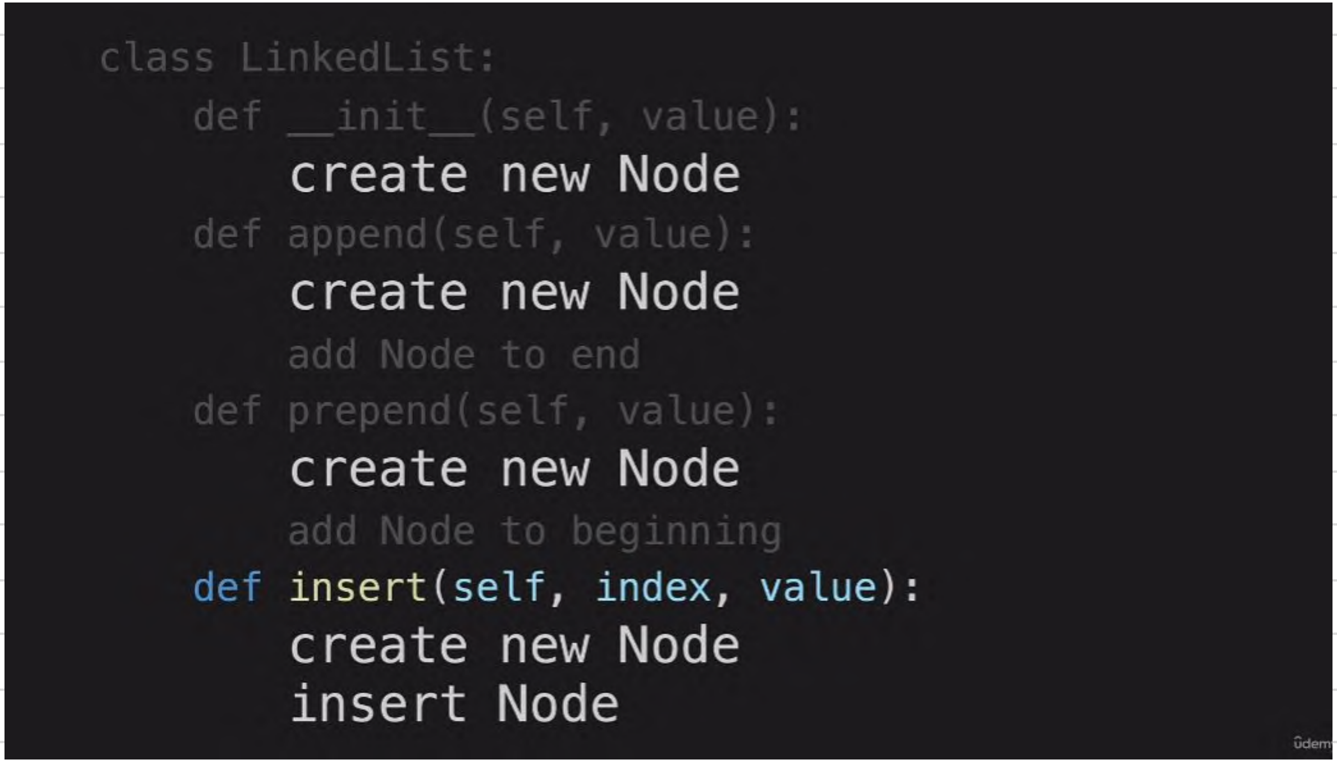
```
class LinkedList:
    def __init__(self, value):
    def append(self, value):
    def prepend(self, value):
    def insert(self, index, value):
```

Linked List: Class



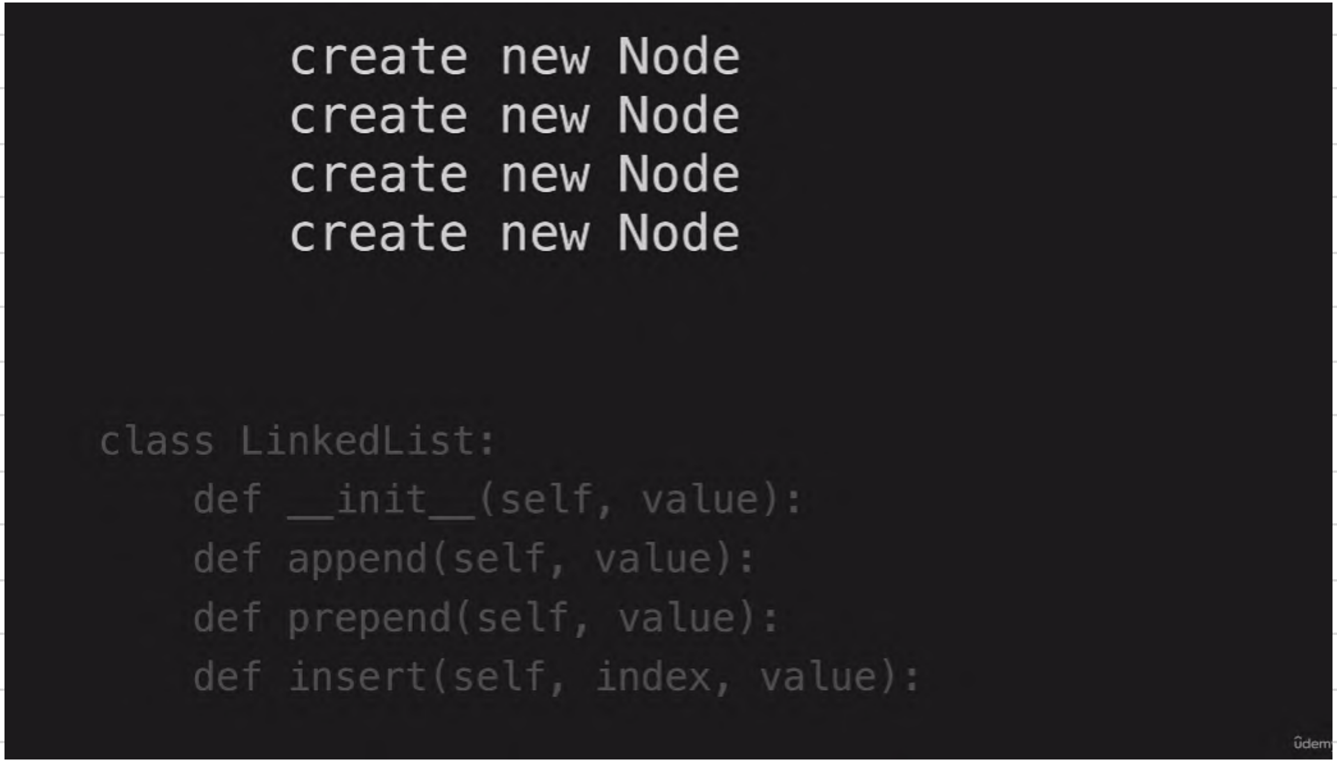
```
class LinkedList:
    def __init__(self, value):
    def append(self, value):
    def prepend(self, value):
    def insert(self, index, value)
```

Linked List: Class



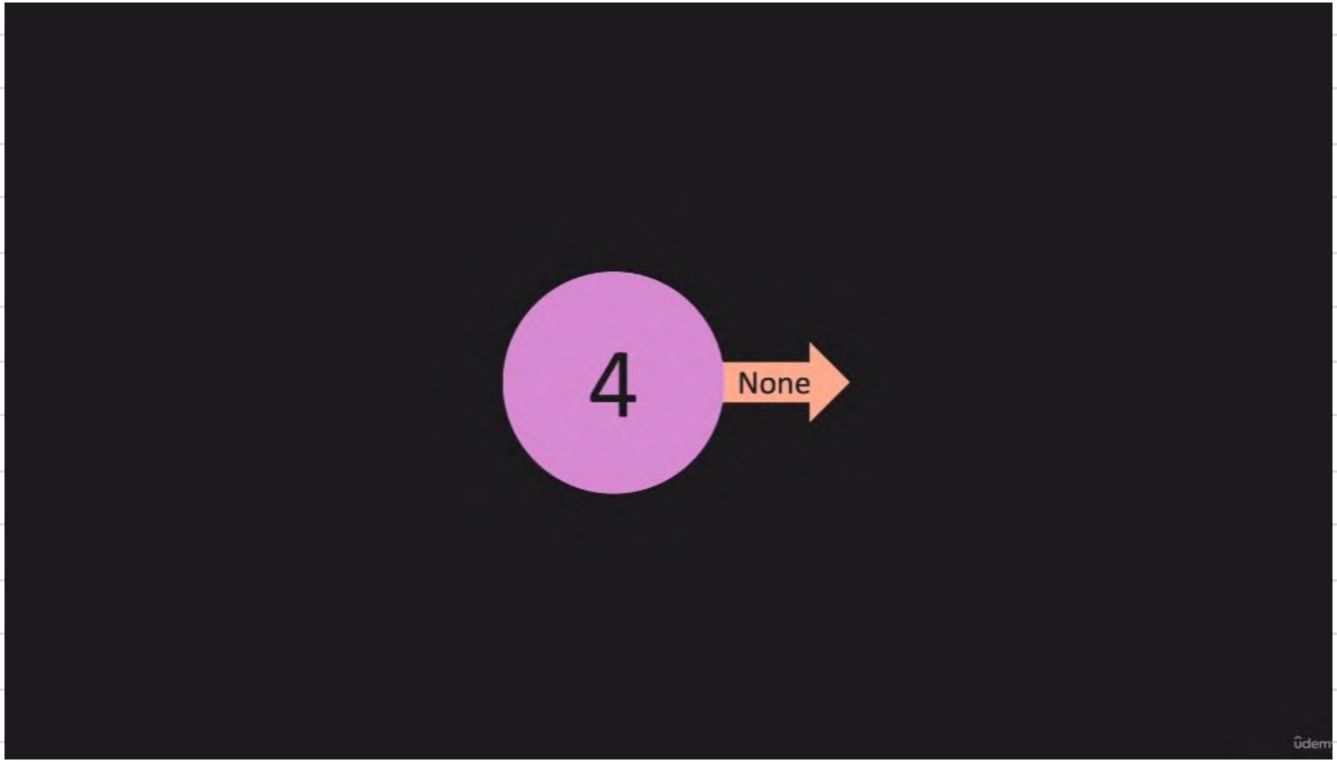
```
class LinkedList:
    def __init__(self, value):
        create new Node
    def append(self, value):
        create new Node
        add Node to end
    def prepend(self, value):
        create new Node
        add Node to beginning
    def insert(self, index, value):
        create new Node
        insert Node
```

Linked List: Class

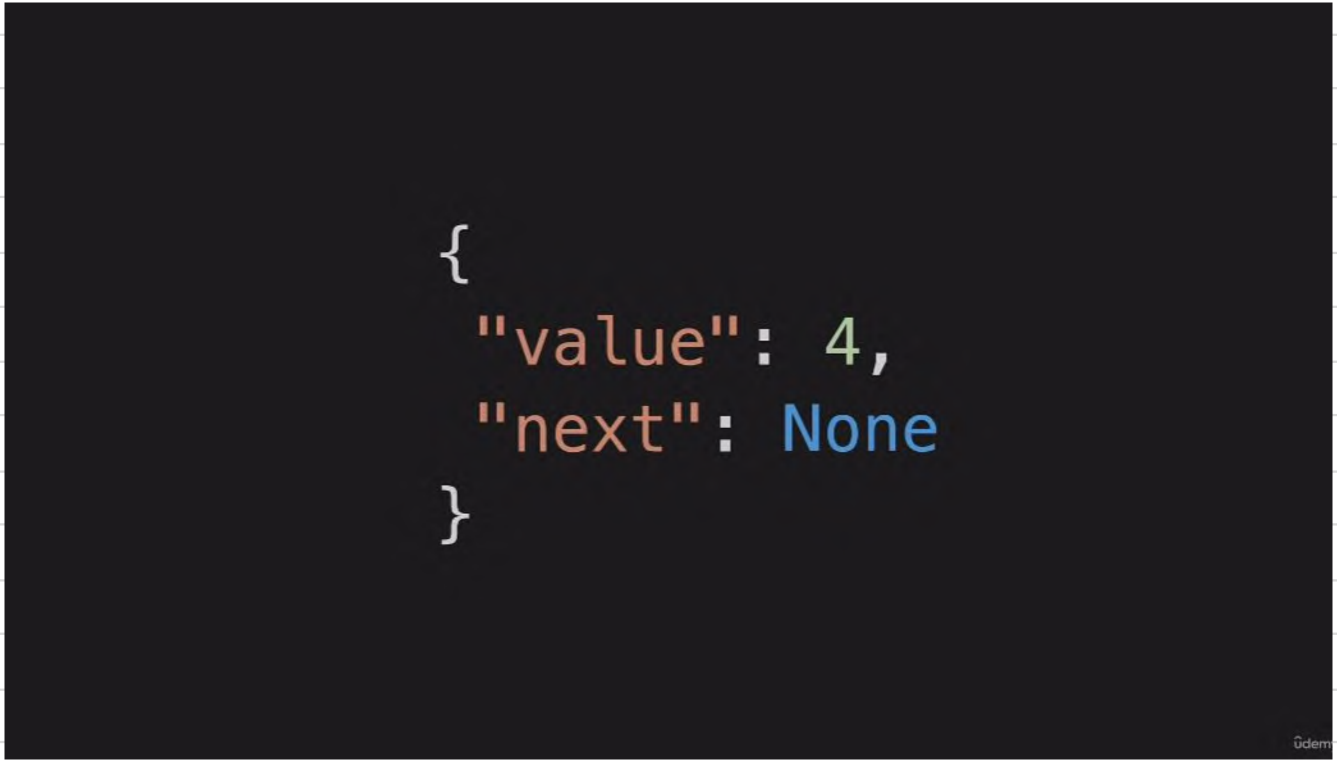


```
      create new Node
      create new Node
      create new Node
      create new Node



class LinkedList:
    def __init__(self, value):
    def append(self, value):
    def prepend(self, value):
    def insert(self, index, value):
```

Ingat kembali Node

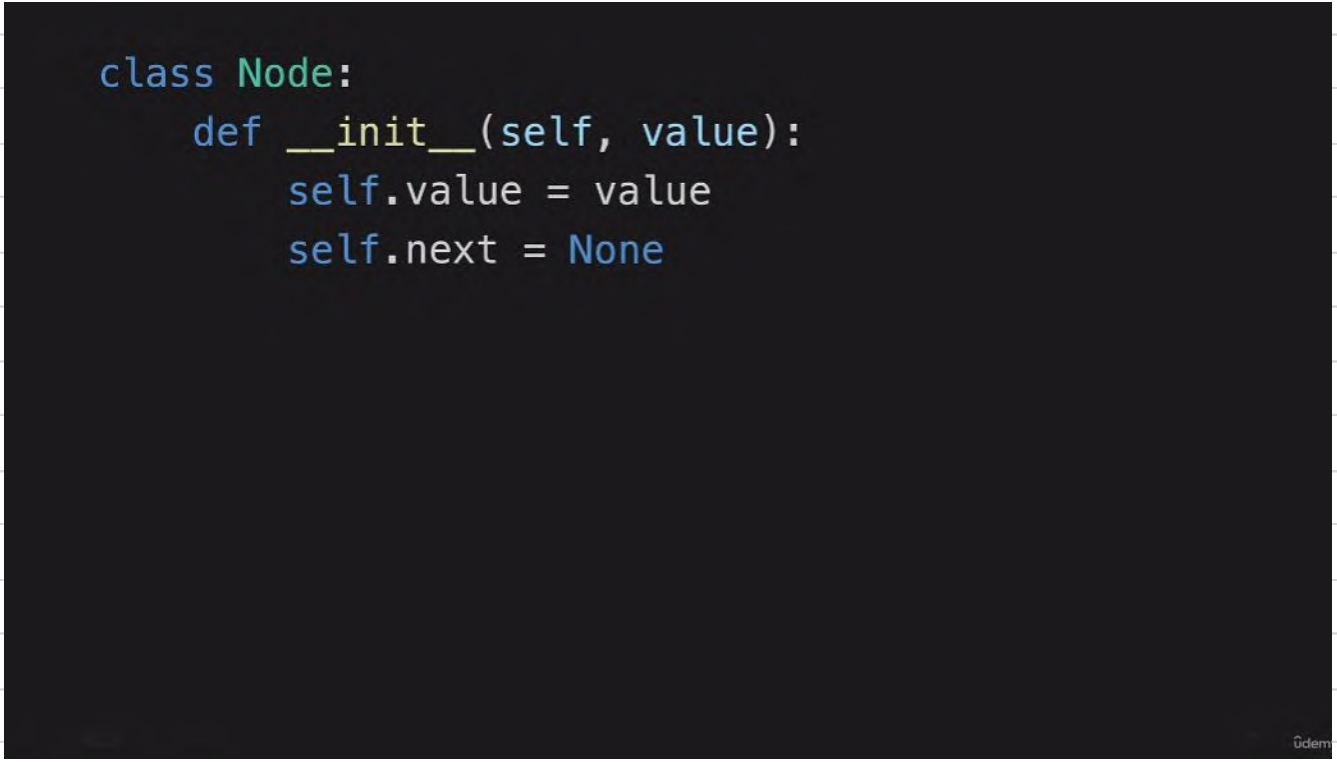


Ingat kembali Node



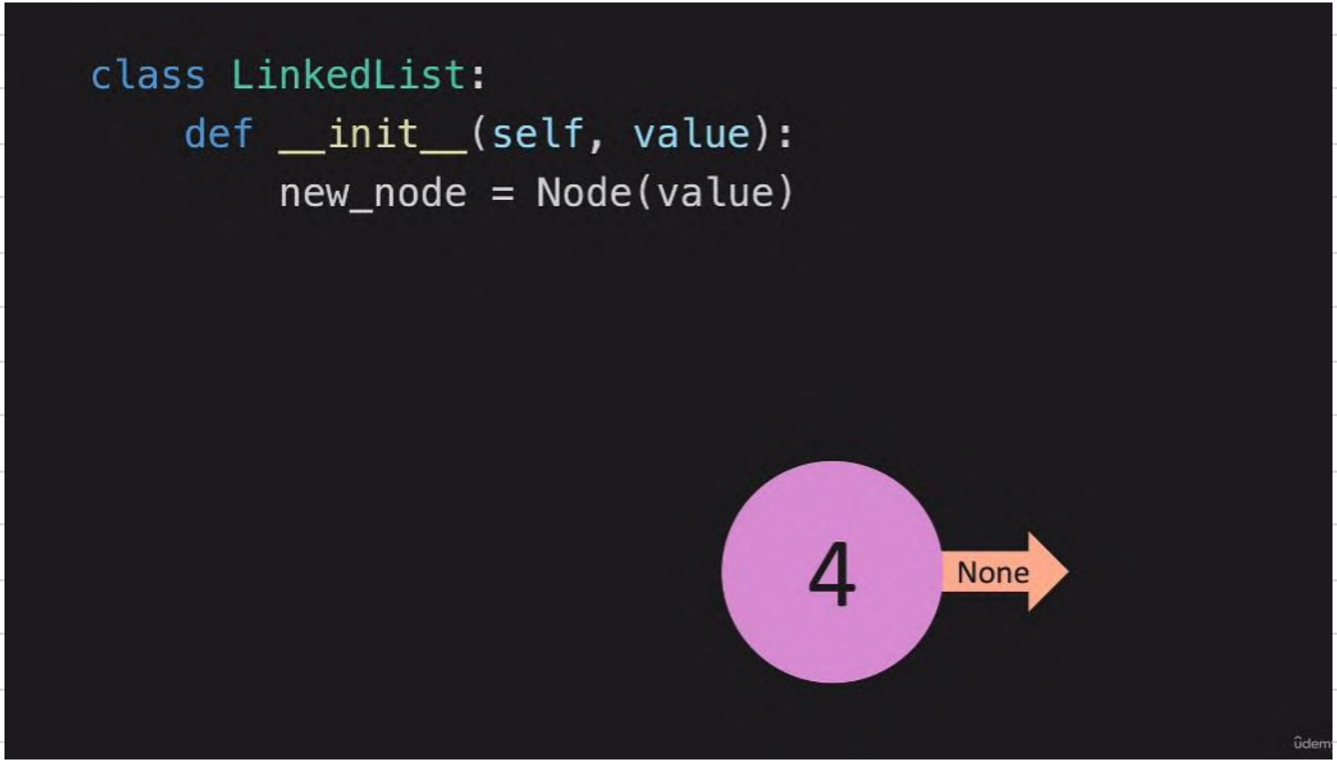
```
{
 "value": 4,
 "next": None
}
```

Node: Class



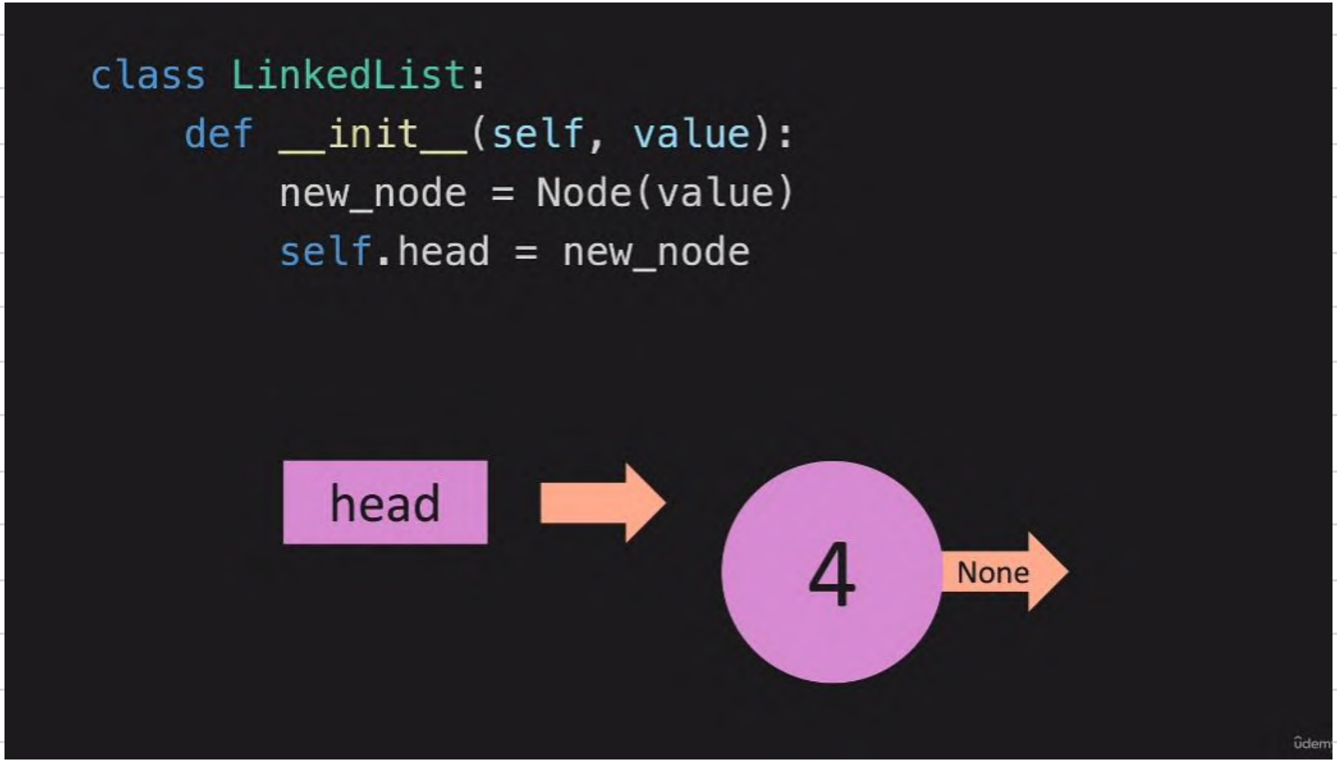
```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None
```

Node: Class



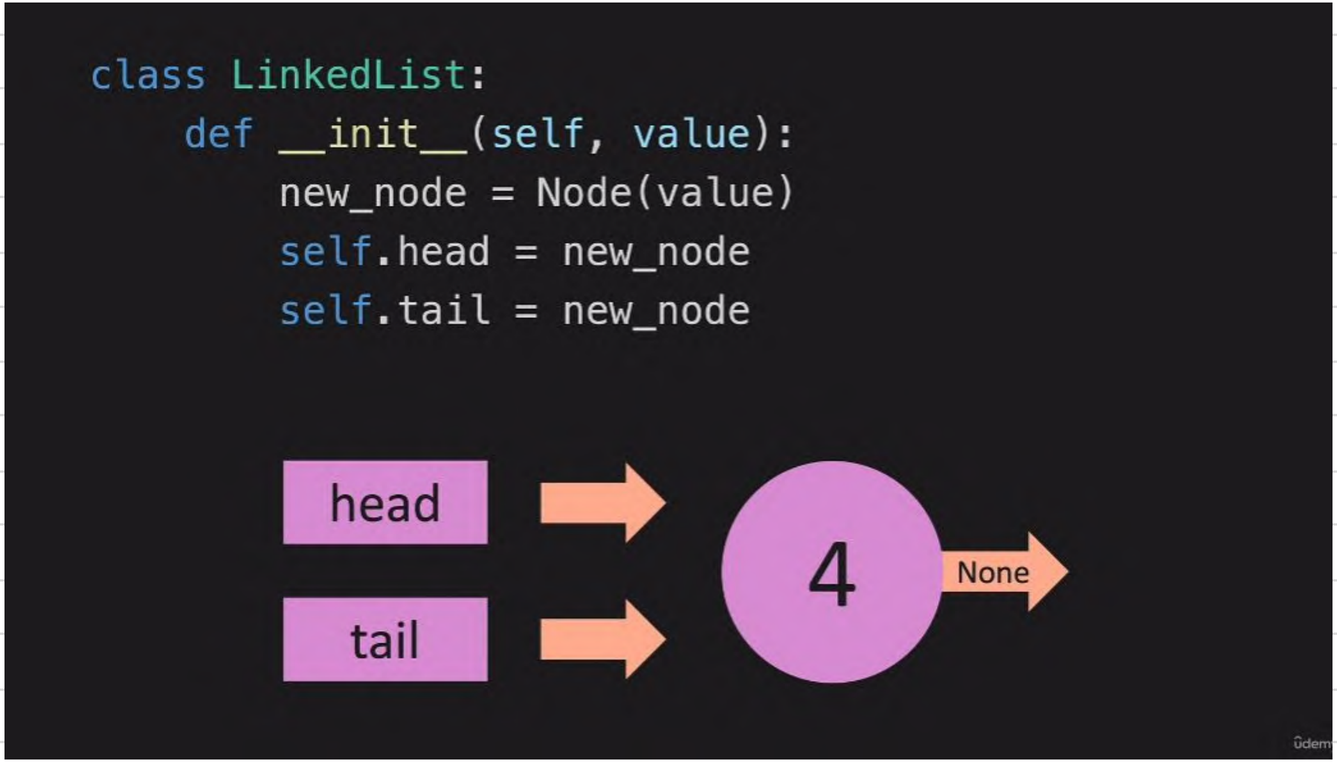
```
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
```

Node: Class



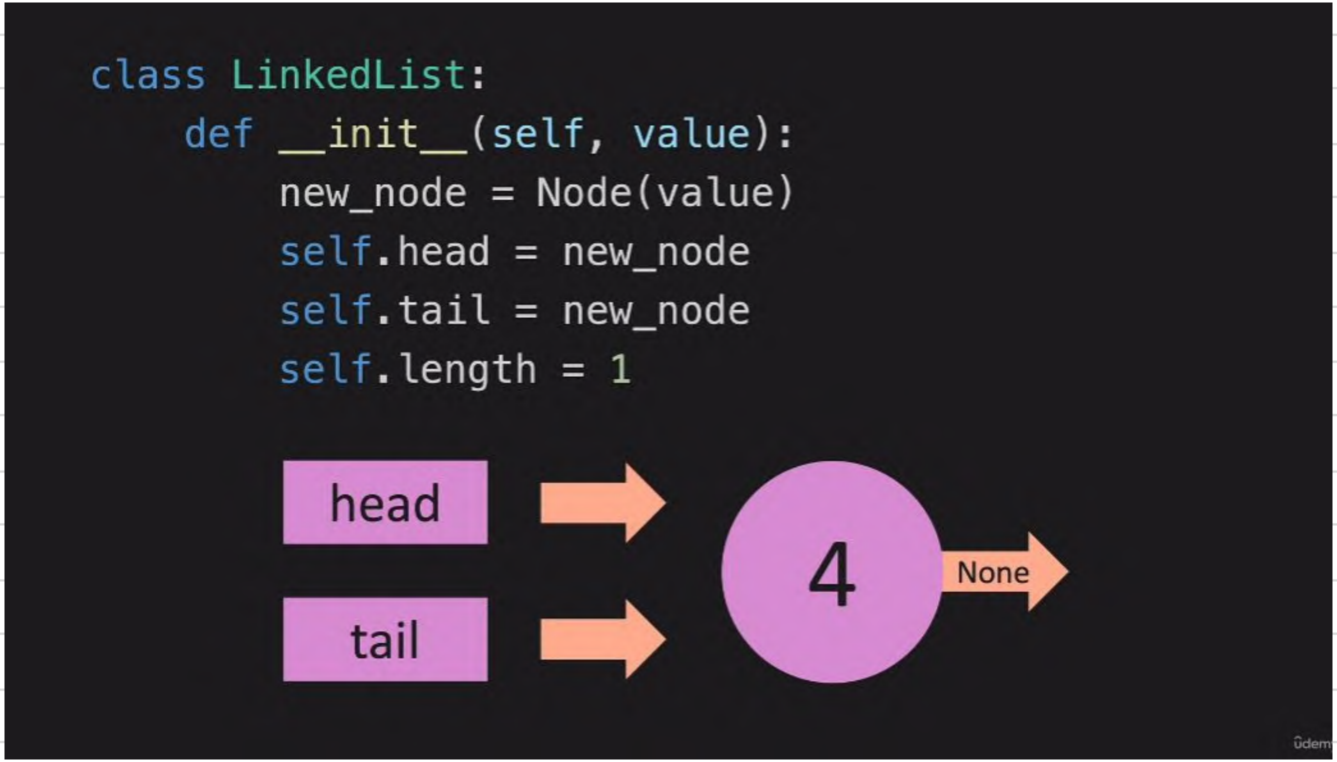
```python
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
```

Node: Class



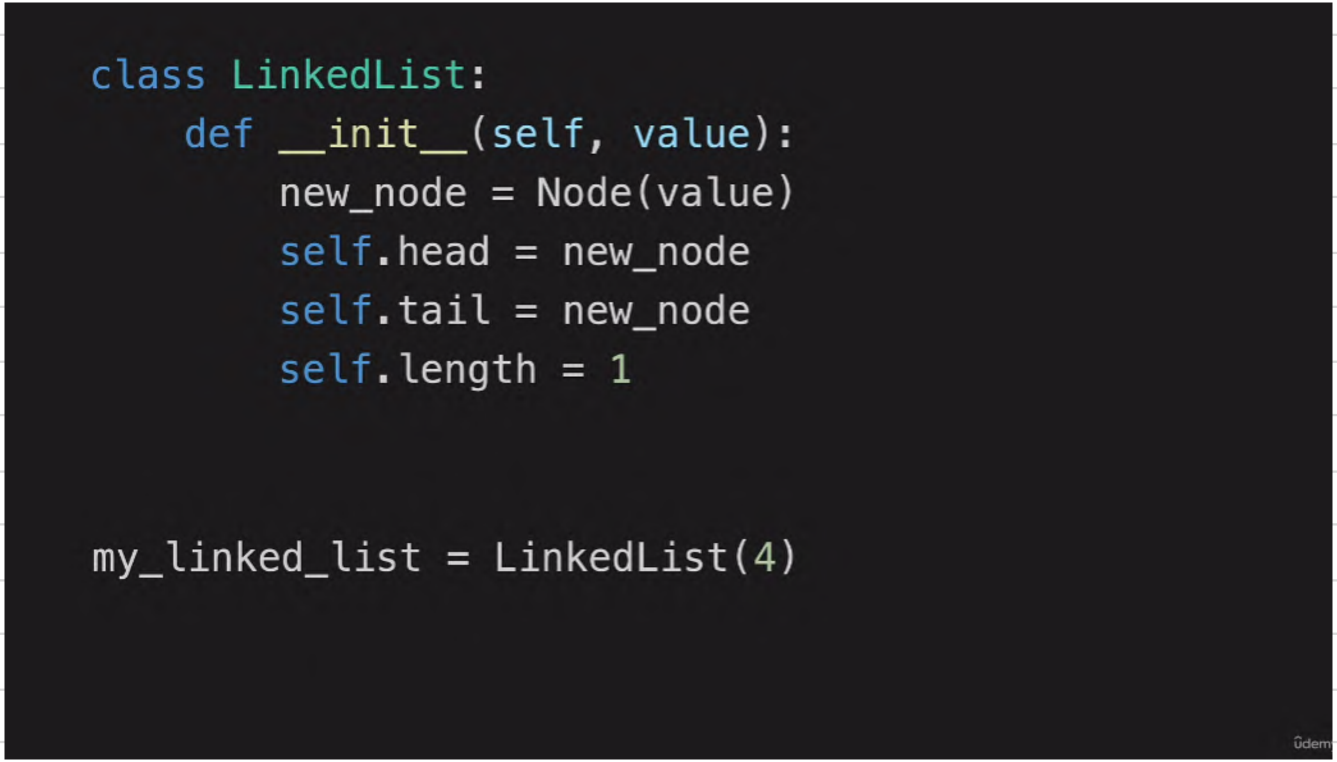
```python
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
```

Node: Class



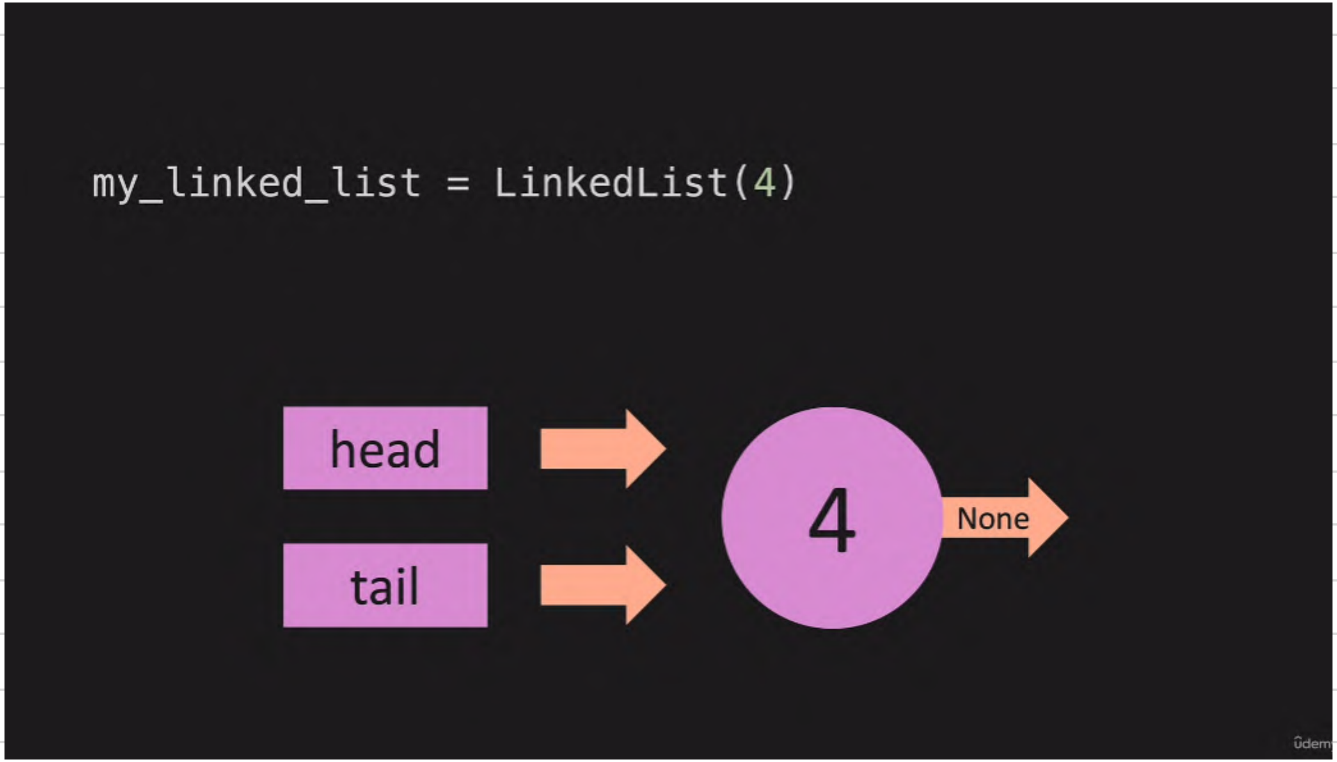
```
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1
```

Node: Class



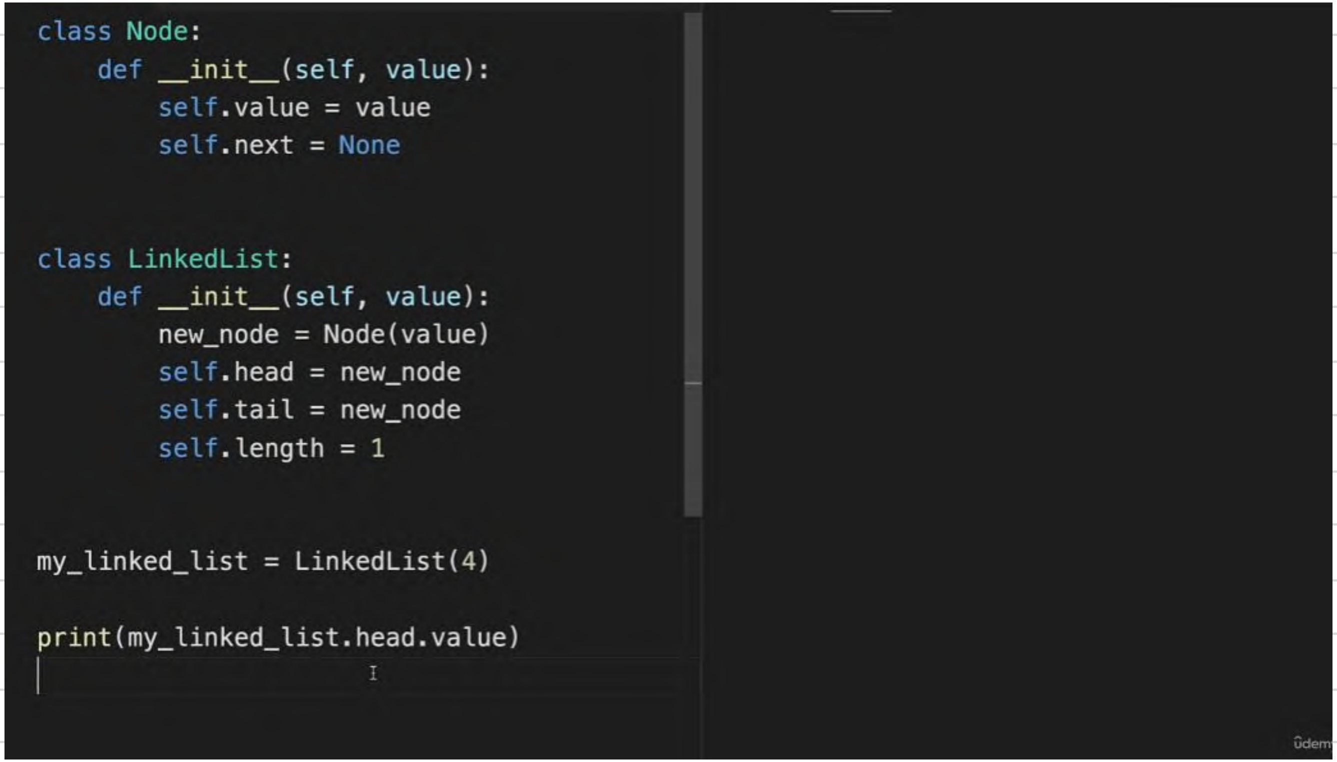
```
class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1


my_linked_list = LinkedList(4)
```

Node: Class



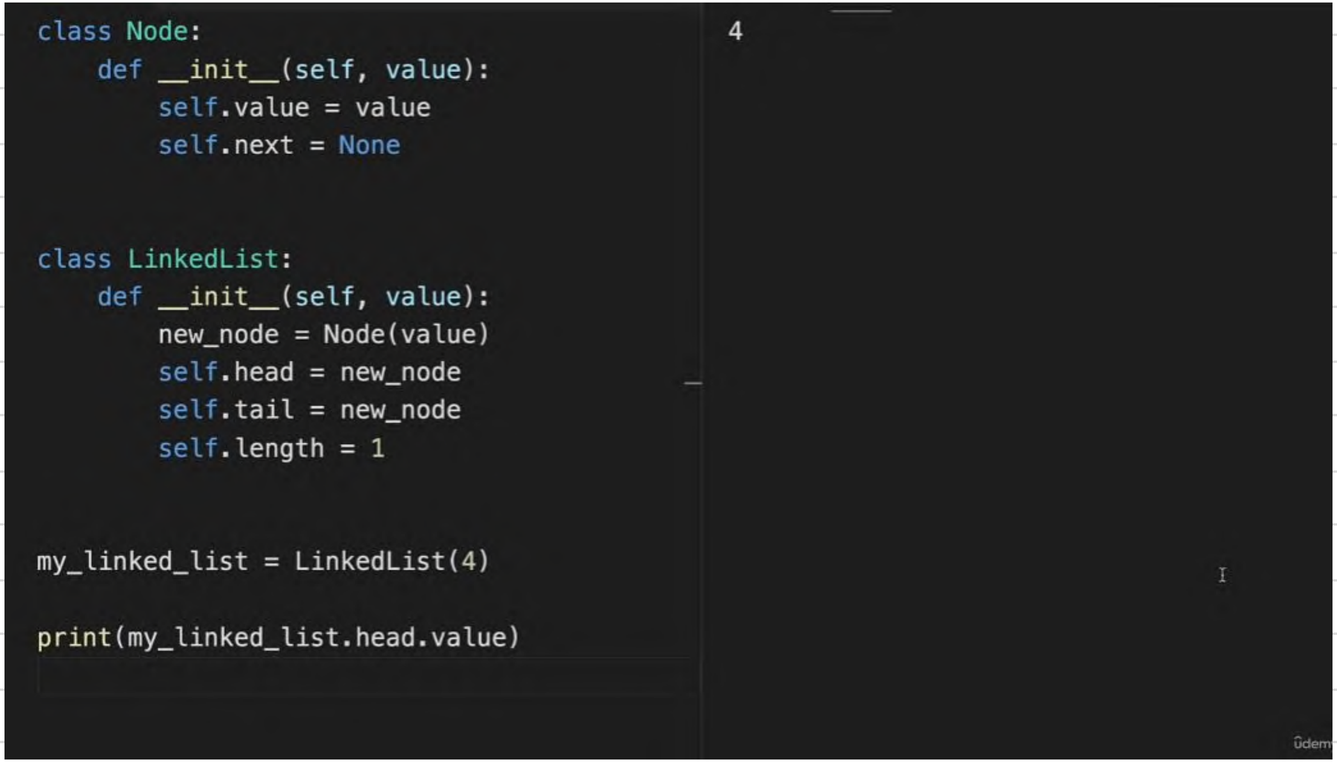
```
my_linked_list = LinkedList(4)
```

Linked List : Constructor



```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None


class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1


my_linked_list = LinkedList(4)

print(my_linked_list.head.value)
```

