Data Structures

Display Nodes

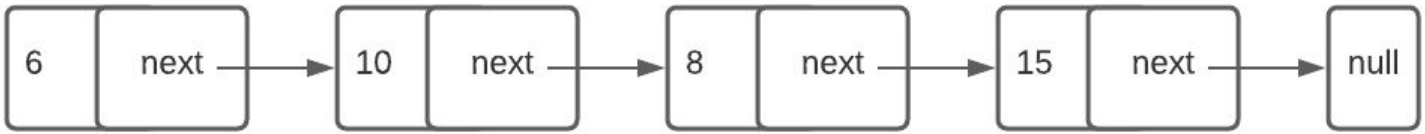
Mostafa S. Ibrahim
Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

So far

- We learned how to create and do manual navigation of the items
- Let’s write a function that prints all values starting from the head (4 here)

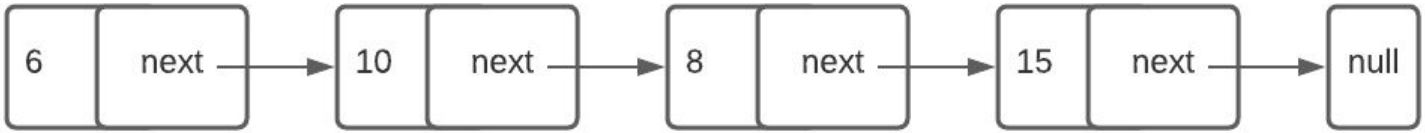
```
print(node1.next.next.next.data)     # 15
print(node2.next.next.data)          # 15
print(node3.next.data)               # 15
print(node4.data)                    # 15
```



Printing The Node Chain

- Let’s build over the last code
- From main:
  - print_lst(node1);
- The output is: 6->10->8->15->
- Take 10 minutes to trace and verify

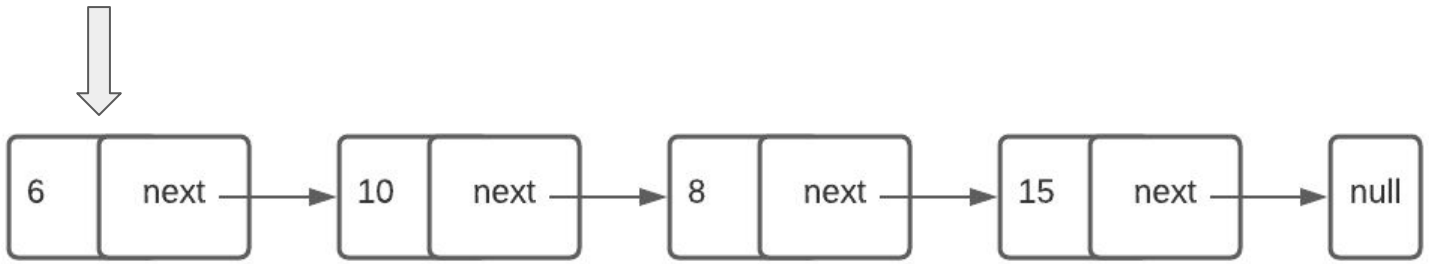
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Trace

- The head node initially points to node 1
  - Let's call it a **pointer node**
    - Its reference points to a specific location
  - Does it **equal/point** to None? No
  - Print the value ⇒ 6
  - What is head.next? node2
  - Set head = node2

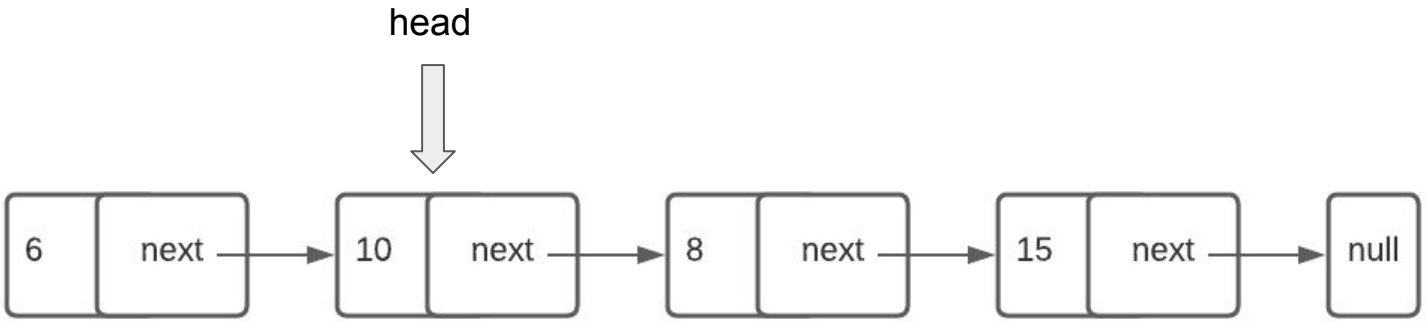
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Trace

