Nodes and Linked Lists

Node

- A node is a basic data structure.
- A node stores:
  - Data
  - One or more pointer to other elements (helps to link nodes together)

Class Node

```
// Class node
public class Node {
    private String data; // Data to store
    private Node next; // Pointer to next node in list
}
```

Pointers Exercise 1

Use the Node.java file provided. Make a diagram to represent the following code using nodes and pointers (analize one line at the time).

```
Node node1 = new Node("a");

Node node2 = new Node("b");

node1.setNext(node2);

node2.setNext(new Node("c"));

node2 = new Node("d");

Node node3 = new Node("e", node2);
```

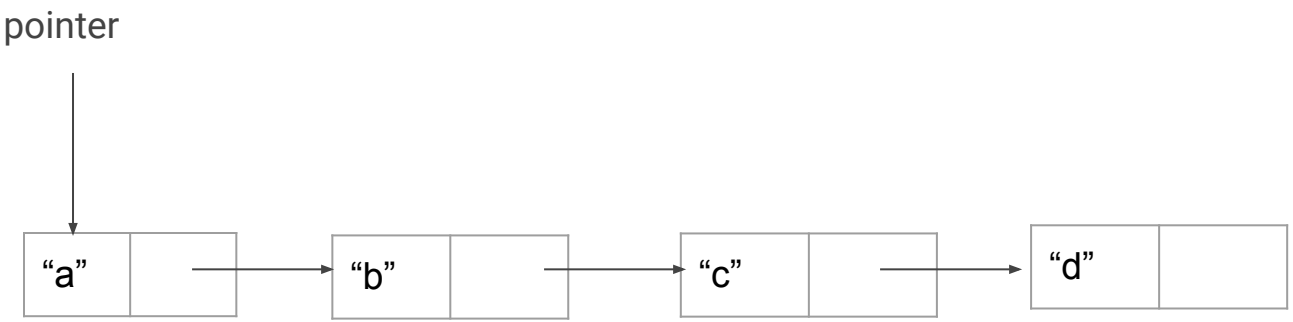
Pointers Exercise 2

Use the previous diagram and do the following:

```
node2.setNext(node1);

node1 = node3;
```

Pointers Exercise 3



Write a few lines of code to perform the following steps:

1. Create a new Node variable set it to point to the node with the “b” in it.
2. Create a new Node variable and instantiate it to a new Node with a value of “e”.
3. Write the code to insert this new Node between the “b” and the “c”

Linked List

How would you access the linked list chain?

How would you traverse the elements in a linked list?

Class Linked List

How would you access the linked list chain?

We need a pointer to track the first element of the list.

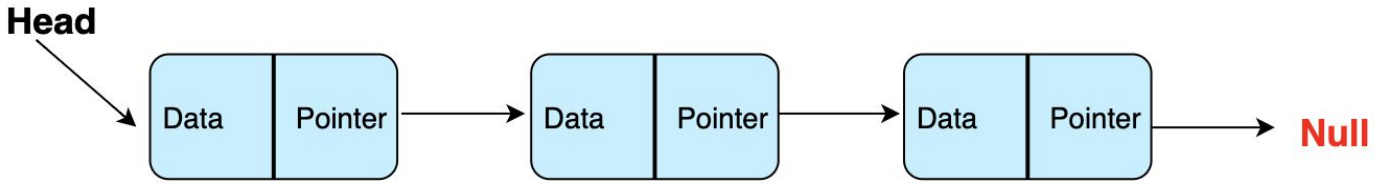
public Node head; // head of the linked list

How would you traverse the elements in a linked list?

Having the first element, we can go over the next elements in the list.

Linked List

- It is a linear data structure made of a chain of nodes.
- Each node contains a value and a pointer to the next node in the chain.
- It has a Head pointer which points to the first node
- The last element point to null



Linked List Characteristics

- The size increases dynamically
- No need to know the size of the element when we create a linked list
- Easy to insert/delete (change pointers)
- Linked list uses extra memory to store links

Types of Link List

Singly: It is a list where each node has data and a reference pointer to its next node.

Doubly: Each node in this list has 3 attributes which are data, next node reference and previous node reference.

Applications of Linked List

- In music players: Your playlist may be created using a linked list.
- Photo gallery applications were you can access the previous/next picture.
- URLs that have previous/next buttons to navigate between pages

Linked List Operations

- Insertion : adds a new element to the linked list
- Deletion : delete existing element form the linked list
- Searching : search for an element by its value in the linked list
- Traversal : traverse all elements starting from head in the linked list

Insert

- Inserting new node at the beginning
- Inserting new node at the end
- Inserting new node at random position of the linked list.

Delete

- Deleting node at the beginning
- Deleting node at the end
- Deleting node at random position of the linked list.