

ساختمان داده:

لیست پیوندی



@ShayanAryania

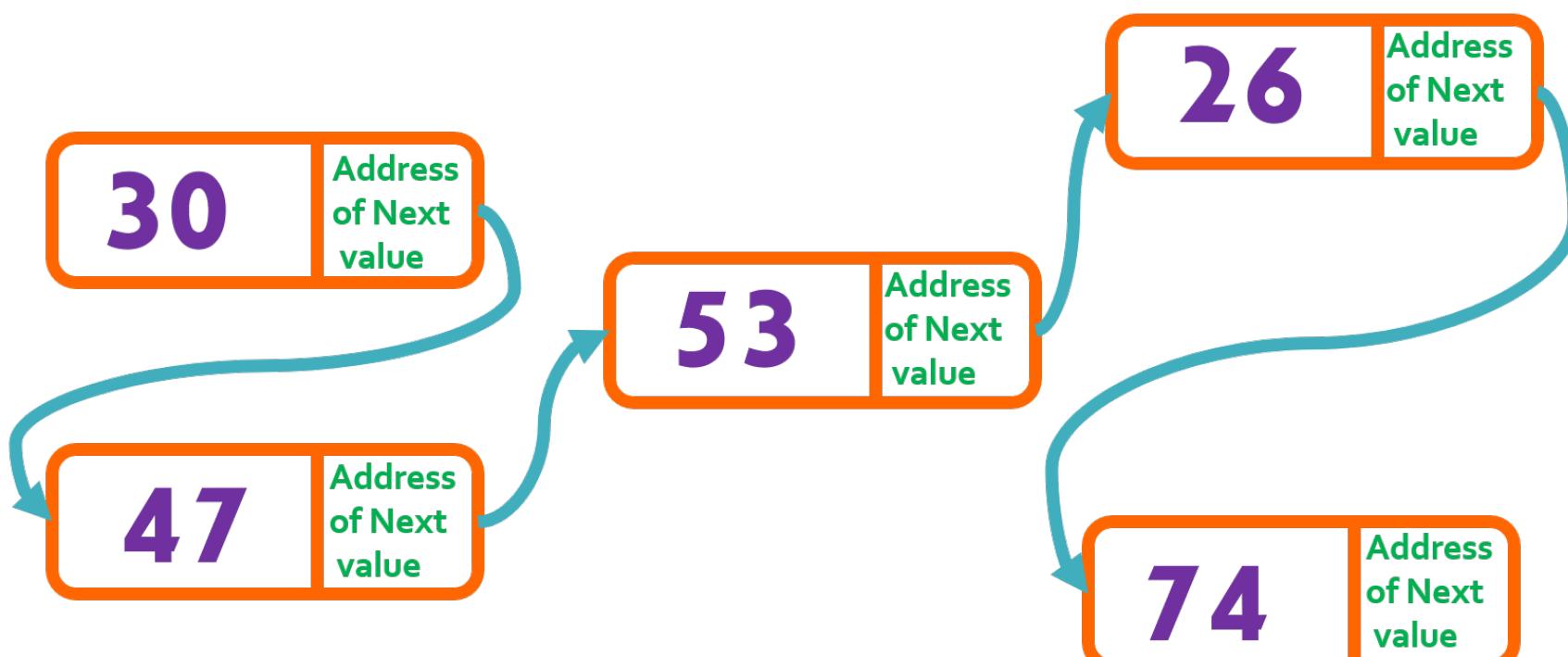
toooooooooo>>



لیست پیوندی:

ساختاری شامل دنباله‌ای از عناصر است که هر عنصر دارای اشاره‌گری به عنصر بعدی در دنباله است.

لیست پیوندی از جهاتی شبیه به آرایه است ولی مزیت مهم فهرست پیوندی نسبت به آرایه‌ها این است که ترتیب قرار گرفتن داده‌ها در آن با ترتیب قرار گرفتن آن‌ها در حافظه متفاوت است.





