

Linked Lists



“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

DISCLAIMER

You will likely never have a compelling reason to actually use a linked list as a web developer

Linked Lists matter because
~~for some reason they still get~~
~~used in programming~~
~~interviews~~

data structures matter

Some Other Lists

ARRAYS

```
[ :one, 2, "three" ]  
int list[3] = {1, 2, 3};
```

LITERALLY EVERYTHING IN LISP

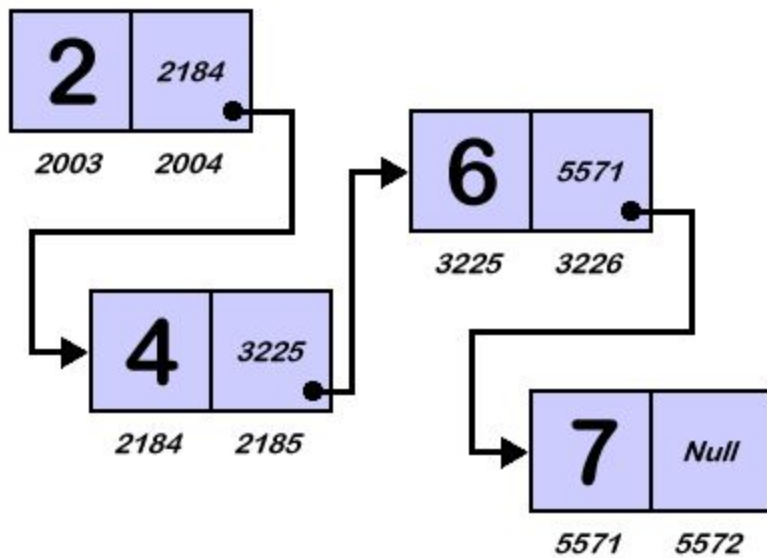
```
(map 'string 'shout (lisp is weird))
```

```
<ul>  
  <li>HTML</li>  
  <li>does</li>  
  <li>too</li>  
</ul>
```

Notice the C array

```
int list[3] = {1, 2, 3};  
const char *strings[3][3] = {"one", "two", "thre"};
```

- size set in advance,
can't be changed
- only one data type per
array
- that sucks



Linked Lists aren't in the
standard library of either
Ruby or JavaScript

So let's write our own
implementation


```

1  class Node
2    attr_accessor :value, :next
3
4    def initialize(value, next_node = nil)
5      @value = value
6      @next = next_node
7    end
8
9    def to_s
10     value.to_s
11 end

```

```

1  require_relative 'node'
2
3  class LinkedList
4    include Enumerable
5
6    attr_accessor :head
7
8    def each
9      # if @head is nil, the second half of the && expression won't run
10     @head && yield(@head)
11
12     next_node = @head.next
13     while next_node
14       yield(next_node)
15       next_node = next_node.next
16     end
17   end
18
19   def initialize(head = Node.new)
20     @head = head
21   end
22
23   # add a node to the front of the list
24   def unshift(new_head)
25     new_head.next = @head
26     @head = new_head
27   end
28
29   # remove and return the node from the front of the list
30   def shift
31     old_head = @head
32     @head = @head.next
33     old_head
34   end

```