AN INTRODUCTION TO

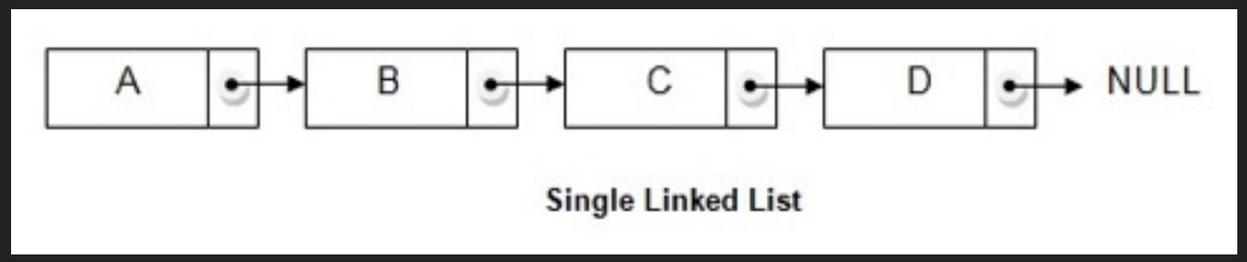
LINKED LISTS

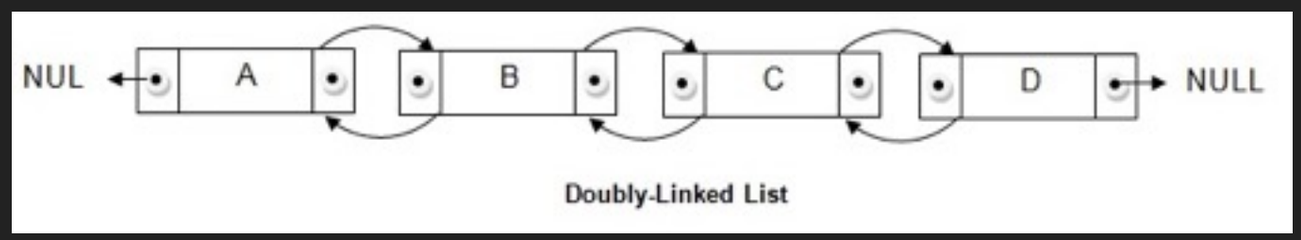
DEFINITION

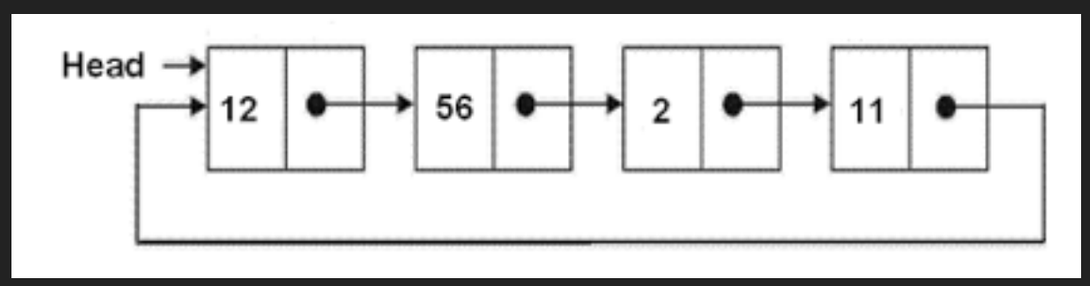
"In computer science, a linked list is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of DATA and a REFERENCE (in other words, a link) to the next node in the sequence; more complex variants add additional links. This structure allows for efficient insertion or removal of elements from any position in the sequence."

Wikipedia

TYPES OF LINKED LISTS

- Singly Linked List
  - 

Single Linked List
- Doubly Linked List
  - 

Doubly-Linked List
- Circular Linked List
  - 

ADVANTAGES

- Dynamic data structure - can grow or shrink memory usage at runtime*
- Non-contiguous memory allocation
- Efficient memory utilization
- Insertions and deletions in Constant Time

DISADVANTAGES

- Space Requirements (pointers take up space)
- No Random Access - Only Sequential
- Time Complexity
- Reverse Traversing Is Difficult
- Heap Space Restriction