Linked List : Constructor



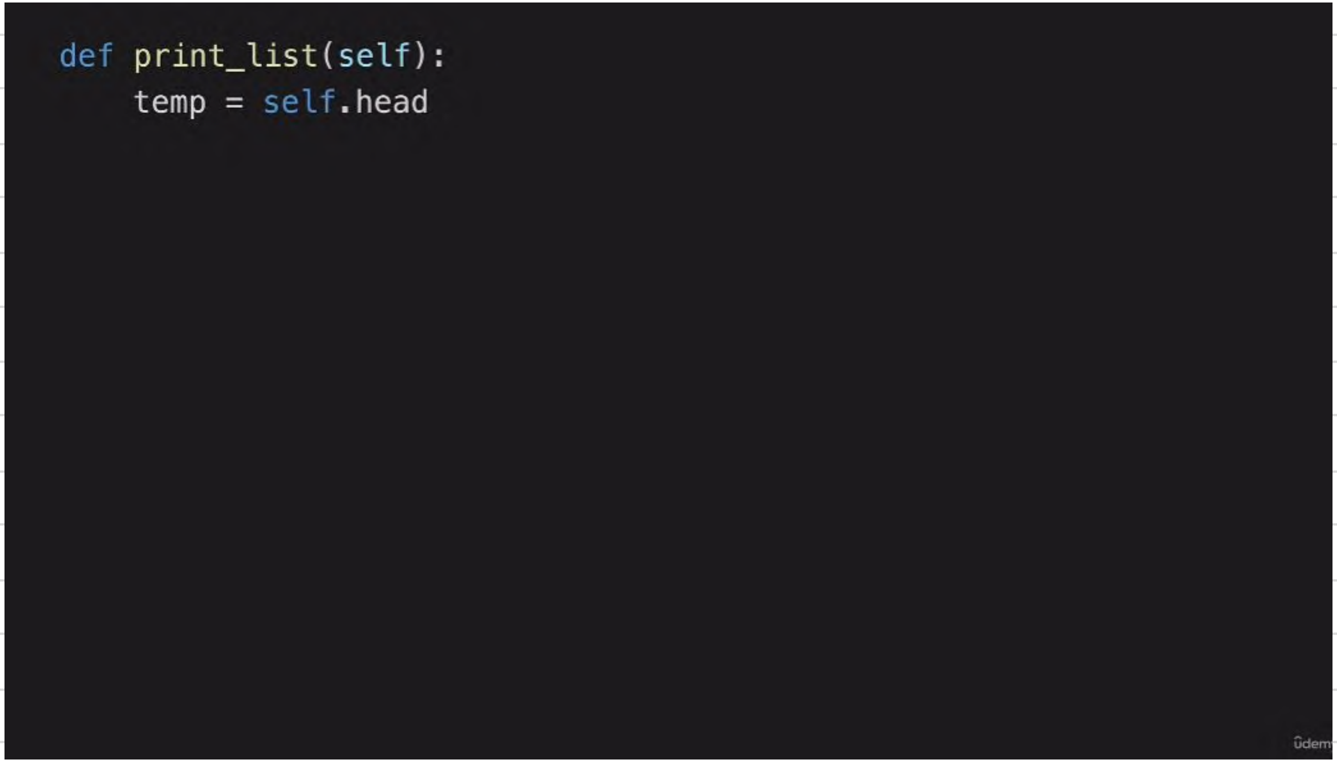
```
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None


class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1


my_linked_list = LinkedList(4)

print(my_linked_list.head.value)
```

```
4
```

05

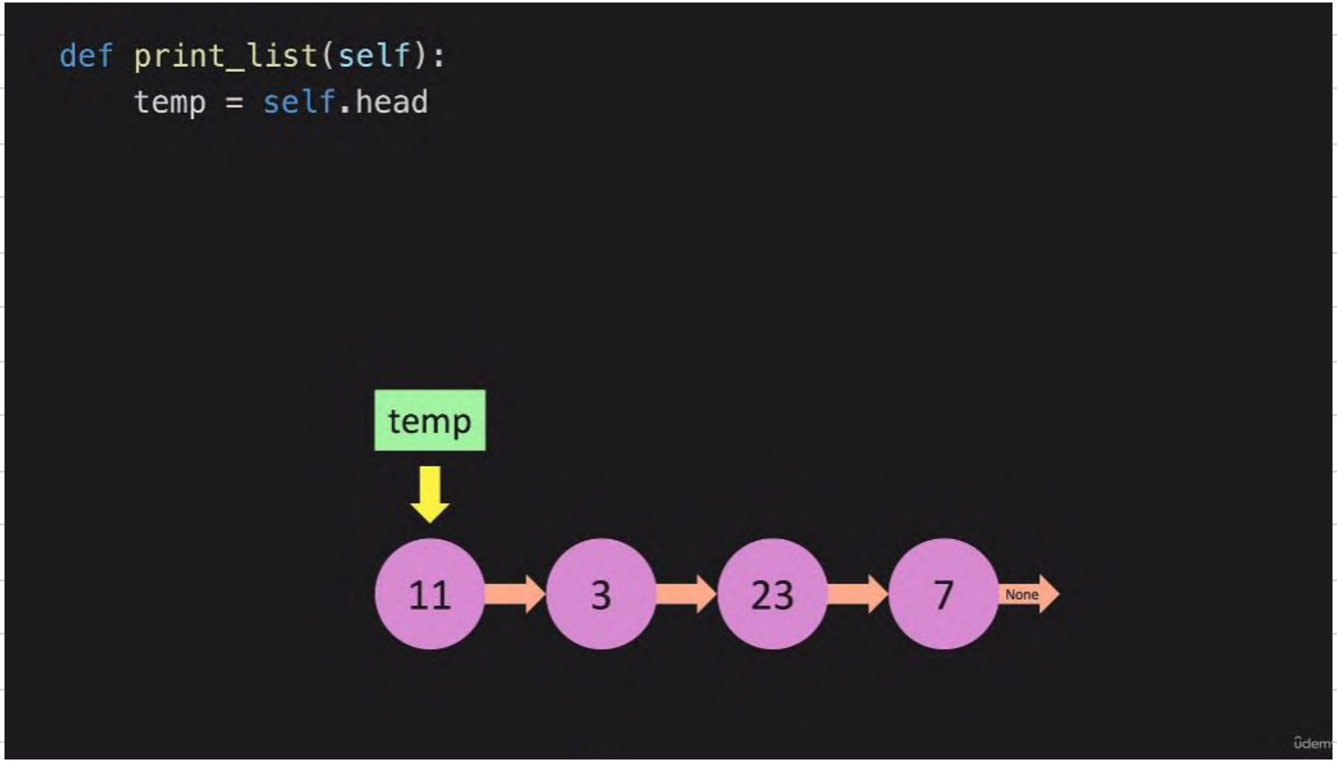
Linked List Print List

Linked List : Print List



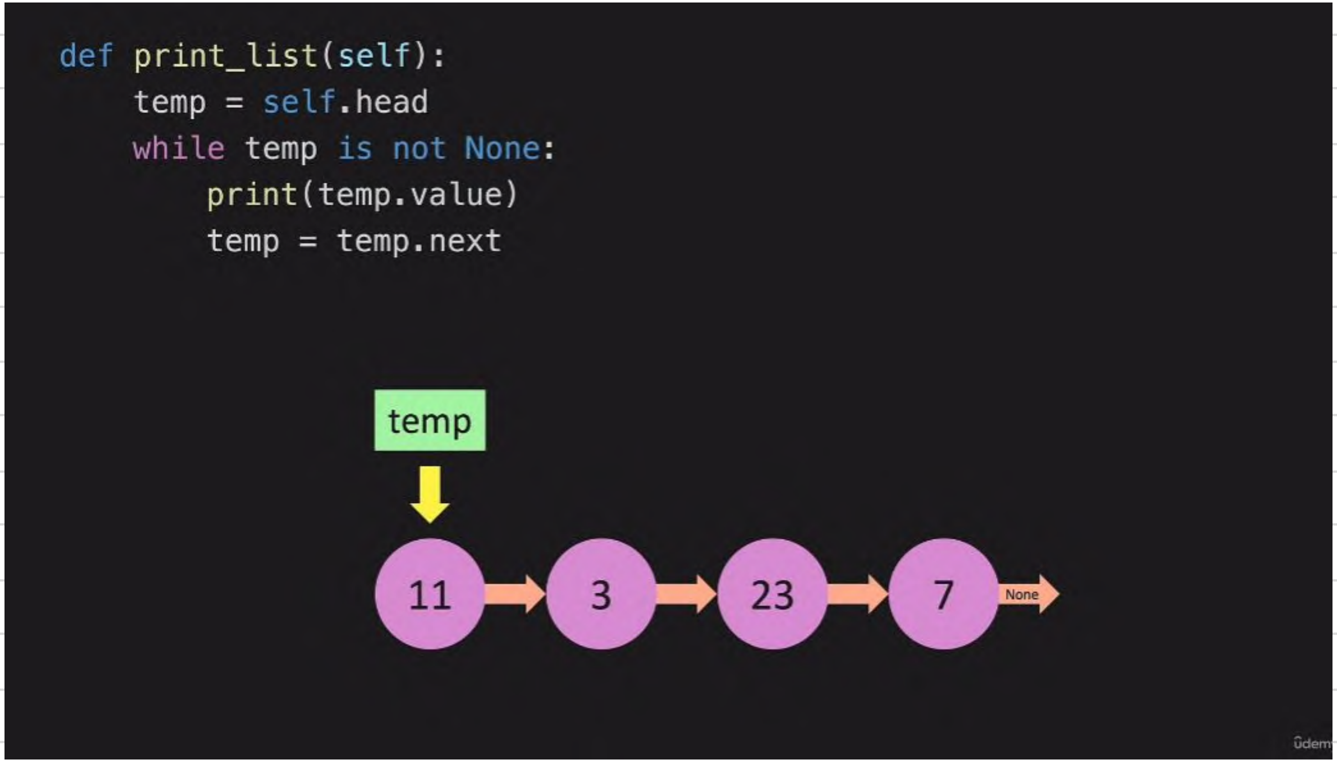
```
def print_list(self):
    temp = self.head
```

Linked List : Print List



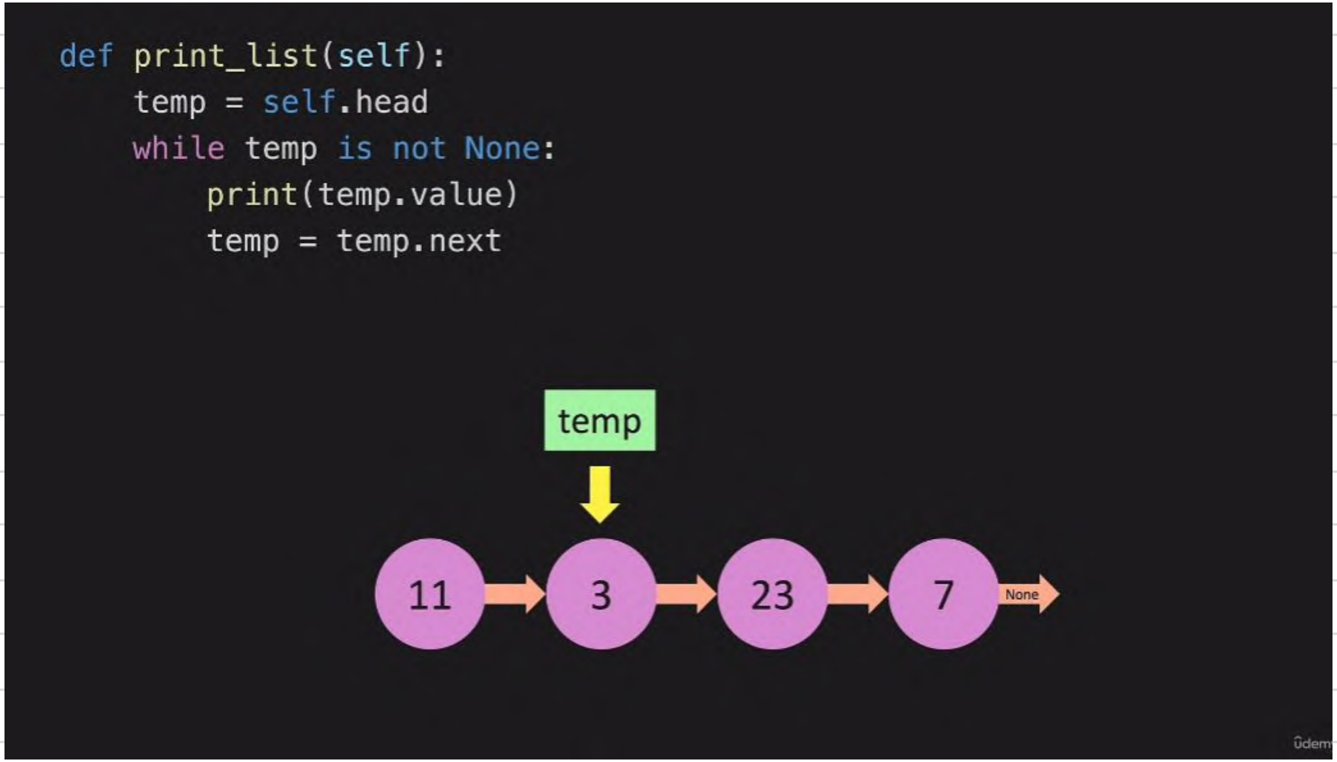
```
def print_list(self):
    temp = self.head
```

Linked List : Print List



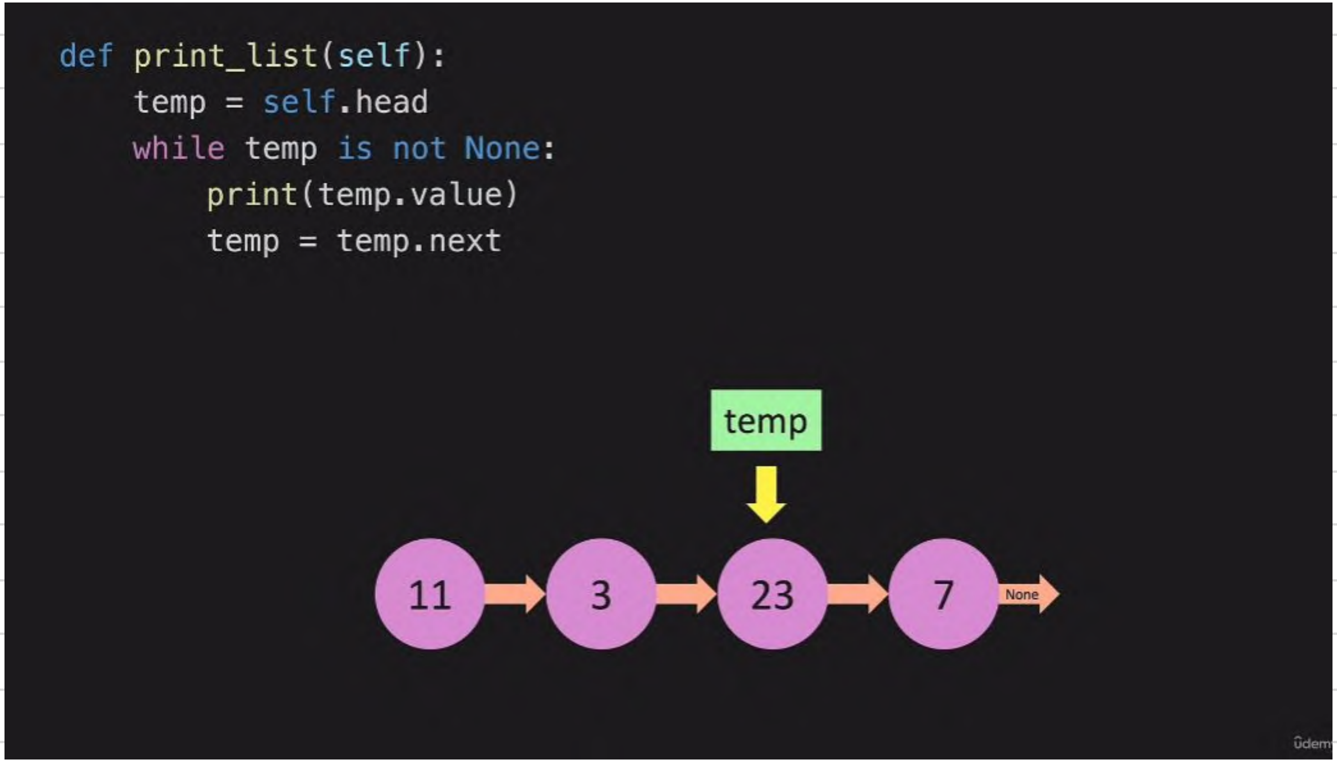
```
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```

Linked List : Print List



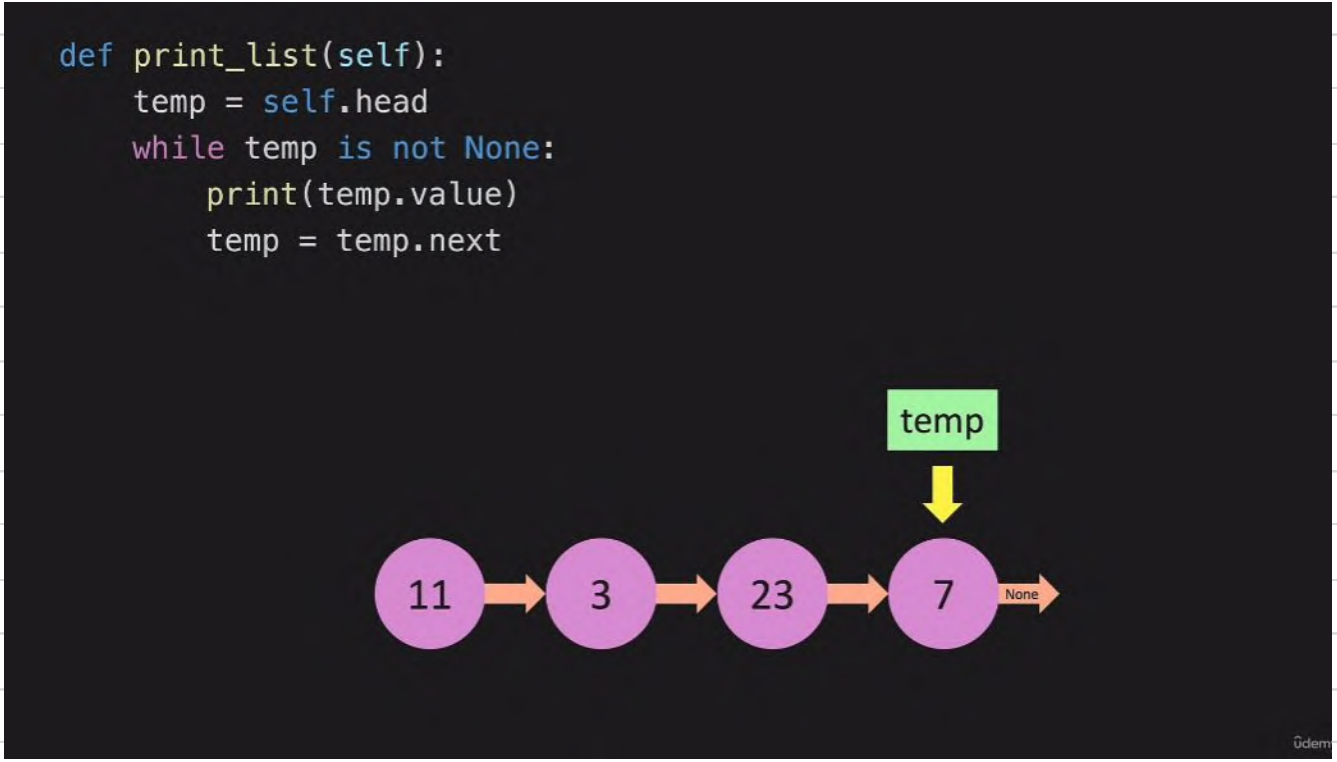
```
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```

Linked List : Print List



```
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```

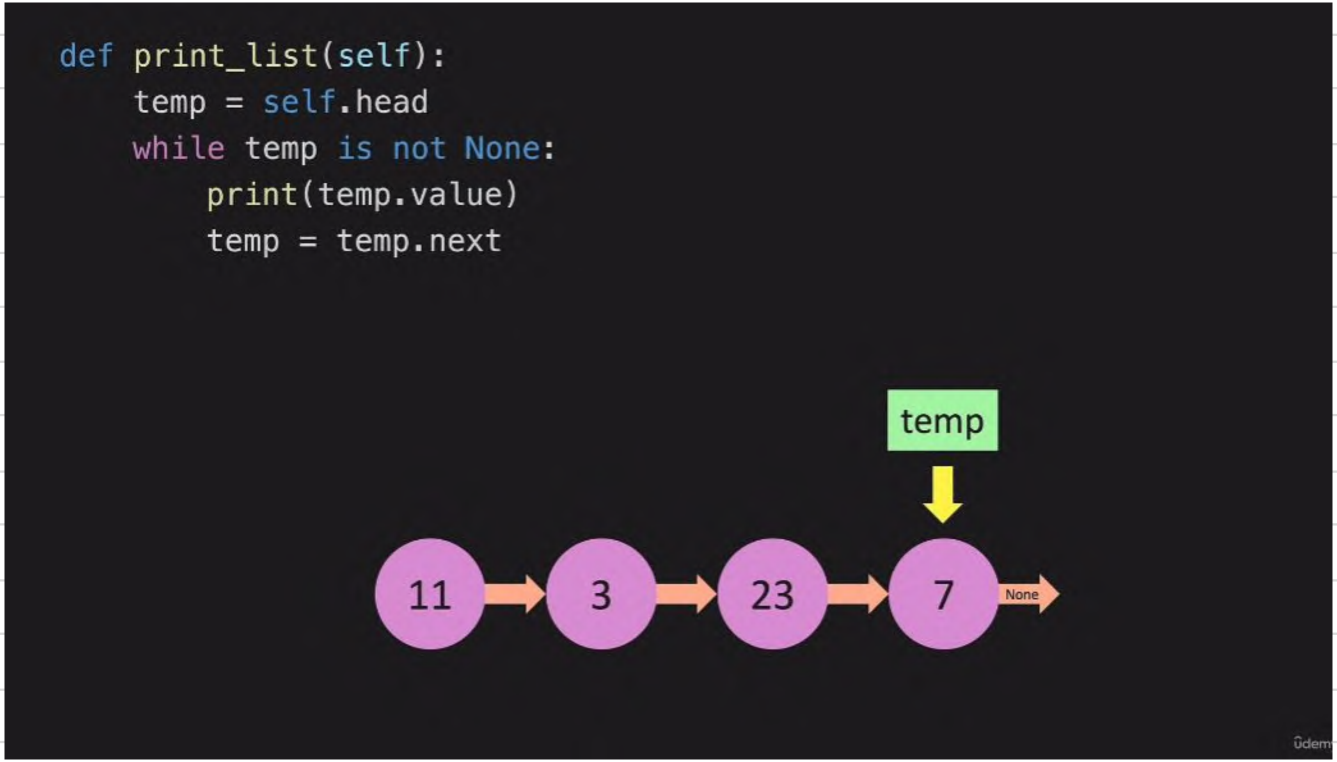
Linked List : Print List



```
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```

Linked List : Print List

Dimana posisi head pada akhir dari Print List?



```
def print_list(self):
    temp = self.head
    while temp is not None:
        print(temp.value)
        temp = temp.next
```

06

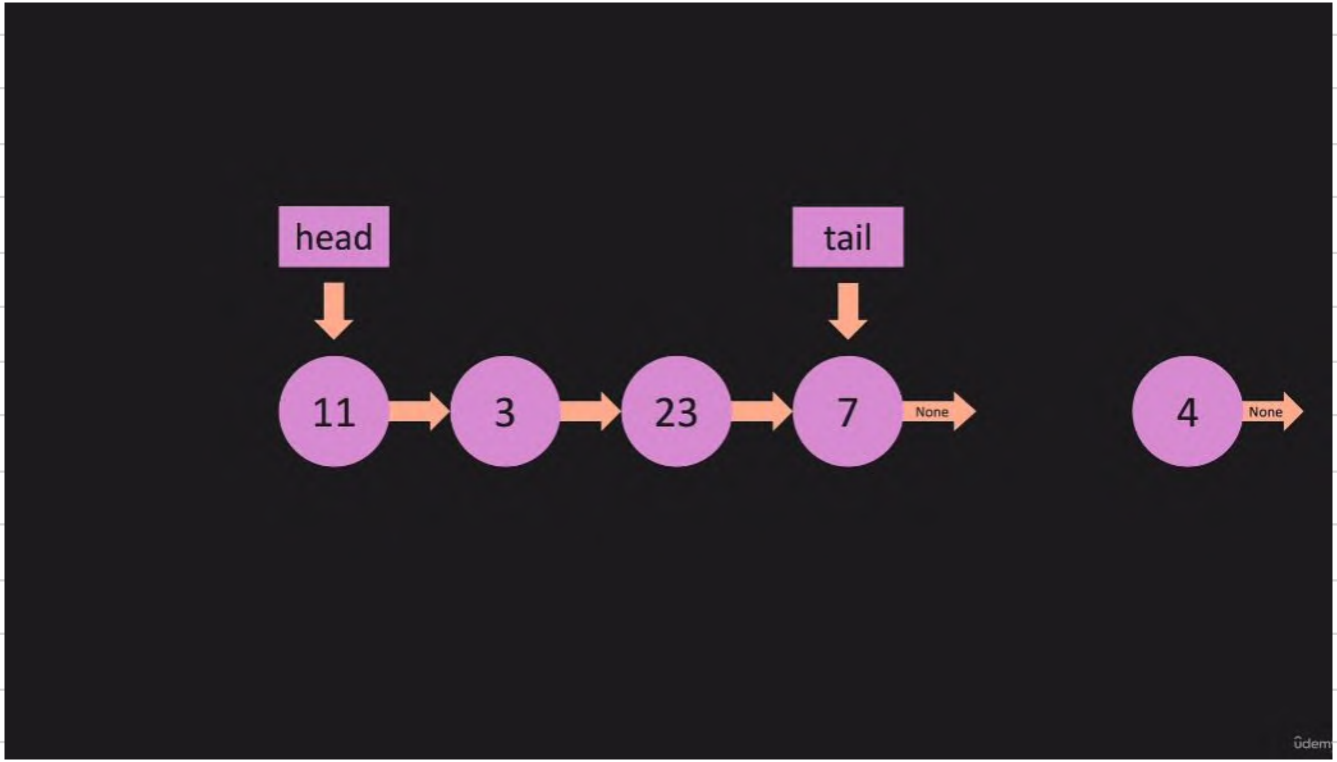
Linked List Append

Linked List : Append

- Apa itu Append…?

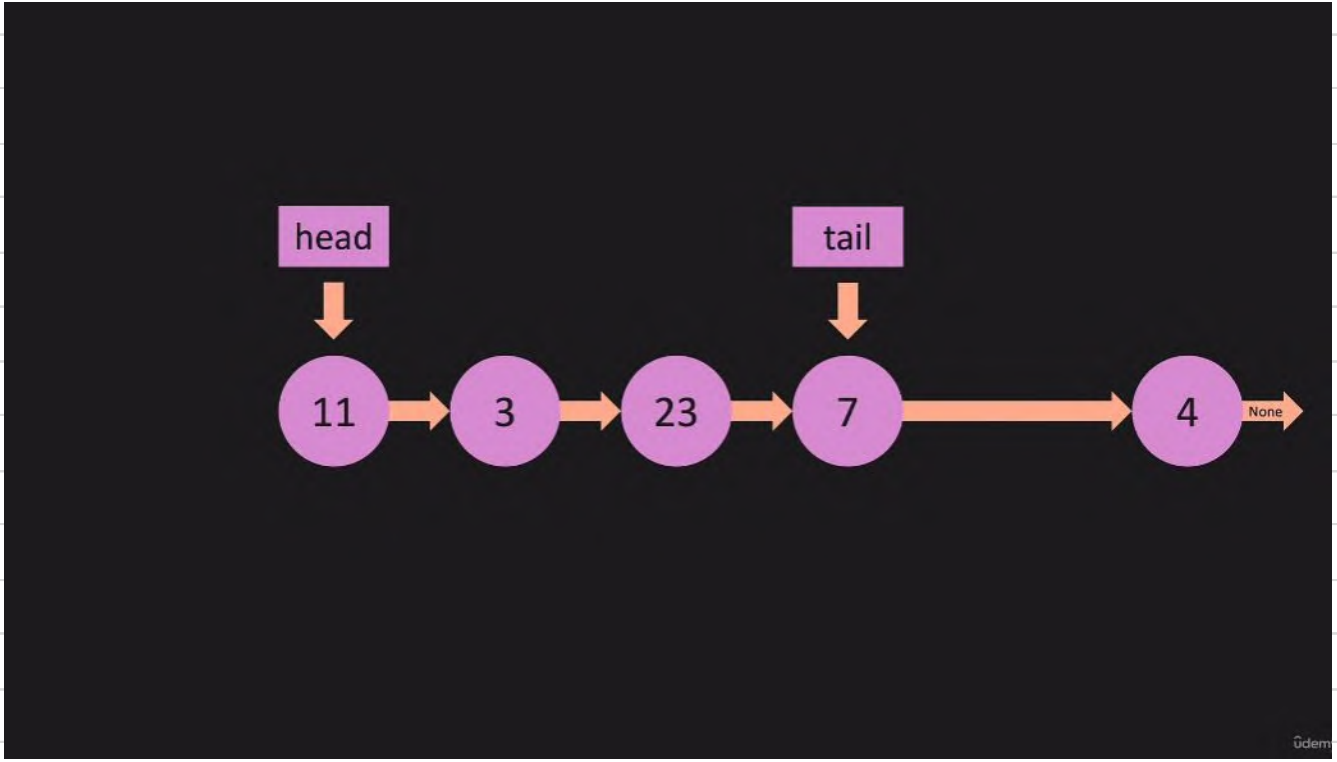
Buat node baru

Linked List : Append



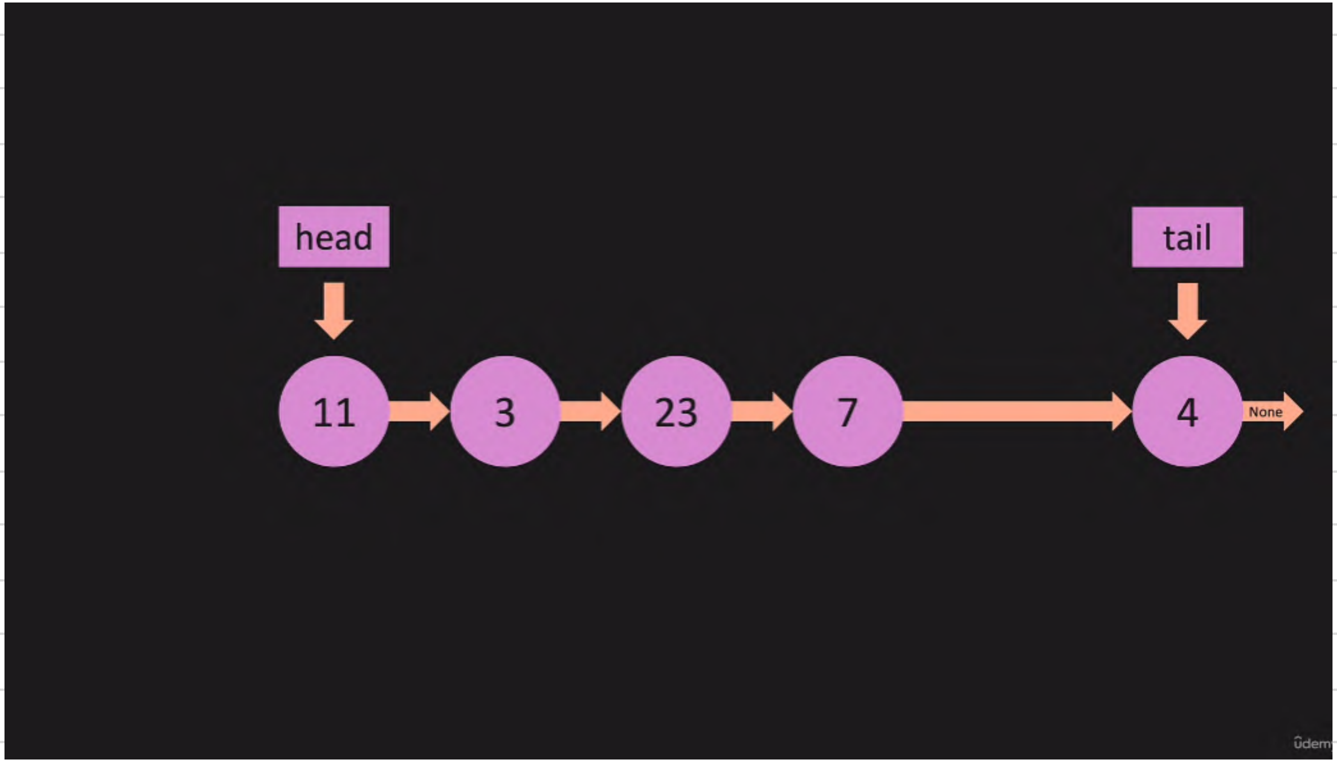
Arahkan tail.next ke node baru

Linked List : Append



Arahkan tail ke node baru

Linked List : Append

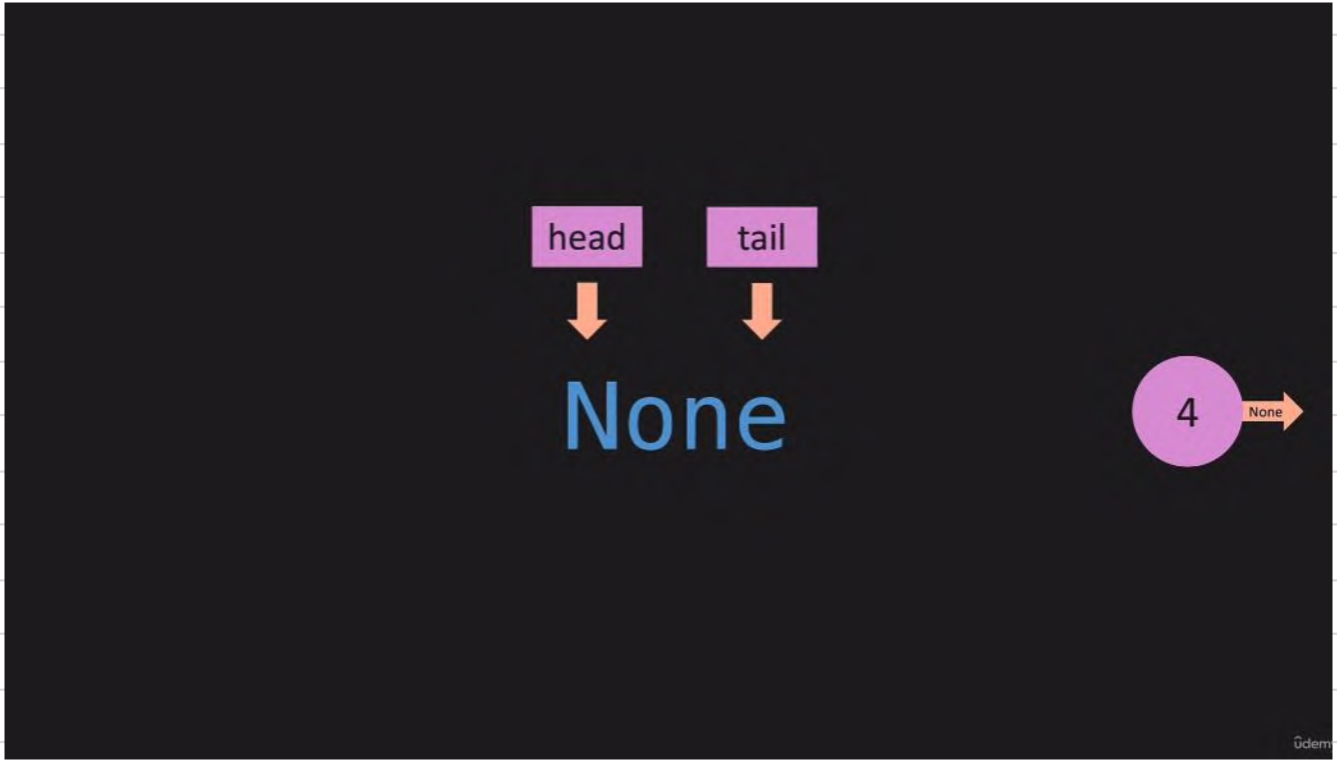


Linked List : Append

- Namun ada beberapa kondisi yang harus mendapatkan perhatian khusus

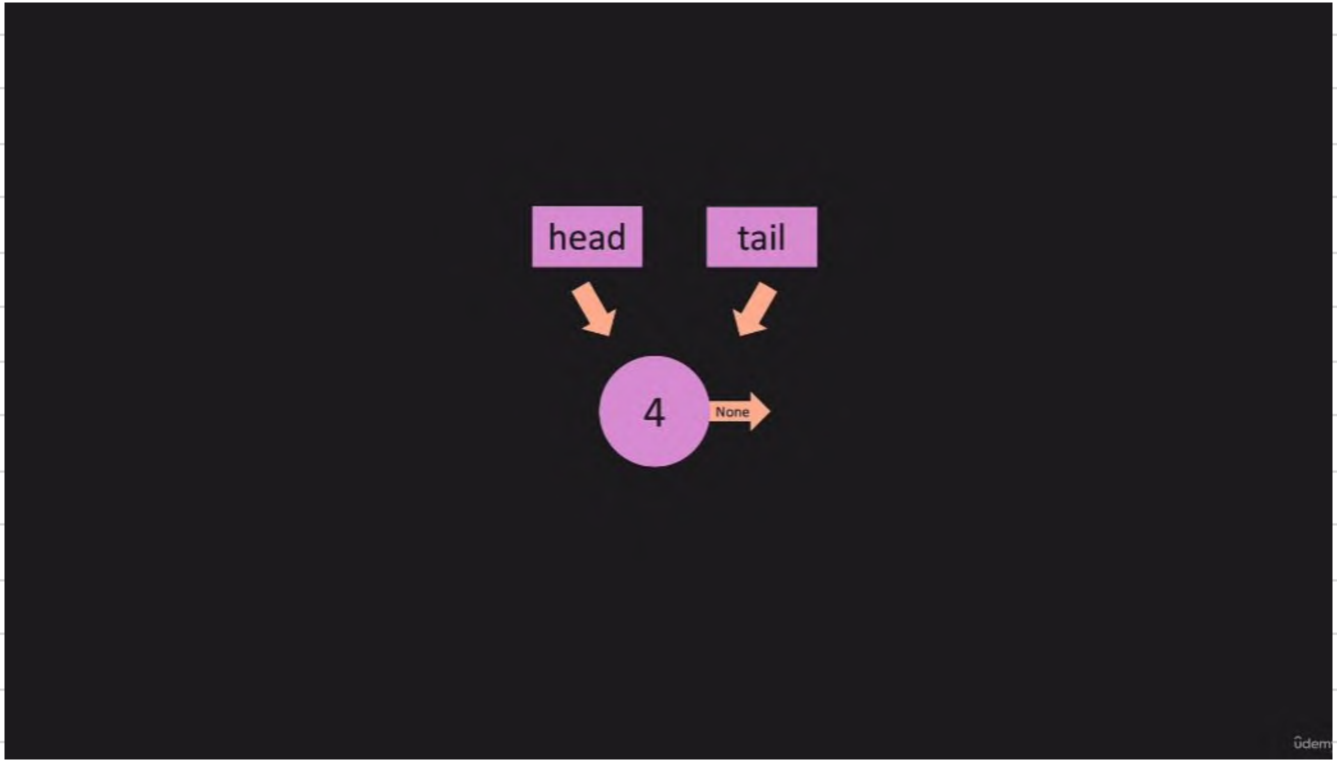
Bagaimana append bekerja?

Linked List : Append

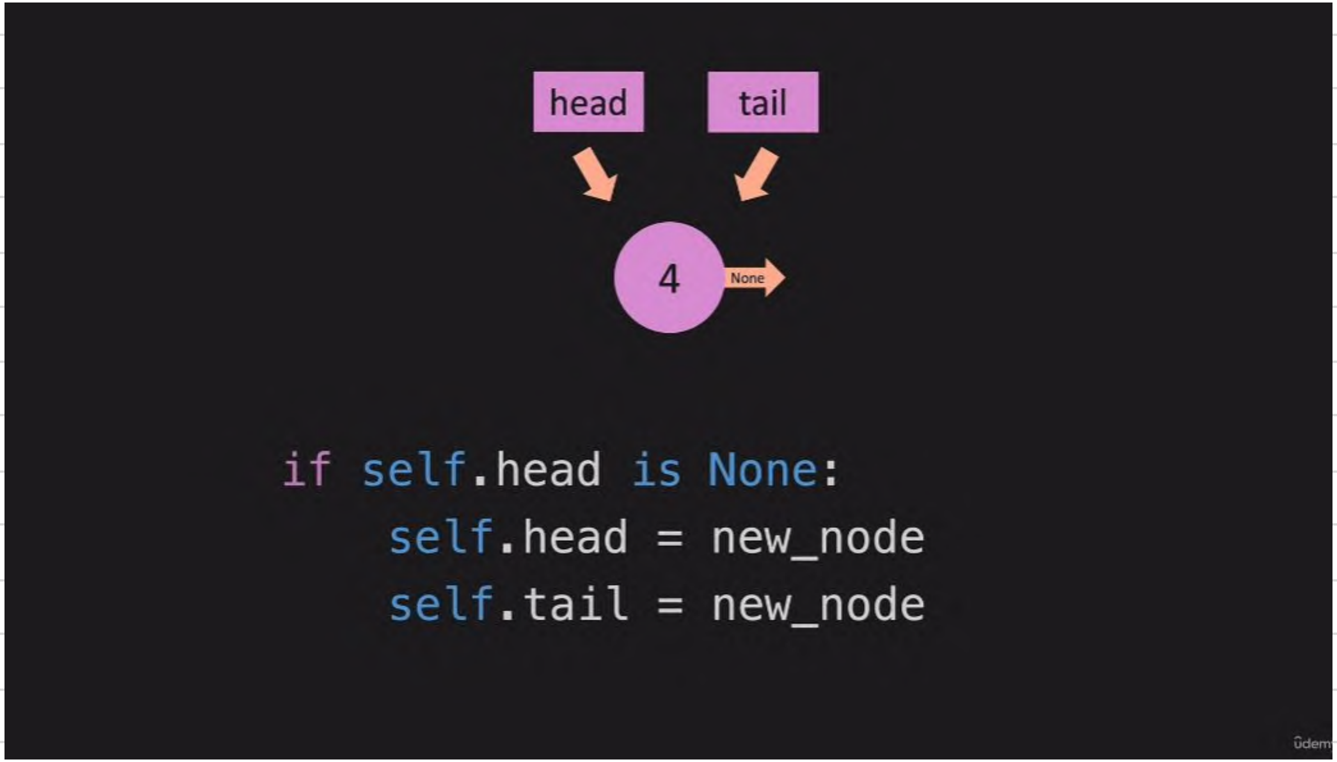


Bagaimana append bekerja?

