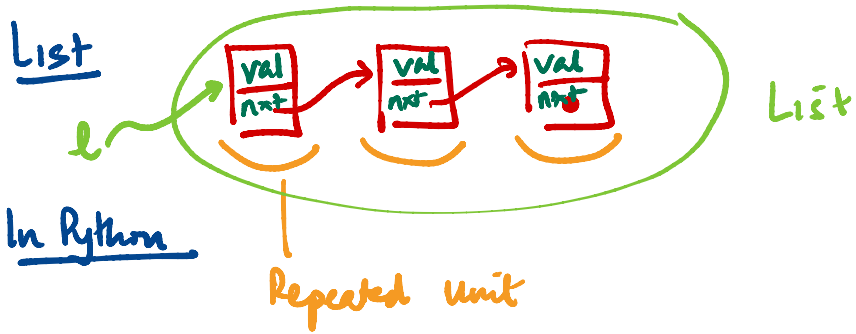
Lecture 13, 26 September 2024

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Programming and Data Structures with Python

Abstract data types, classes, objects



Use only one class to represent a Node

List points to the first Node

class List:

__init__

l = [] ✓ For simplicity

l = [3,8,7]

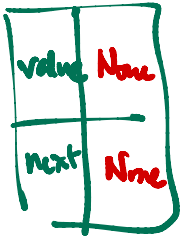
l = List() → __init__
sets up empty list

What does an empty list look like?

Use None

next is None – no next node

Empty list



None is the empty list

No type!

l=List() ⇝ l=None

Cannot say l.append(),
l.insert() ...

Empty list

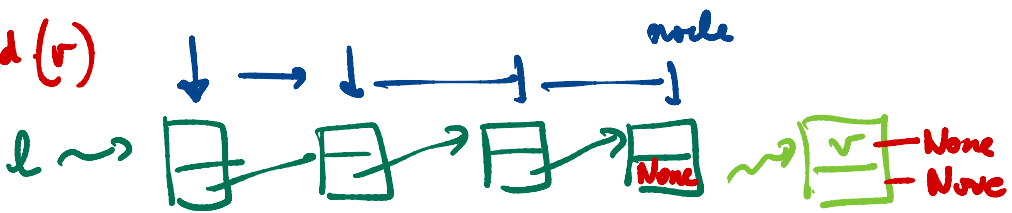
Singleton list

Never allow Value = None anywhere else

```
class List:
    def __init__(self):
        self.value = None
        self.next = None
    def isempty(self):
        return (self.value == None)
```

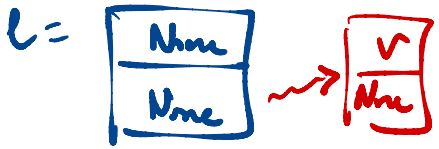
```
l.isempty()
l.append(v)
?
```

L.append(v)

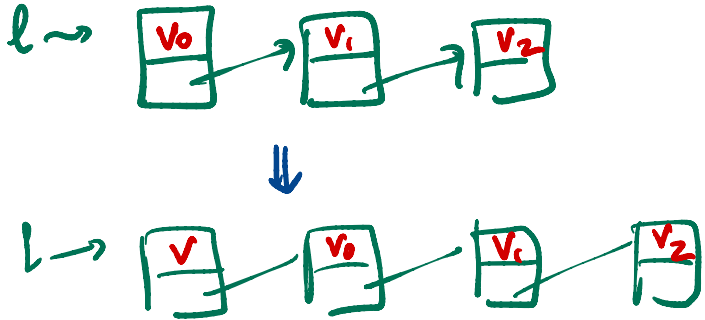


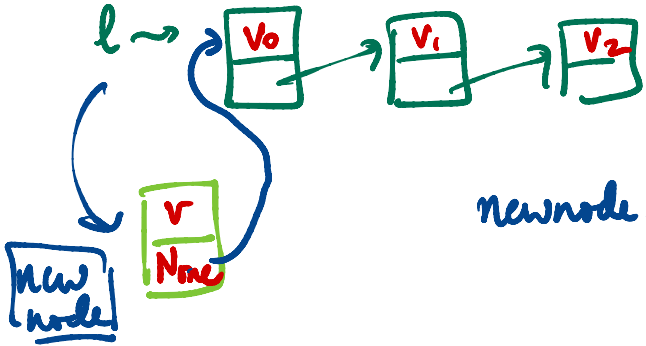

```
node=l
While node.next != None:
    node = node.next

node.next = List()
node.next.value = v
```

```
def append(self, v):
    node = self
    if self.isempty():
        self.value = v
        return
    node = self  # Not empty!
    while node.next != None
        node = node.next
```



l.insert(v) (Python's list.insert(0,v))





newnode.next = l

l = newnode

All objects are mutable

Recall

```
def myappend(l,v):
    l = l + [v]
```

vs

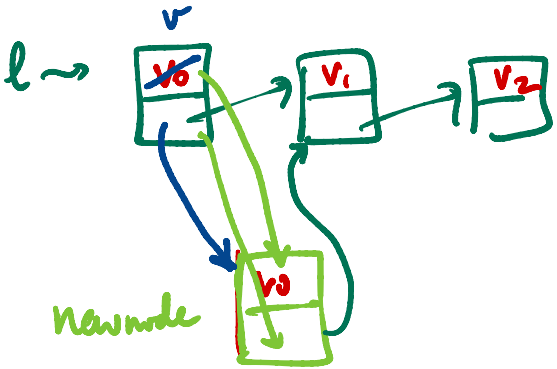
```
l.append(v)
```

`l[0]=v`

```
newnode.next = l
l = newnode
```

has same problem!

Instead



```
def insert(self,v)
    if self.isempty():
        self.value = v
        return
    newnode = List()
    newnode.value = l.value
    newnode.next = l.next
    l.value = v
    l.next = newnode
    return
```

① Make __init__ more general

__init__ takes a Python list and converts it to a List

```
def __init__(self, initlist = []):
    self.value = None
    self.next = None
    for x in initlist:
        self.append(x)
```

l = [3,6,2]

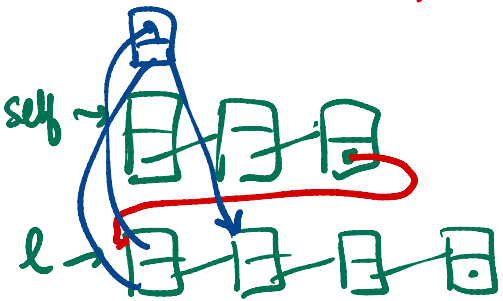
```
l = []
l.append(3)
l.append(6)
l.append(2)
```

```
def extend(self, l):
    for x in l:
        self.append(x)
    return
```

Python list

def extend(self, l)

List()




```
if l.isempty():
    return
if self.isempty():
    self.value = l.value
    self.next = l.next
    return
node = self
while node.next != None:
    node = node.next
~~node.next = l~~
if not(l.isempty()):
    node.next = l
```

def __str__(self):

Convert List() to Python list and do str(list)