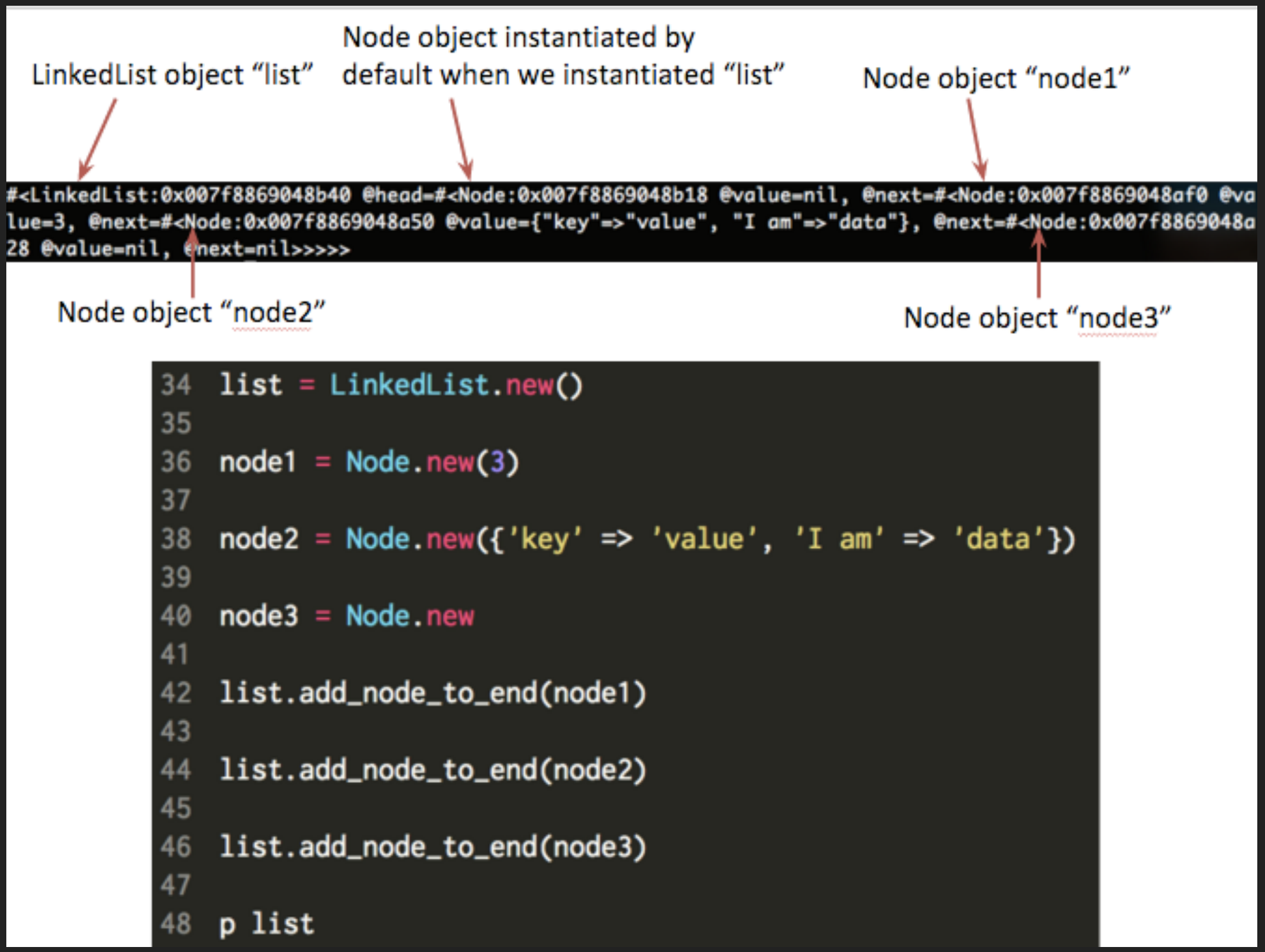
RUBY IMPLEMENTATION PART 1

```ruby
class Node

  attr_accessor :next, :value

  def initialize(value = nil, next_node = nil)
    @value = value
    @next = next_node
  end

end
```

RUBY IMPLEMENTATION PART 2

```ruby
class LinkedList

  attr_accessor :head

  def initialize(head = Node.new)
    @head = head
  end

  def add_node_to_end(node = Node.new)
    next_node = @head.next
    if next_node == nil
      @head.next = node
    else
      until next_node.next == nil
        next_node = next_node.next
      end
      next_node.next = node
    end
  end

end
```

RUBY IMPLEMENTATION PART 3



WHEN TO USE LINKED LISTS

- If your data is easily represented in one dimension, and the number of elements is unknown or is expected to change often throughout the operation of your program, a linked list is more efficient.
- If you expect to be regularly adding or subtracting elements, especially if you need to maintain a sorted order, the versatility of the linked list will be of greater benefit.

WHEN TO USE STATIC ARRAYS

- If your data is best represented using a multidimensional structure, or the number of elements is known in advance and will remain consistent, an array is best.
- If your data will be searched and accessed often but will change infrequently, the array offers the least overhead for your expected operations.

HELPFUL RESOURCES

- Wikipedia
  - https://en.wikipedia.org/wiki/Linked_list
- Stanford CS Education Library
  - http://cslibrary.stanford.edu/103/
  - http://cslibrary.stanford.edu/105/
- Implementing a Linked List in Ruby
  - http://matt.weppler.me/2013/08/14/implementing-a-linked-list-in-ruby.html

fin