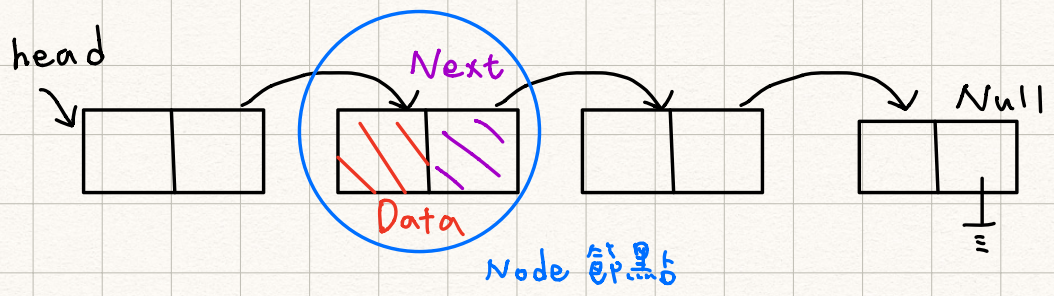
鏈結串列 Linked List



必須定義兩個 class { 1. Node
2. Linked List

物件

```
Node head;

head.next = ..;
```

用 . 來訪問成員變數

類別指標

```
Node *node;

node->next = ..;
node->data = ..;
```

用 → 來訪問成員變數

注意!!

1. head 只有一個
2. `Node* x = new Node(123);`
   此種指標有兩種. (1) 是 x → next
   (2) 是 x 本身

x → next 裡面存一個 List, 即 □□ 的記憶體位置

```
class Node
{
  public:
     int Data ;          // 一筆資料
     Node * next ;       // 一個指向 Node 的指標
   Node(int data)
   {
      Next = NULL ;
      Data = data ;
   }
} ;
```

新增/刪除 節點

最前端 / 末端

NULL

```
class LinkedList
{
    Node * head = new Node (1);
    push_front (int d)
    {
        Node * newNode = new Node (d);
        cout << "Push Front" << d << endl;
        if ( head -> next == NULL)
        {
            head -> next = newNode ;
            return ;
        }
```

```
    newNode -> next = head -> next;
    head -> next = newNode;
}

push_back (int d)
{
    cout << "Push Last : " << d << endl;
    Node * newNode = new Node(d);
    Node * lastNode = new Node(d);

    if (head -> next == NULL)
    {
        head -> next = newNode;
        return;
    }

    lastNode = head -> next;

    while (lastNode -> next != NULL)
        lastNode = lastNode -> next;

    lastNode -> next = newNode;
}
```

```
pop_front()
{
    Node* tempNode = new Node(1);

    if(head->next == NULL)
    {
        cout << "empty" << endl;
        return;
    }

    tempNode = head->next;
    cout << tempNode->Data << endl;

    head->next = tempNode->next;

    delete tempNode;
}
```