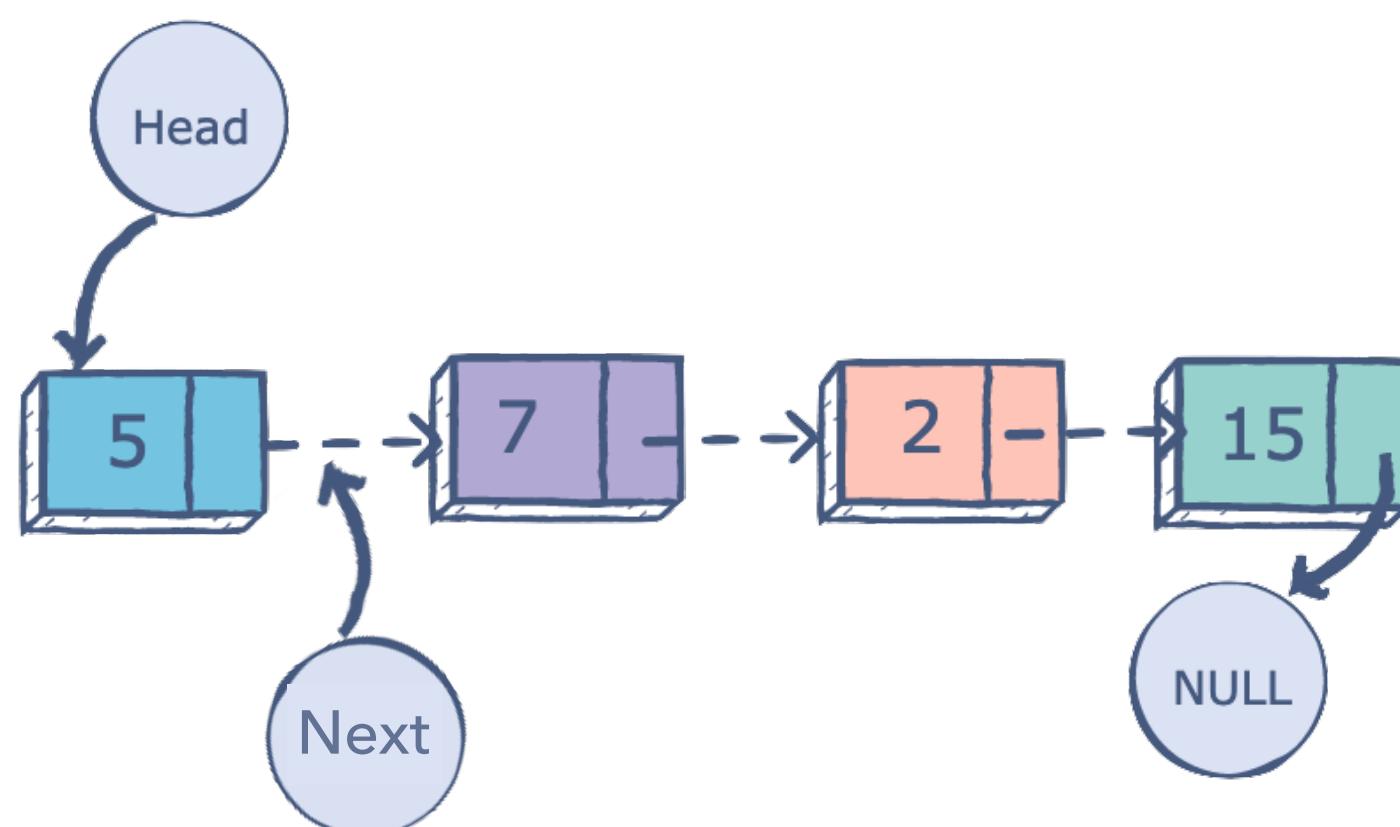
همان طور که در شکل فوق مشخص است، نقاط زیر در یک لیست پیوندی مهم هستند:

لیست پیوندی شامل یک عنصر پیوندی است که **شروع (Head)** نامیده می شود.

