

Lecture 15, 10 October 2024

- Raising an exception in `List()`
- Inserting a negative value raises `ValueError`
- Use string formatting to add negative value to error message

```
In [1]: class List:
def __init__(self, initlist = []):
    self.value = None
    self.next = None
    for x in initlist:
        self.append(x)
    return

def isempty(self):
    return(self.value == None)

def appendi(self, v): # append, iterative
    if v < 0:
        raise ValueError("Negative input:{0}".format(v))
    if self.isempty():
        self.value = v
        return

    temp = self
    while temp.next != None:
        temp = temp.next

    temp.next = List()
    temp.next.value = v
    return

def appendr(self, v): # append, recursive
    if v < 0:
        raise ValueError("Negative input:{0}".format(v))
    if self.isempty():
        self.value = v
    elif self.next == None:
        self.next = List([v])
    else:
        self.next.appendr(v)
    return

def append(self, v):
    self.appendr(v)
    return

def insert(self, v):
    if v < 0:
        raise ValueError("Negative input:{0}".format(v))
    if self.isempty():
        self.value = v
        return

    newnode = List()
    newnode.value = v

    # Exchange values in self and newnode
    (self.value, newnode.value) = (newnode.value, self.value)

    # Switch links
    (self.next, newnode.next) = (newnode, self.next)

    return

def delete(self, v): # delete, recursive
    if self.isempty():
        return

    if self.value == v:
        self.value = None
        if self.next != None:
            self.value = self.next.value
            self.next = self.next.next
        return
    else:
        if self.next != None:
            self.next.delete(v)
            if self.next.value == None:
                self.next = None
```

```

    return

def __str__(self):
    # Iteratively create a Python list from linked list
    # and convert that to a string
    selflist = []
    if self.isempty():
        return(str(selflist))

    temp = self
    selflist.append(temp.value)

    while temp.next != None:
        temp = temp.next
        selflist.append(temp.value)

    return(str(selflist))

```

```
In [2]: l = List([1,-2,3])
print(l)
```

```

-----
ValueError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 l = List([1,-2,3])
      2 print(l)

Cell In[1], line 6, in List.__init__(self, initlist)
      4 self.next = None
      5 for x in initlist:
----> 6     self.append(x)
      7 return

Cell In[1], line 39, in List.append(self, v)
     38 def append(self,v):
--> 39     self.appendr(v)
     40     return

Cell In[1], line 29, in List.appendr(self, v)
     27 def appendr(self,v): # append, recursive
     28     if v < 0:
--> 29         raise ValueError("Negative input:{}".format(v))
     30     if self.isempty():
     31         self.value = v

ValueError: Negative input:-2

```

```
In [3]: try:
        l = List([1,-2,3])
        except ValueError:
            print("oops")
```

oops

```
In [4]: try:
        l = List([1,-2,3])
        except ValueError as errmsg: # Saves error value in errmsg
            print(errormsg)
```

Negative input:-2

- String formatting -- positional arguments

```
In [5]: x = 7
        y = 22
        message = "First one is {}, second one is {}"
        message.format(x,y)
```

Out[5]: 'First one is 22, second one is 7'

- String formatting with output specification

```
In [6]: "Value {:.2f}".format(747.3)
```

Out[6]: 'Value 747.30'

```
In [7]: "Value {:.2f} {:.7f}".format(99999999947.3444,22)
```

Out[7]: 'Value 99999999947.34 99999999947.3444061'

- Modifying behaviour of `print()`

```
In [8]: x = 5  
y = 7  
print(x,y,sep=": ")
```

5:7

```
In [9]: l = list(range(20))
```

```
In [10]: for x in l:  
    print(x,end=" ")  
    print("Where are we?")
```

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, Where are we?

```
In [11]: for x in l:  
    print(x,end=" ")  
    print()  
    print("Where are we?")
```

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
Where are we?