- Now the head pointer is at node 2
  - Is None? No
  - Print the value ⇒ 10
  - What is head.next? node3
  - Set head = node3

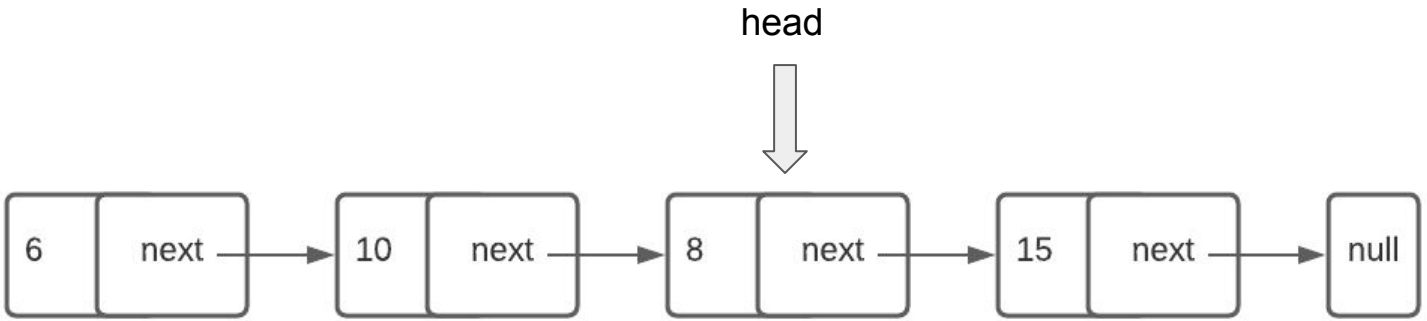
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Trace

- The pointer is pointing to node 3
  - Is None? No
  - Print the value ⇒ 8
  - What is head.next? node4
  - Set head = node4

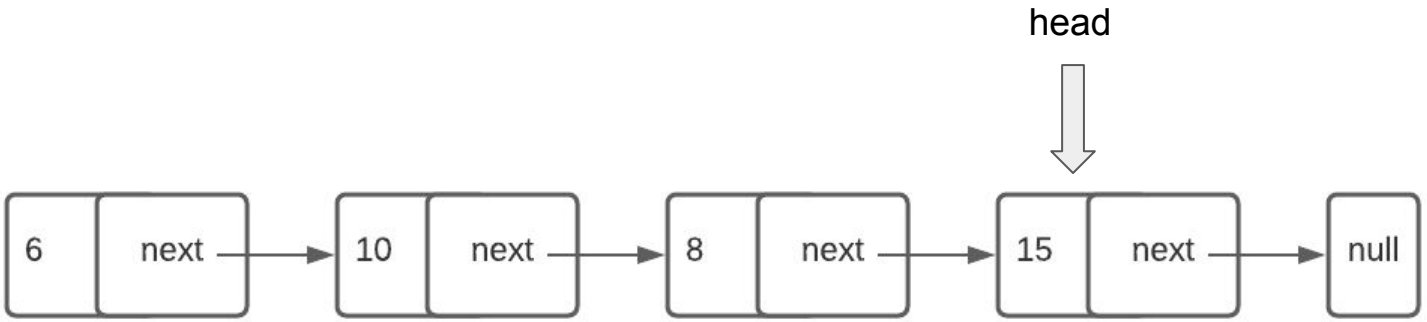
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Trace

- The pointer is pointing to node 4
  - Is None? No
  - Print the value ⇒ 15
  - What is head.next? null
  - Set head = null

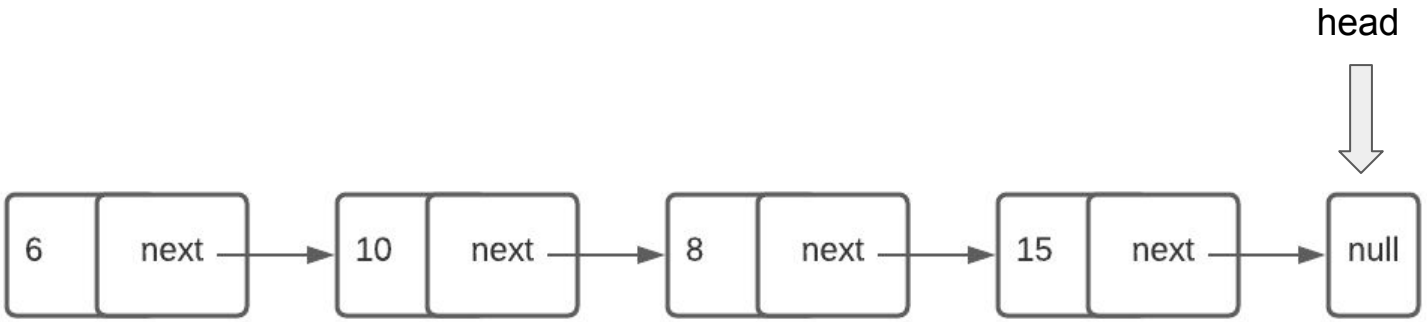
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Trace