هر لینک فیلد های داده ای و همچنین یک فیلد پیوند دارد که **بعدی (Next)** نامیده می شود.

هر لینک با استفاده از پیوند Next به پیوند بعدی خود مربوط می شود.

هر لینک که یک پیوند NULL داشته باشد به معنی انتهای لیست است.





نکات :

فهرست پیوندی دارای این ویژگی است که درج و حذف گره‌ها در هر نقطه‌ای از فهرست، با تعداد ثابتی از عملیات امکان‌پذیر است.

در این نوع ساختمان داده برای دسترسی به هر عنصر زمان بیشتری (نسبت به آرایه) صرف می‌شود.

حداکثر تعداد لیست پیوندی که می‌توان ایجاد کرد، بستگی به مقدار حافظه سیستم دارد.





کلیت پیاده سازی لیست پیوندی در C++

در این کد، لیست را با مجموعه ایی از **Node**‌ها پایه ریزی کرده ایم، که هر **Node** دارای ۲ بخش است.

بخشی دارای اطلاعات و بخشی دیگر دارای اشاره گری به **Node** بعدی است.

```
template <typename T> // Because we want to store any type of data
    (int, string, ..)
class ListNode {
public :
    T & data; //data piece
    ListNode *next; // pointer to next
    ListNode (T & data) : data(data), next(nullptr) {}
};
```





C++

List.h

The screenshot shows the Xcode interface with the project 'Linked_Memory' open. The 'List.h' file is selected in the sidebar. The code implements a linked list template:

```
1 // List.h
2 //
3 // Created by Shayan Aryania
4
5 #pragma once
6
7
8 template <typename T> // Because we want to store any type of data (int, string, ..)
9
10 class List {
11     public :
12         const T & operator[](unsigned index);
13         void InsertAtFront(const T & data);
14
15     private :
16         class ListNode {
17             public :
18                 const T & data; //data piece
19                 ListNode *next; // pointer to next
20                 ListNode(const T & data) : data(data), next(nullptr) {}
21         };
22
23         ListNode *head_; //Head pointer for our list
24     };
25
26 #include "list.hpp"
27
```

The code includes a private ListNode class with a constructor taking a const T & parameter and initializing both data and next pointers. The main List class has a public operator[] method for indexed access and an InsertAtFront method for inserting elements at the front of the list.



oooooo|oooo>>



C++

List.hpp

The screenshot shows the Xcode IDE interface with the following details:

- Title Bar:** Shows the project name "Linked_Memory" and the file "main". It also indicates "Finished running Linked_Memory : Linked_Memory" and has a warning count of "2".
- File Navigator:** Shows the file structure: Linked_Memory > Linked_Memory > h List > operator[](index).
- Code Editor:** Displays the content of List.hpp. The code is a template class for a linked list. It includes methods for indexing and inserting at the front.

```
1 // List.hpp
2 //
3 // Created by Shayan Aryania
4
5 template <typename T>
6
7 const T & List<T> :: operator[](unsigned index){
8     //start a 'thru' pointer to advanced thru the list
9     ListNode *thru = head_;
10    // loop untill the end of the list
11    while (index>0 && thru->next != nullptr) {
12        thru = thru->next;
13        index--;
14    }
15
16    return thru->data; //return the data
17 }
18
19 template <typename T>
20
21 void List<T> :: InsertAtFront(const T & data) {
22     // create a new list node on the heap :
23     ListNode *node = new ListNode(data);
24
25     //set the new nodes next pointer point the current
26     //head of the list:
27     node->next = head_;
28
29     //set the list's head pointer to the new code:
30     head_ = node;
31 }
```

- Status Bar:** Shows "Line: 13 Col: 17" and other standard status bar icons.



