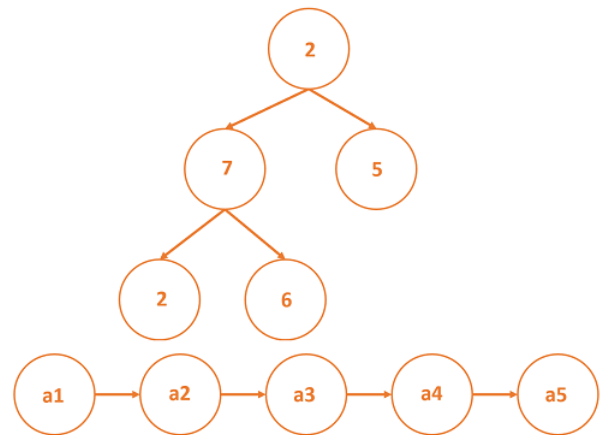


# Nodes

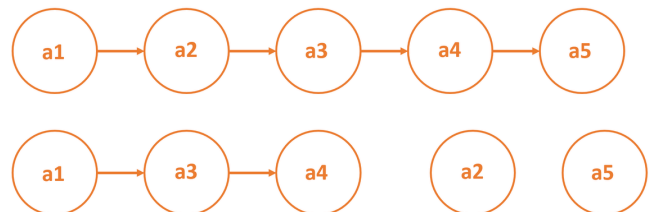
## Node: An individual part of a larger data structure

Nodes are a basic data structure which contain data and one or more links to other nodes. Nodes can be used to represent a tree structure or a linked list. In such structures where nodes are used, it is possible to traverse from one node to another node.



## Orphaned nodes

Nodes that have no links pointing to them except for the head node, are considered “orphaned.” In the illustration, if the nodes **a2** and **a5** are removed, they will be orphaned.



## Null node link

Data structures containing nodes have typically two bits of information stored in a node: data and link to next node.

The first part is a value and the second part is an address of sorts pointing to the next node. In this way, a system of nodes is created. A **NULL** value in the link part of a node's info denotes that the path or data structure contains no further nodes.

## Python Node implementation

A Node is a data structure that stores a value that can be of any data type and has a pointer to another node. The implementation of a Node class in a programming language such as Python, should have methods to get the value that is stored in the Node, to get the next node, and to set a link to the next node.

```
class Node:
    def __init__(self, value,
next_node=None):
        self.value = value
        self.next_node = next_node

    def set_next_node(self, next_node):
        self.next_node = next_node

    def get_next_node(self):
        return self.next_node

    def get_value(self):
        return self.value
```

 **Print**    **Share** ▼