Linked List : Append

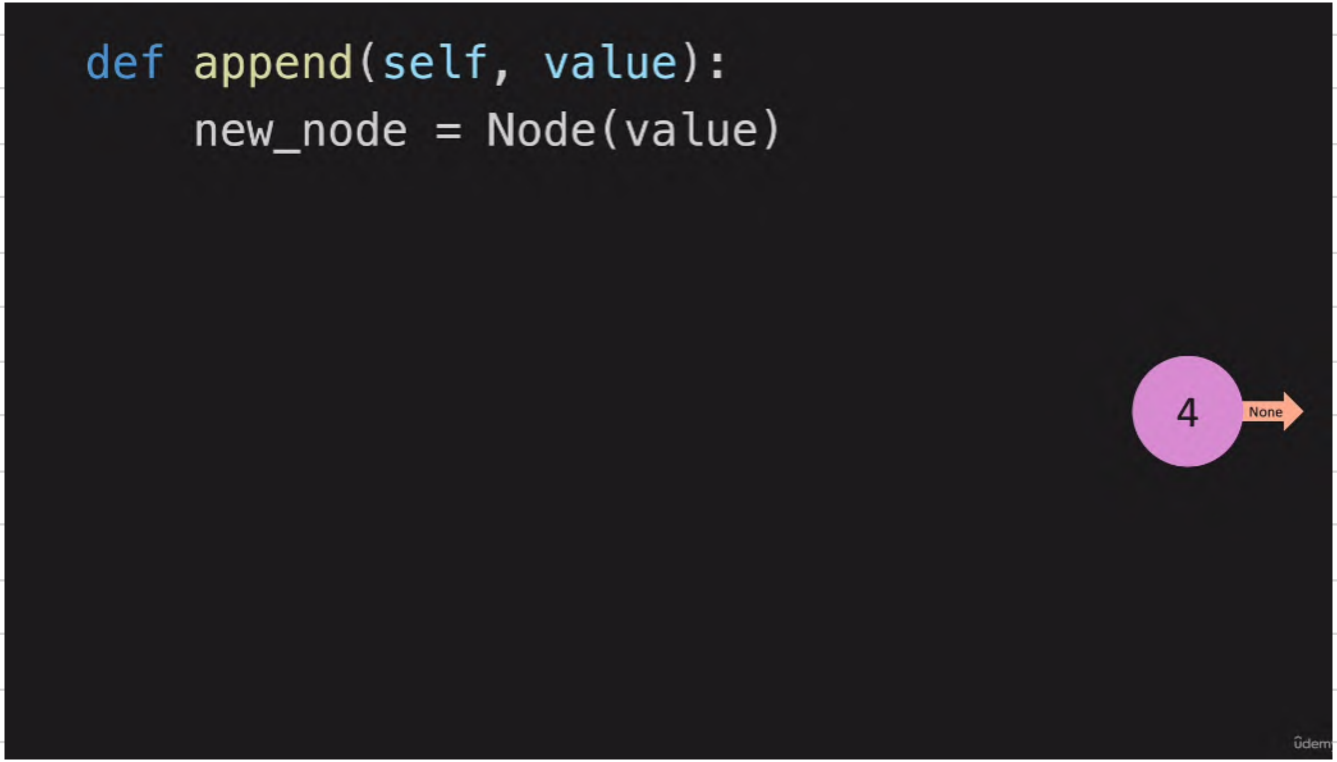


Linked List : Append



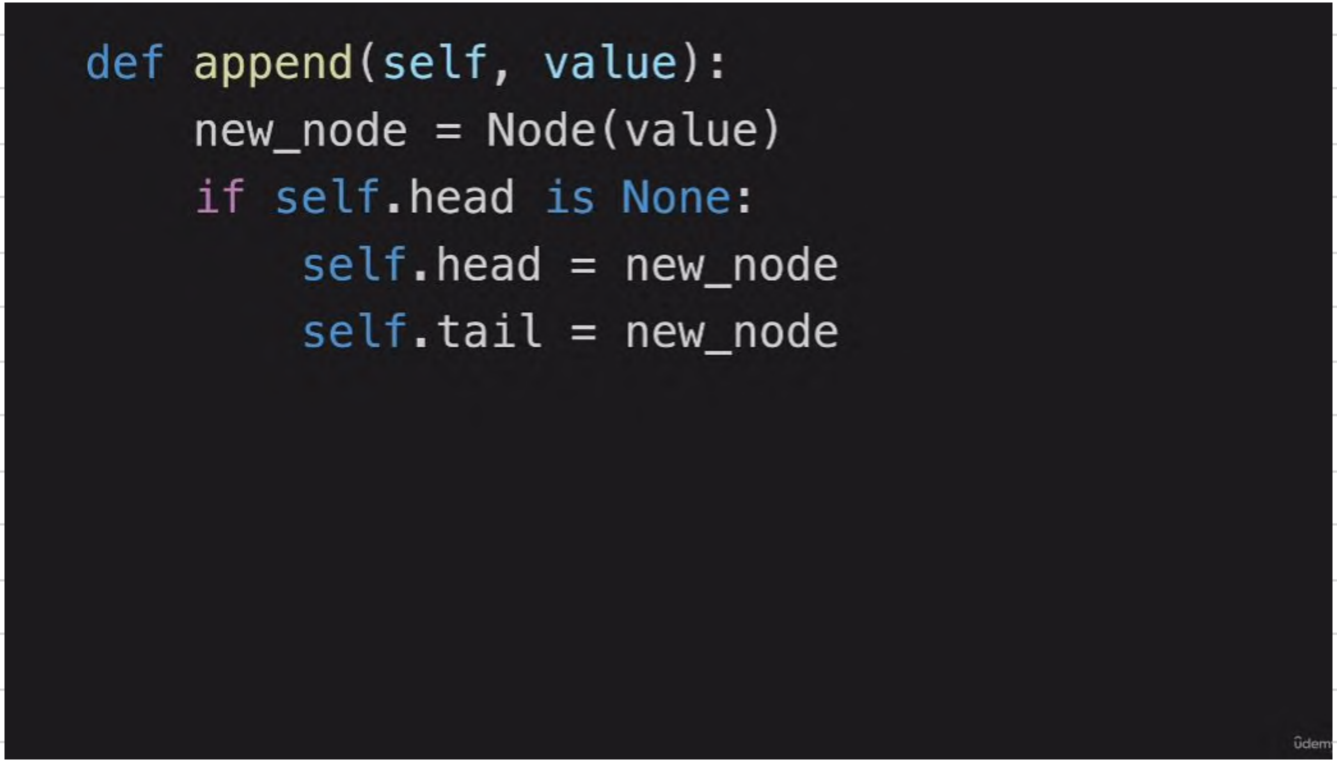
```
if self.head is None:
    self.head = new_node
    self.tail = new_node
```

Linked List : Append Function



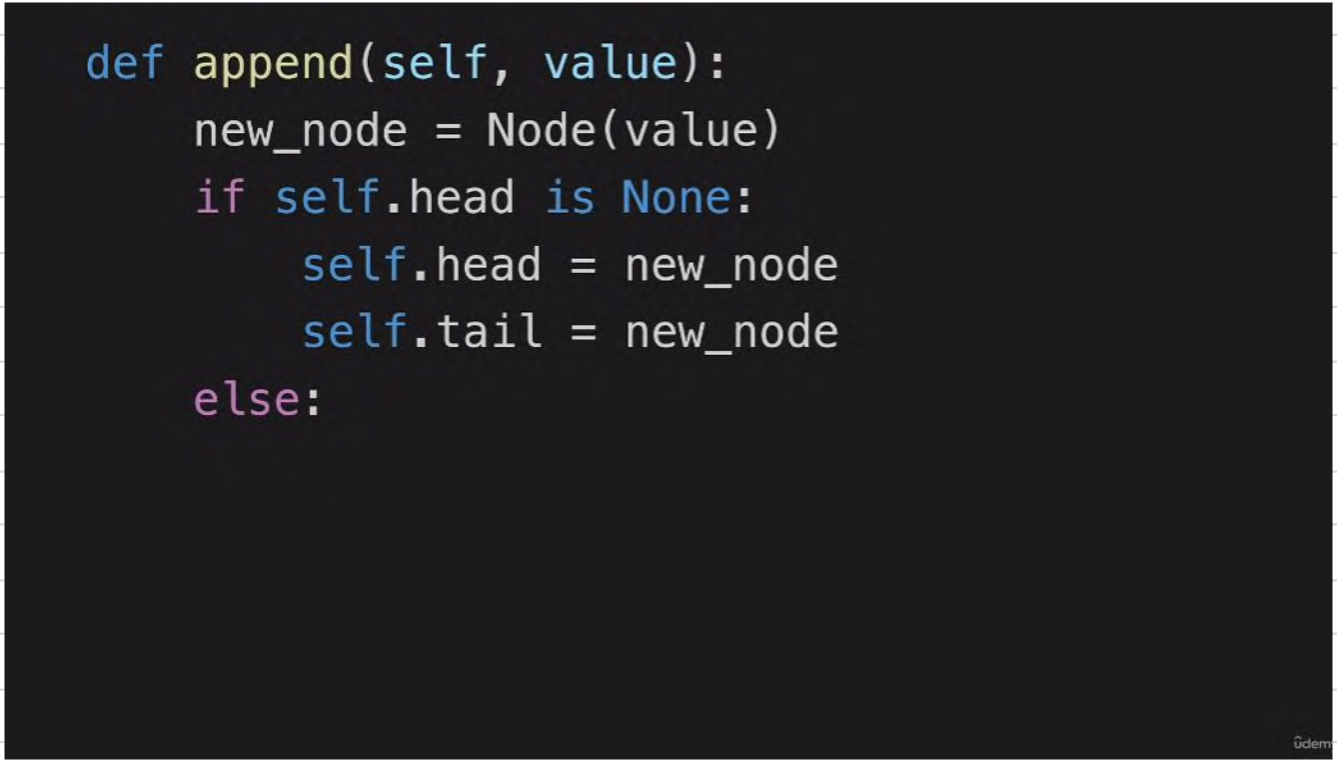
```
def append(self, value):
    new_node = Node(value)
```

Linked List : Append Function



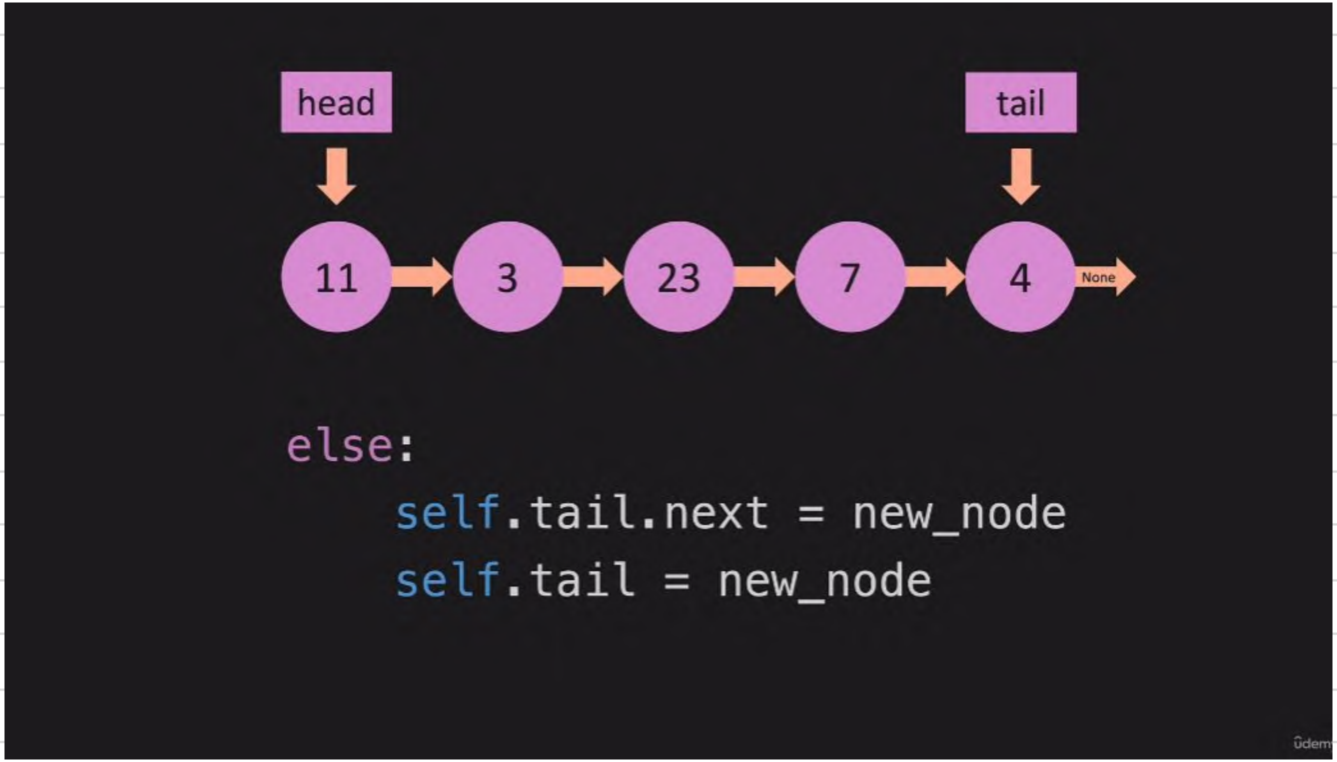
```
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
```

Linked List : Append Function



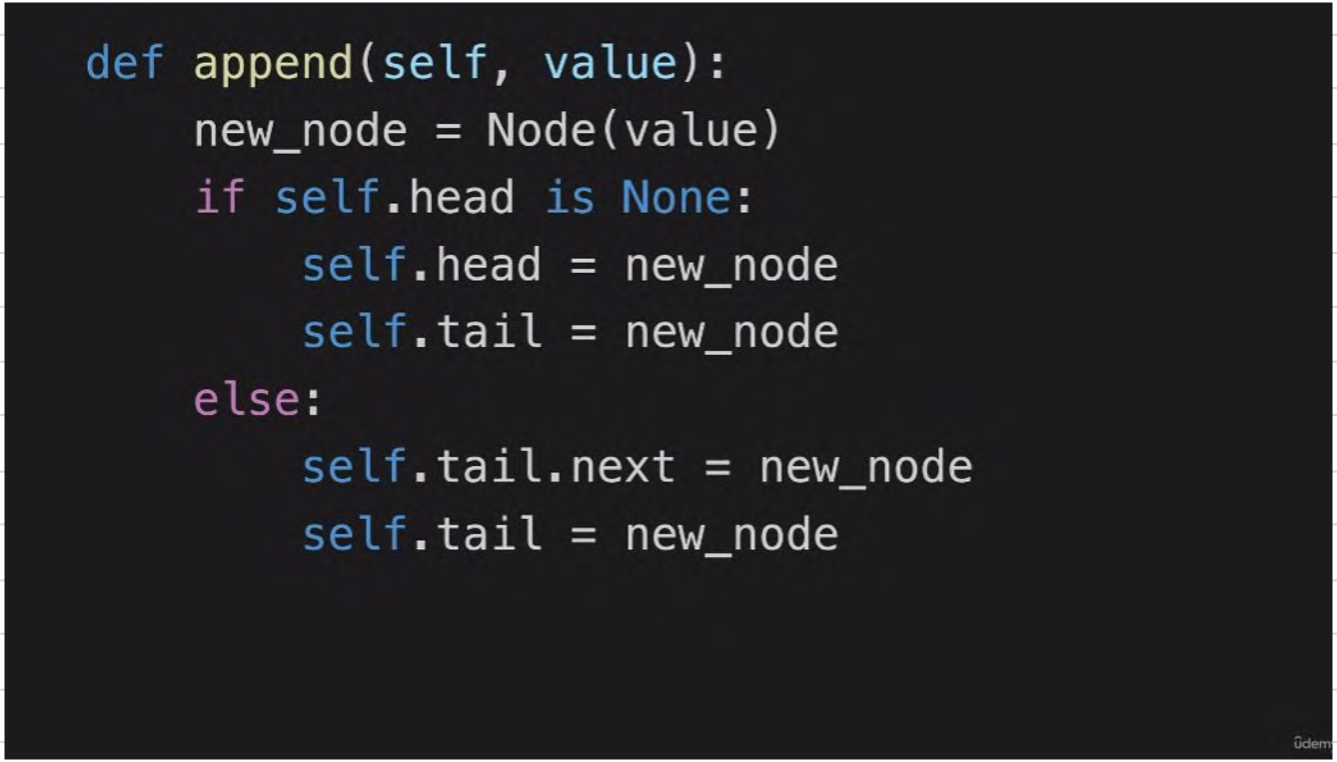
```
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
    else:
```

Linked List : Append Function



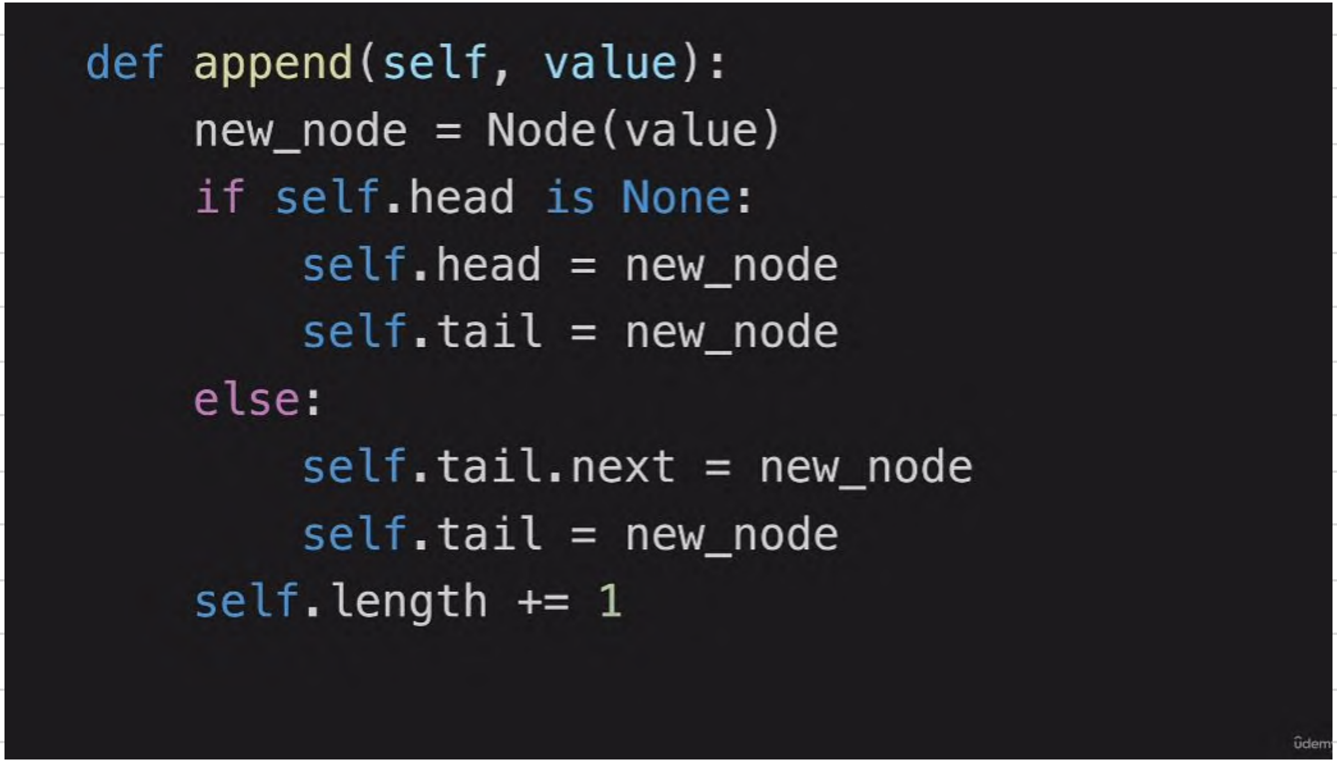
```
else:
    self.tail.next = new_node
    self.tail = new_node
```

Linked List : Append Function



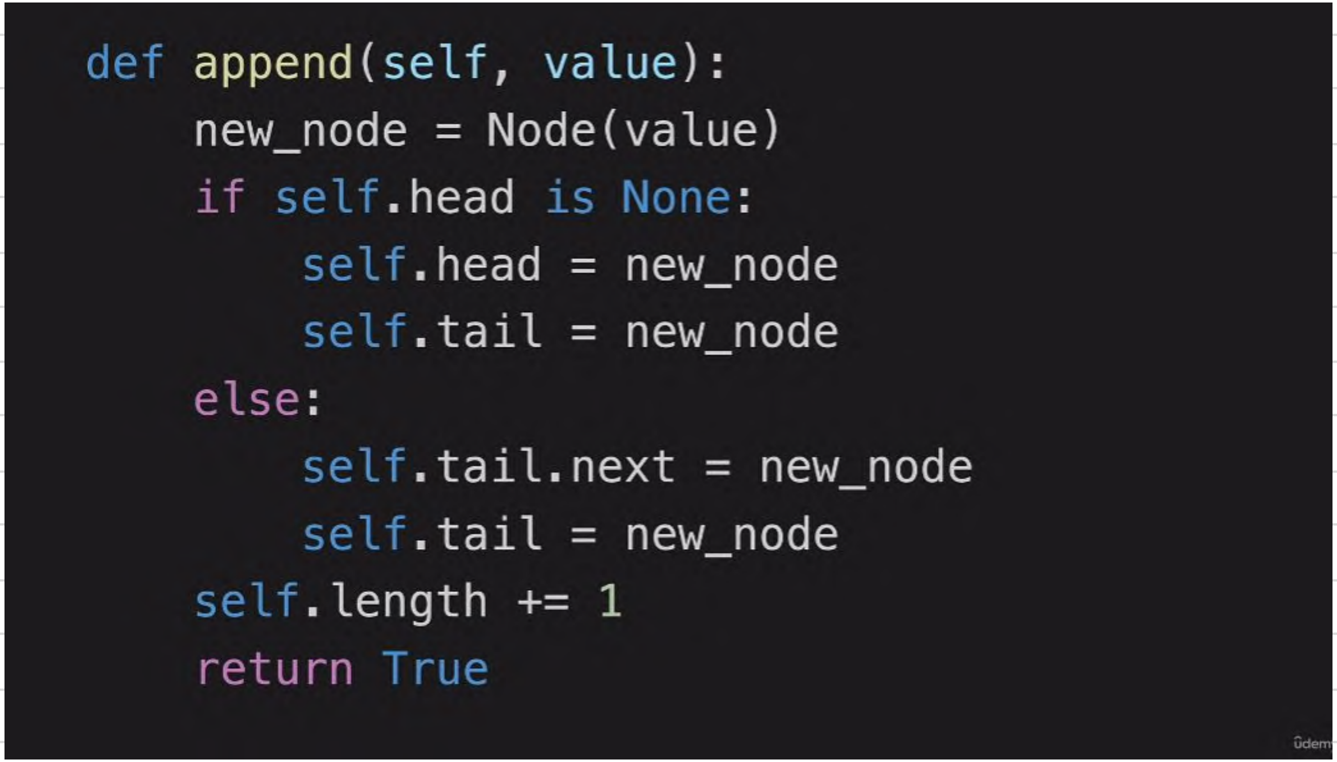
```
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
    else:
        self.tail.next = new_node
        self.tail = new_node
```

Linked List : Append Function



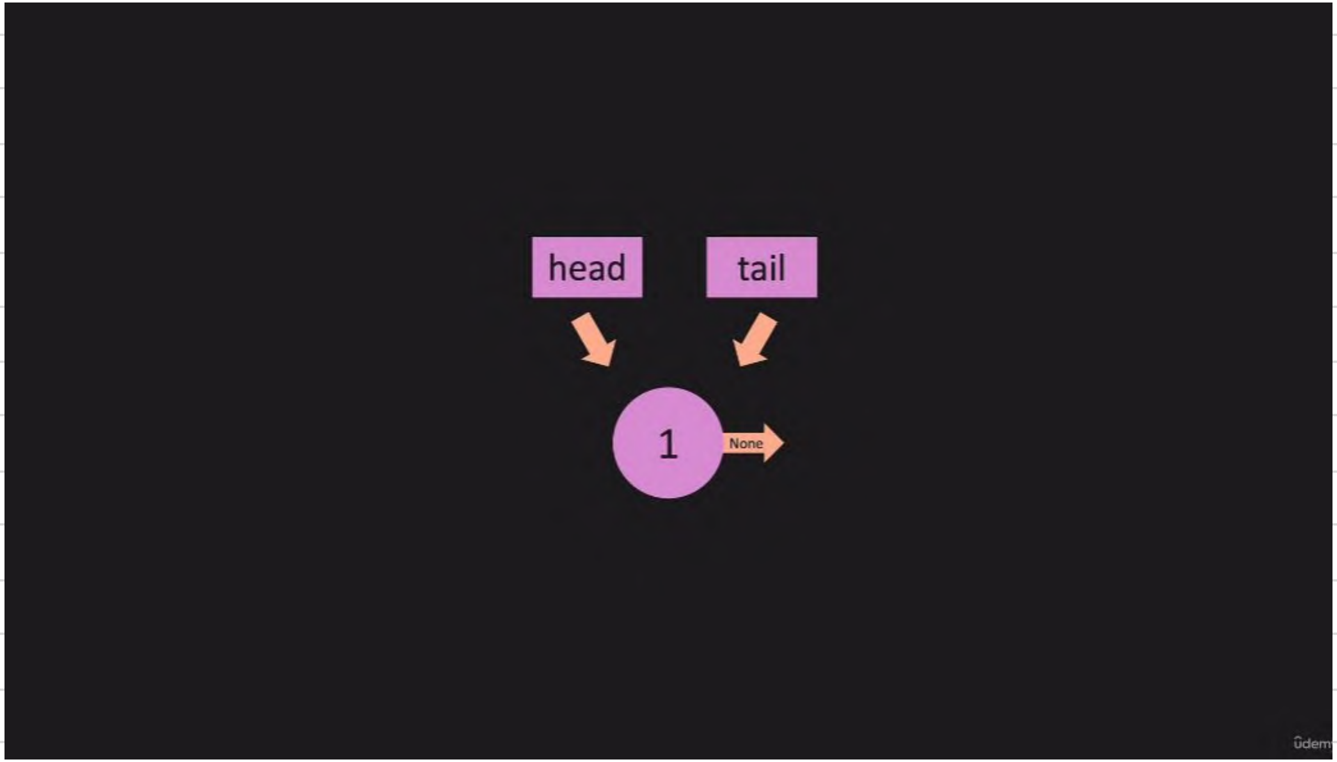
```
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
    else:
        self.tail.next = new_node
        self.tail = new_node
    self.length += 1
```

Linked List : Append Function

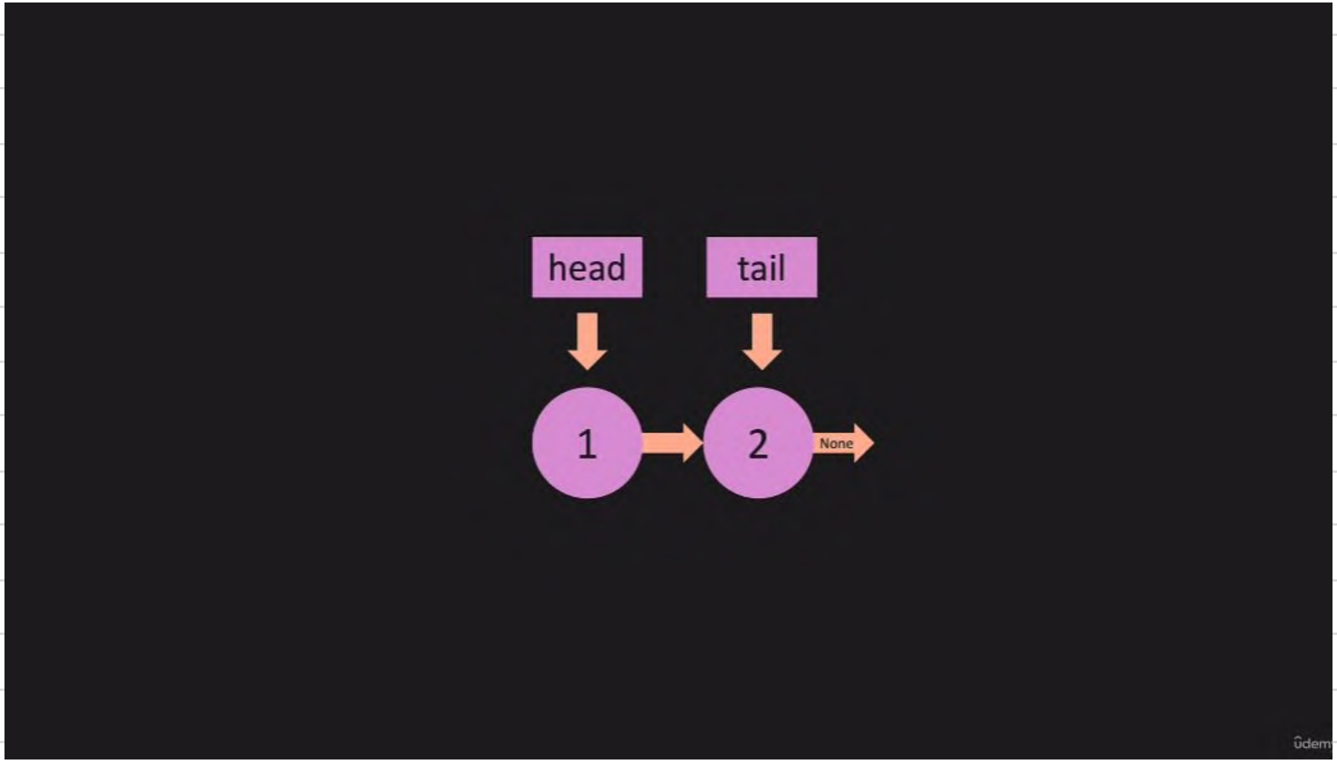


```
def append(self, value):
    new_node = Node(value)
    if self.head is None:
        self.head = new_node
        self.tail = new_node
    else:
        self.tail.next = new_node
        self.tail = new_node
    self.length += 1
    return True
```

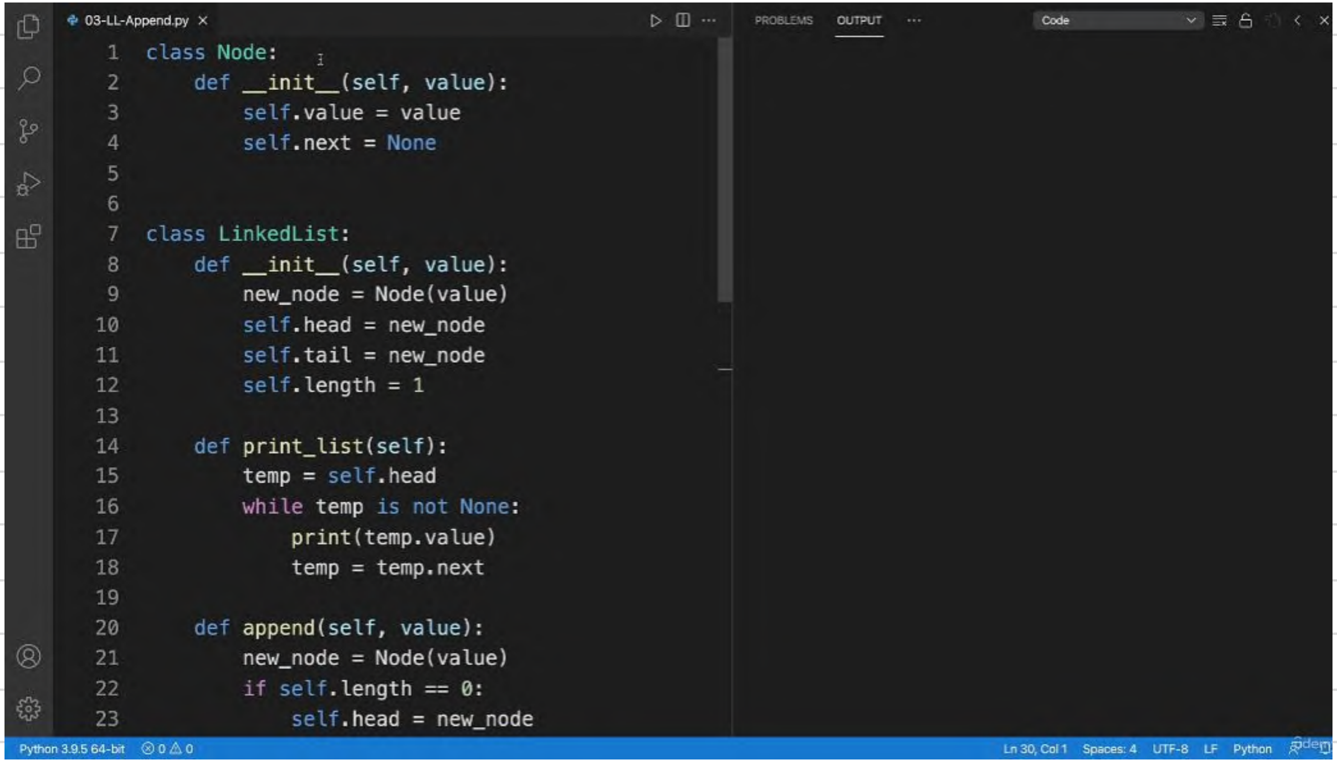
Linked List : Append Skenario Percobaan



Linked List : Append Skenario Percobaan



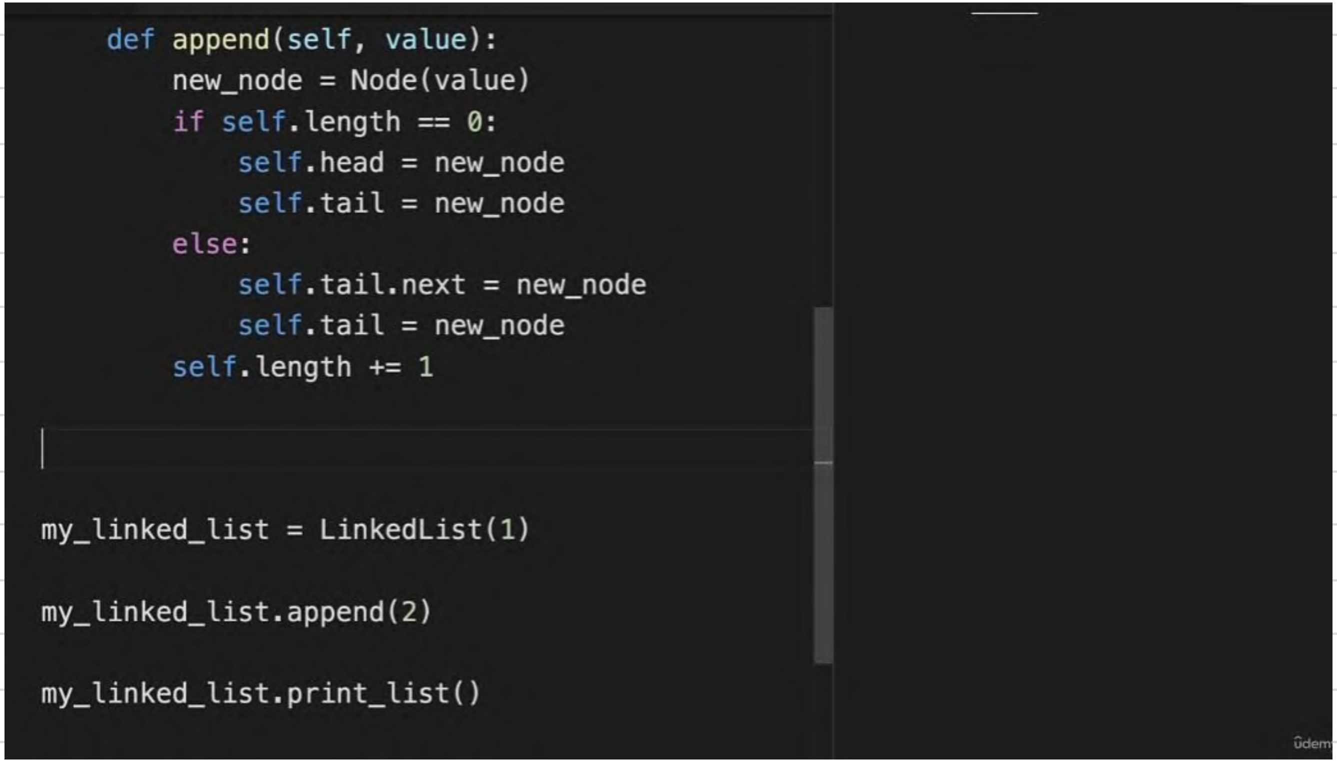
Linked List : Append Skenario Percobaan



03-LL-Append.py

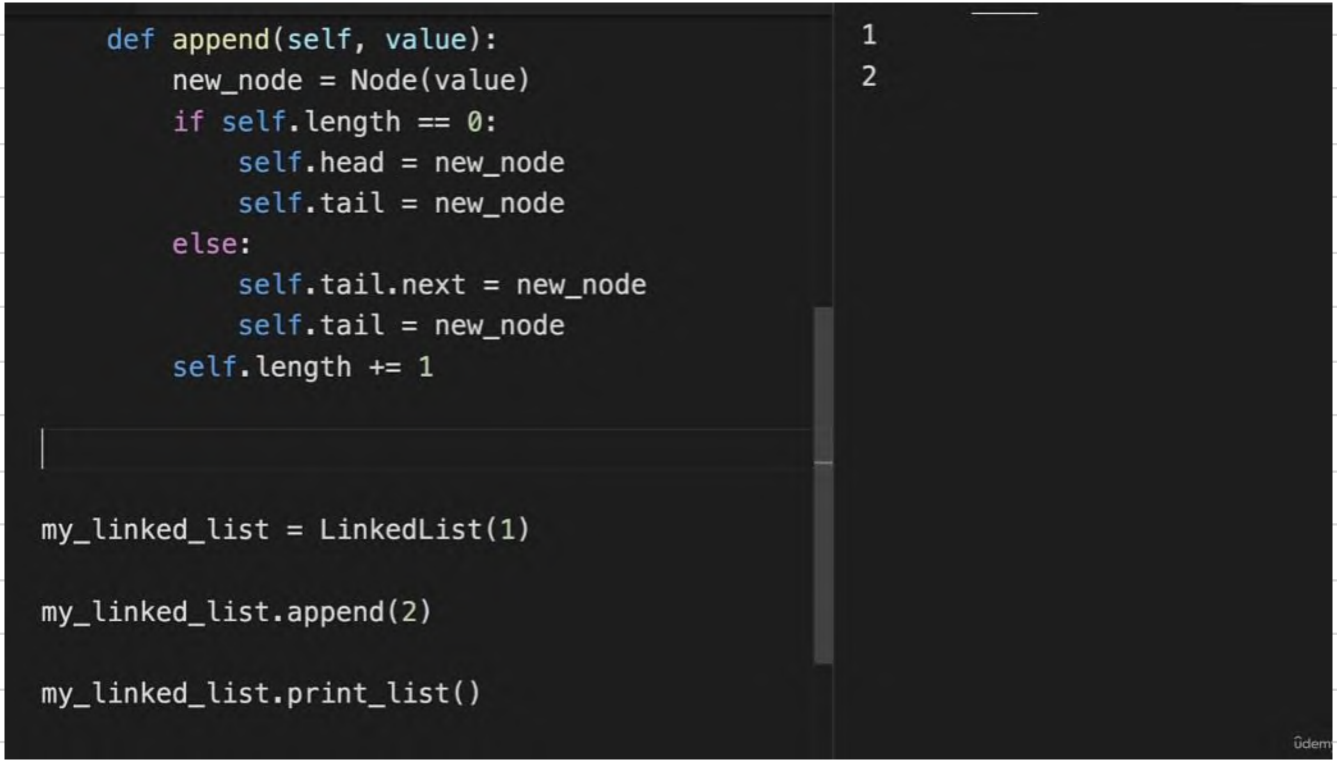
```python
class Node:
    def __init__(self, value):
        self.value = value
        self.next = None


class LinkedList:
    def __init__(self, value):
        new_node = Node(value)
        self.head = new_node
        self.tail = new_node
        self.length = 1

    def print_list(self):
        temp = self.head
        while temp is not None:
            print(temp.value)
            temp = temp.next

    def append(self, value):
        new_node = Node(value)
        if self.length == 0:
            self.head = new_node
```

Linked List : Append Skenario Percobaan



```
    def append(self, value):
        new_node = Node(value)
        if self.length == 0:
            self.head = new_node
            self.tail = new_node
        else:
            self.tail.next = new_node
            self.tail = new_node
        self.length += 1



my_linked_list = LinkedList(1)

my_linked_list.append(2)

my_linked_list.print_list()
```

Linked List : Append Skenario Percobaan



```
def append(self, value):
    new_node = Node(value)
    if self.length == 0:
        self.head = new_node
        self.tail = new_node
    else:
        self.tail.next = new_node
        self.tail = new_node
    self.length += 1


my_linked_list = LinkedList(1)

my_linked_list.append(2)

my_linked_list.print_list()
```

