

Linked Lists in Python

1. Linked Lists

A linked list is a sequence of data structures, which are connected together via links. Each data structure contains a component that links to the next data structure in the line. The entry point into a linked list is called the head of the list. It should be noted that the head is not a separate kind of node, but the reference to the first node. If the list is empty then the head is a null reference.

2. List Constructors: EmptyList and MakeList

The EmptyList constructor creates an empty list, while the MakeList constructor creates a list with a head node and a tail, which can be another list or the empty list.

```
class Node:

    def __init__(self, data=None):

        self.data = data

        self.next = None
```

```
class LinkedList:

    def __init__(self):

        self.head = None
```

```
def EmptyList():

    return LinkedList()
```

Linked Lists in Python

```
def MakeList(element, lst):  
  
    new_node = Node(element)  
  
    new_node.next = lst.head  
  
    lst.head = new_node  
  
    return lst
```

3. List Selectors/Accessors: First, Rest, IsEmpty

The First function returns the first element of the list, Rest returns the list excluding the first element, and IsEmpty checks if the list is empty.

```
def First(lst):  
  
    return lst.head.data  
  
  
def Rest(lst):  
  
    rest = LinkedList()  
  
    rest.head = lst.head.next  
  
    return rest  
  
  
def IsEmpty(lst):  
  
    return lst.head is None
```

4. List Mutators

Linked Lists in Python

In Python, objects are passed by reference, so changes to the list inside a function are seen outside. However, if we change the reference to the list, the old list will not be affected. List mutators generally include methods for adding, removing, or otherwise changing the elements of the list.

```
def add_end(lst, element):  
  
    new_node = Node(element)  
  
    if IsEmpty(lst):  
  
        lst.head = new_node  
  
    else:  
  
        current = lst.head  
  
        while current.next:  
  
            current = current.next  
  
        current.next = new_node
```

5. XML Representation of Lists

XML (eXtensible Markup Language) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. An XML representation of a list could look like this:

```
<list>  
  
    <node>  
  
        <data>1</data>
```

Linked Lists in Python

```
<next>

    <node>

        <data>2</data>

        <next>

            <node>

                <data>3</data>

                <next/>

            </node>

        </next>

    </node>

</next>

</node>

</list>
```

6. Recursion

Recursion can be used in various list operations, such as printing the list, finding the length of the list, etc.

```
def print_recursive(node):

    if node is None:

        return

    print(node.data)

    print_recursive(node.next)
```