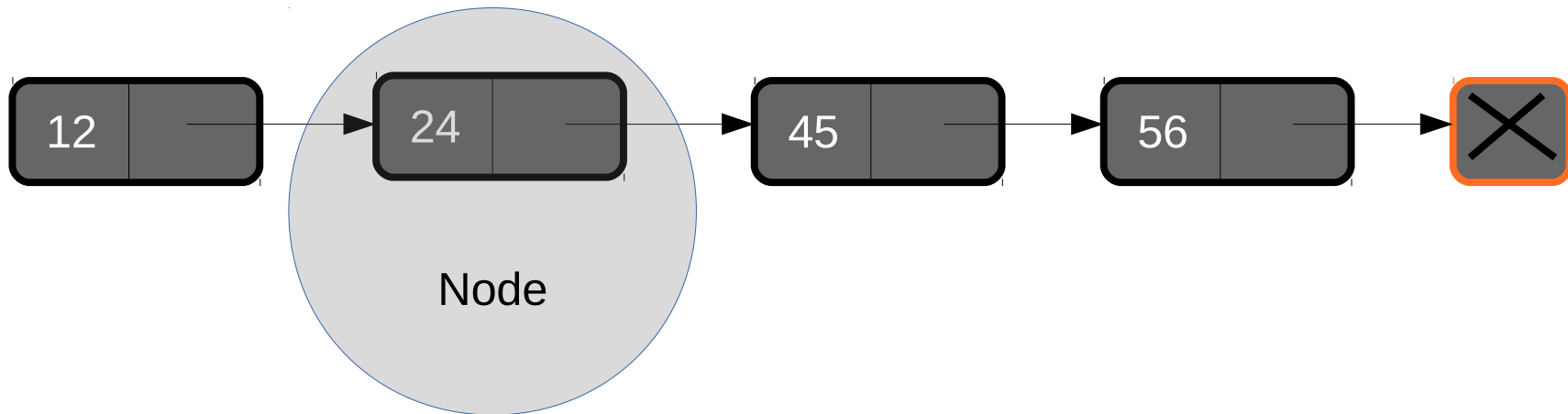


Chapter 2: Linked List

A decorative graphic consisting of a horizontal bar with a gradient from magenta to purple, ending in a double-headed arrow shape pointing to the right.

Abstract

- ✓ A collection of items accessible one after another beginning at the head and ending at the tail is called a list.
- ✓ A **linked list** is a data structure consisting of a group of nodes which together represent a sequence.
- ✓ Under the simplest form, each node is composed of a data and a reference (in other words, a link) to the next node in the sequence.