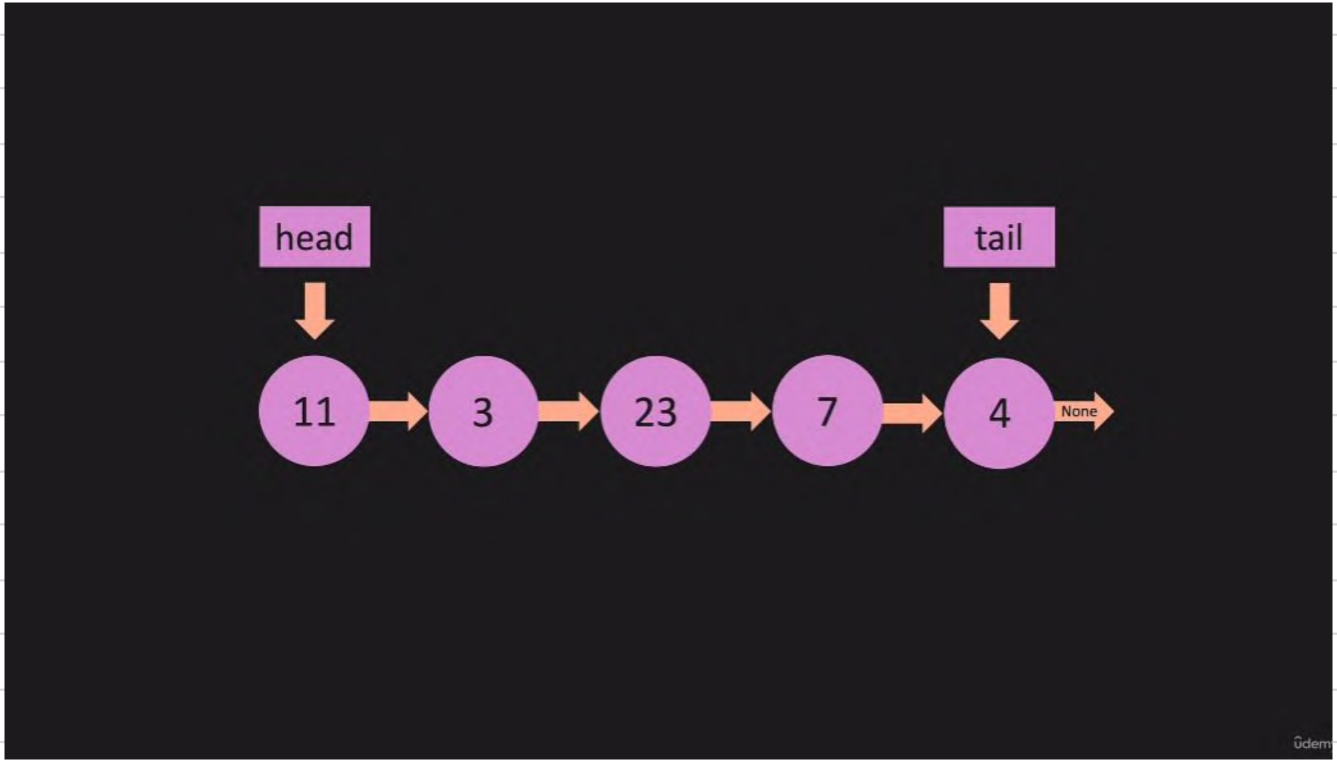
```
1
2
```

07

Linked List Pop

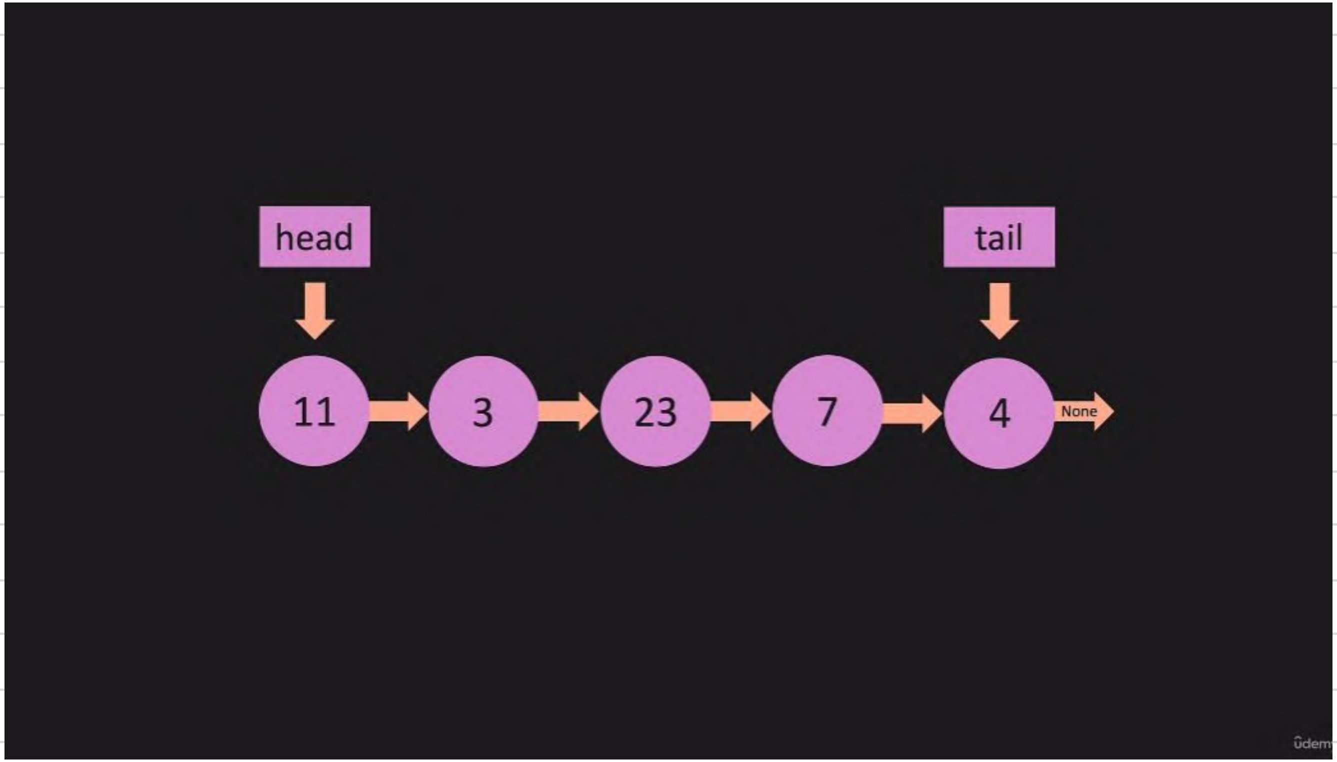
Linked List : Pop

Diketahui suatu list



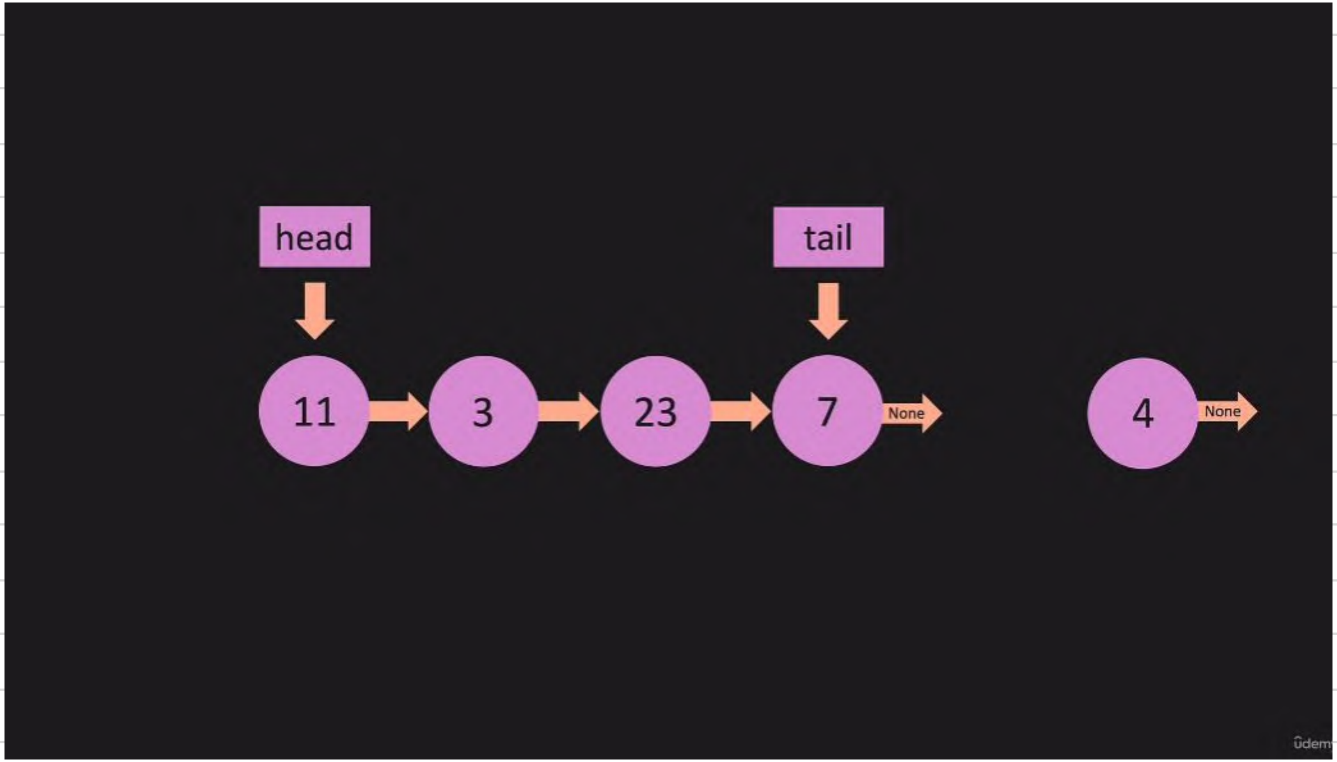
Linked List : Pop

Bagaimana jika ingin menghapus node paling belakang? (Pop)



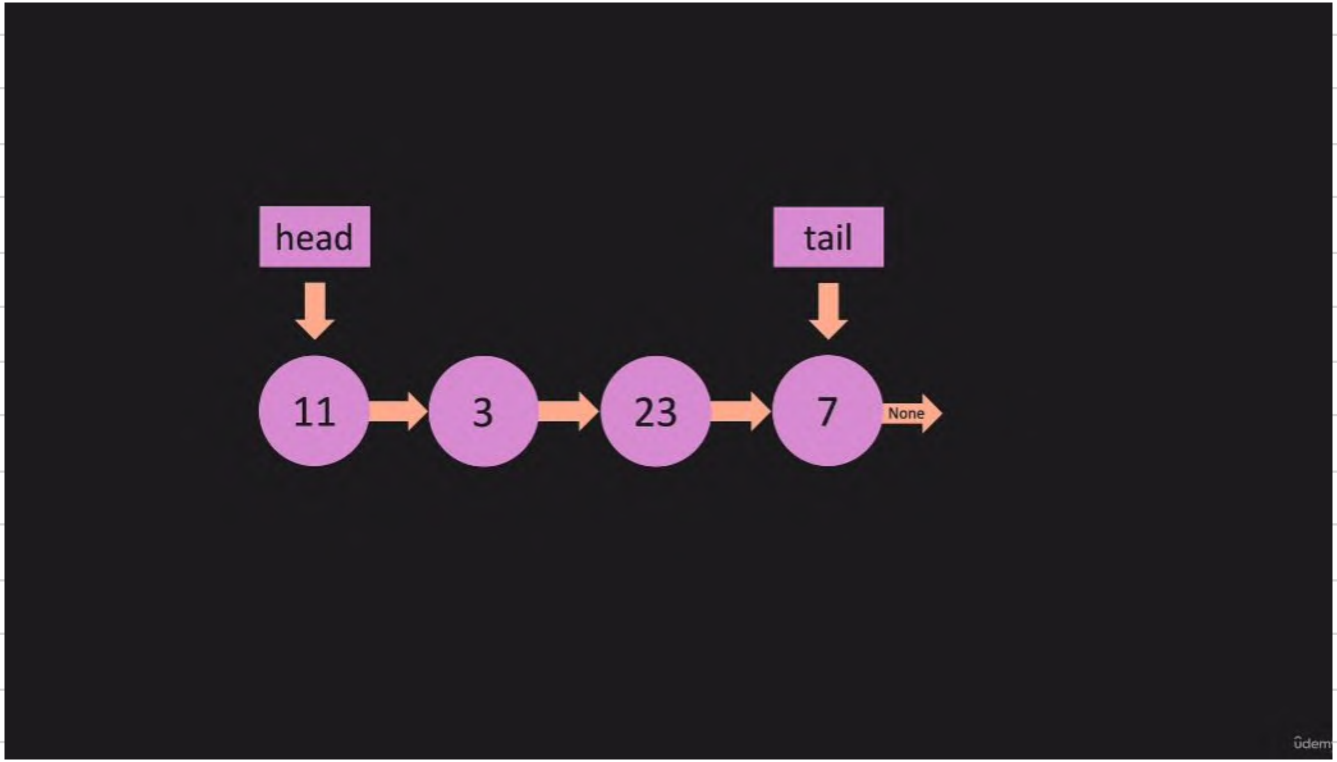
Linked List : Pop

Bagaimana jika ingin menghapus node paling belakang? (Pop)



Linked List : Pop

Bagaimana jika ingin menghapus node paling belakang? (Pop)