000000+000>>



C++

Main.cpp

The screenshot shows the Xcode IDE interface with the following details:

- Project Name:** Linked_Memory
- File:** main.cpp
- Build Status:** Finished running Linked_Memory : Linked_Memory (2 warnings)
- Code Content:** The main.cpp file contains C++ code for a linked list. It includes `<iostream>`, `"list.h"`, and the `std::` namespace. The code creates a `List<int>` object and inserts three elements (3, 2, 1) at the front. It then prints the elements at indices 0, 1, and 2.
- Output Window:** The bottom window shows the terminal output:

```
list[0]: 1
list[1]: 2
list[2]: 3
Program ended with exit code: 0
```
- Bottom Bar:** Shows "Line: 16 Col: 27" and other standard Xcode navigation icons.



oooooooooooo>>



Python

The screenshot shows a Python development environment in a dark-themed code editor. On the left, the file `LinkedList.py` contains the following code:

```
1 # Node class
2 class Node:
3
4     # Function to initialise the node object
5     def __init__(self, data):
6         self.data = data # Assign data
7         self.next = None # Initialize next as null
8
9
10 # Linked List class contains a Node object
11 class LinkedList:
12
13     # Function to initialize head
14     def __init__(self):
15         self.head = None
16
17     # This function prints contents of linked list
18     # starting from head
19     def printList(self):
20         temp = self.head
21         while (temp):
22             print (temp.data)
23             temp = temp.next
24
25
```

On the right, the terminal window shows the execution of the code:

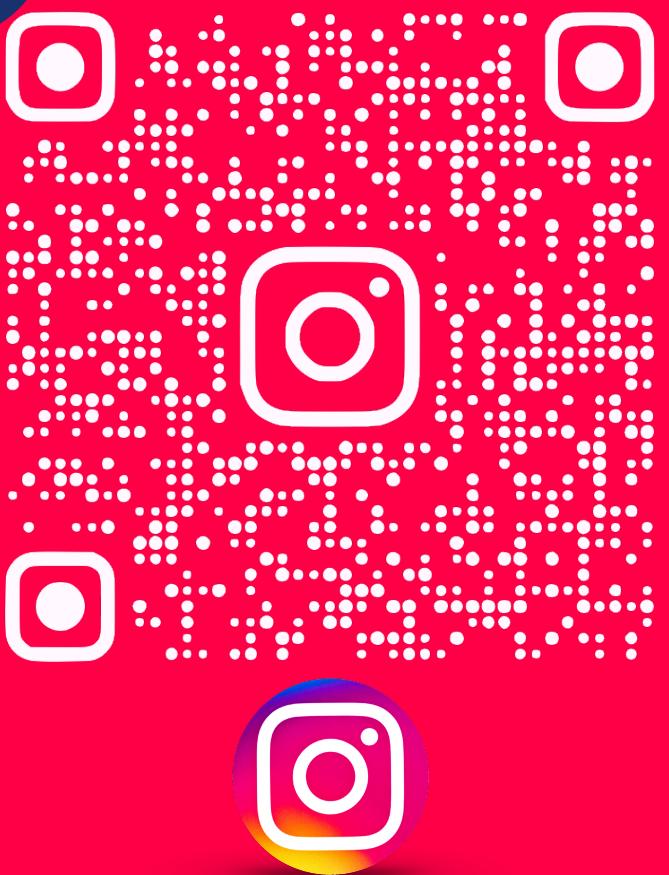
```
shayan@Shayans-MBP Desktop % /usr/bin/env /usr/local/bin/python3.10 /Users/shayan/.vscode/extensions/ms-python.python-2022.0.1814523869/pythonFile
s/lib/python/debugpy/launcher 51090 -- /Users/shayan/Desktop/LinkedList.py
1
2
3
shayan@Shayans-MBP Desktop %
```



00000000+o>>



Follow



www.instagram.com/shayanaryania



www.linkedin.com/in/shayanaryania



Like & Share

Save