



The linked list is a simple linear structure which can be used to implement many ADTs, including lists, stacks, queues etc. In Python some libraries (e.g. `collections`) have their own implementations for the linked list but it is pretty straightforward to create your own.

First we create a class `Node` which stores the given data and holds a link to the next node.

Example 1.1: Creating `Node` class

```
# example1.py

class Node:
    def __init__(self, data, next):
        self.data = data
        self.next = next

if __name__ == "__main__":
    # Nodes can be declared separately, but they can be chained like this
    head = Node(1, Node(20, Node(3, Node(123, None))))
    while (head != None):
        print(head.data)
        head = head.next
```

Example 1.2: Output

```
$ python3 example1.py
1
20
3
123
```

We are done now if we want to keep the implementation it as simple as possible. However it is rather useful to create another class that does the upkeep of the structure for us. Let's create another class `LinkedList` for that purpose.

```
def __init__(self):
    self.tail = Node(None, None)
    self.head = Node(None, self.tail)
    self.len = 0
```

Nodes `head` and `tail` won't have any other purpose than just pointing where the list starts and ends. Some prefer to not implement separate head and tail nodes but with them we prevent some special cases in the code later on. Integer `len` tells the length of the linked list.

Now we can implement all the methods we want to the `LinkedList` class. Two common methods are adding keys to the end of the linked list (`append`) and printing the linked list.

Example 2.1: Summary of the linked list

```
# example2.py

class Node:
    def __init__(self, data, next):
        self.data = data
        self.next = next

class LinkedList:
    def __init__(self):
        self.tail = Node(None, None)
        self.head = Node(None, self.tail)
        self.len = 0

    # prints the linked list
    def print(self):
        # ...

    # add data to the end of the linked list
    def append(self, data):
        # ...

if __name__ == "__main__":
    L = LinkedList()
    L.append(1)
    L.append(50)
    L.append(23)
    L.print()
    L.append(100)
    L.append("foo")
    L.append("bar")
    L.print()
```

Example 2.2: Expected output

```
$ python3 example.py
1 -> 50 -> 23
1 -> 50 -> 23 -> 100 -> foo -> bar
```

You are logged in as Hung Nguyen (Log out)

Search and Moodle Help

Course search

Student Guide (PDF)

Moodle teacher's guide

Moodle in Intra

Accessibility statement

Data retention summary

Get the mobile app

Policies

Copyright © LUT University