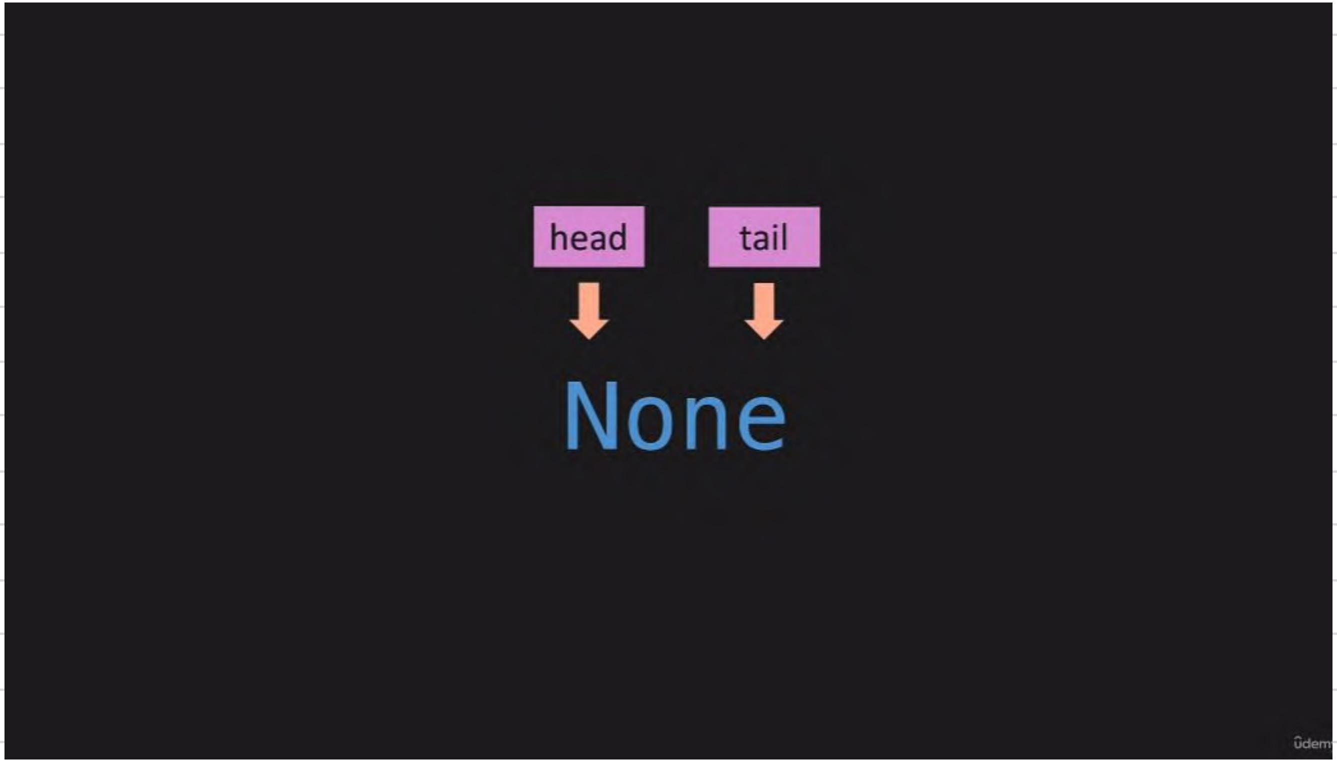


Linked List : Pop

- Namun ada beberapa kondisi yang harus mendapatkan perhatian khusus

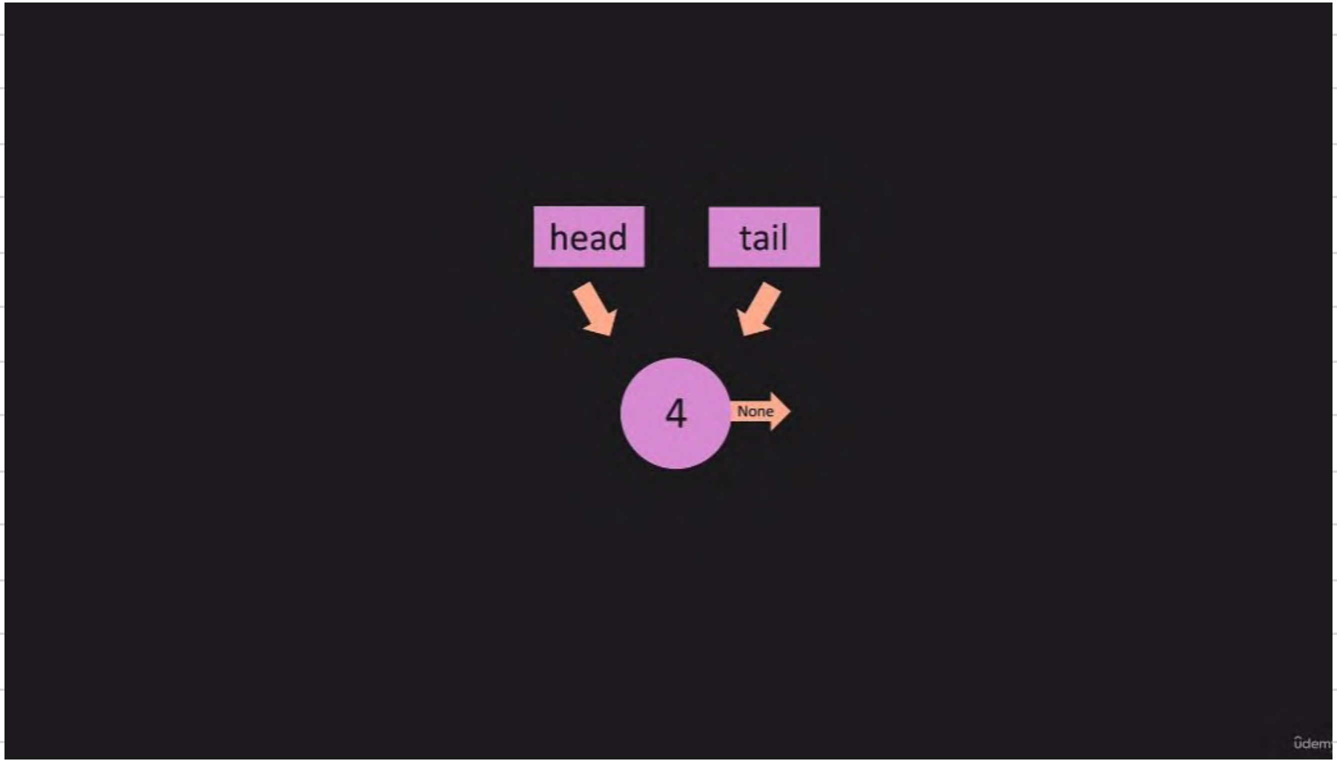
Kasus #1

Linked List : Pop

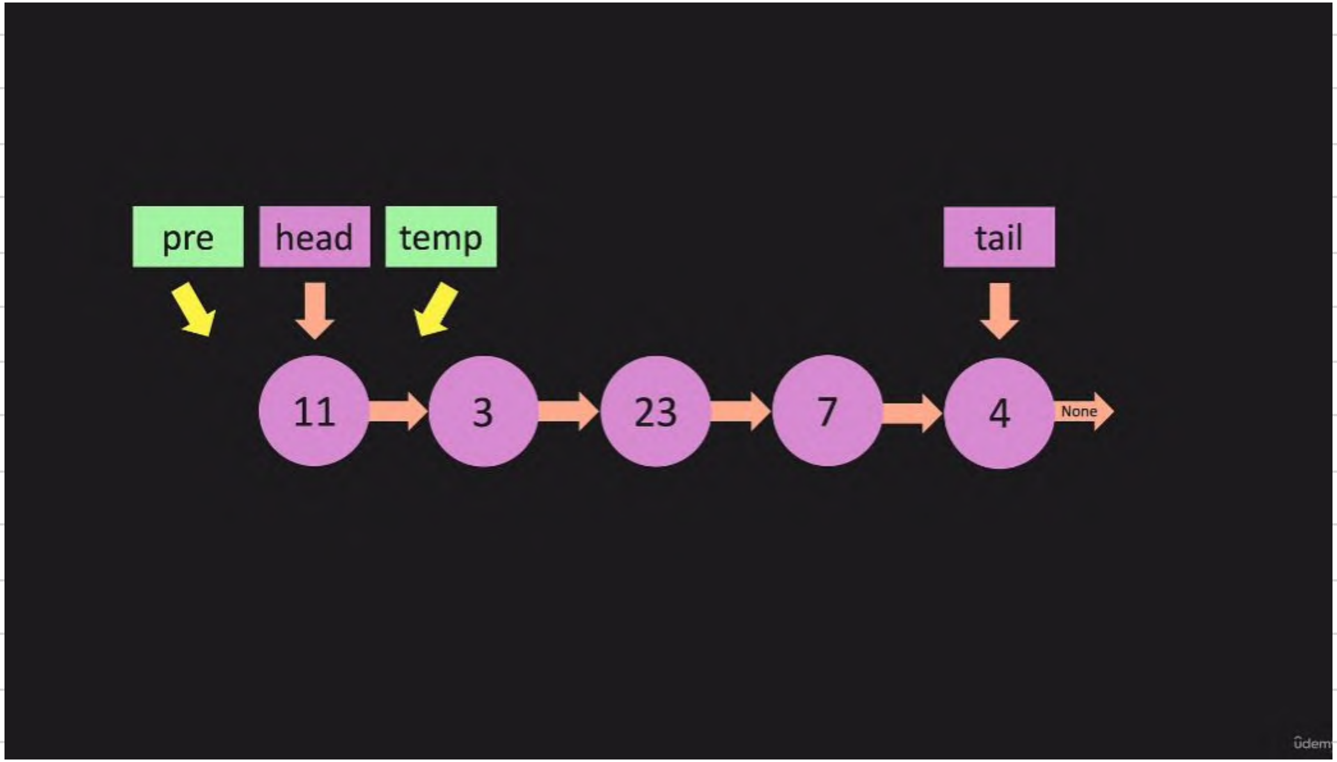


Kasus #2

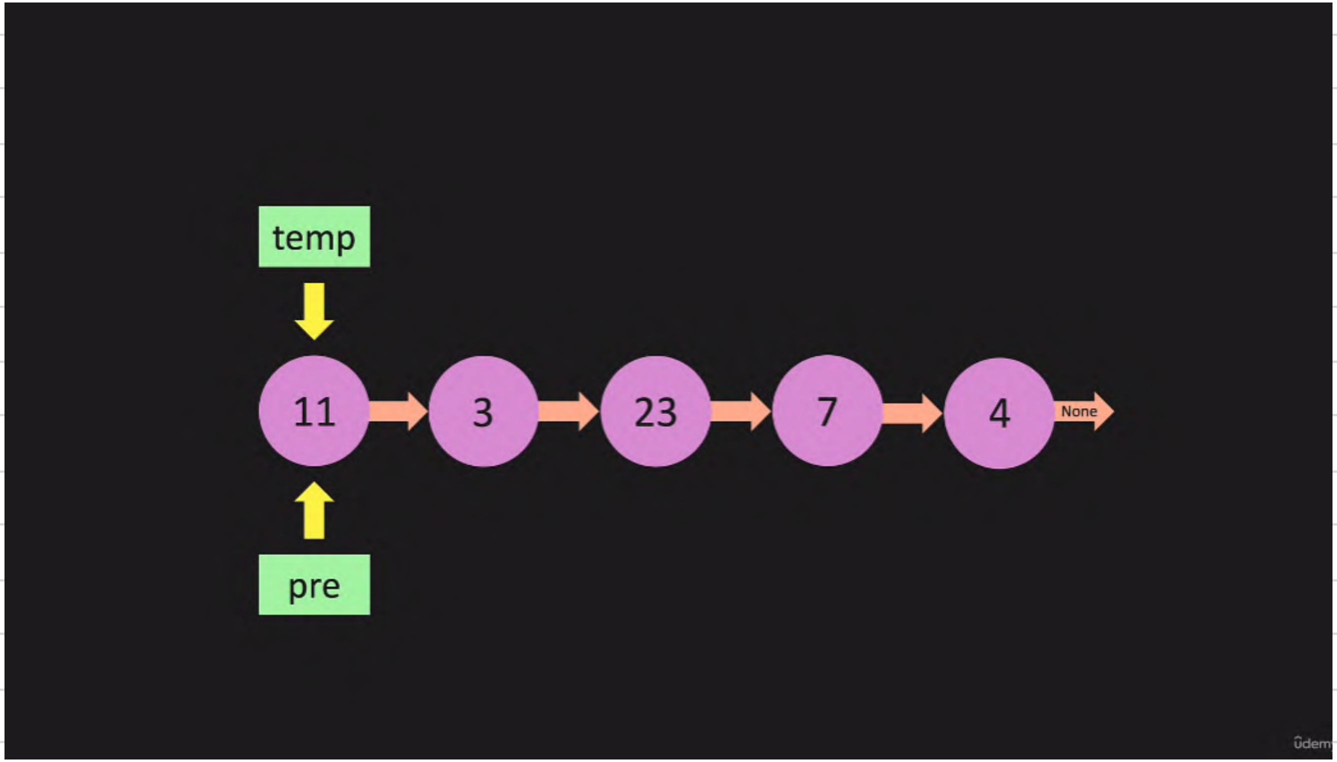
Linked List : Pop



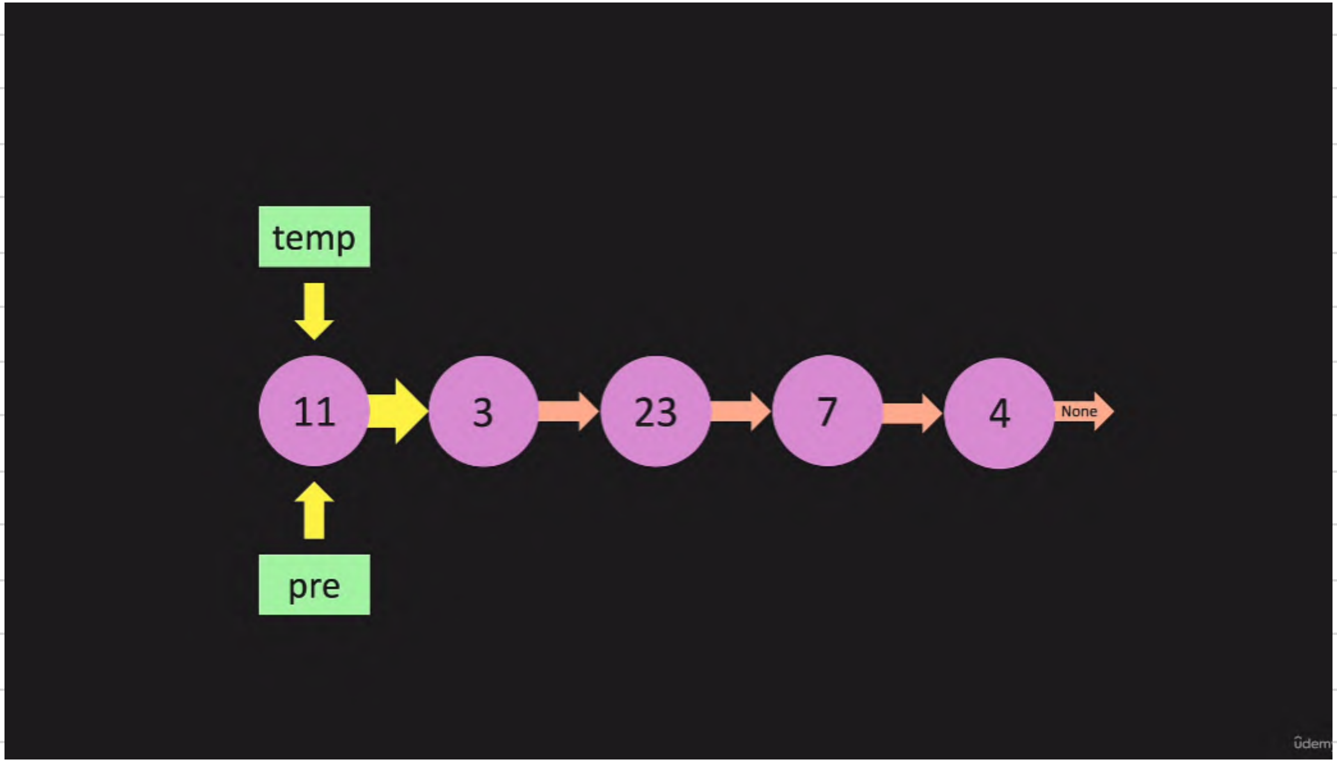
Linked List : Pop – Kasus Umum



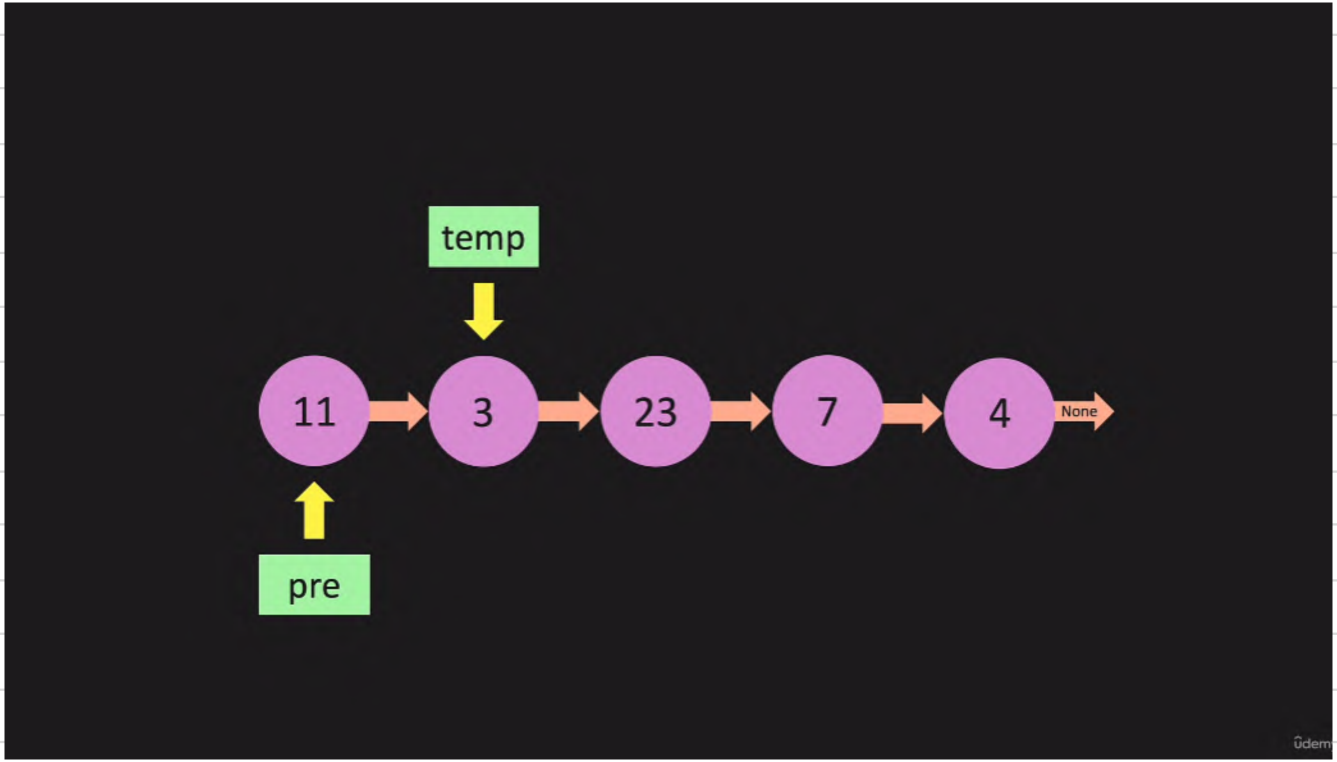
Linked List : Pop – Kasus Umum



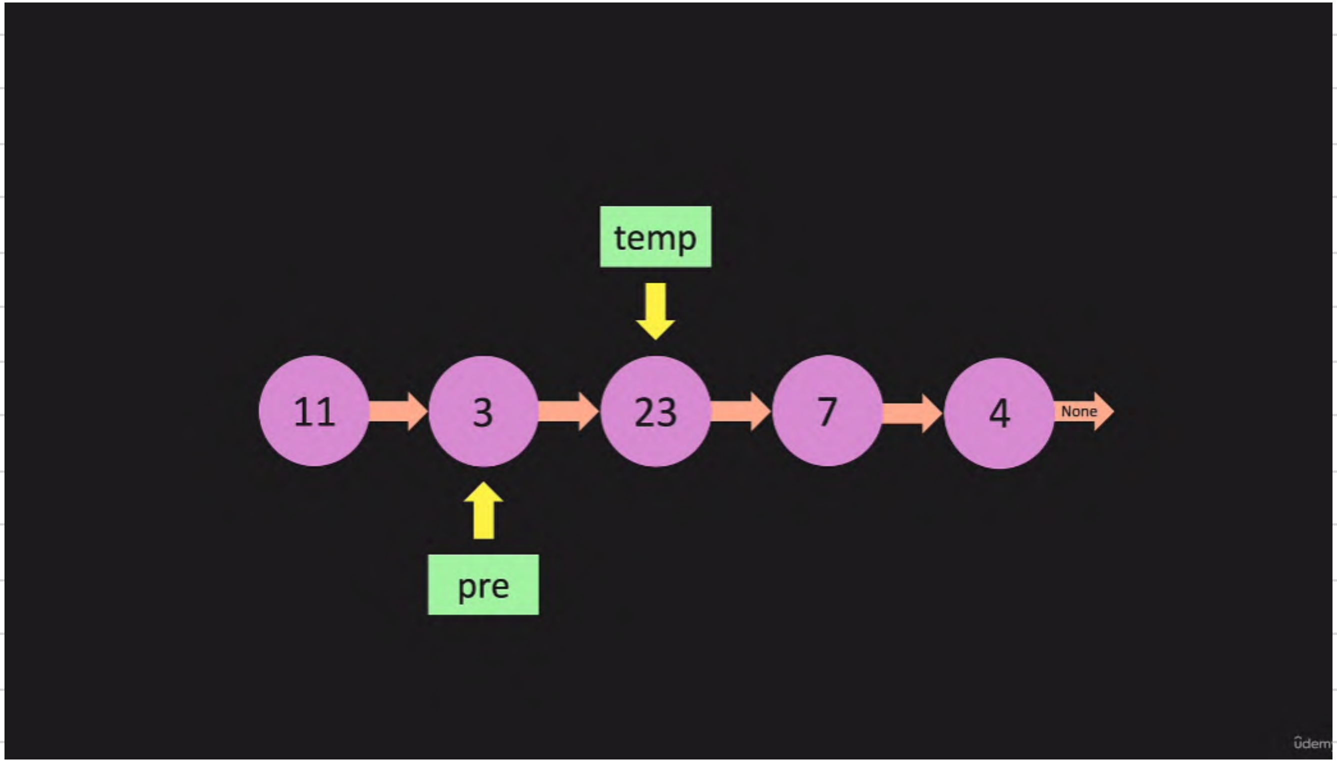
Linked List : Pop – Kasus Umum



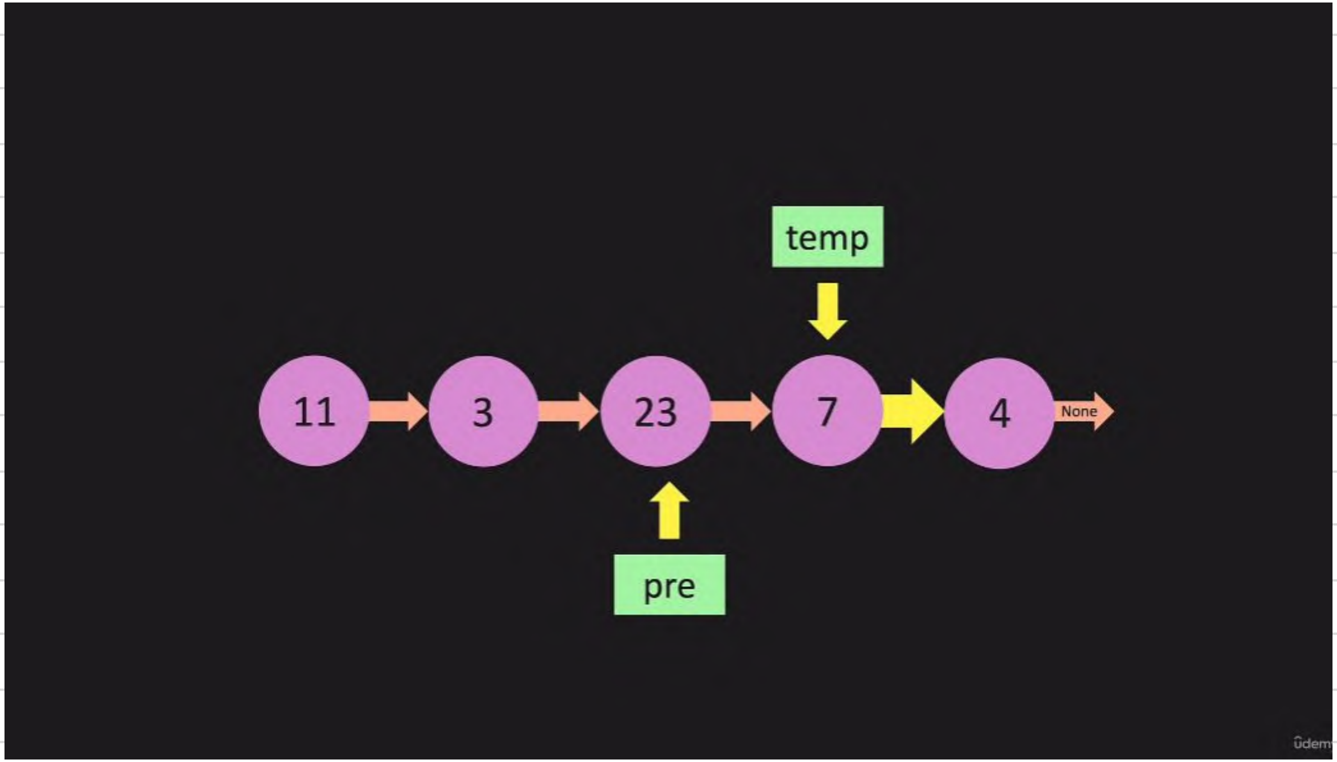
Linked List : Pop – Kasus Umum



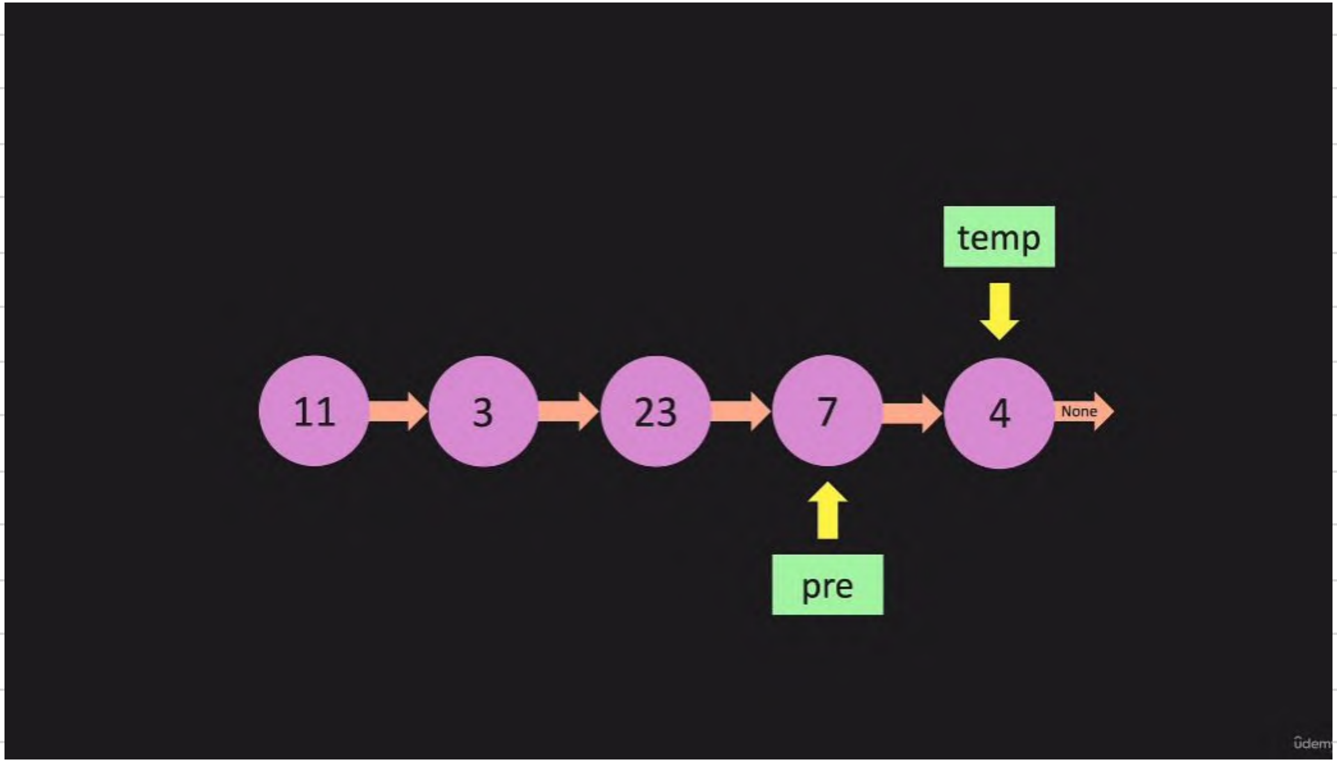
Linked List : Pop – Kasus Umum



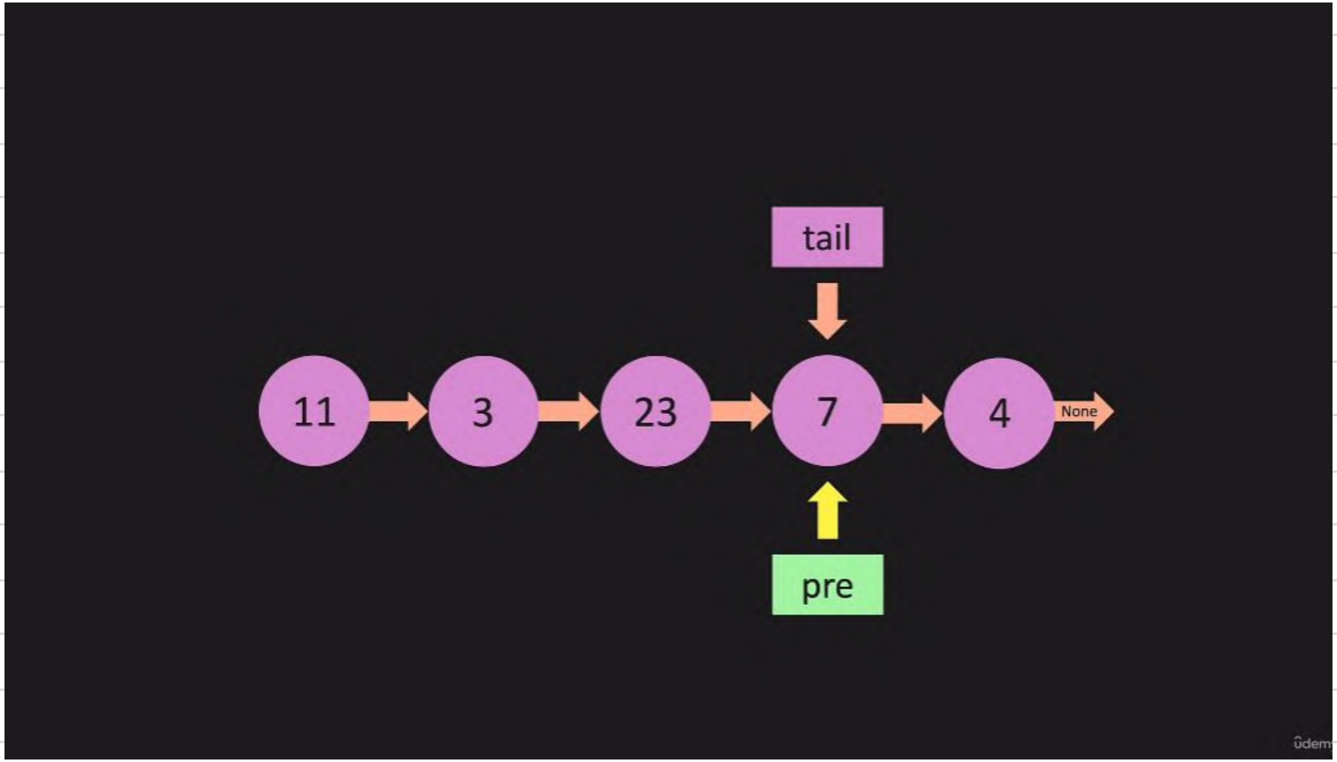
Linked List : Pop – Kasus Umum



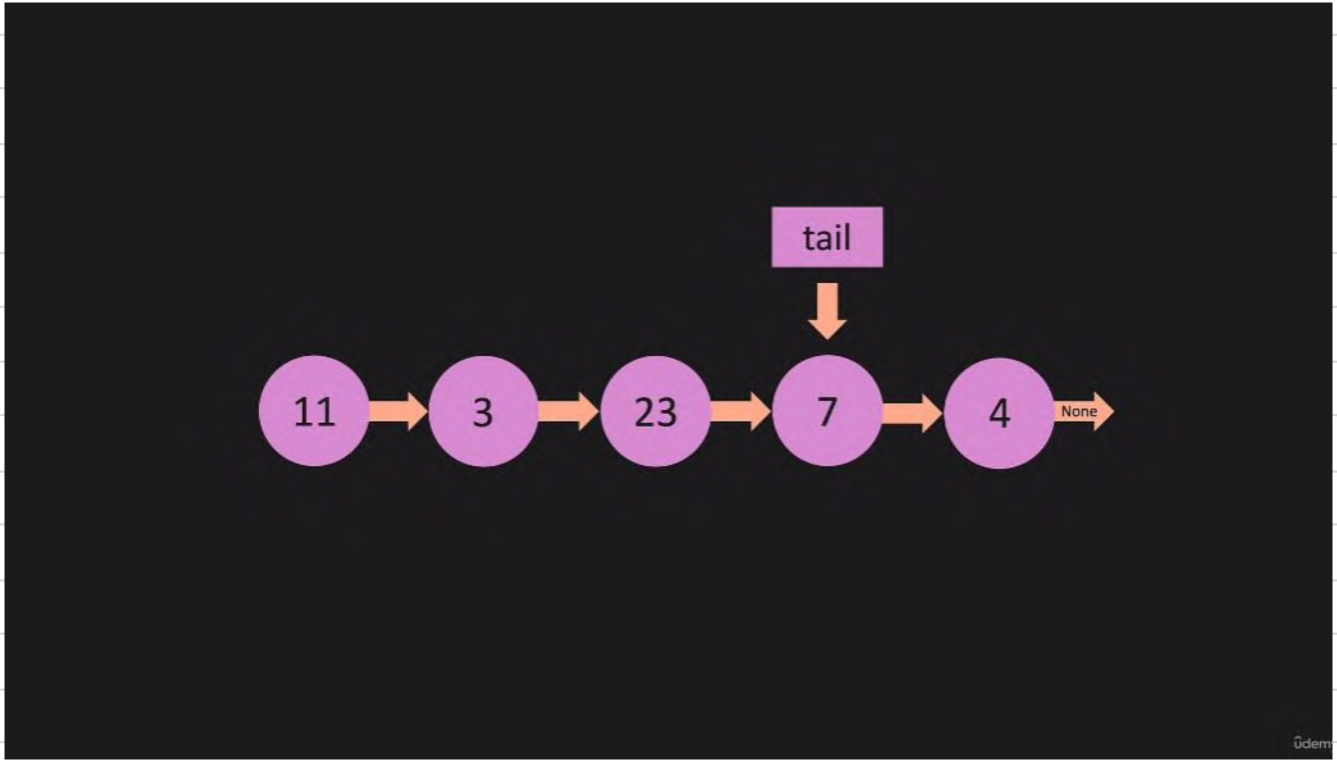
Linked List : Pop – Kasus Umum



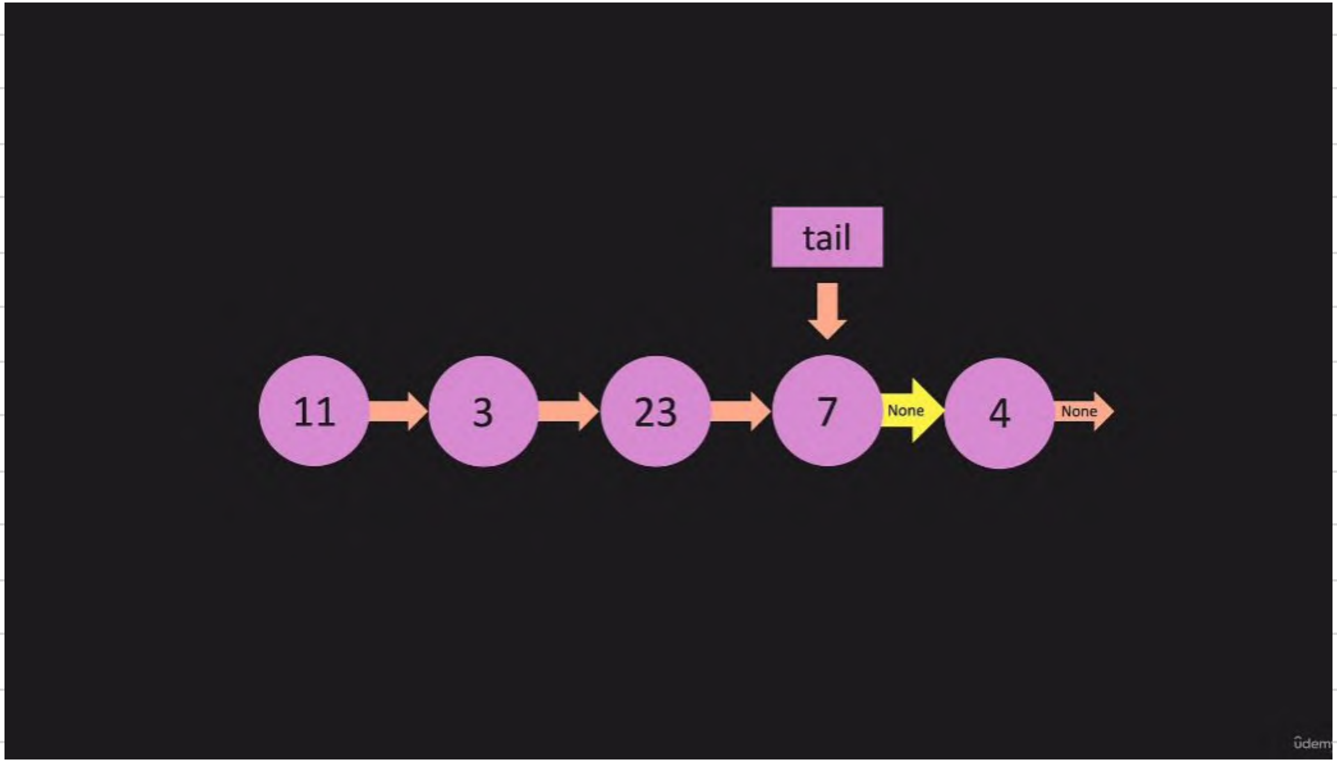
Linked List : Pop – Kasus Umum



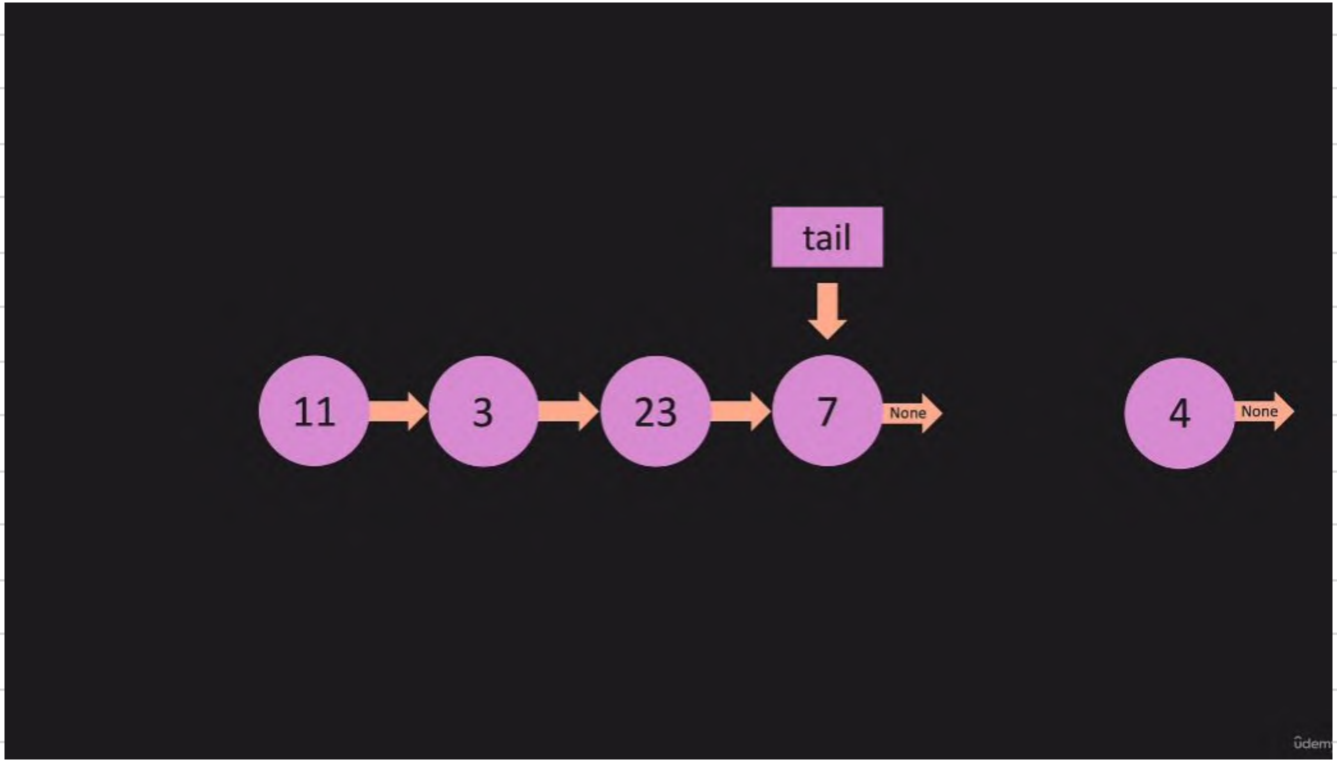
Linked List : Pop – Kasus Umum



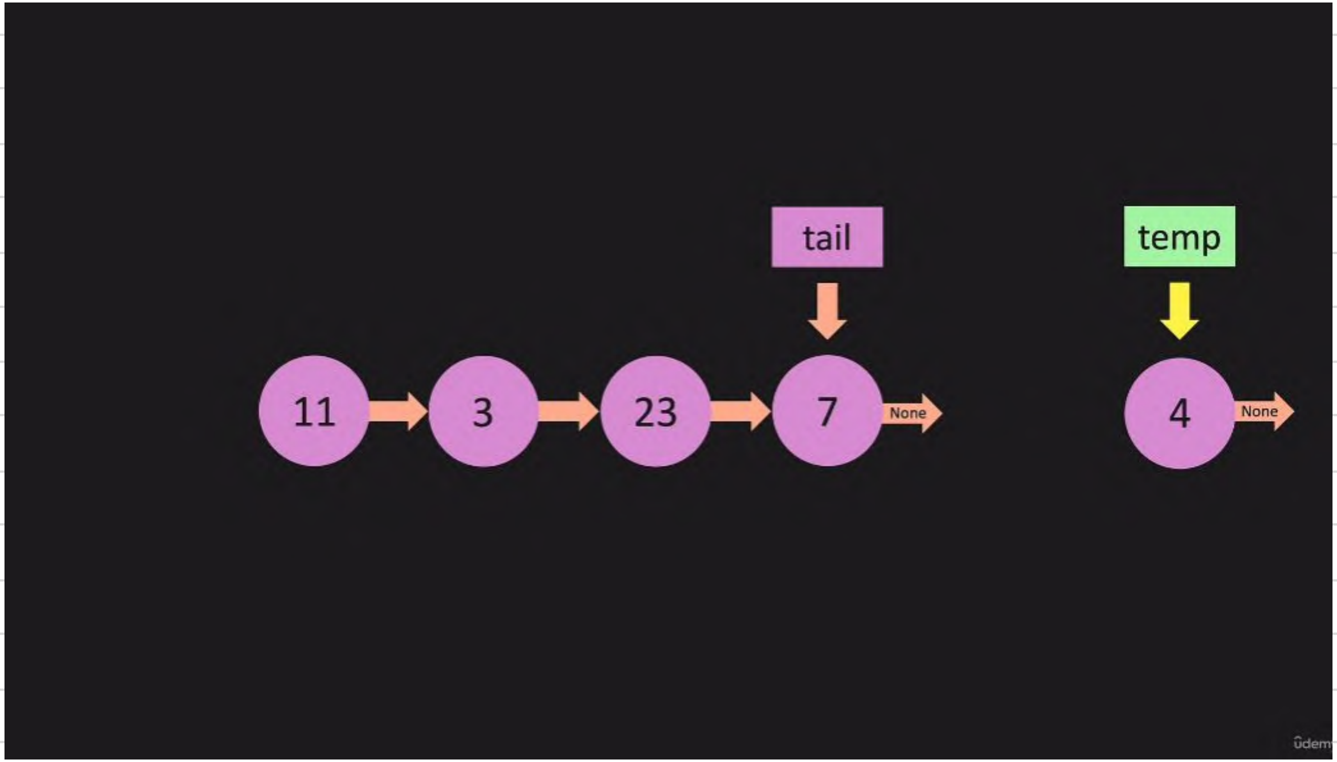
Linked List : Pop – Kasus Umum



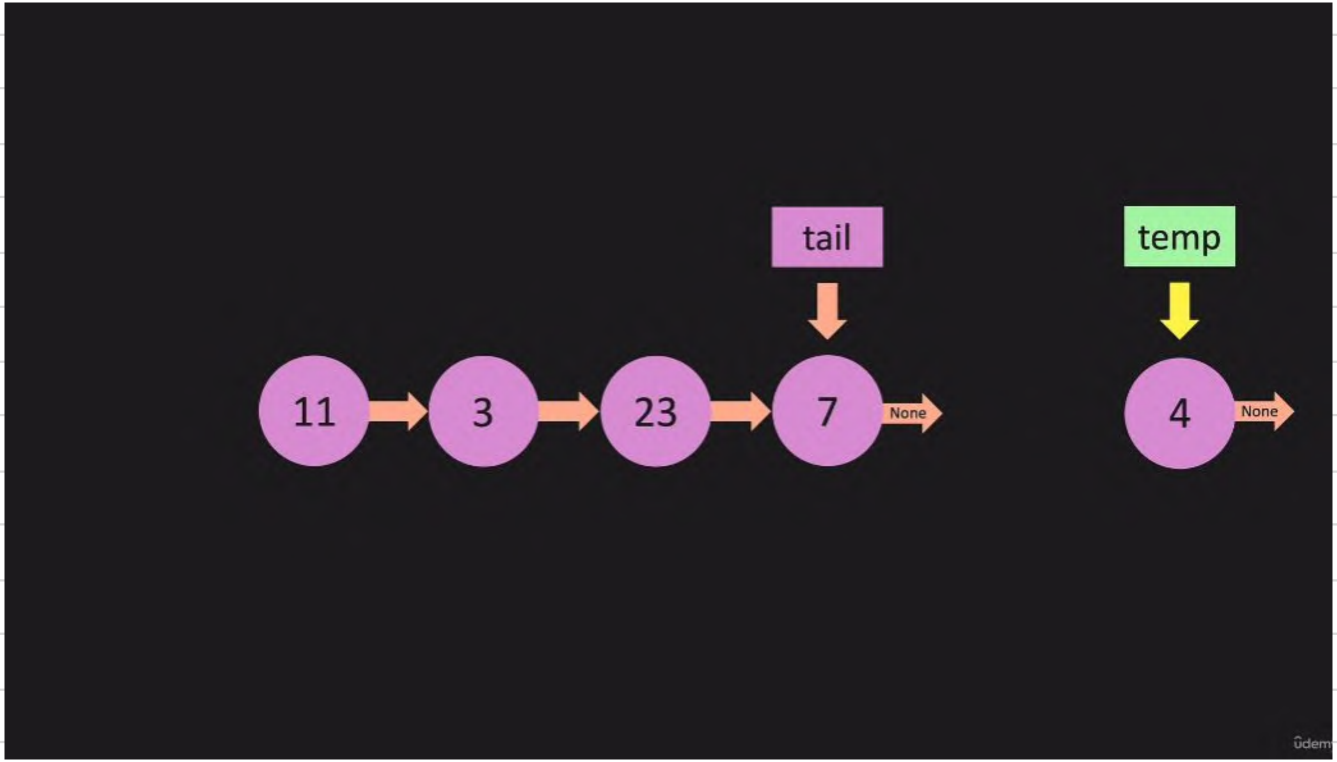
Linked List : Pop – Kasus Umum



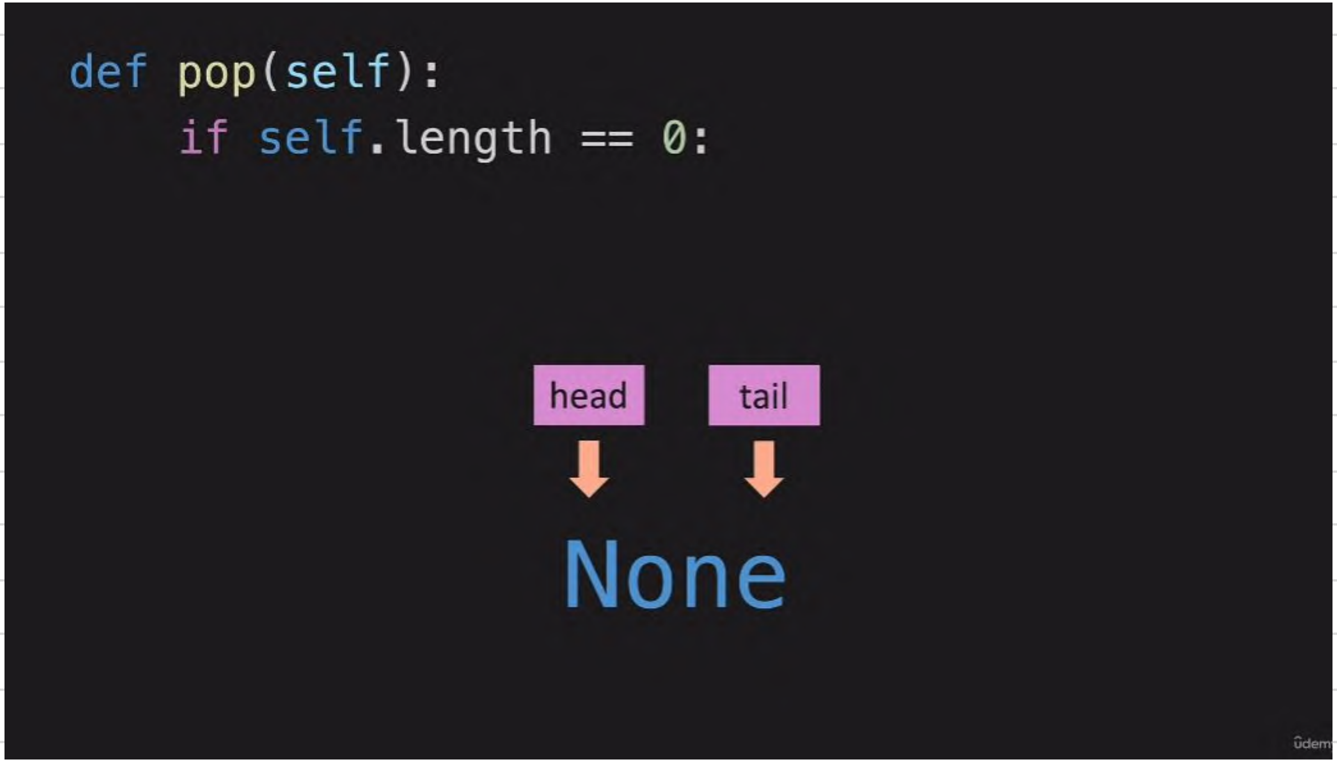
Linked List : Pop – Kasus Umum



Linked List : Pop – Kasus Umum

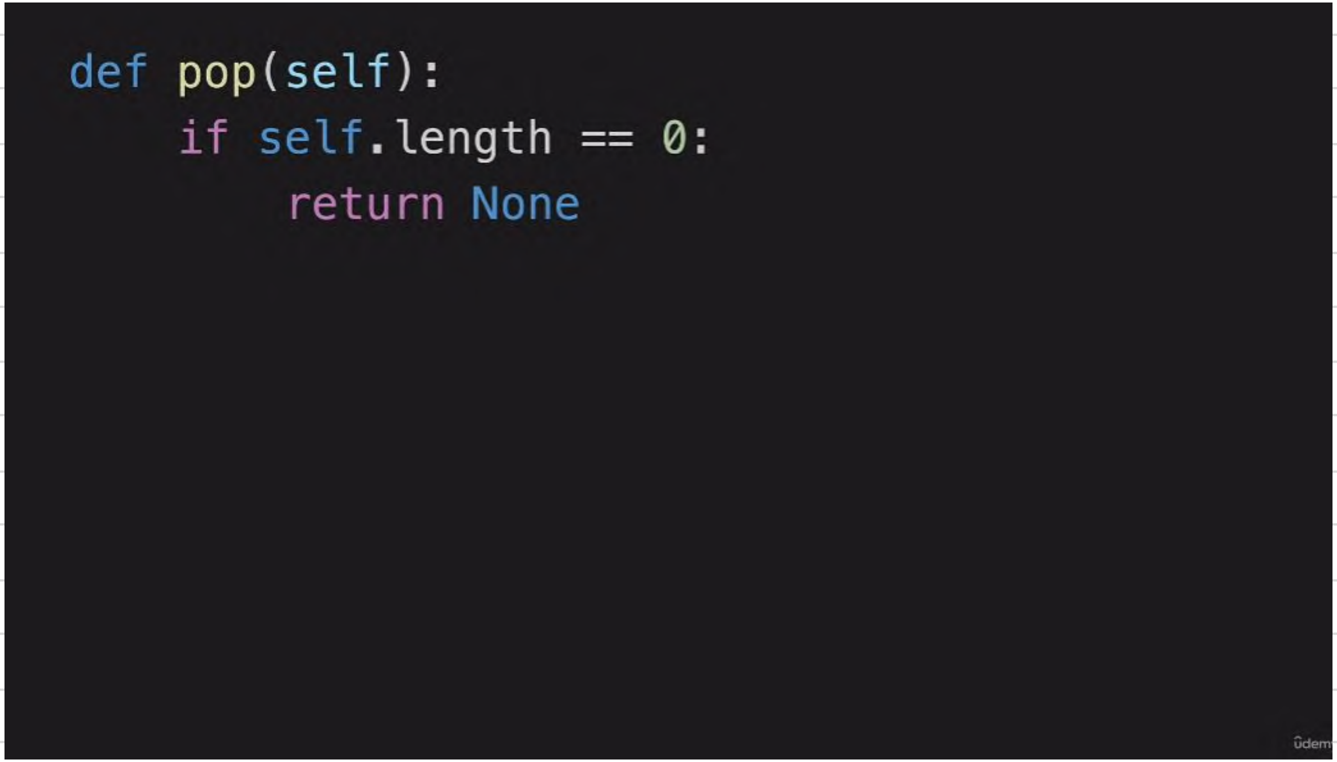


Linked List : Pop Code



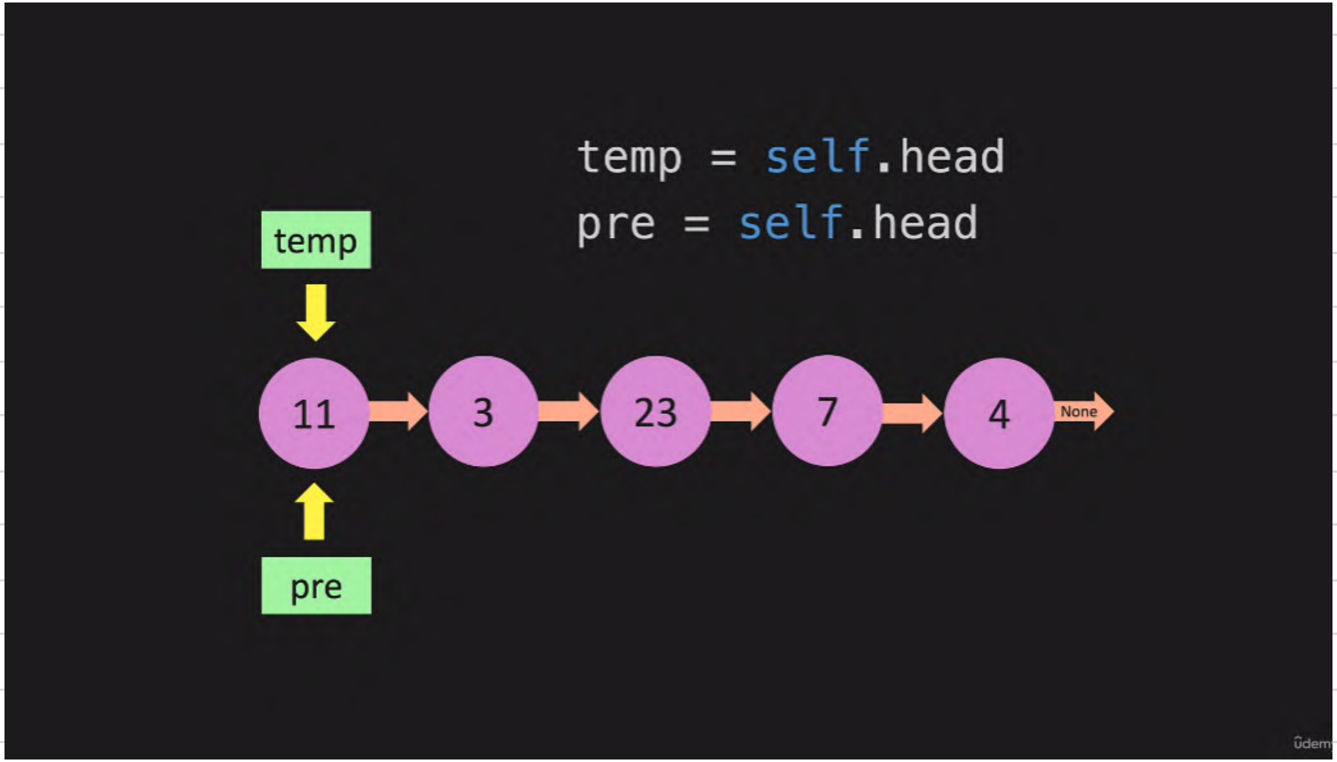
```
def pop(self):
    if self.length == 0:
```

Linked List : Pop Code



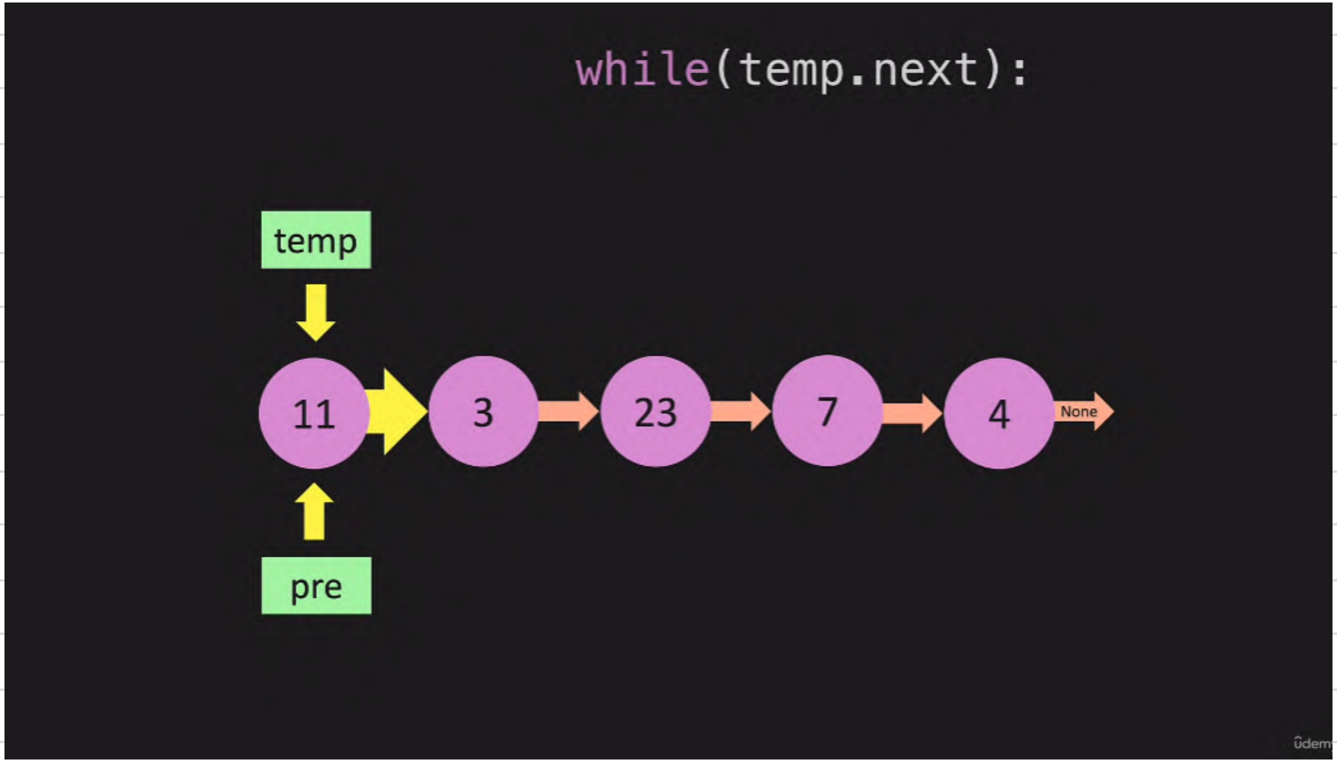
```
def pop(self):
    if self.length == 0:
        return None
```

Linked List : Pop Code



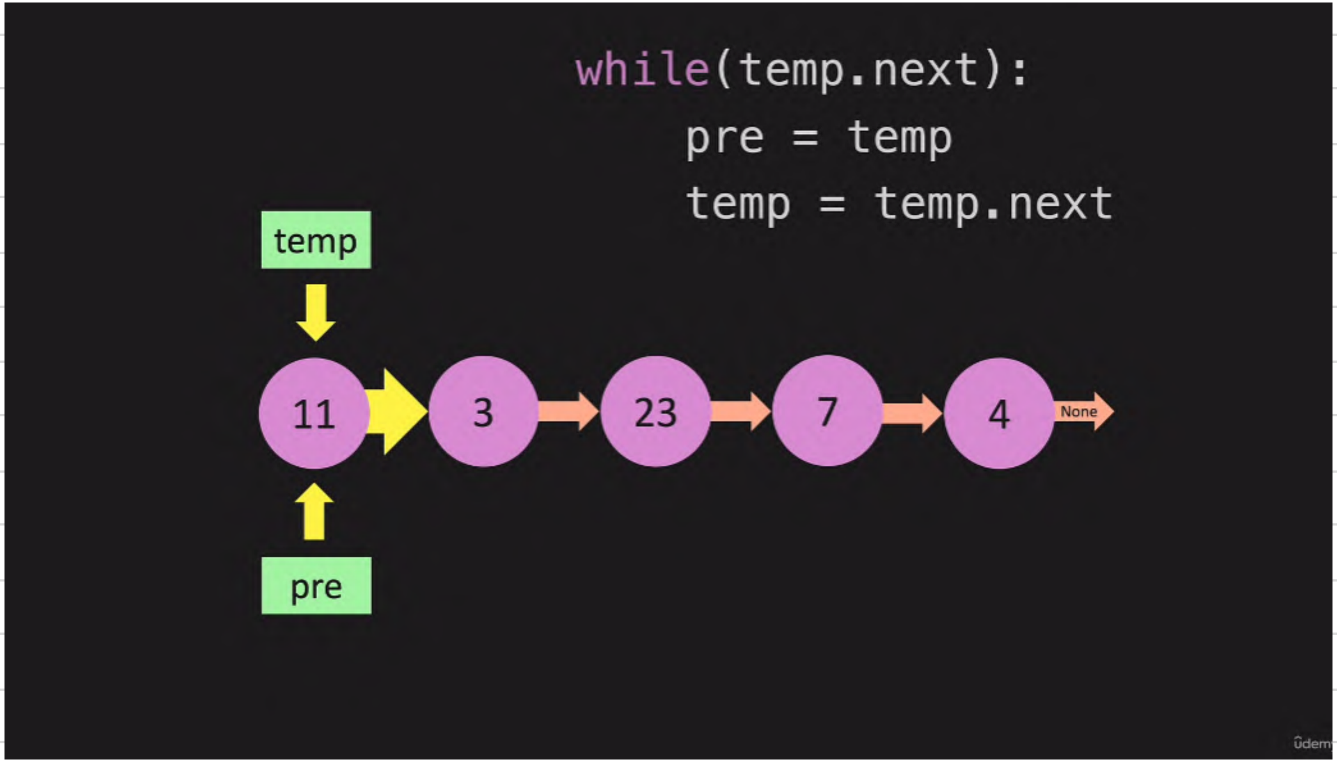
```
temp = self.head
pre = self.head
```

Linked List : Pop Code



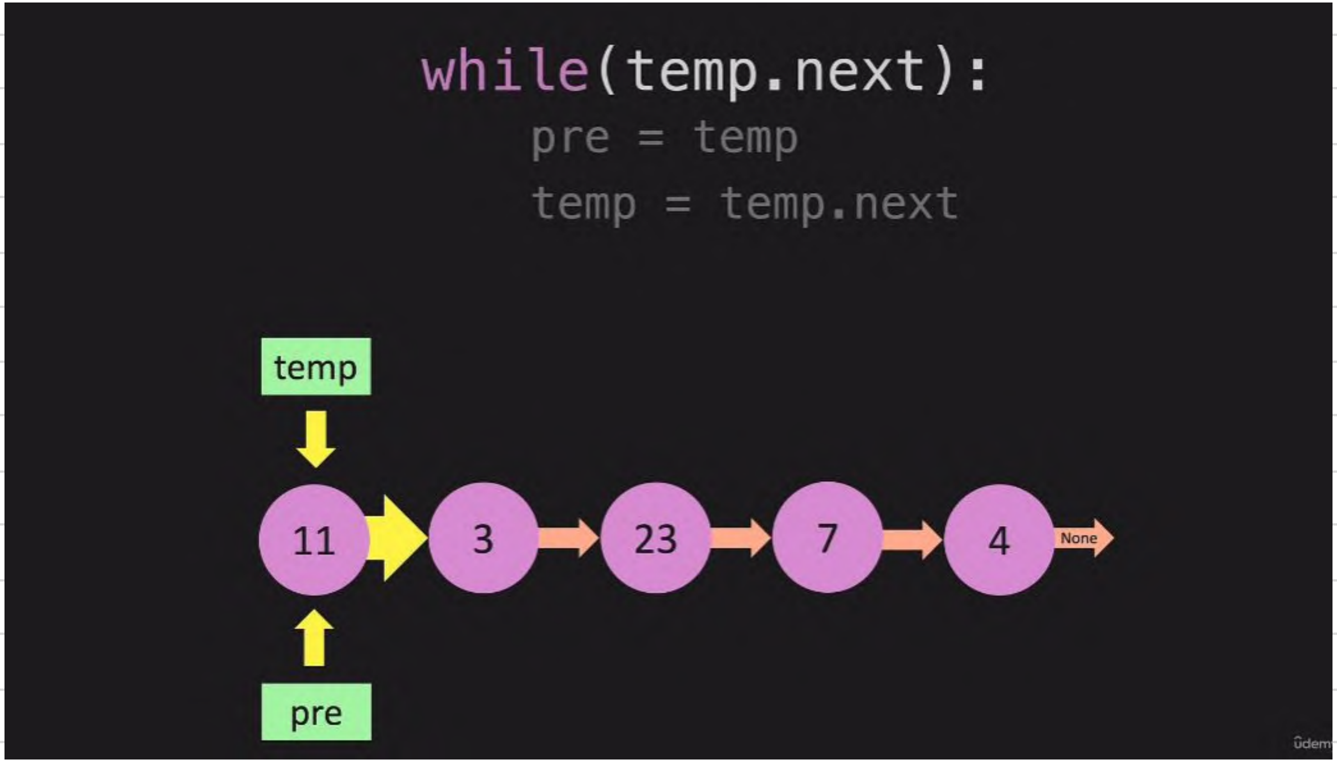
```
while(temp.next):
```

Linked List : Pop Code



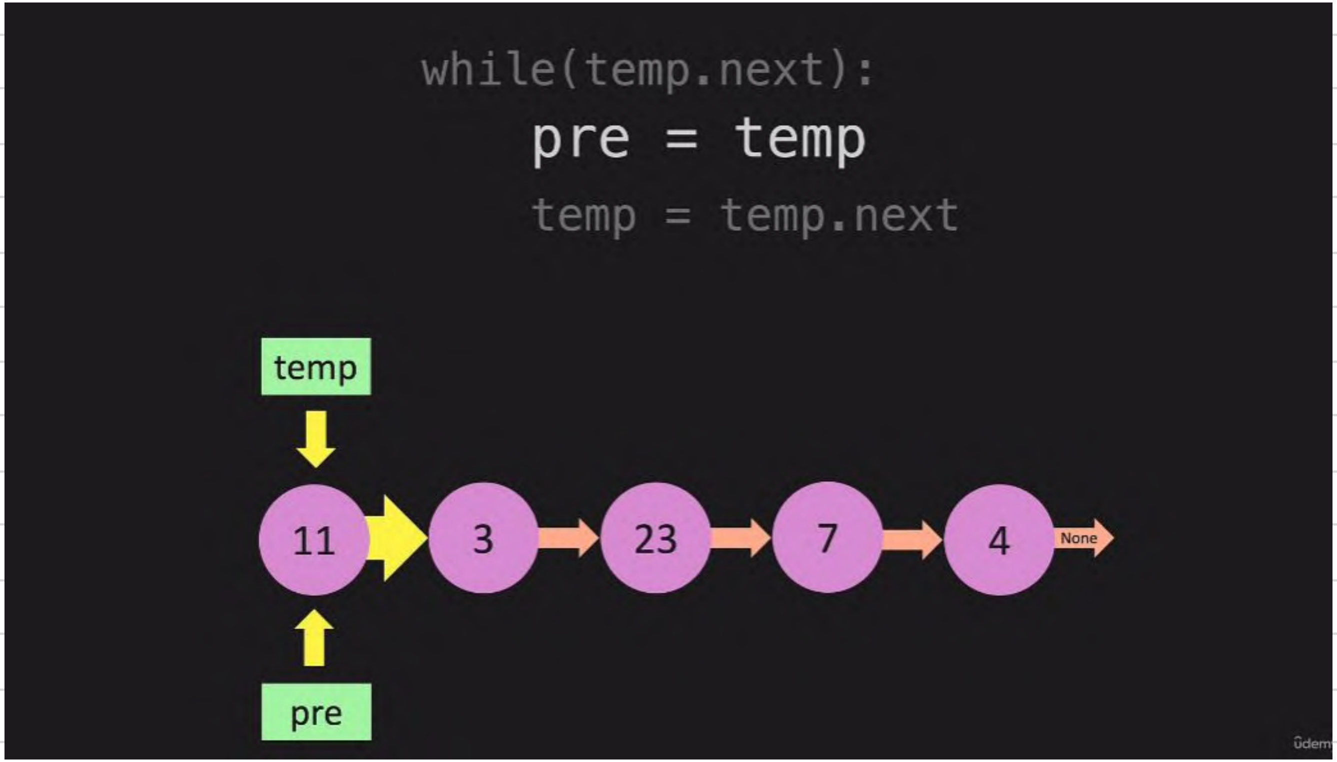
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



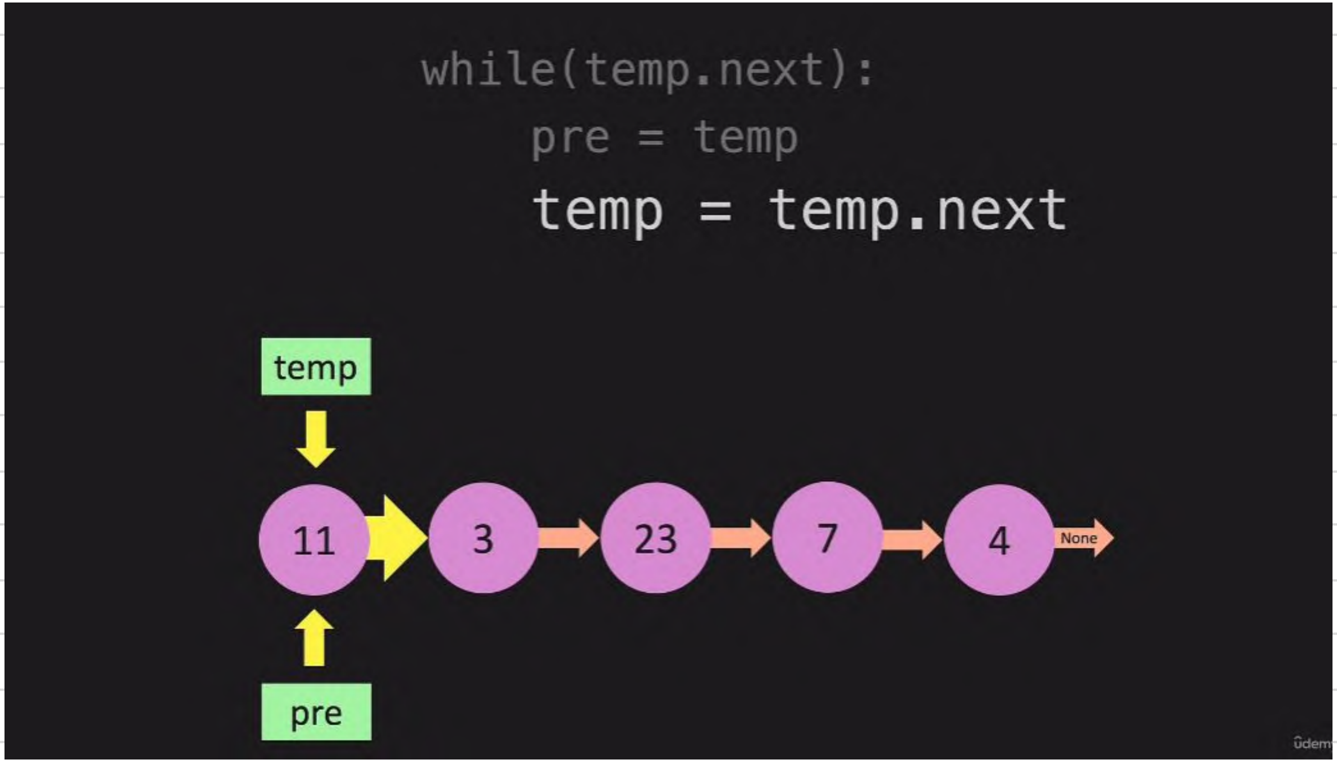
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