Why : Linked List

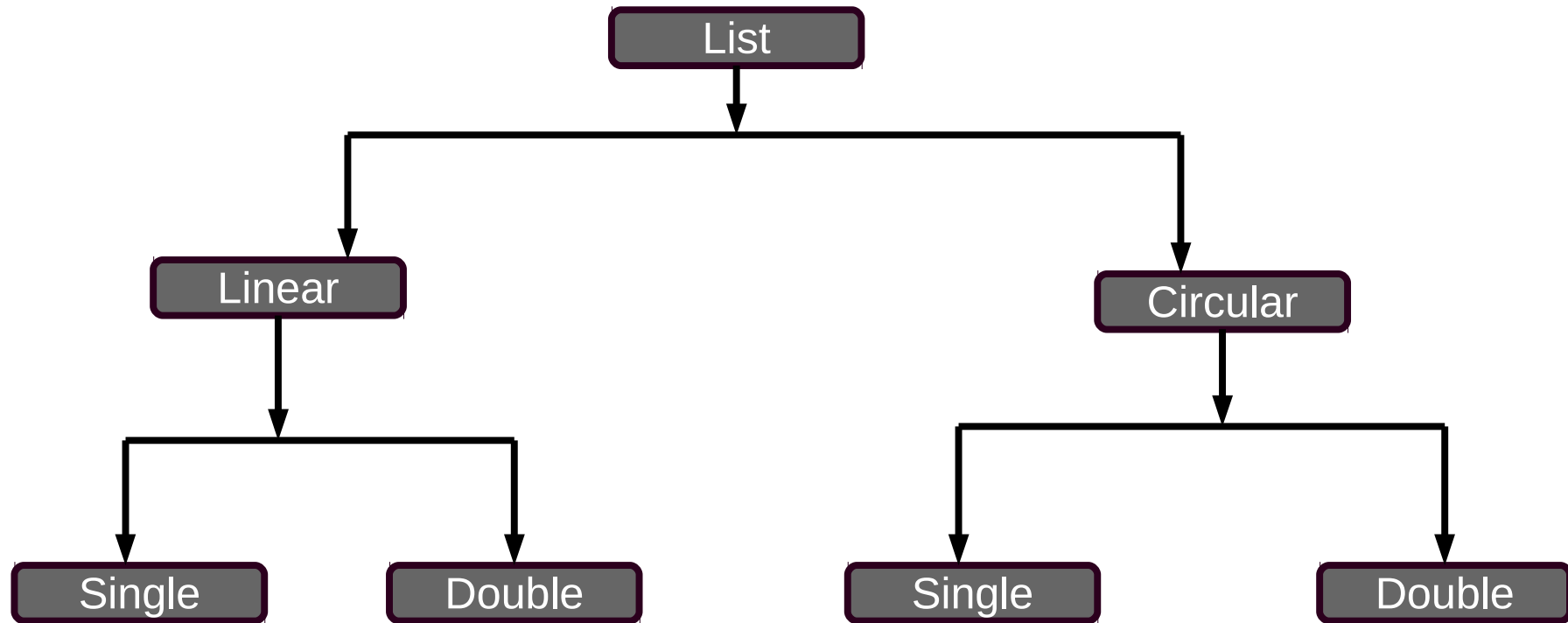


Linked List Vs Arrays

- ✓ Elements can be inserted into linked lists indefinitely, while an array will eventually either fill up or need to be resized.
- ✓ Further memory savings can be achieved.
- ✓ Simple example of a persistent data structure.
- ✓ On the other hand, arrays allow random access, while linked lists allow only sequential access to elements.
- ✓ Another disadvantage of linked lists is the extra storage needed for references, which often makes them impractical for lists of small data items such as characters or boolean values.



Linked List : Types



Singly Linked List

- ✓ The simplest kind of linked list is a singly-linked list (or slist for short), which has one link per node.
- ✓ This link points to the next node in the list, or to a null value or empty list if it is the final node.

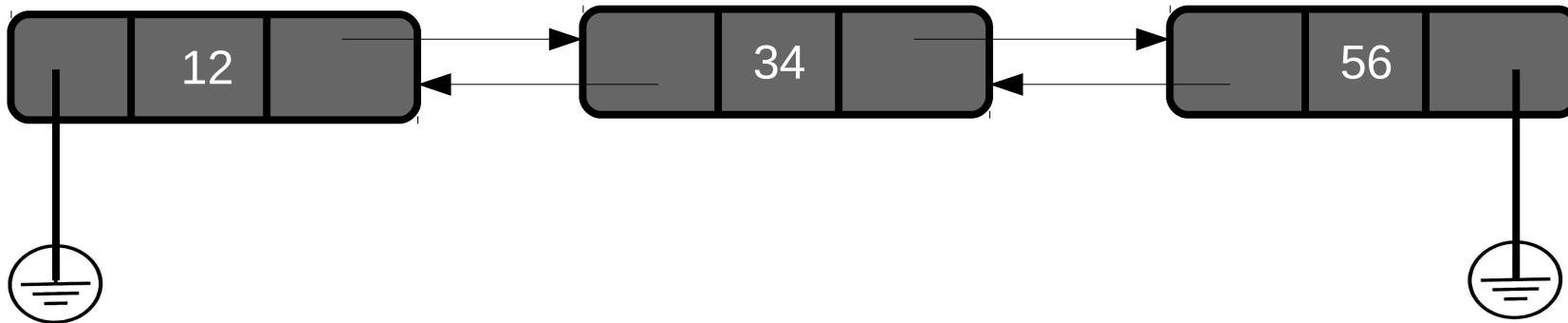
Example



Doubly Linked List

- ✓ A variant of a linked list in which each item has a link to the previous item as well as the next.
- ✓ This allows easily accessing list items backward as well as forward and deleting any item in constant time, also known as two-way linked list, symmetrically linked list.

Example

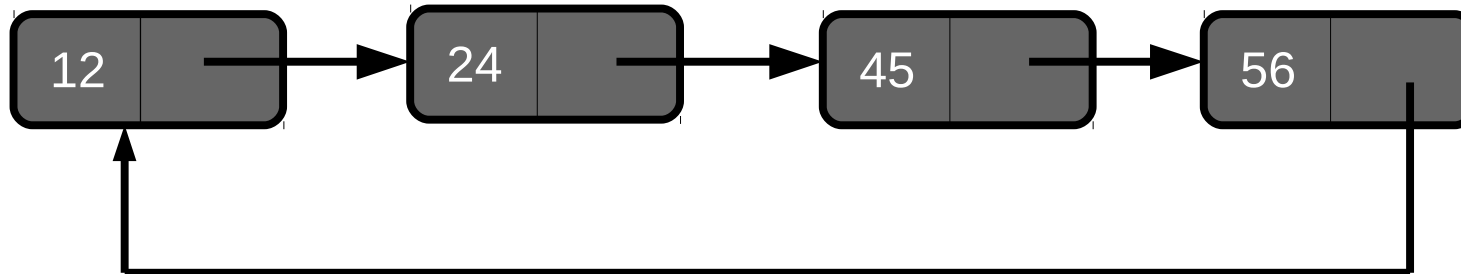


Singly Circular Linked List



- ✓ Similar to an ordinary singly-linked list, except that the next link of the last node points back to the first node.

