به نام خدایی که از رگ گردن نزدیک تر است

پس از آشنایی اولیه با شیءگرایی یادگرفتید که دستورهای struct و class به ما امکان ساخت تایپ داده دلخواهمان را می دهند و ما می توانیم هرچیزی (تقریبا هرچیزی) که دلمان می خواهد را درونشان بگذاریم و آن را بسته بندی کنیم.

و یکی از چیزهایی که می توانیم درون کلاس یا استراکت بگذاریم اشاره گری از نوع خودش است. بدین صورت هر شیء از کلاس ما می تواند به شیء دیگری اشاره کند.

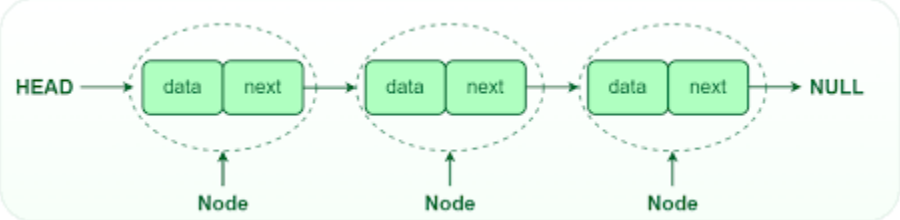
```
struct myStruct{

    myStruct* ptr;
}
```

```
class myClass{

    myClass* ptr;
}
```

این به ما این امکان را می دهد که زنجیره ای از این اشیاء درست کنیم.



این زنجیره به Linked list معروف است و به طور سنتی به هر یک از اشیاء آن node یا گره گفته می شود. همچنین بنابر سنت اشاره گر به اولین گره head و درصورت وجود اشاره گر به آخرین گره end/tail نامیده می شوند.

شکل بالا نمایانگر ساده ترین نوع لیست پیوندی به نام singly linked list است که تنها امکان پیمایش (دیدن گره ها) در یک جهت را می دهد.

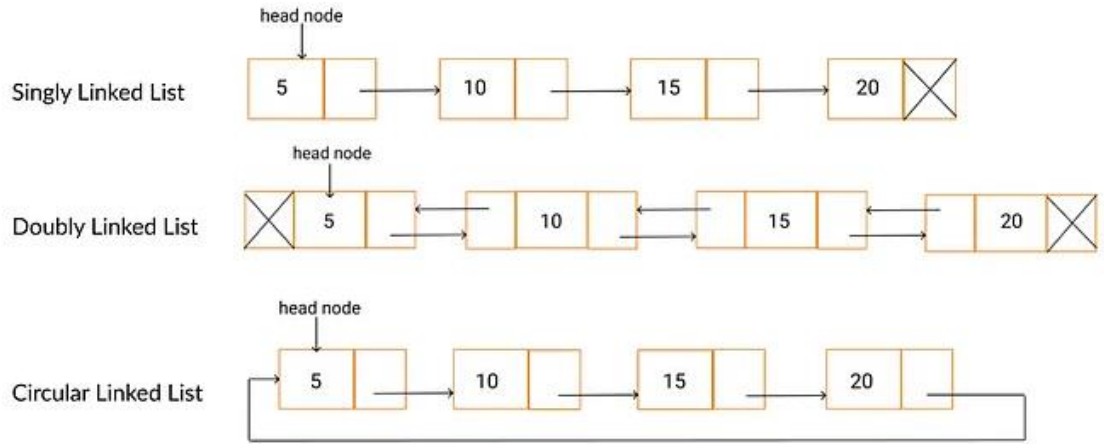
توجه داشته باشید که اشاره گر آخرین گره باید حتما NULL گردد در غیر این صورت رفتار غیرقابل پیش بینی از خود نشان می دهد و پیمایش ما را دچار مشکل می کند.

```
void traverseLinkedList(Node* head)
{
    Node* cur = head; //cur is abbreviation of current
    while(cur->next != NULL){
        std::std::cout << cur->value << ' ';
        cur = cur->next;//this goes to the next node
    }
}
```