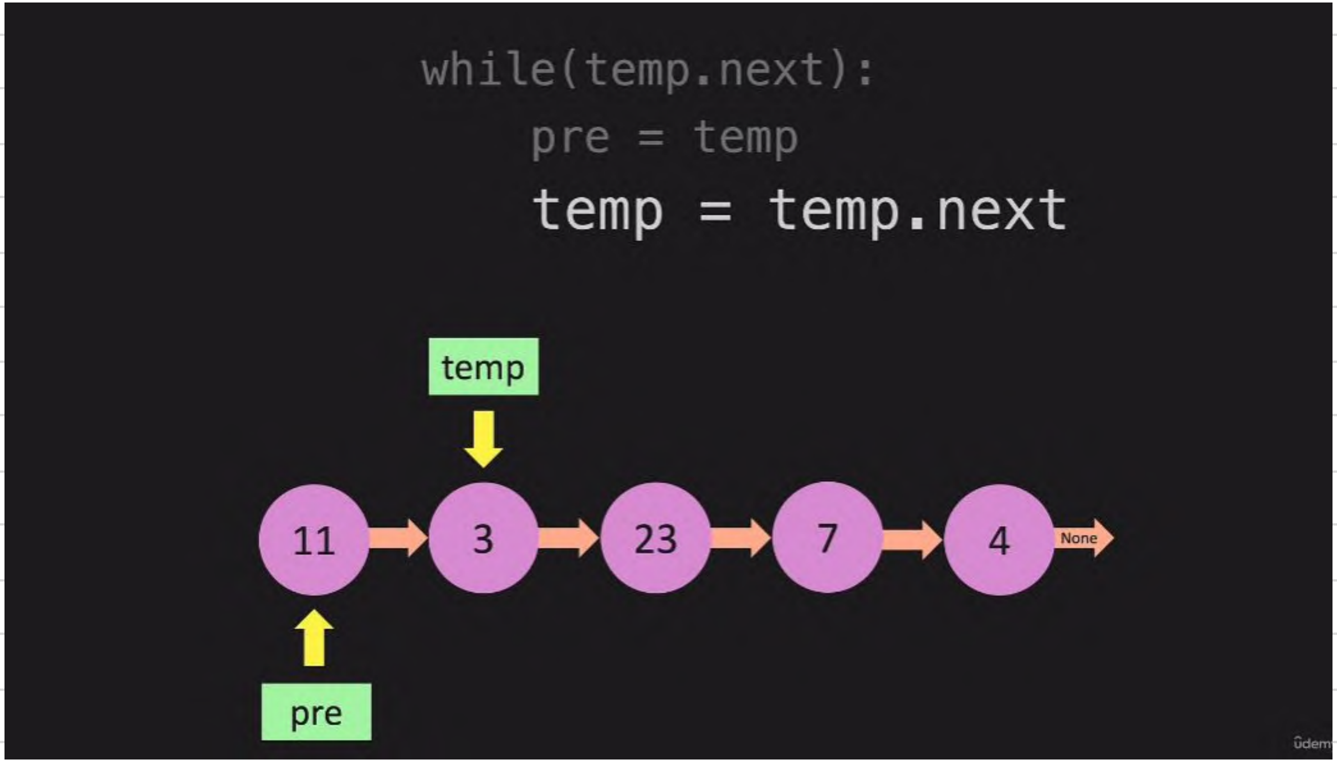
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



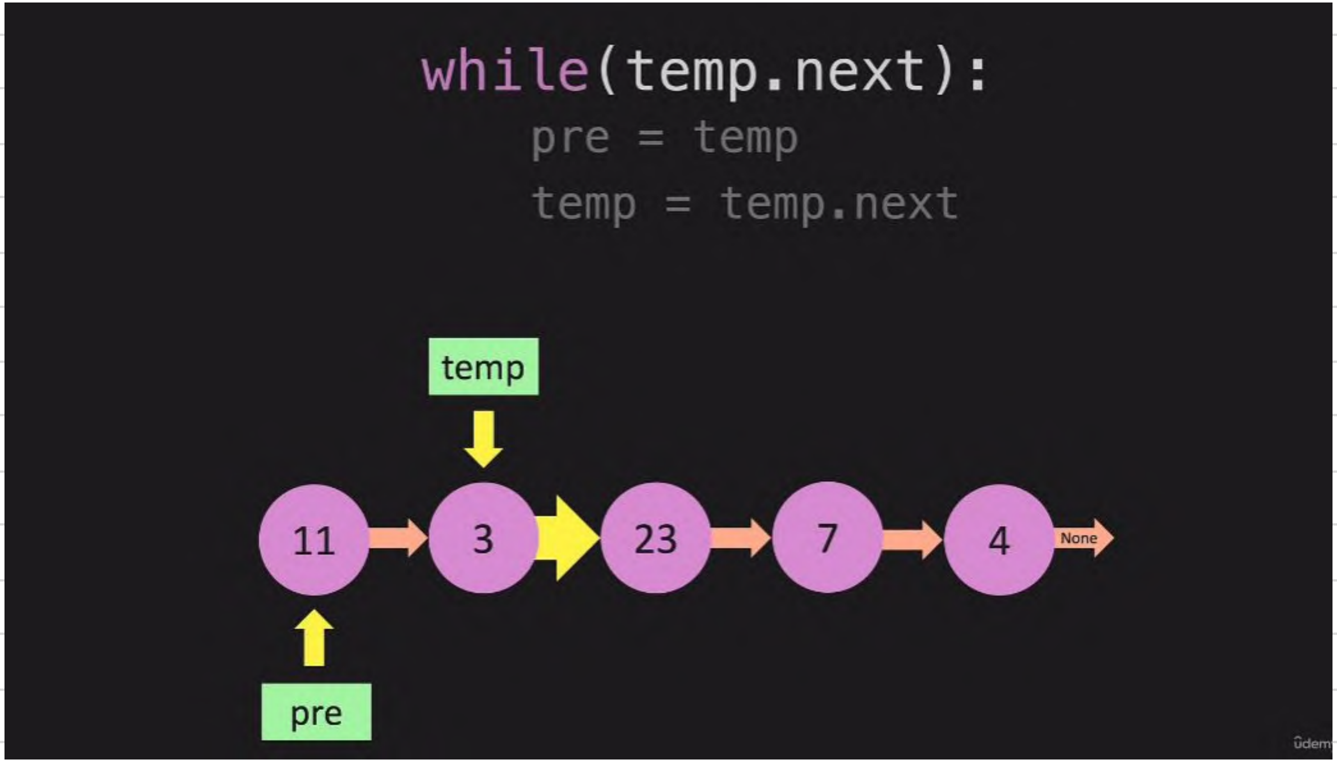
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



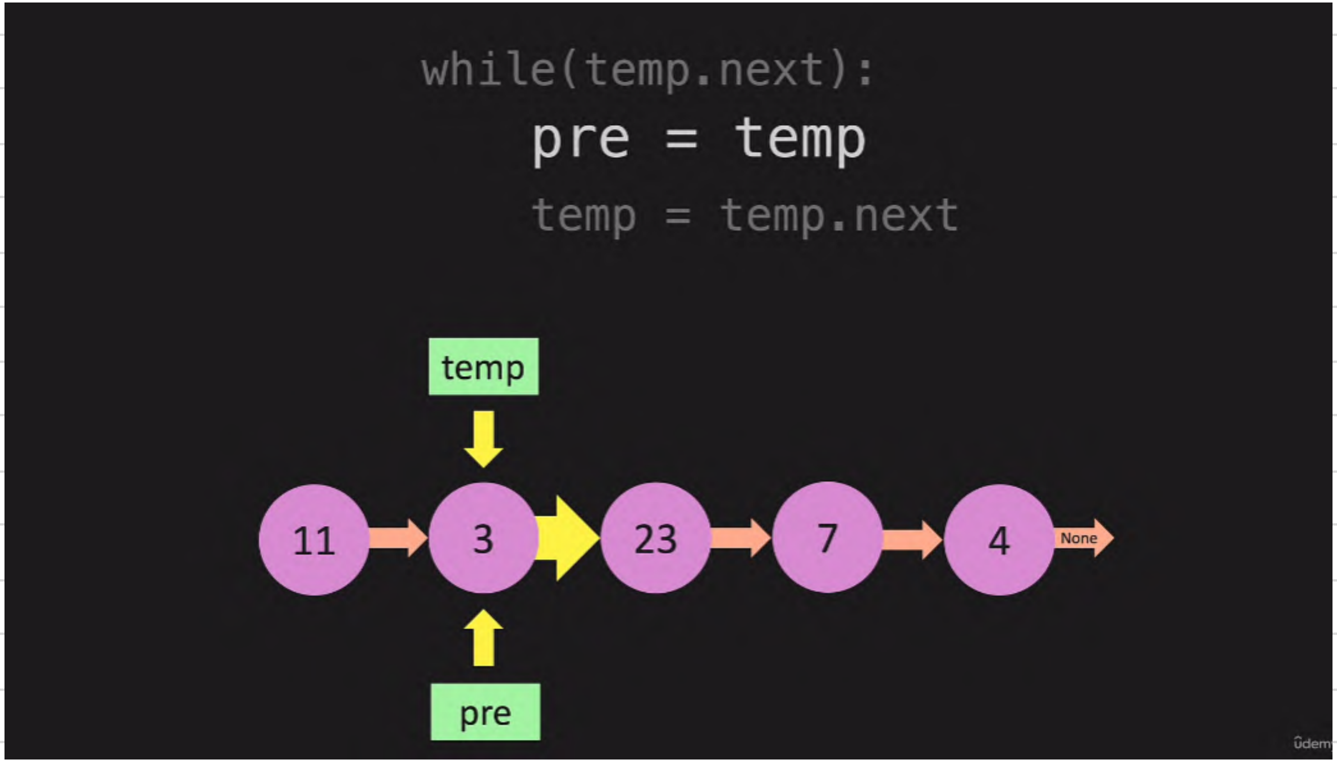
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



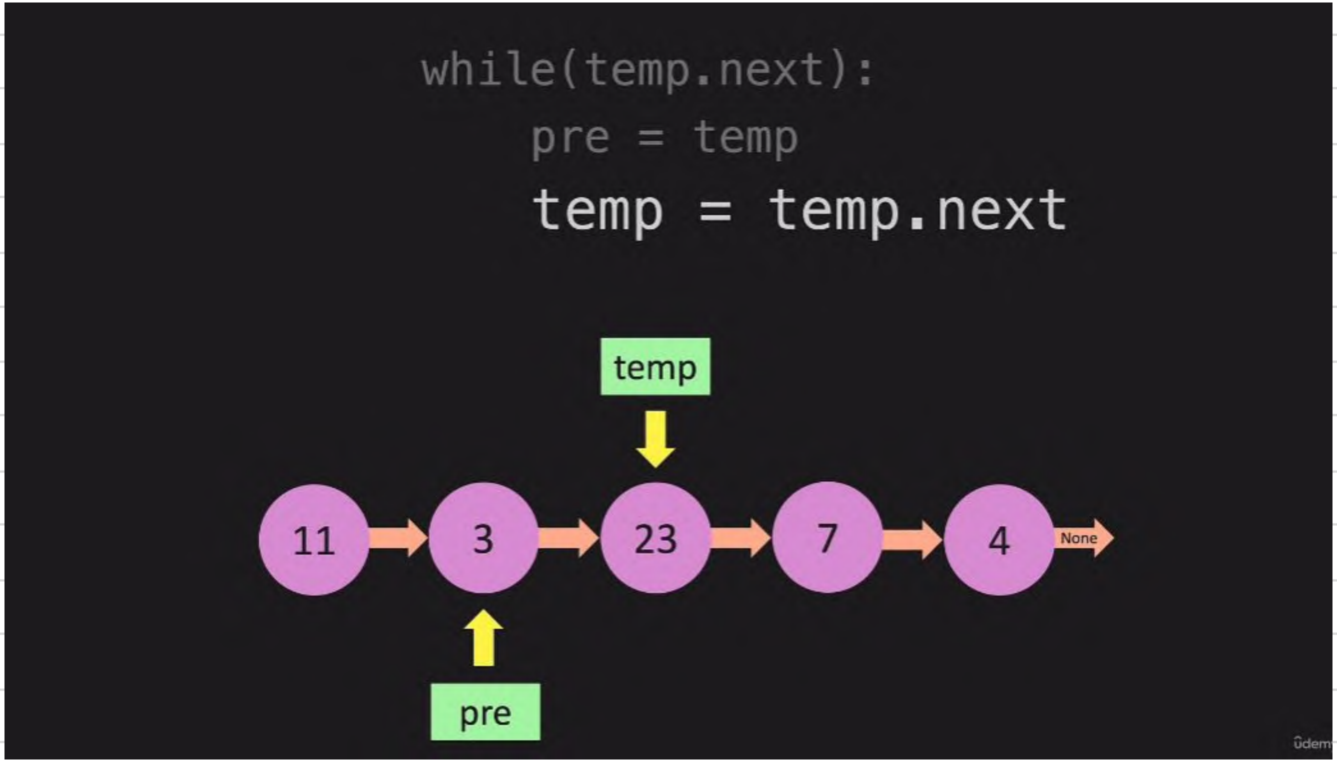
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



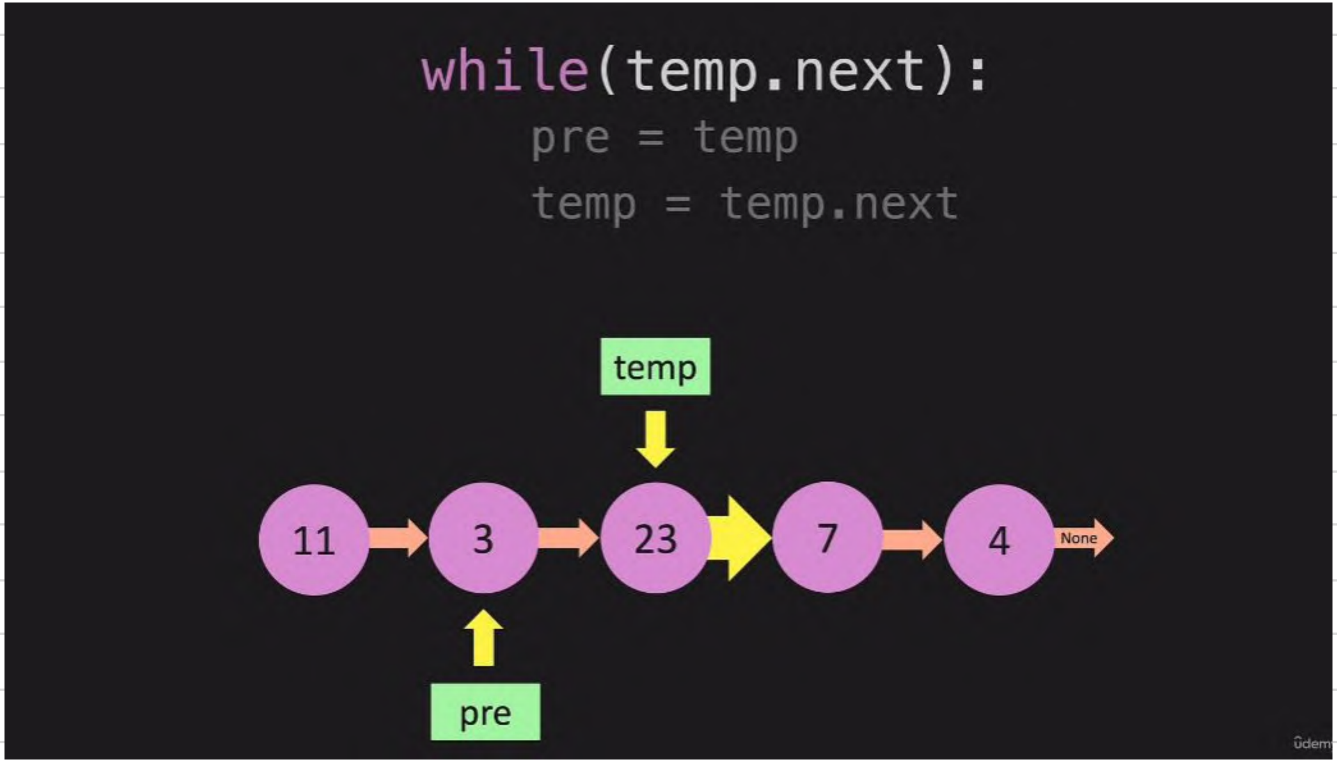
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



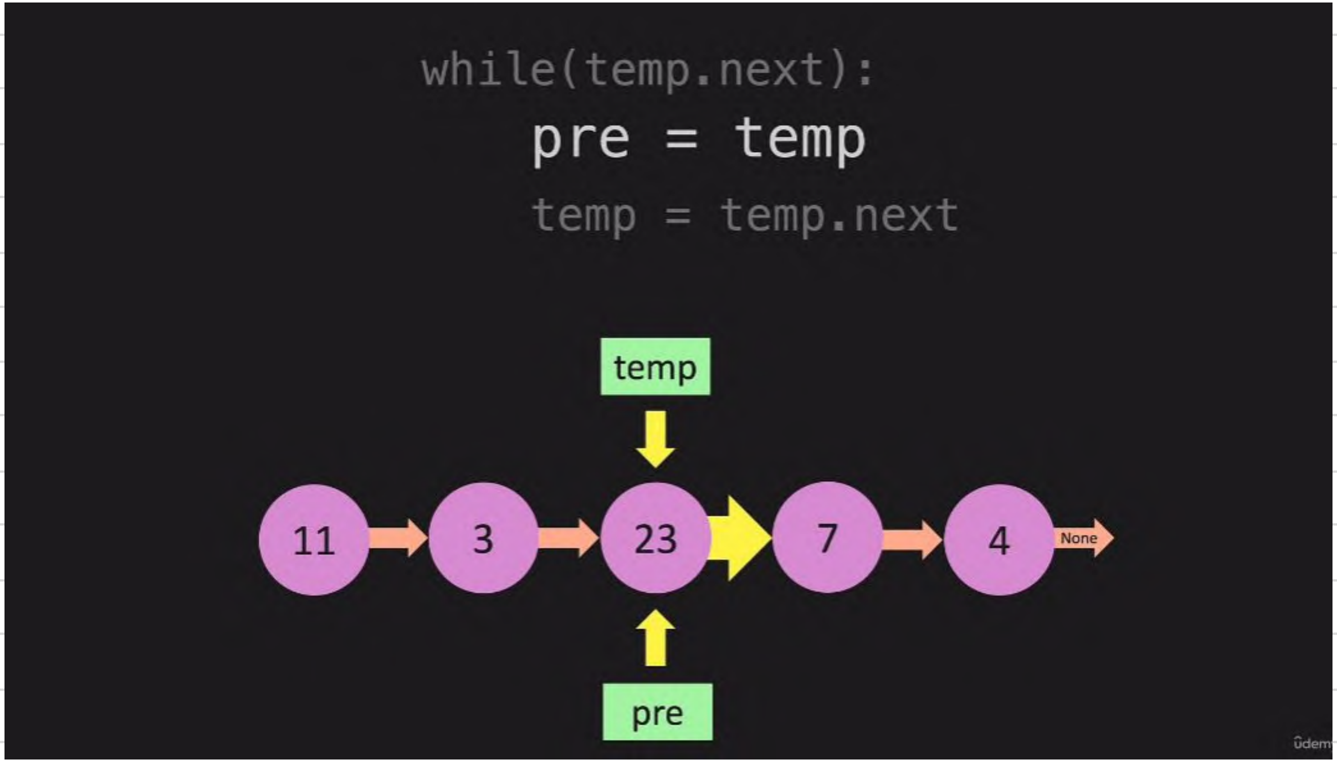
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



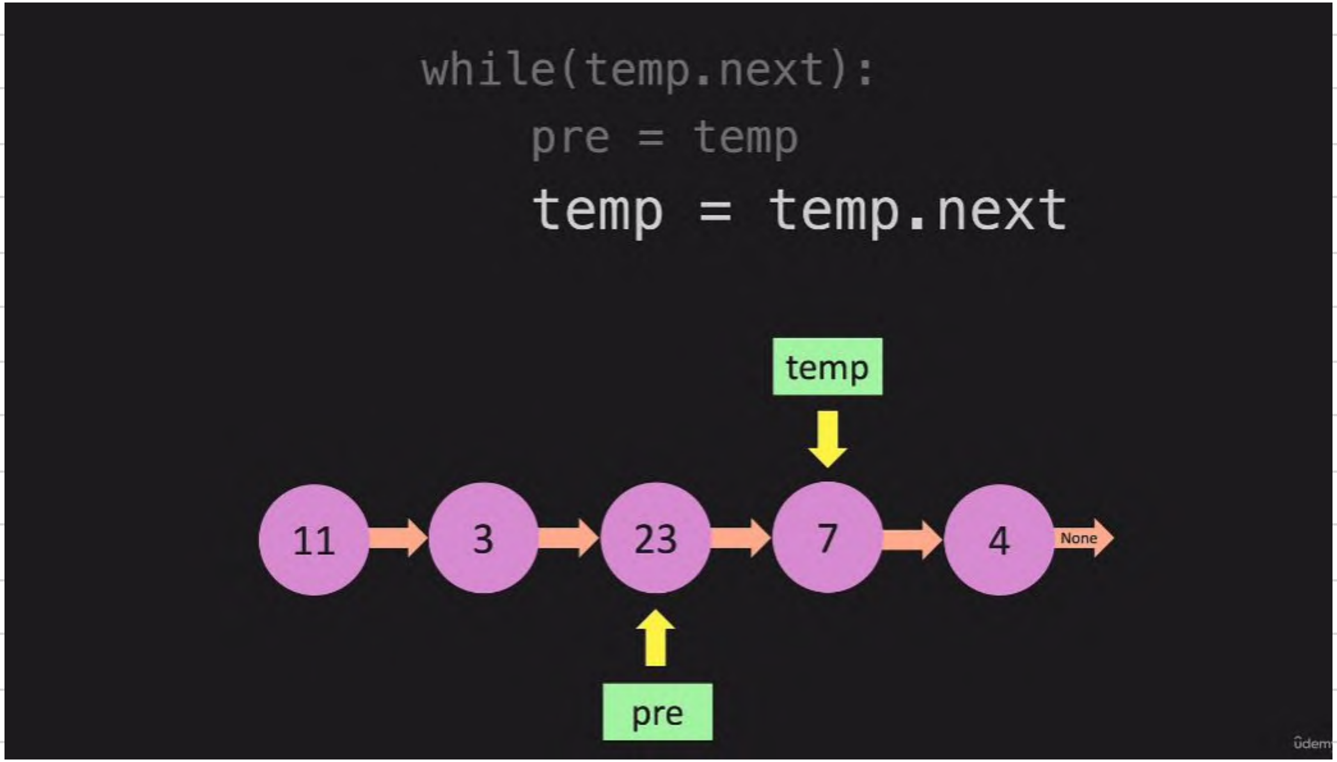
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



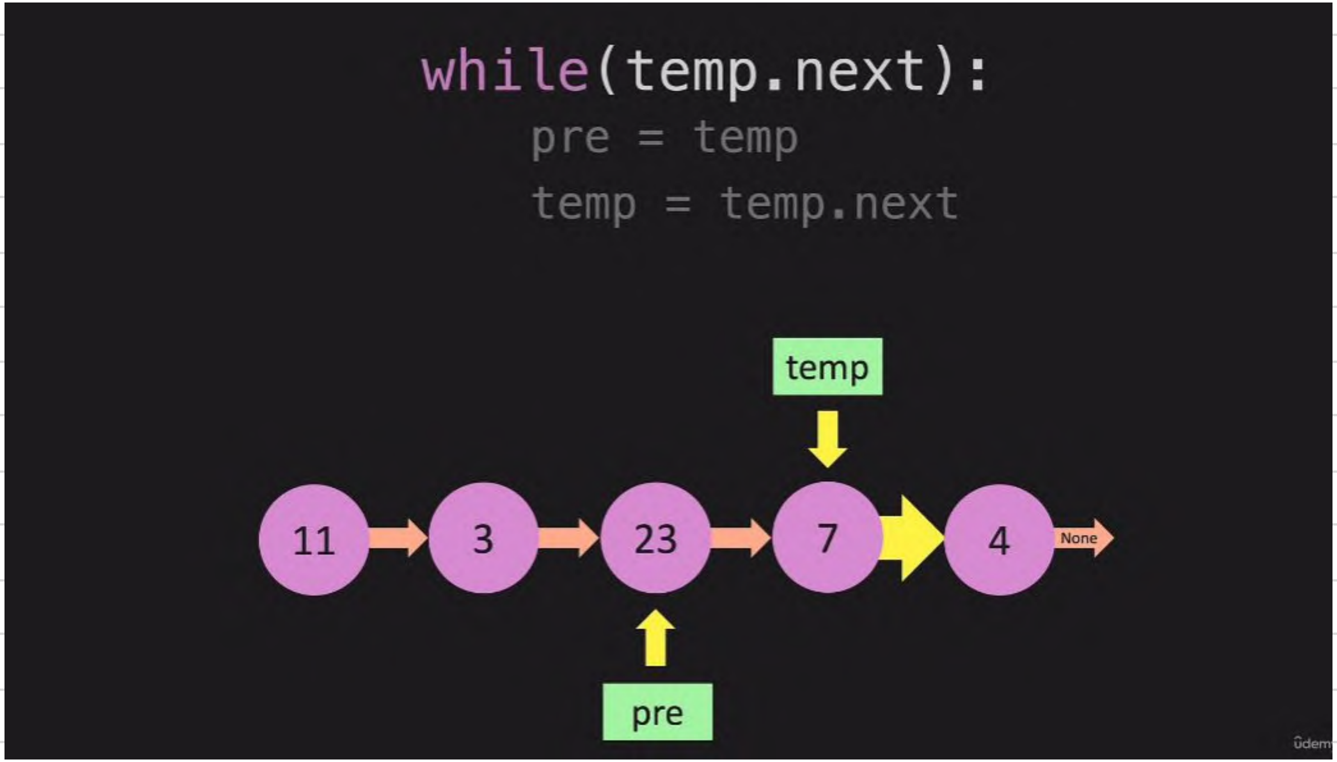
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



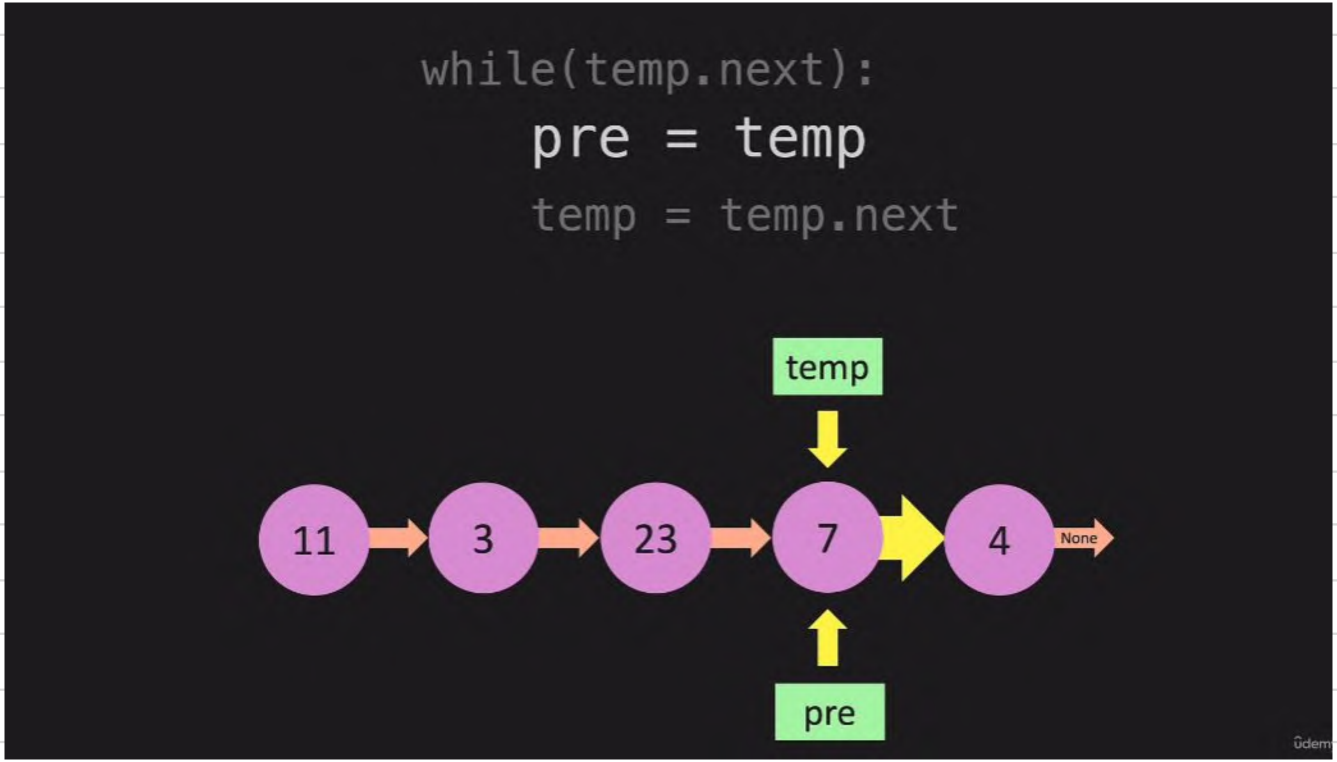
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



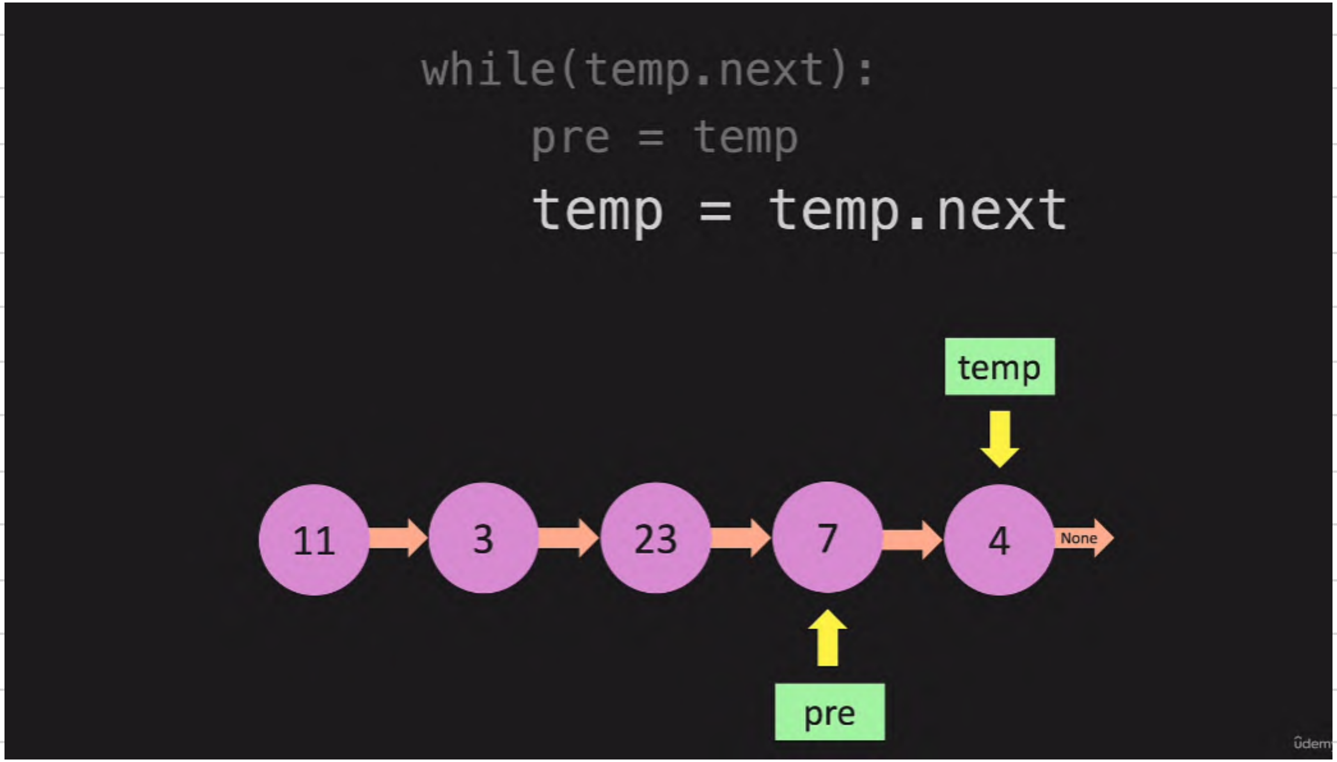
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



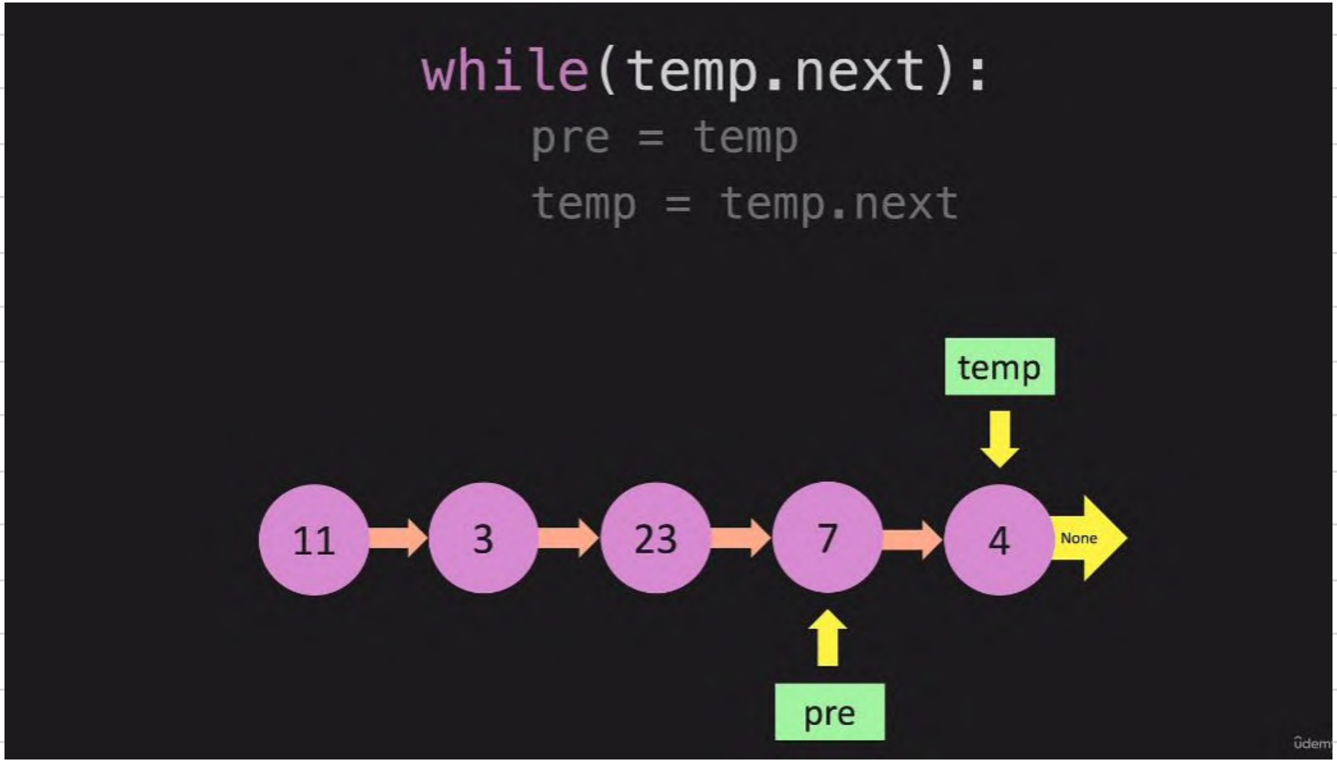
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



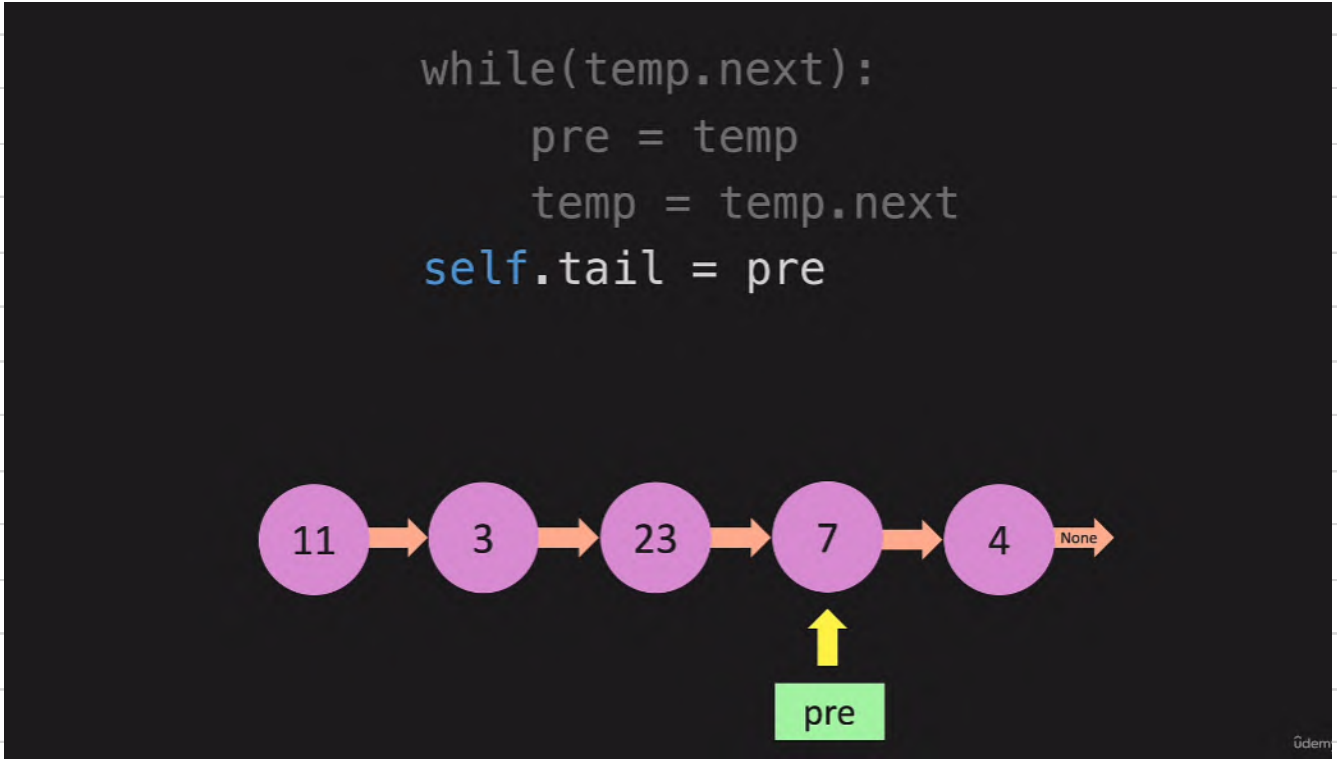
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



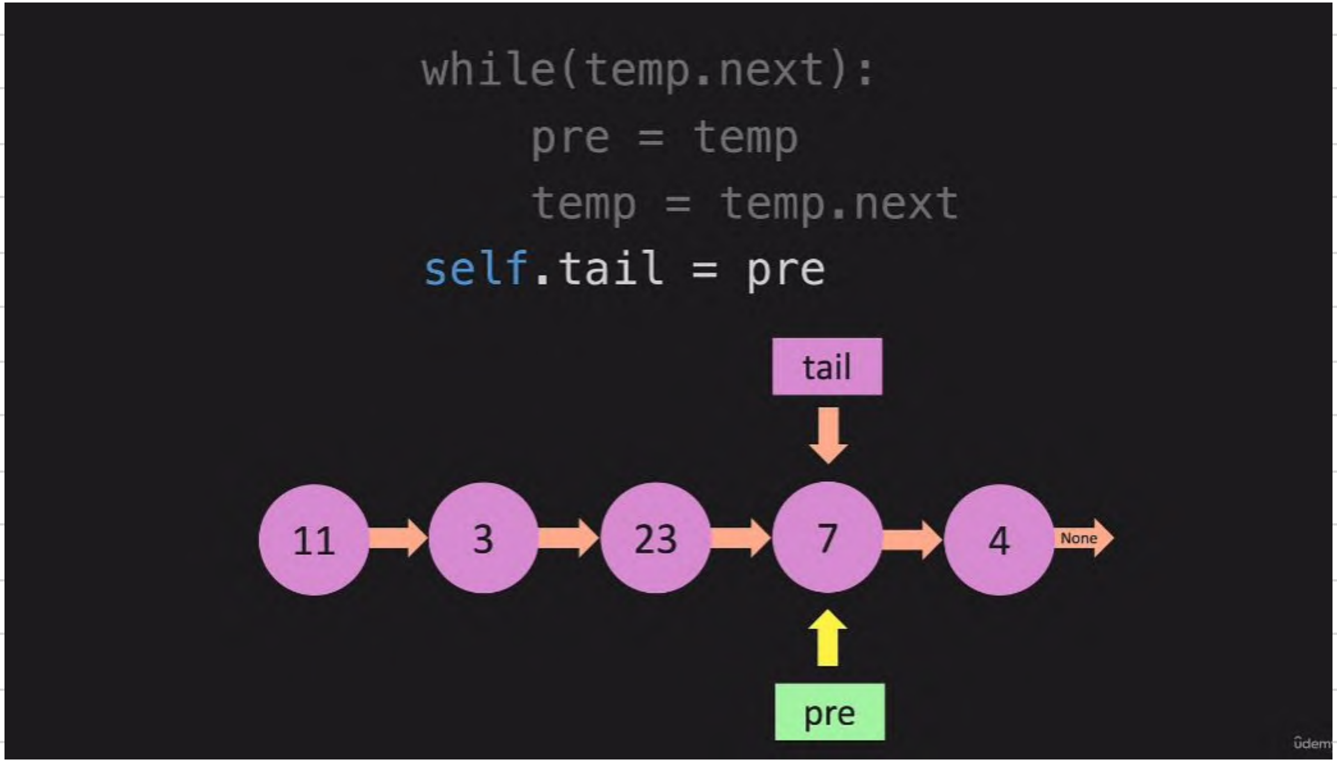
```
while(temp.next):
    pre = temp
    temp = temp.next
```

Linked List : Pop Code



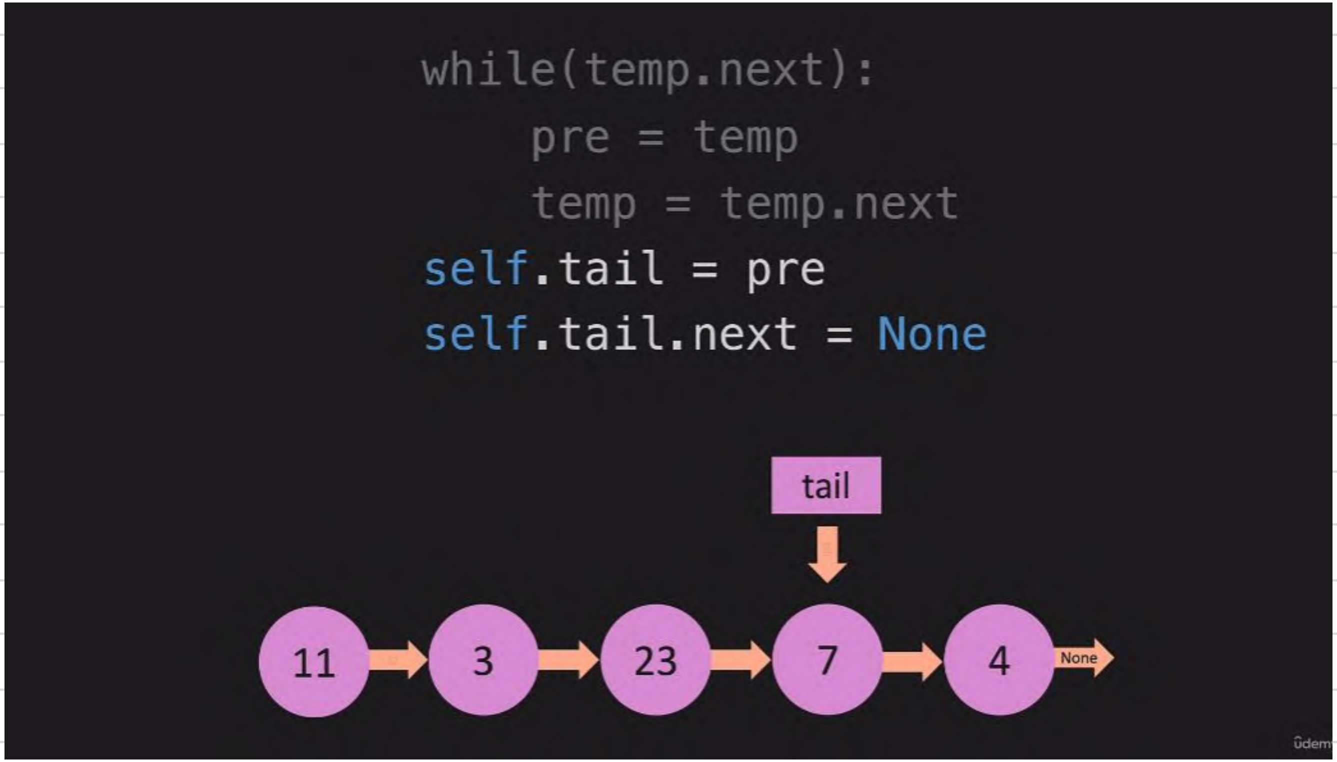
```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
```