- The pointer is pointing to **None**
  - Is None? Yes. STOP
- This code is very fundamental
  - Make sure you completely understand it
- Your turn:
  - Rewrite this code recursively

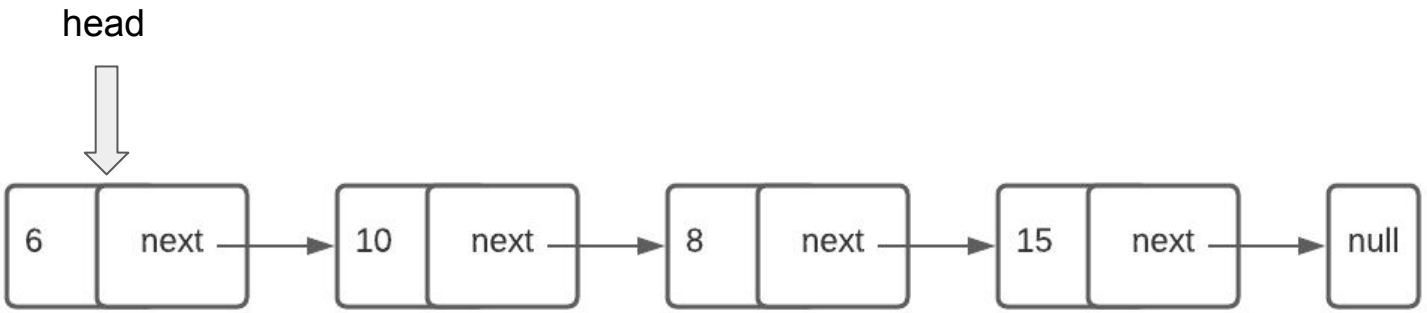
```
def print_lst(head):
    while head is not None:
        print(head.data, end='->')
        head = head.next
    print()
```



Printing Nodes Chain: Recursively

- This is exactly like printing a list recursively
  - Print
  - Call the next element (node.next)
- Similarly, try to print reversed
  - 15 8 10 6

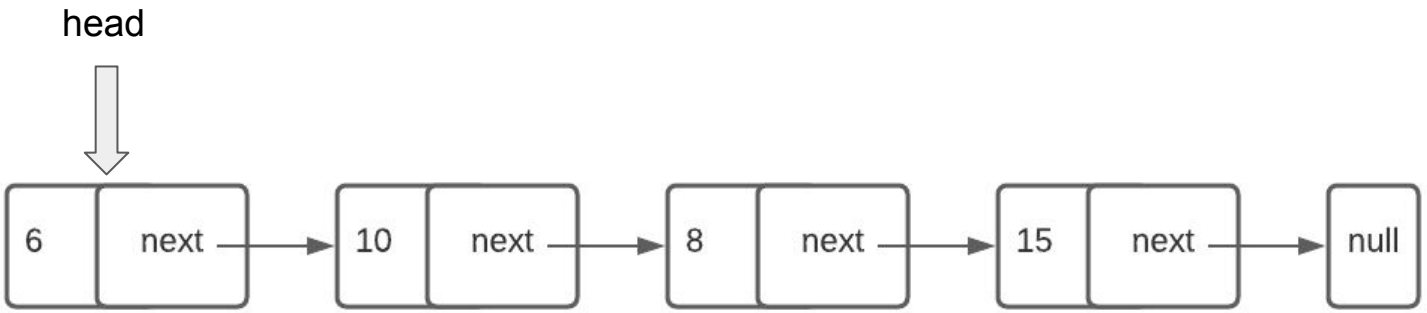
```
def print_rec(head):
    if head is not None:
        print(head.data, end='->')
        print_rec(head.next)
```



Printing Nodes Chain: Recursively

- The trick is to first call it recursively, THEN print
  - Then once we call is almost finished, it prints

```
def print_rec_reversed(head):
    if head is not None:
        print_rec_reversed(head.next)
        print(head.data, end='->')
```



Your turn

- Understand the code very well
- Play with the code
- Try to implement the following ideas
  - Function find(value) that **searches** for a node with the given value
    - If the value is found, return the node
    - Otherwise return None

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."