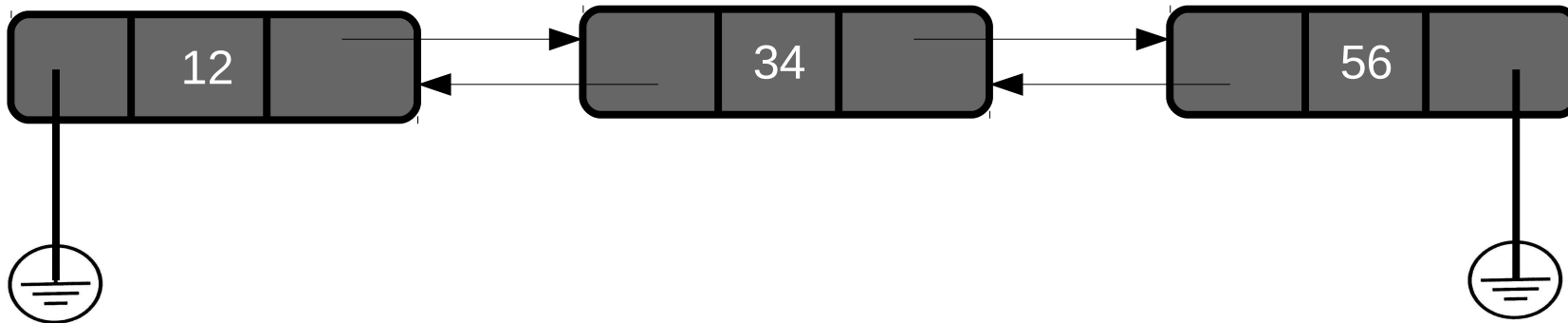
Example



Doubly Linked List

- ✓ A variant of a linked list in which each item has a link to the previous item as well as the next.
- ✓ This allows easily accessing list items backward as well as forward and deleting any item in constant time, also known as two-way linked list, symmetrically linked list.

Example

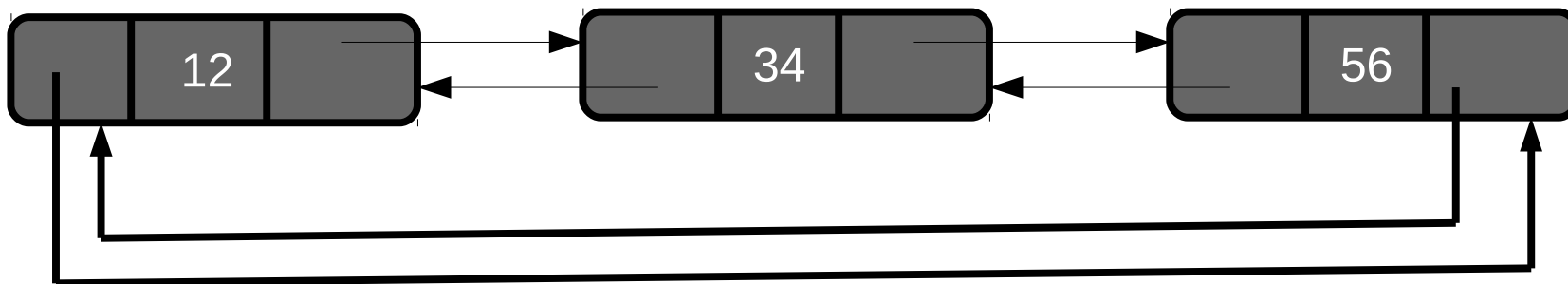


Doubly Circular Linked List



- ✓ Similar to a doubly-linked list, except that the previous link of the first node points to the last node and the next link of the last node points to the first node.

Example





TradeOffs



Double Linked List Vs Single Linked List

<i>Single Linked List</i>	<i>Double Linked List</i>
Less space per node	More space per node
Elementary operations Less expensive	Elementary operations more expensive
Bit difficult to manipulate	Easier to manipulate

Circular Linked List Vs Linear Linked List

- ✓ Allows quick access to the first and last records through a single pointer (the address of the last element).
- ✓ Their main disadvantage is the complexity of iteration, which has subtle special cases.





Operations : Linked List



1. Creation
2. Insertion
3. Deletion
4. Searching
5. Sorting
6. Listing
7. Destroying etc...





Applications



Used in the implementation of other Data Structures

1. Stacks
2. Queues