Linked List : Pop Code



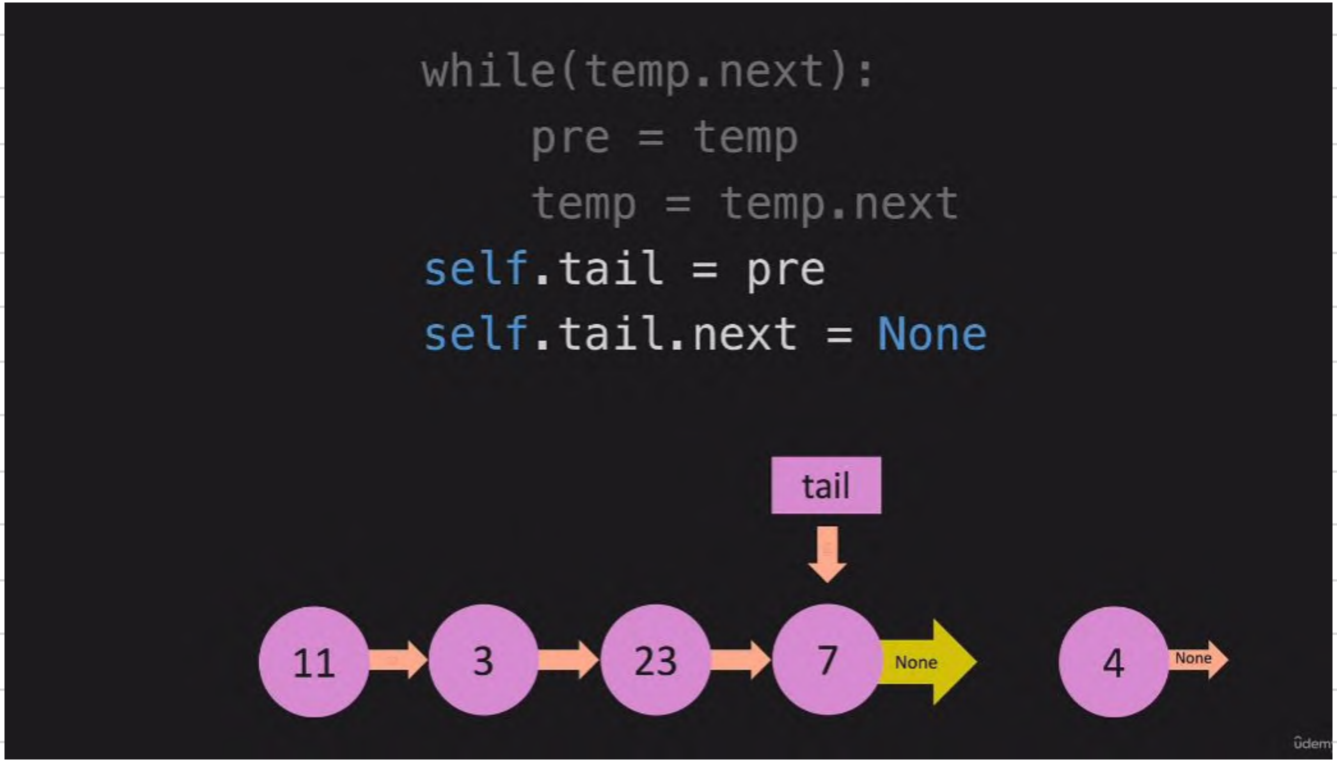
```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
```

Linked List : Pop Code



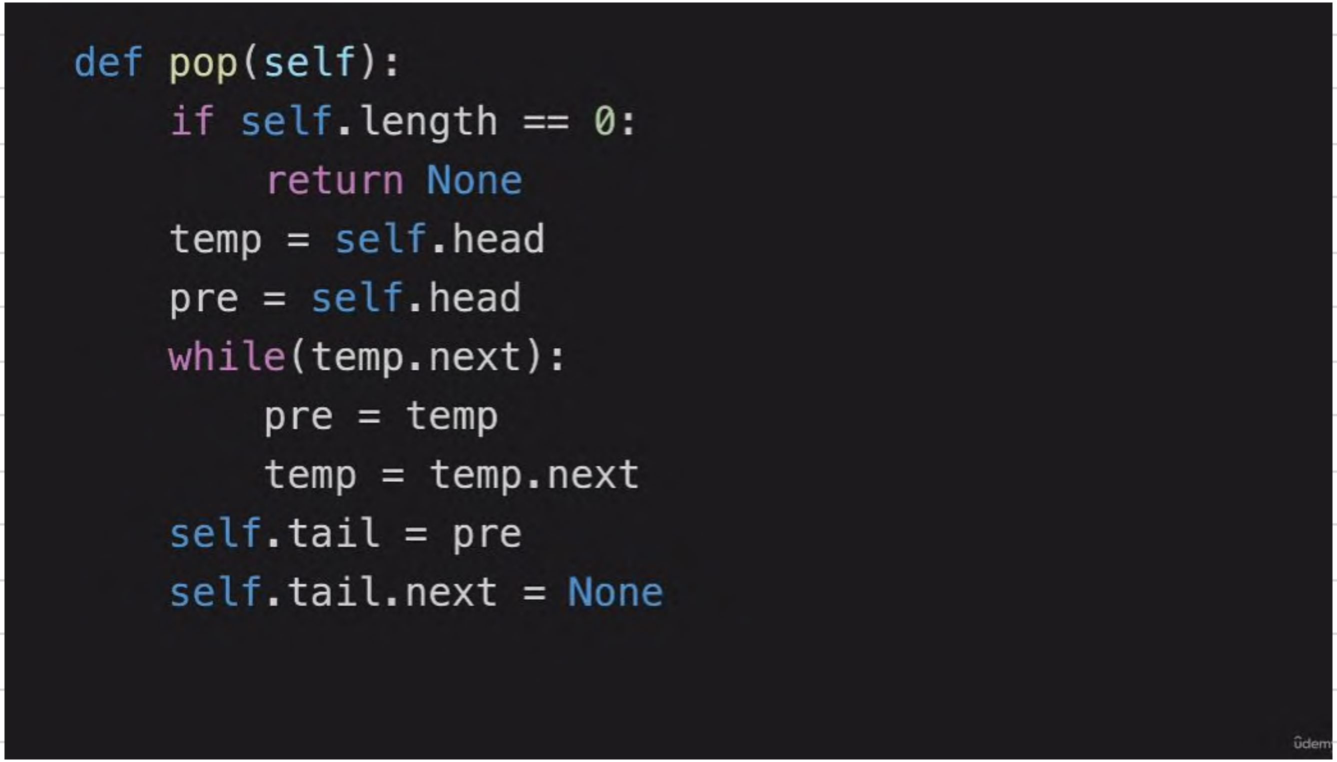
```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
self.tail.next = None
```

Linked List : Pop Code



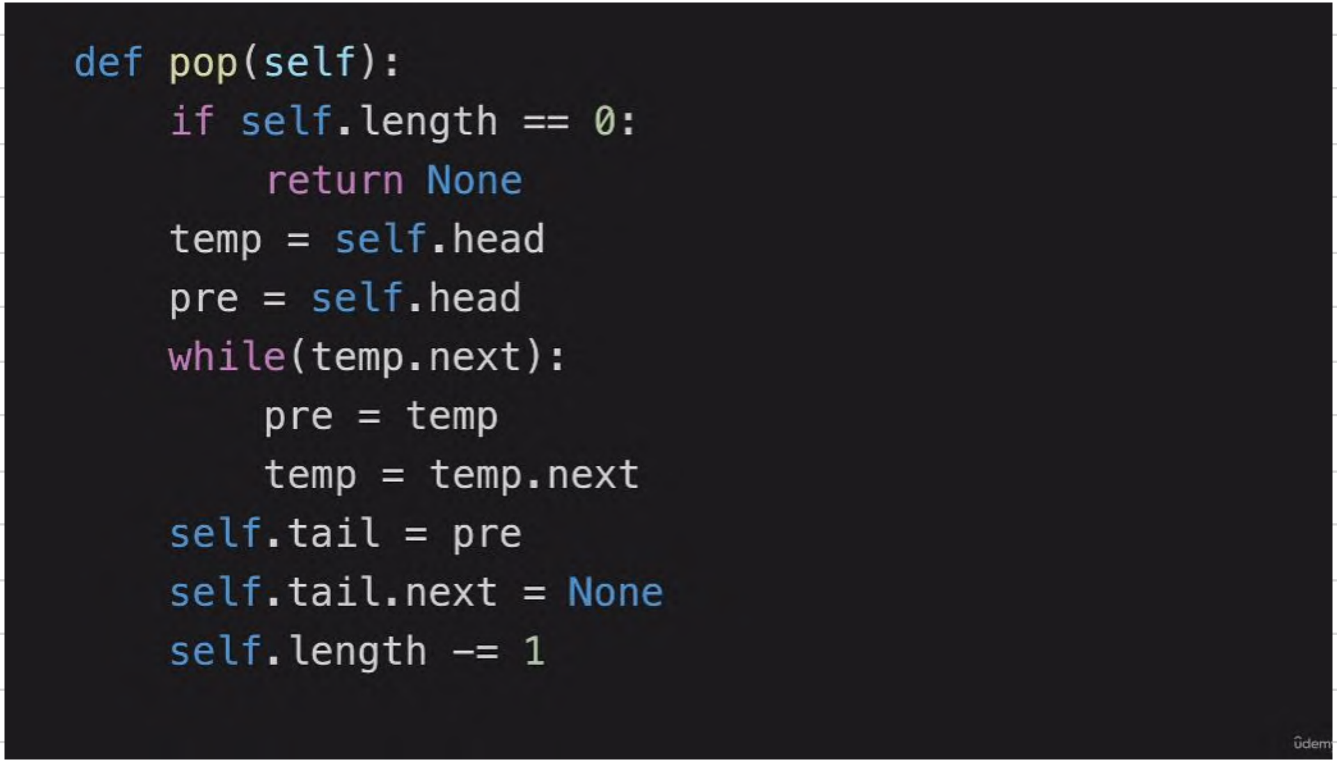
```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
self.tail.next = None
```

Linked List : Pop Code



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
```

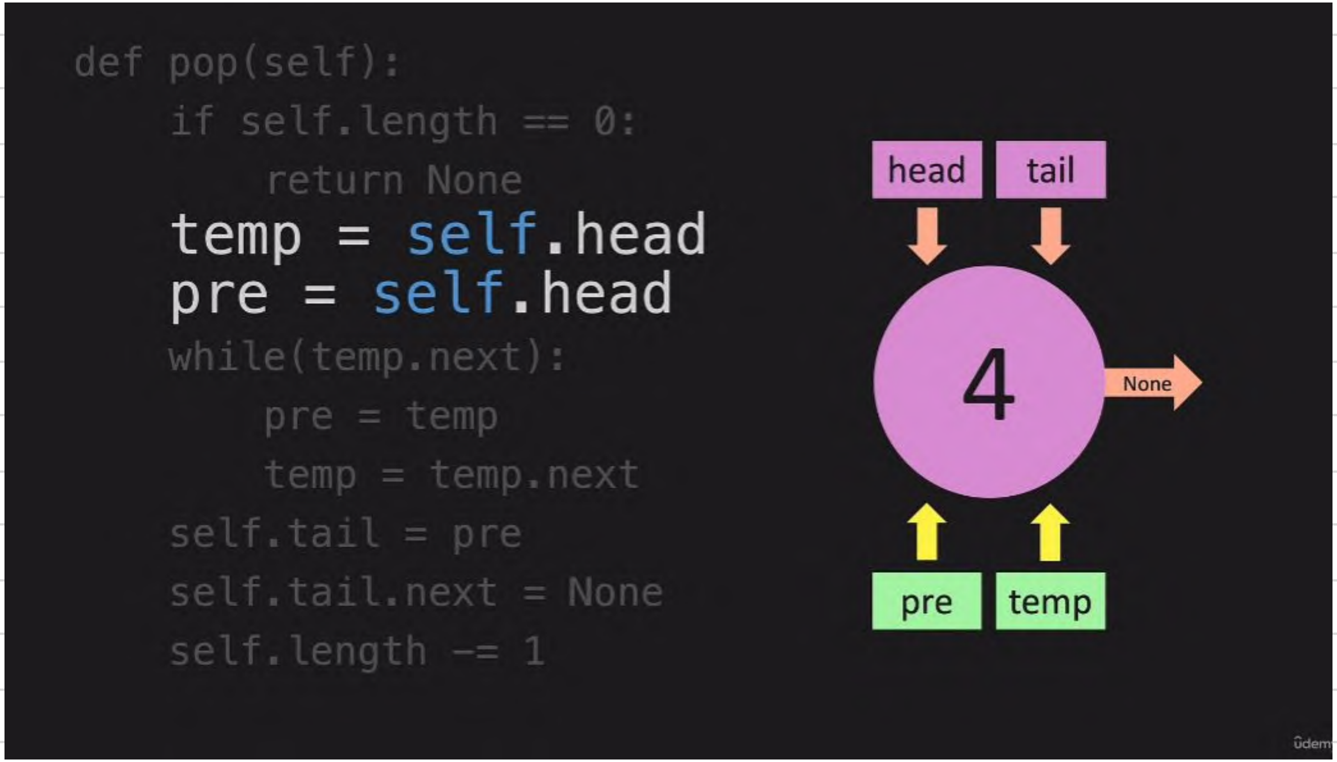
Linked List : Pop Code



```python
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
```

Kasus #2

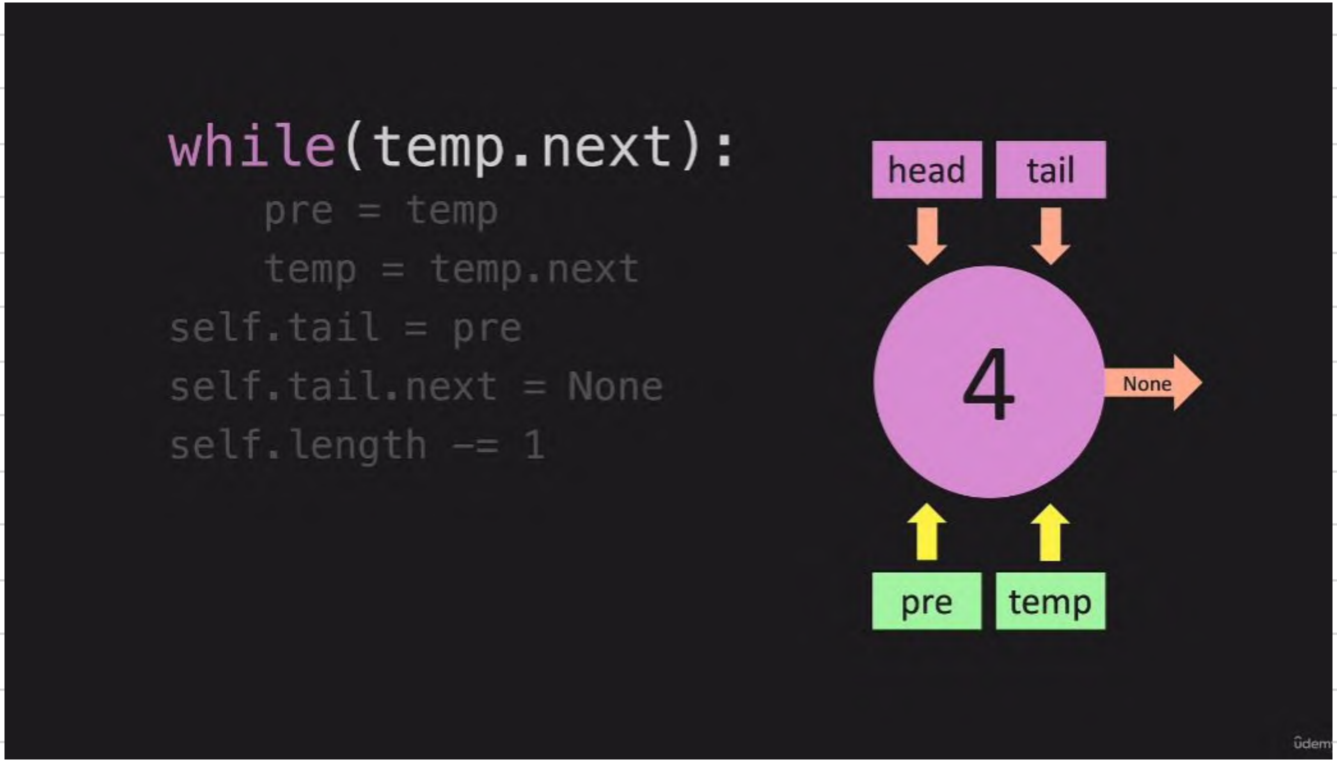
Linked List : Pop Code



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
```

Kasus #2

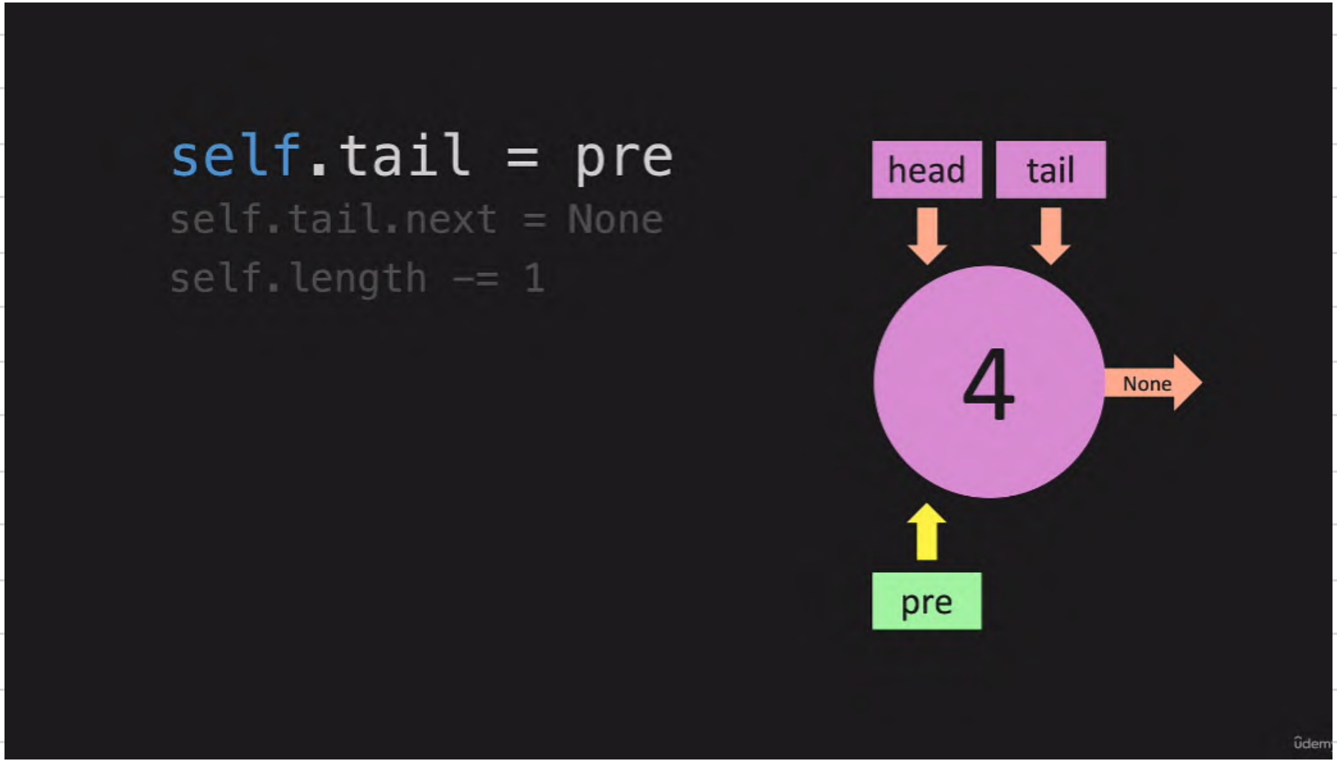
Linked List : Pop Code



```
while(temp.next):
    pre = temp
    temp = temp.next
self.tail = pre
self.tail.next = None
self.length -= 1
```

Kasus #2

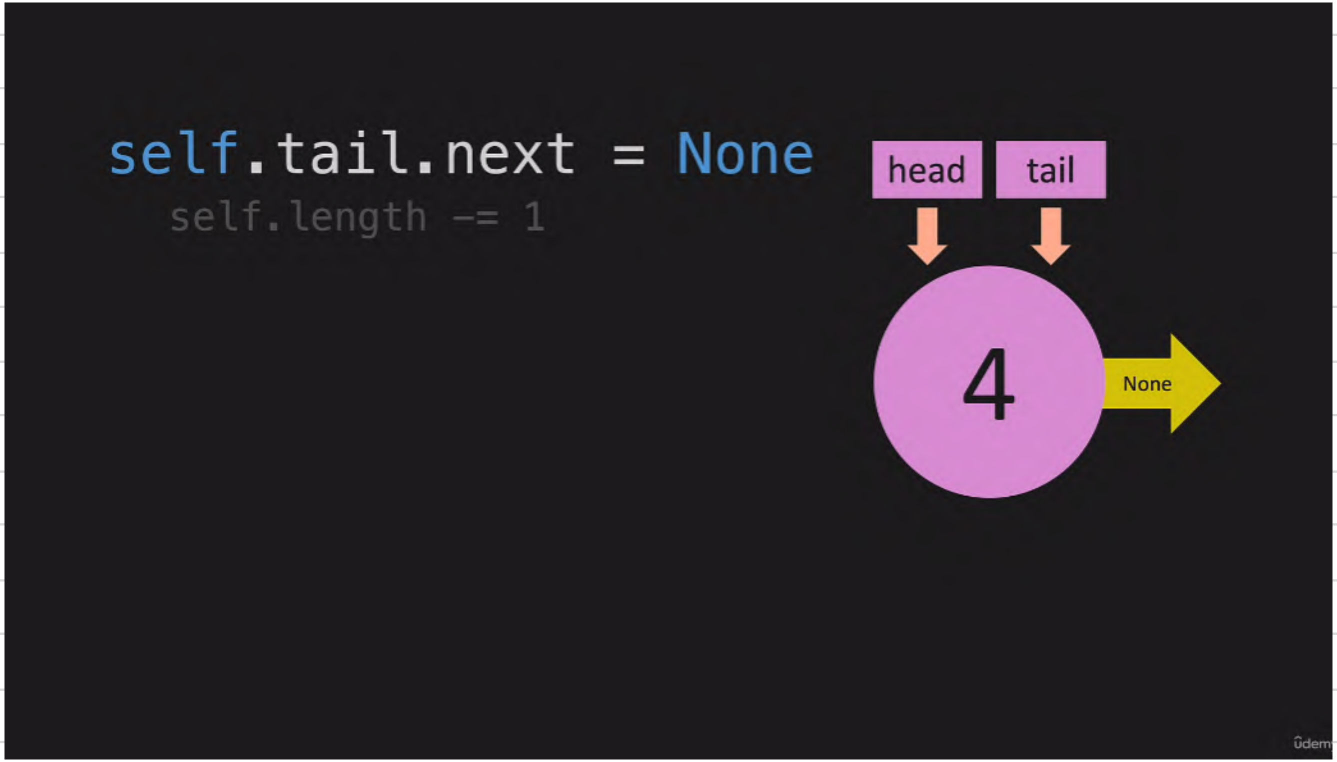
Linked List : Pop Code



```
self.tail = pre
self.tail.next = None
self.length -= 1
```

Kasus #2

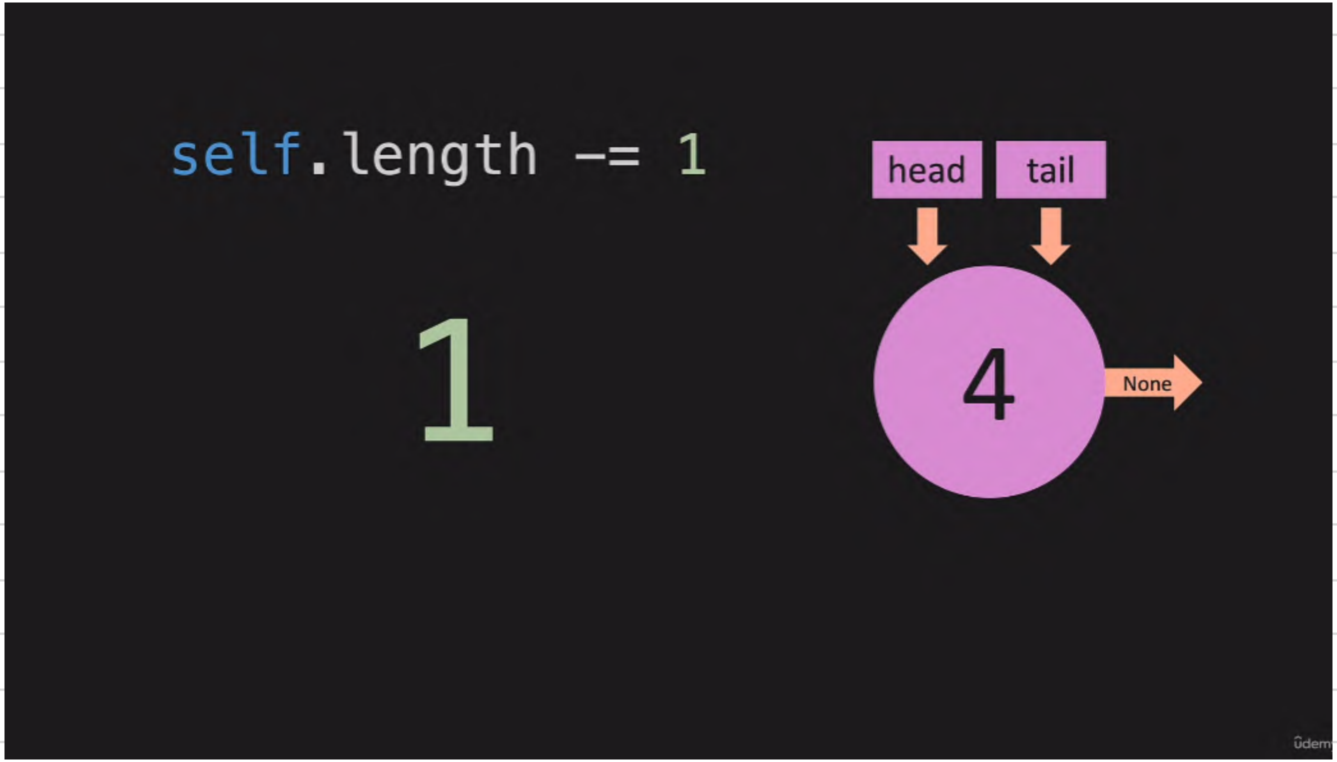
Linked List : Pop Code



```
self.tail.next = None
    self.length -= 1
```

Kasus #2

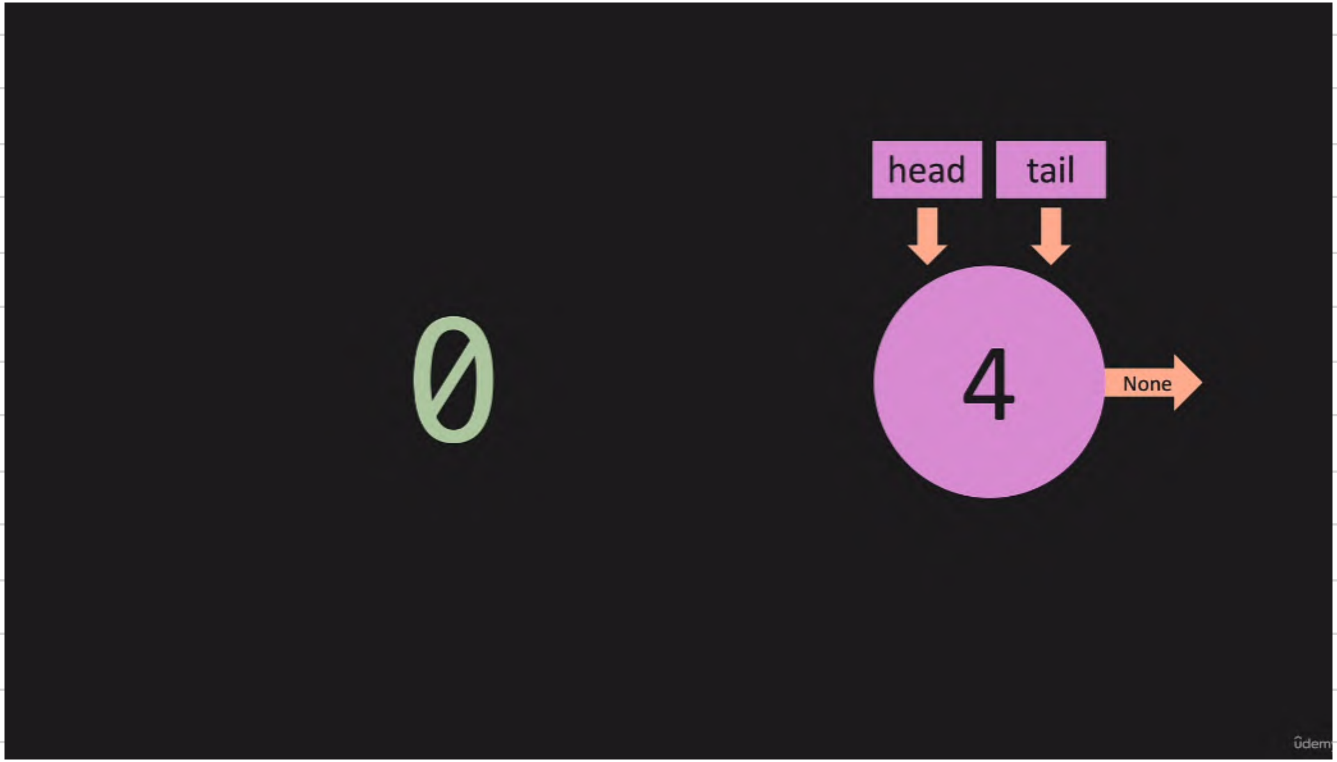
Linked List : Pop Code



```
self.length -= 1
```

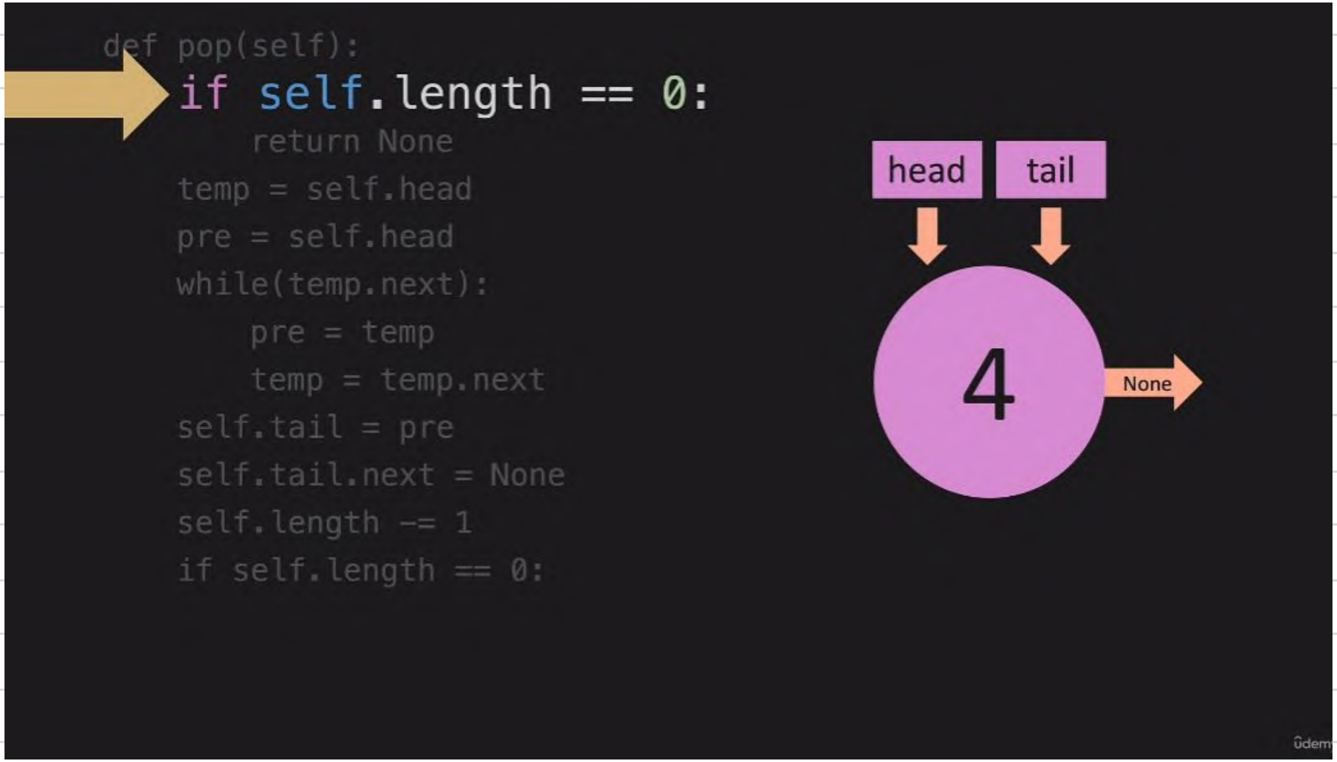
Linked List : Pop Code

Kasus #2
Kondisi menjadi tidak valid
Length = 0 padahal ada 1 node



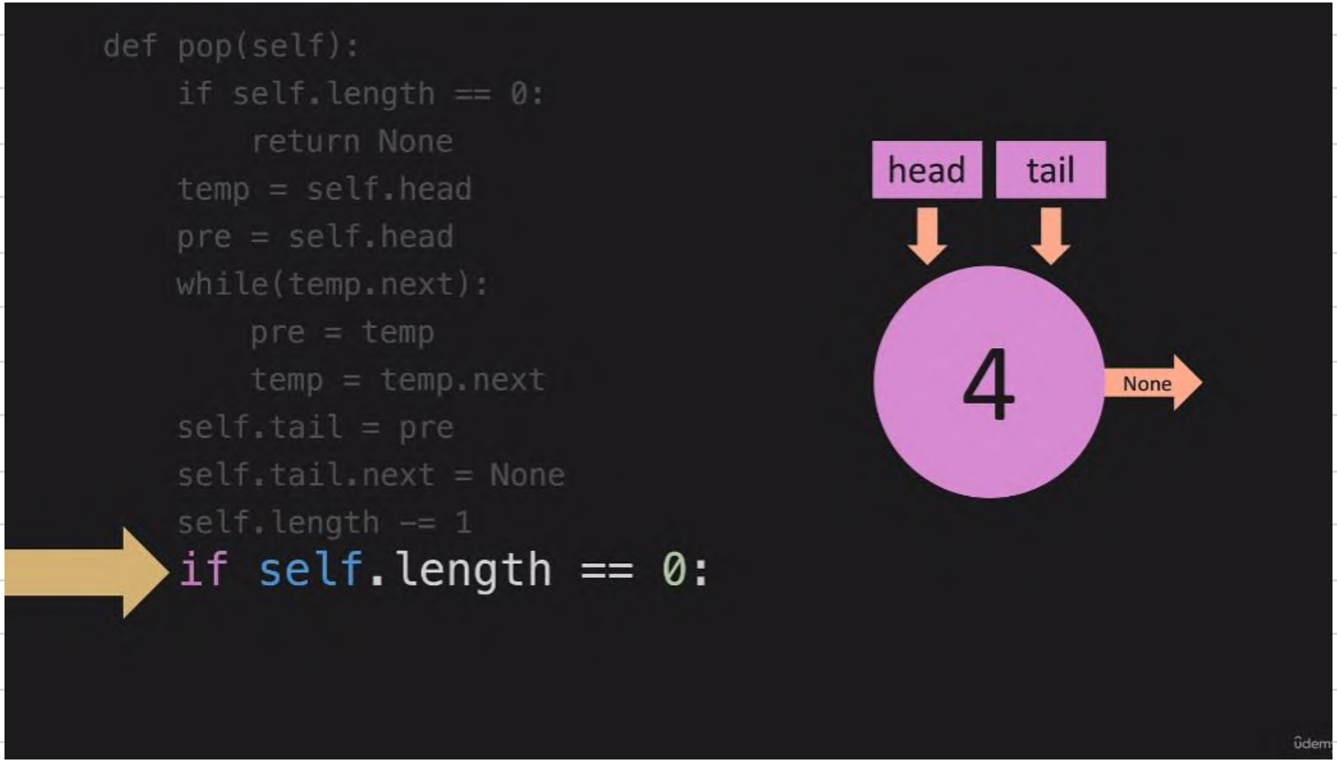
Kasus #2
Solusi

Linked List : Pop Code


```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
```

Linked List : Pop Code

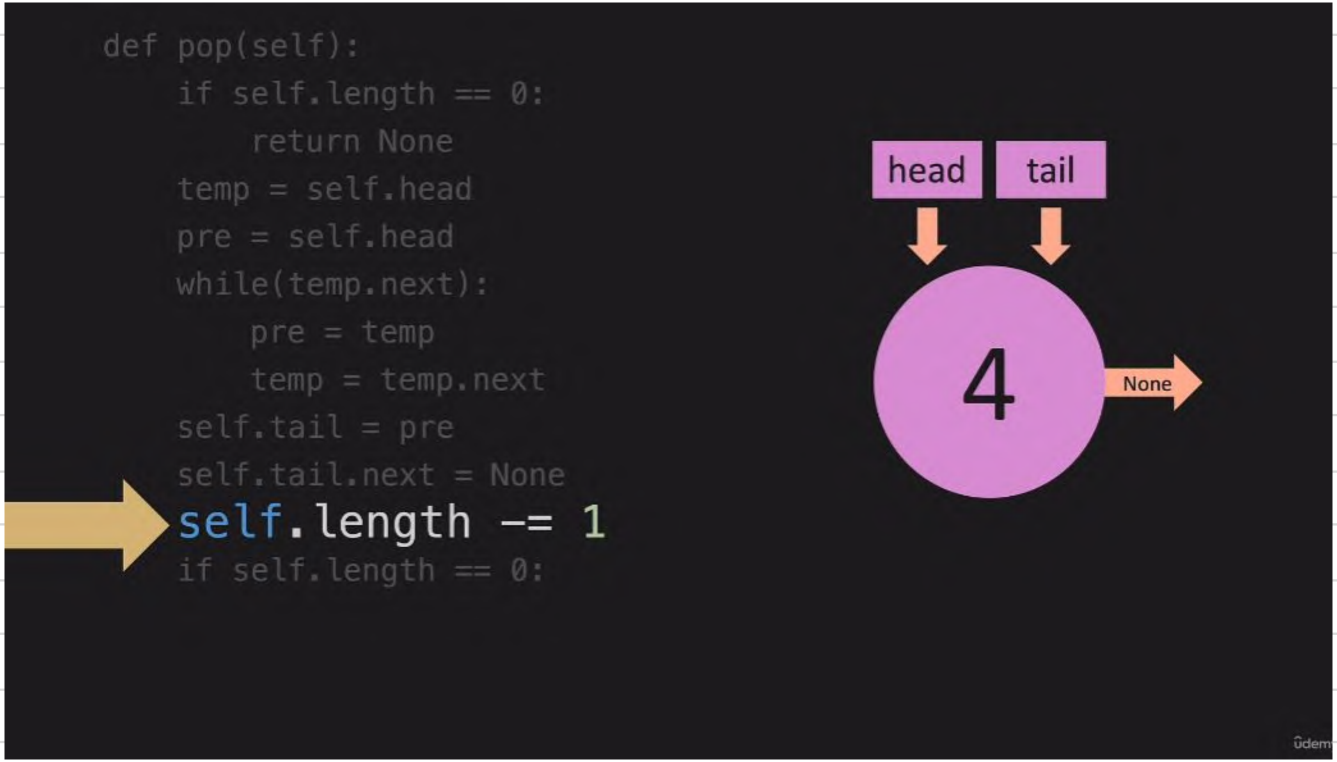
Kasus #2
Solusi



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
```

Linked List : Pop Code

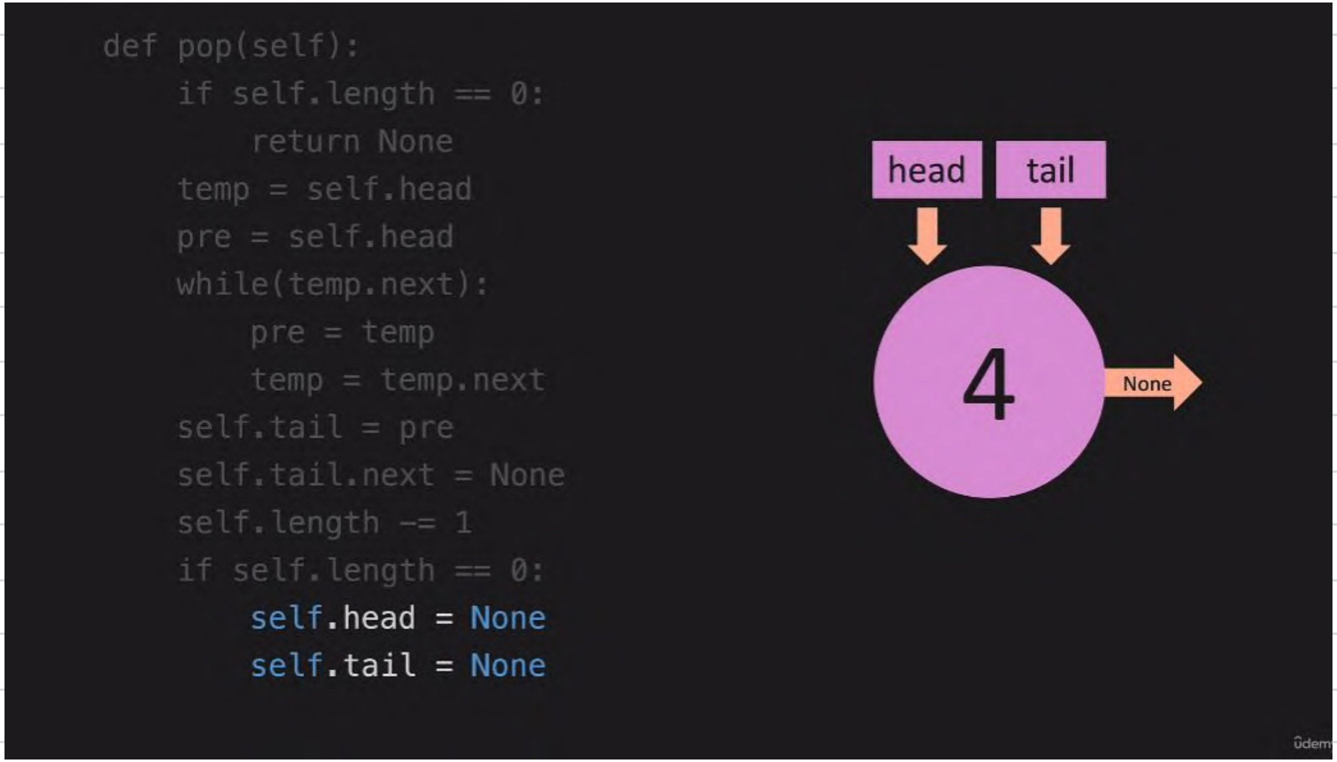
Kasus #2
Solusi



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
```