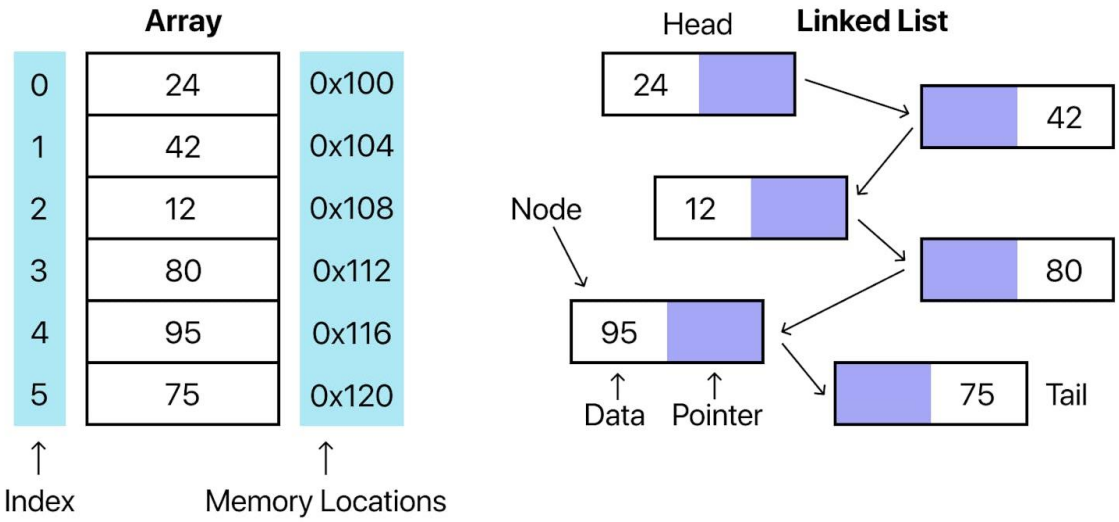
با اضافه کردن پوینتری دیگر و یا اتصال پوینتر آخرین گره به اولین گره می توان لیست های پیوندی دوطرفه (doubly linked list) و حلقوی/دایروی را به جود آورد.

```
class Node{

    Node* next;
    Node* previous;
}
```

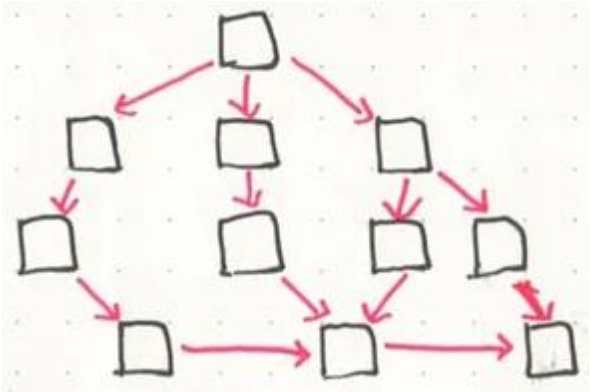


تفاوت لیست های پیوندی و آرایه ها:



توجه کنید که برخلاف آرایه، در لیست های پیوندی امکان دسترسی تصادفی به داده ها برحسب اندیسشان وجود ندارد زیرا هر یک از گره ها در هرجایی از حافظه می توانند قرار داشته باشند.

گراف: حال اگر ما به جای تنها یک اشاره گر به گره بعدی دو و یا تعداد بیشتری اشاره گر قرار دهیم این امکان به لیست ما داده می شود که در هر مرحله به گره های بیشتری اشاره کند و بدین صورت می توانیم گرافها را بسازیم.



درخت: حالت خاصی از گراف که تمامی گره ها (راس ها) به حداقل یکی گره دیگر متصل بوده و هیچ دوری در آن وجود نداشته باشد ساختمان داده ای موسوم به tree یا درخت است.

```
class Node{

    Node* right;
    Node* left;
    Node* parent;
}
```