Linked List : Pop Code

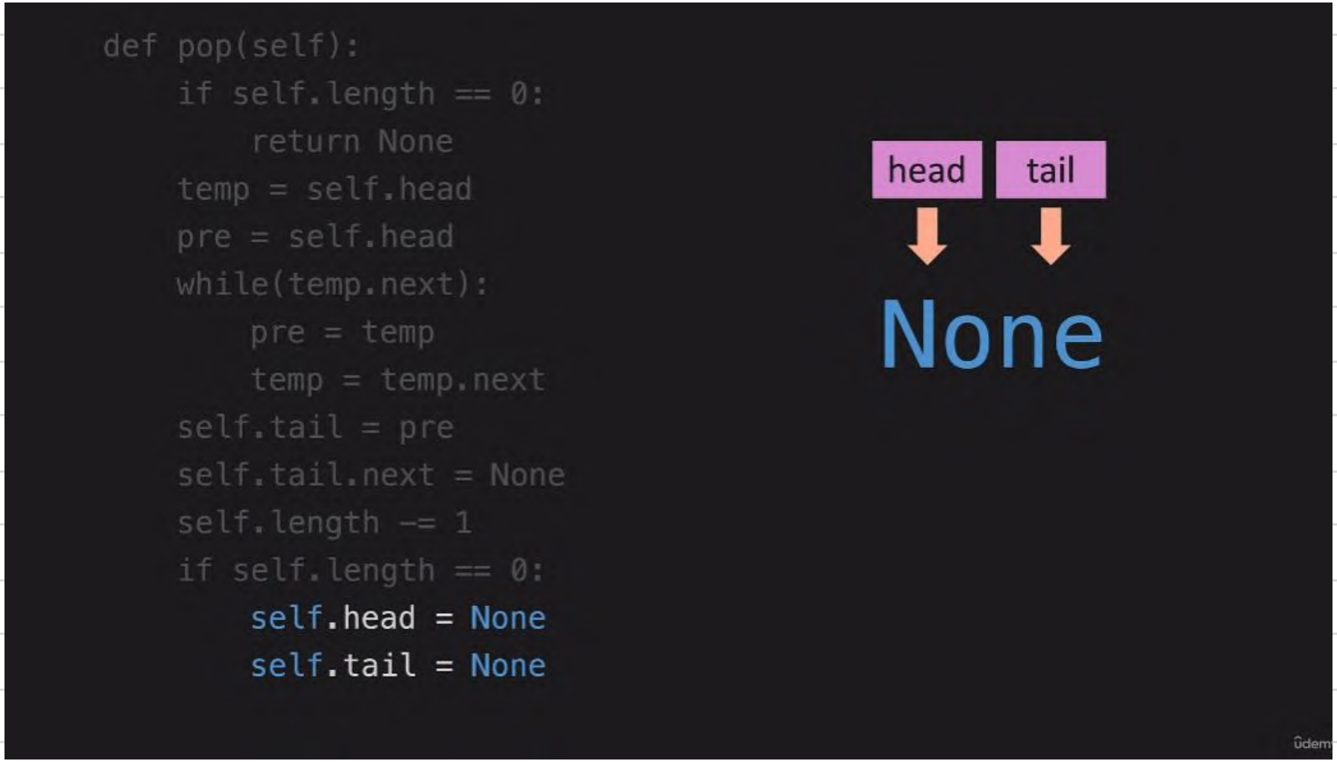
Kasus #2
Solusi



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
        self.head = None
        self.tail = None
```

Linked List : Pop Code

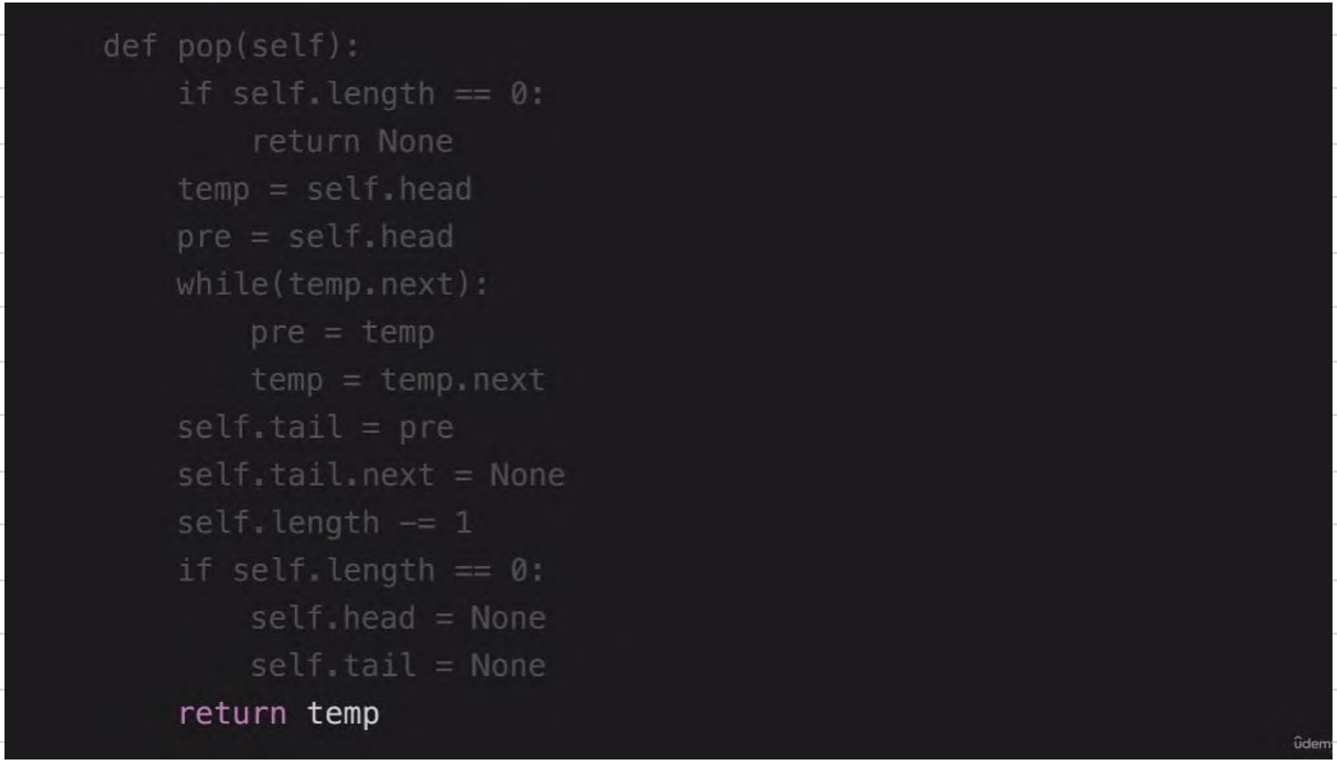
Kasus #2
Solusi



```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
        self.head = None
        self.tail = None
```

Linked List : Pop Code

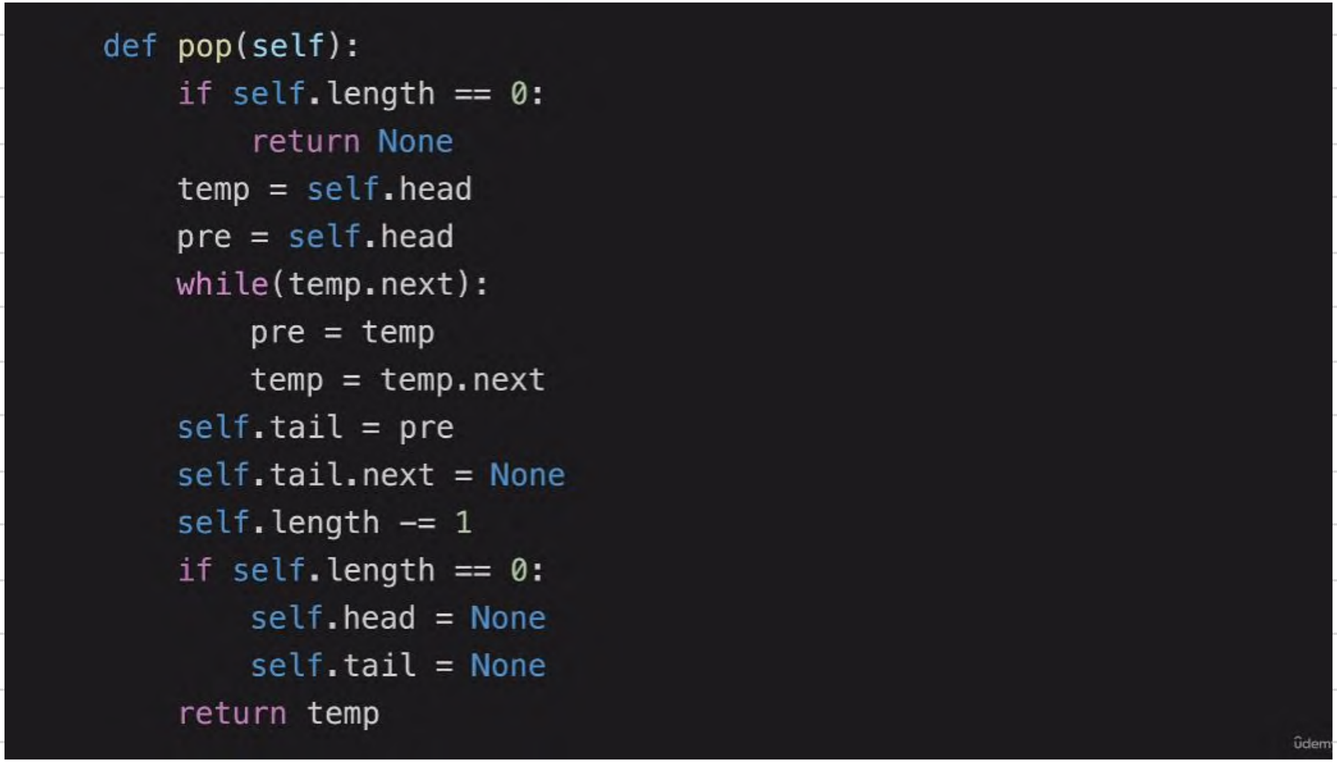
Kasus #2
Solusi



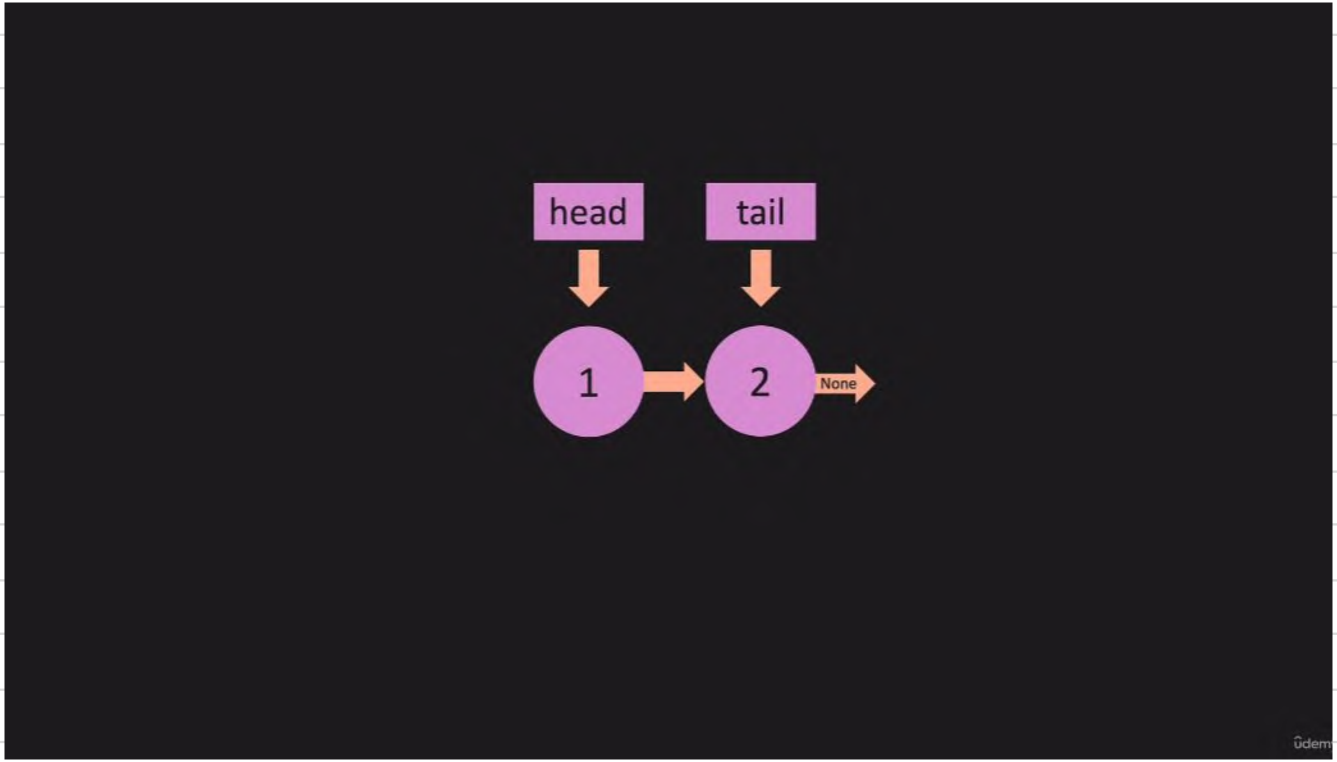
```
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
        self.head = None
        self.tail = None
    return temp
```

Linked List : Pop Code

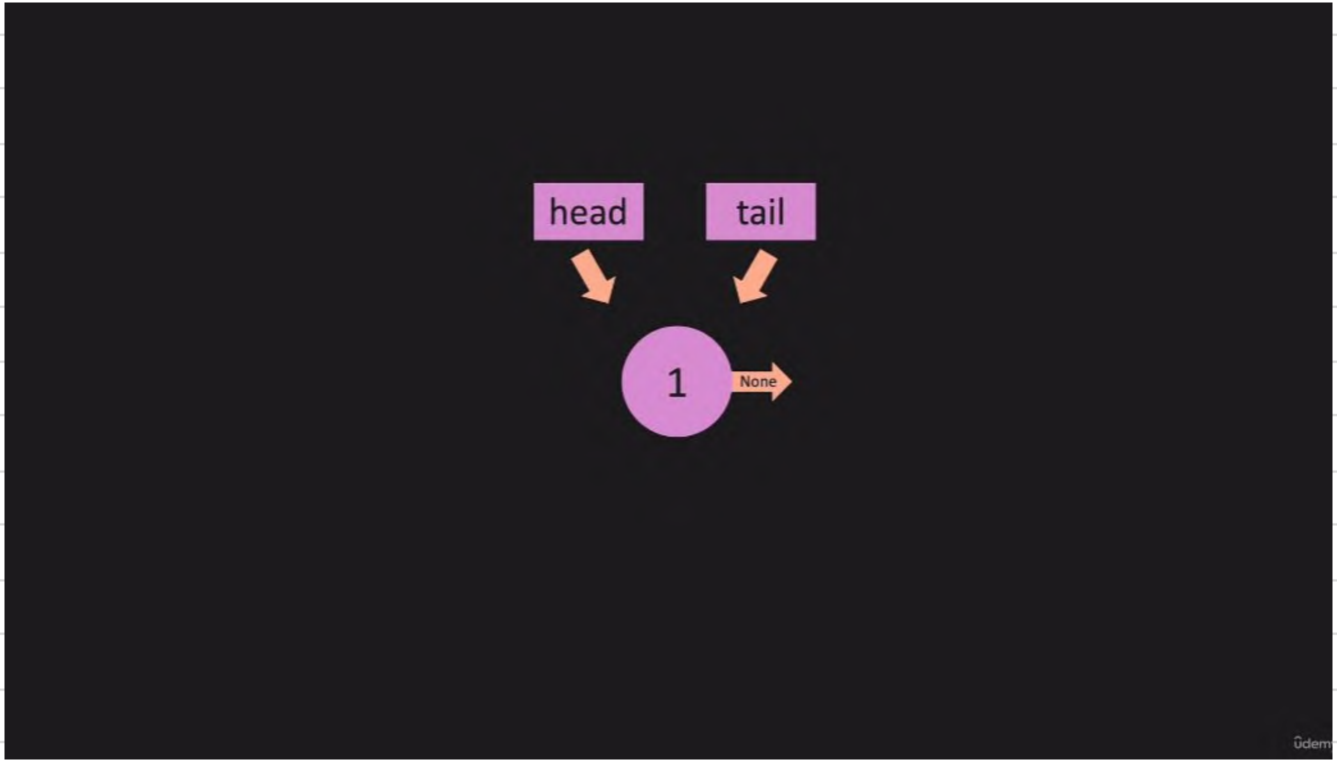
Kasus #2
Solusi


```python
def pop(self):
    if self.length == 0:
        return None
    temp = self.head
    pre = self.head
    while(temp.next):
        pre = temp
        temp = temp.next
    self.tail = pre
    self.tail.next = None
    self.length -= 1
    if self.length == 0:
        self.head = None
        self.tail = None
    return temp
```

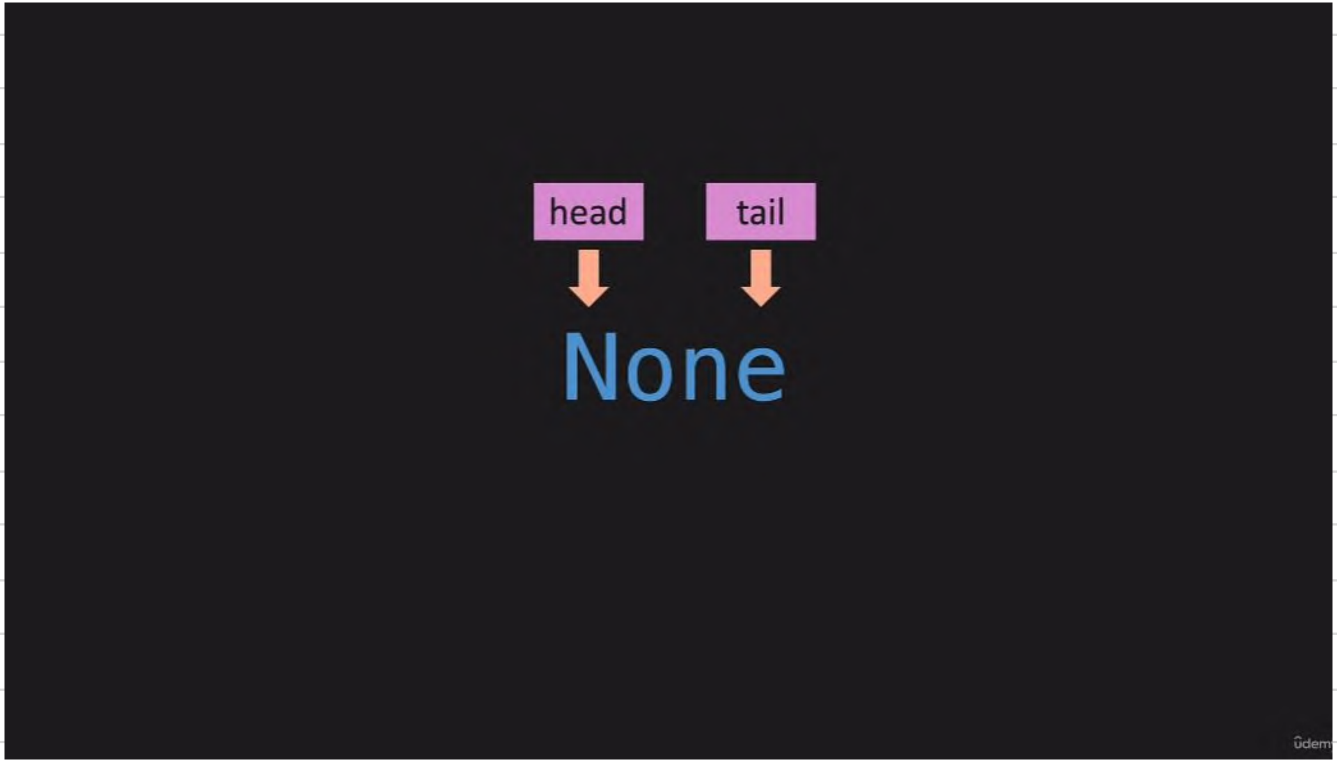
Linked List : Pop Skenario Percobaan



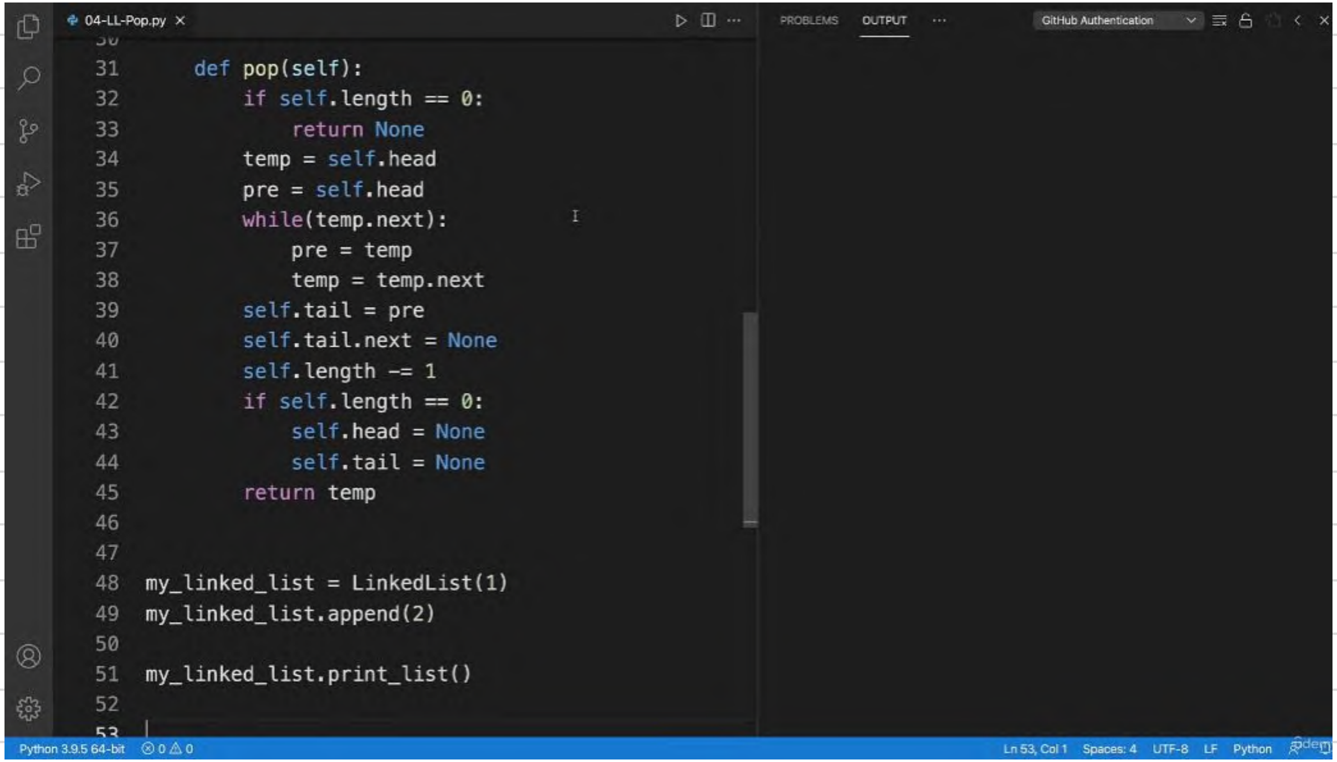
Linked List : Pop Skenario Percobaan



Linked List : Pop Skenario Percobaan

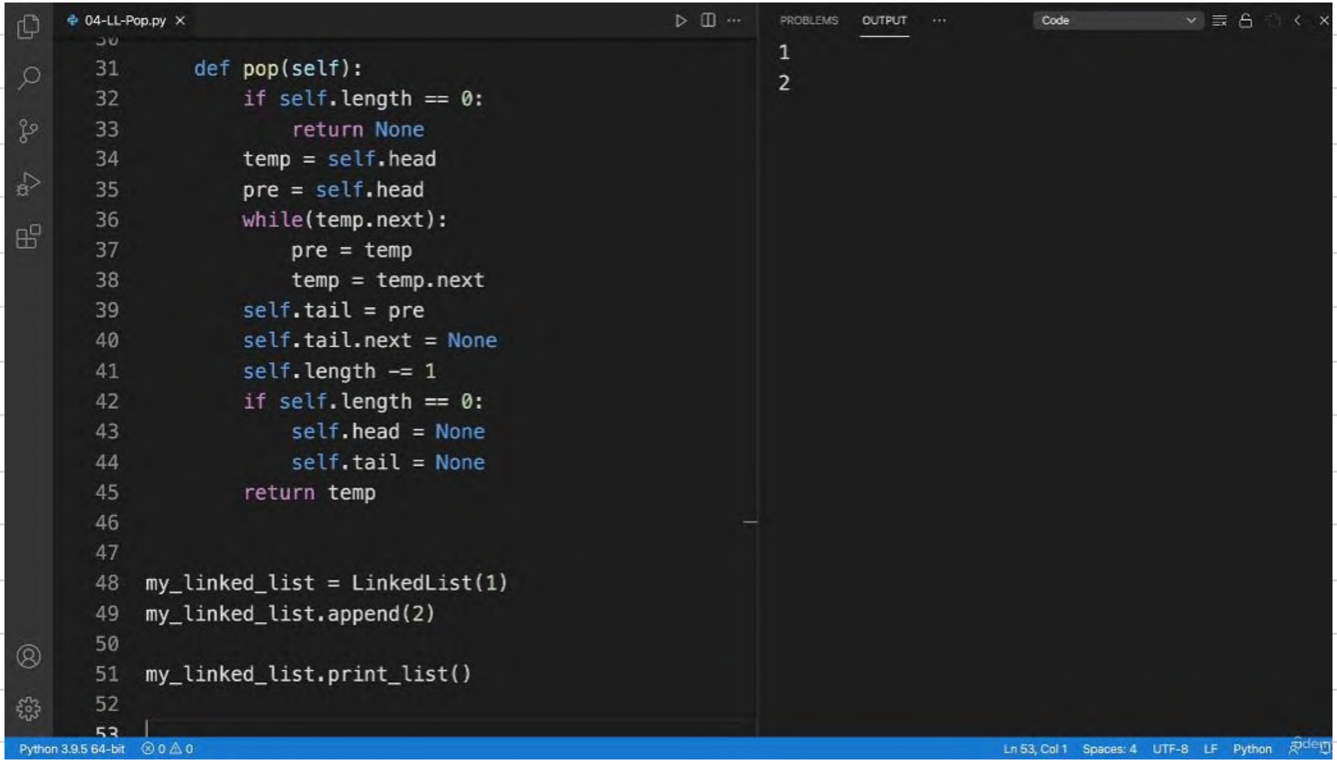


Linked List : Pop Code



```
    def pop(self):
        if self.length == 0:
            return None
        temp = self.head
        pre = self.head
        while(temp.next):
            pre = temp
            temp = temp.next
        self.tail = pre
        self.tail.next = None
        self.length -= 1
        if self.length == 0:
            self.head = None
            self.tail = None
        return temp


my_linked_list = LinkedList(1)
my_linked_list.append(2)

my_linked_list.print_list()

```

Linked List : Pop Code

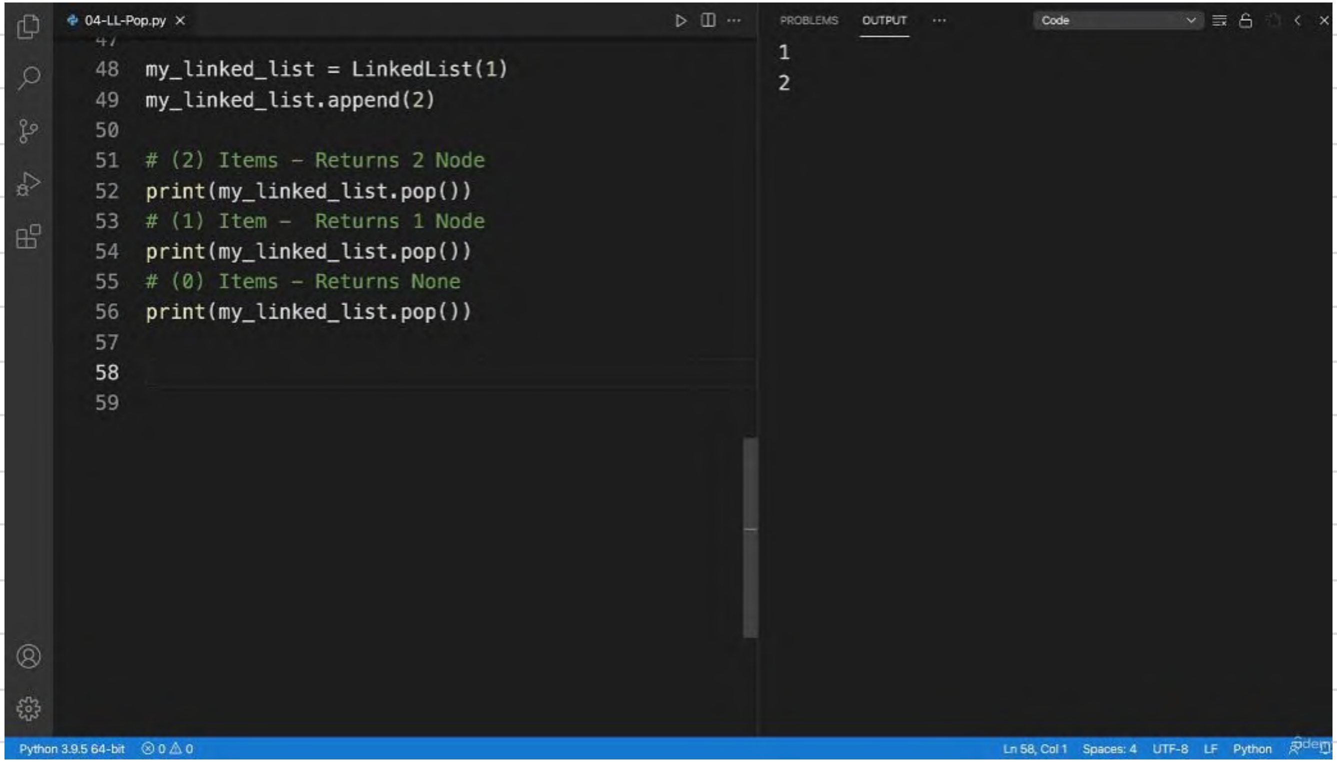


04-LL-Pop.py

```python
    def pop(self):
        if self.length == 0:
            return None
        temp = self.head
        pre = self.head
        while(temp.next):
            pre = temp
            temp = temp.next
        self.tail = pre
        self.tail.next = None
        self.length -= 1
        if self.length == 0:
            self.head = None
            self.tail = None
        return temp


my_linked_list = LinkedList(1)
my_linked_list.append(2)

my_linked_list.print_list()

```

```
1
2
```

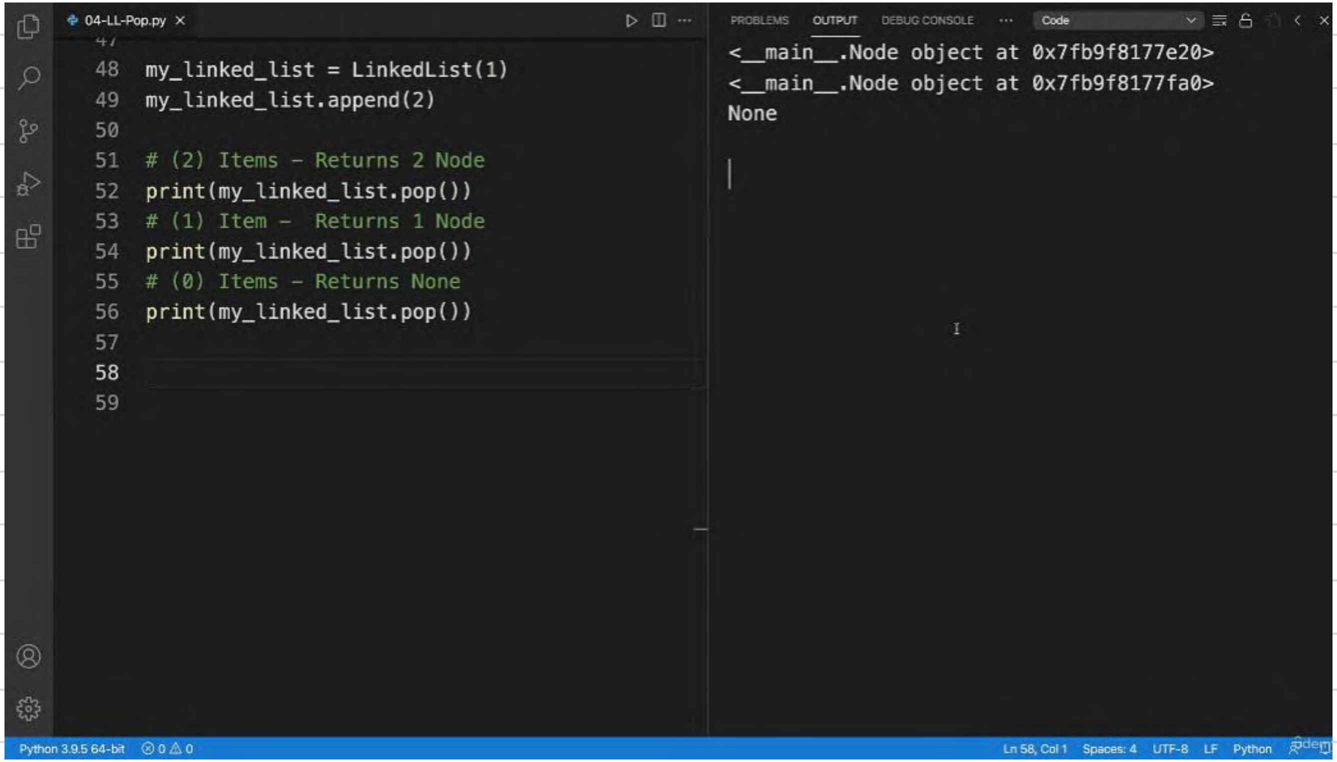
Linked List : Pop Code



```python
my_linked_list = LinkedList(1)
my_linked_list.append(2)

# (2) Items - Returns 2 Node
print(my_linked_list.pop())
# (1) Item -  Returns 1 Node
print(my_linked_list.pop())
# (0) Items - Returns None
print(my_linked_list.pop())
```

```
1
2
```

Linked List : Pop Code

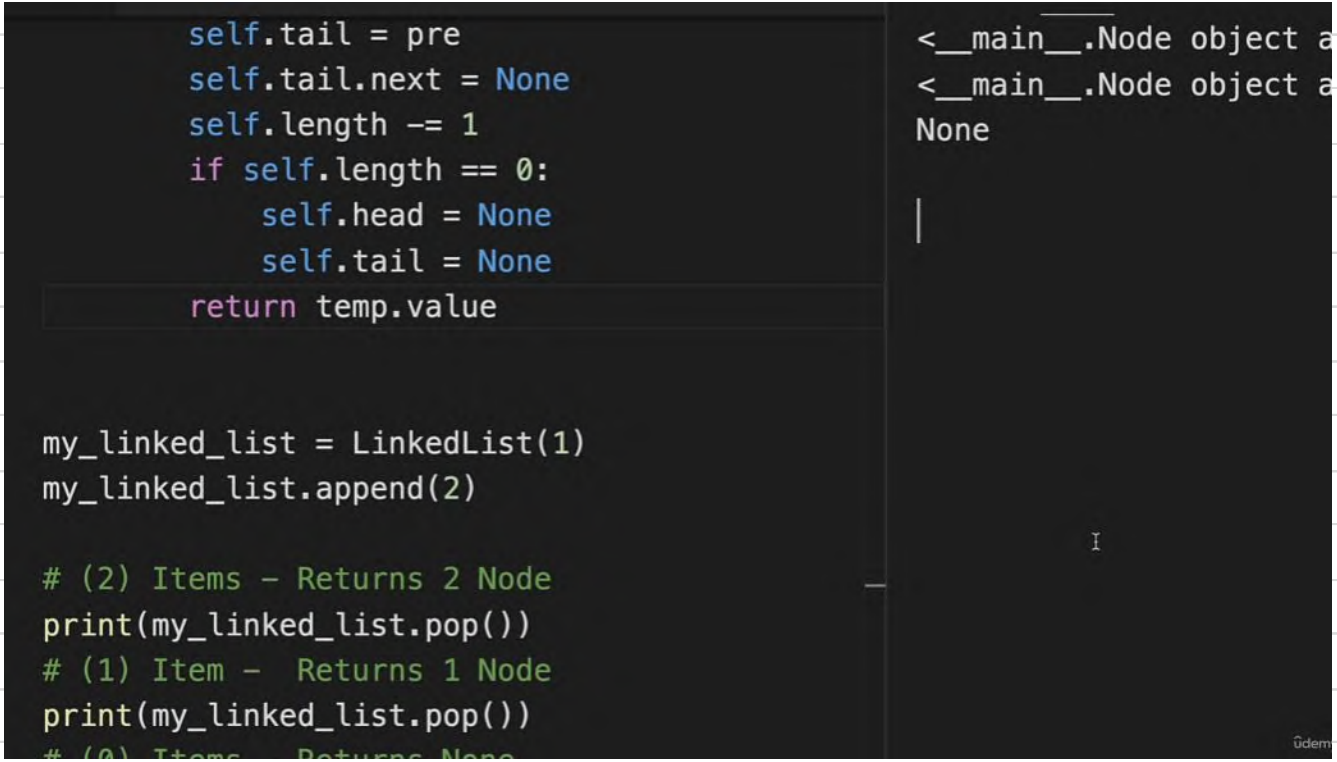


04-LL-Pop.py

```python
my_linked_list = LinkedList(1)
my_linked_list.append(2)

# (2) Items - Returns 2 Node
print(my_linked_list.pop())
# (1) Item -  Returns 1 Node
print(my_linked_list.pop())
# (0) Items - Returns None
print(my_linked_list.pop())
```

```
<__main__.Node object at 0x7fb9f8177e20>
<__main__.Node object at 0x7fb9f8177fa0>
None
```

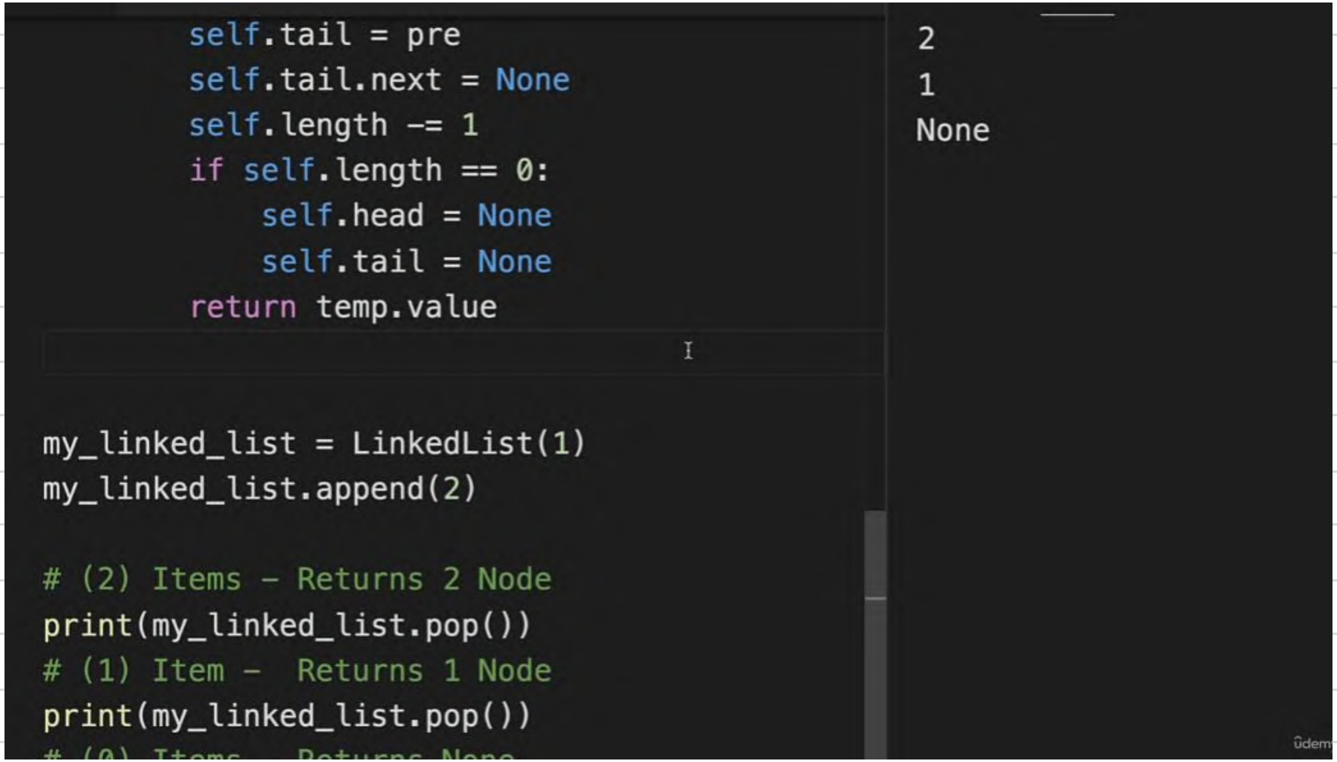
Linked List : Pop Code



```
        self.tail = pre
        self.tail.next = None
        self.length -= 1
        if self.length == 0:
            self.head = None
            self.tail = None
        return temp.value


my_linked_list = LinkedList(1)
my_linked_list.append(2)

# (2) Items - Returns 2 Node
print(my_linked_list.pop())
# (1) Item -  Returns 1 Node
print(my_linked_list.pop())
```

```
<__main__.Node object a
<__main__.Node object a
None
```

Linked List : Pop Code



```
            self.tail = pre
            self.tail.next = None
            self.length -= 1
            if self.length == 0:
                self.head = None
                self.tail = None
            return temp.value


my_linked_list = LinkedList(1)
my_linked_list.append(2)

# (2) Items - Returns 2 Node
print(my_linked_list.pop())
# (1) Item -  Returns 1 Node
print(my_linked_list.pop())
```

```
2
1
None
```

Terima Kasih

Ada **Pertanyaan?**