



Douglas Davison
Student Tutor

Making FRIENDS with singly-linked lists



TODAY'S

Agenda



What are Singly-Linked Lists? Why use them?

Linked Lists vs. Arrays

Structure: nodes

Structure: the list

How we use and change Singly-Linked Lists
(and using them to make friends)

Applause (5min)

```
3 require File.expand_path("../../../../../tmp/spec/spec", __FILE__)
4 # Prevent database truncation of the test database
5 abort("The Rails environment is running in production mode")
6 require 'spec_helper'
7 require 'rspec/rails'

8
9 require 'capybara/rspec'
10 require 'capybara/rails'

11 Capybara.javascript_driver = :webkit
12 Category.delete_all; Category.create!
13 Shoulda::Matchers.configure do |config|
14   config.integrate do |with|
15     with.test_framework :rspec
16     with.library :rails
17   end
18 end
19
20 # Add additional requires below this line if you need them.
21
22 # Requires supporting ruby files with custom matchers and
23 # run as spec files with their own跑动器. See
24 # in _spec.rb will both be required and run
25 # run twice. It is recommended that you do not name
26 # end with _spec.rb. You can configure this with the
27 # option on the command line -l
28 # option on the command line -l
29
30 # mongoid
31
32 # buffer
```

TODAYS

Agenda

2 minutes: Super-quick photo shoot (while we're all here!)

What are Singly-Linked Lists? Why use them?

Linked Lists vs. Arrays

Structure: nodes

Structure: the list

How we use and change Singly-Linked Lists (and using them to make friends)

5 minutes: audience photo shoot (while we're all here!)

Do
this
part
quickly



```
#define L_tmpnam          _TNAMAX
#define TMP_MAX             32
#define SEEK_SET             0
#define SEEK_CUR             1
#define SEEK_END             2
#define stdin                _Files[0]
#define stdout               _Files[1]
#define stderr               _Files[2]
/* type definitions */

#ifndef _SIZET
#define _SIZET
typedef _Sizet size_t;
#endif
typedef struct {
    unsigned long _Off;      /* system dependent */
} fpos_t;
typedef struct {
    unsigned short _Mode;
    short _Handle;
    unsigned char *_Buf, *_Bend, *_Next;
    unsigned char *_Rend, *_Rsave, *_Wend;
    unsigned char _Back[2], _Cbuf, _Nback;
    char *_Tmpnam;
} FILE;
/* declarations */

void clearerr(FILE *);
int fclose(FILE *);
int feof(FILE *);
int ferror(FILE *);
int fflush(FILE *);
int fgetc(FILE *)■
int fgetpos(FILE *, fpos_t *);
char *fgets(char *, int, FILE *);
FILE *fopen(const char *, const char *);
```

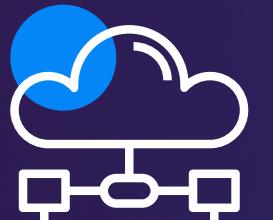


CONCLUSIONS

**Results were
incredible**

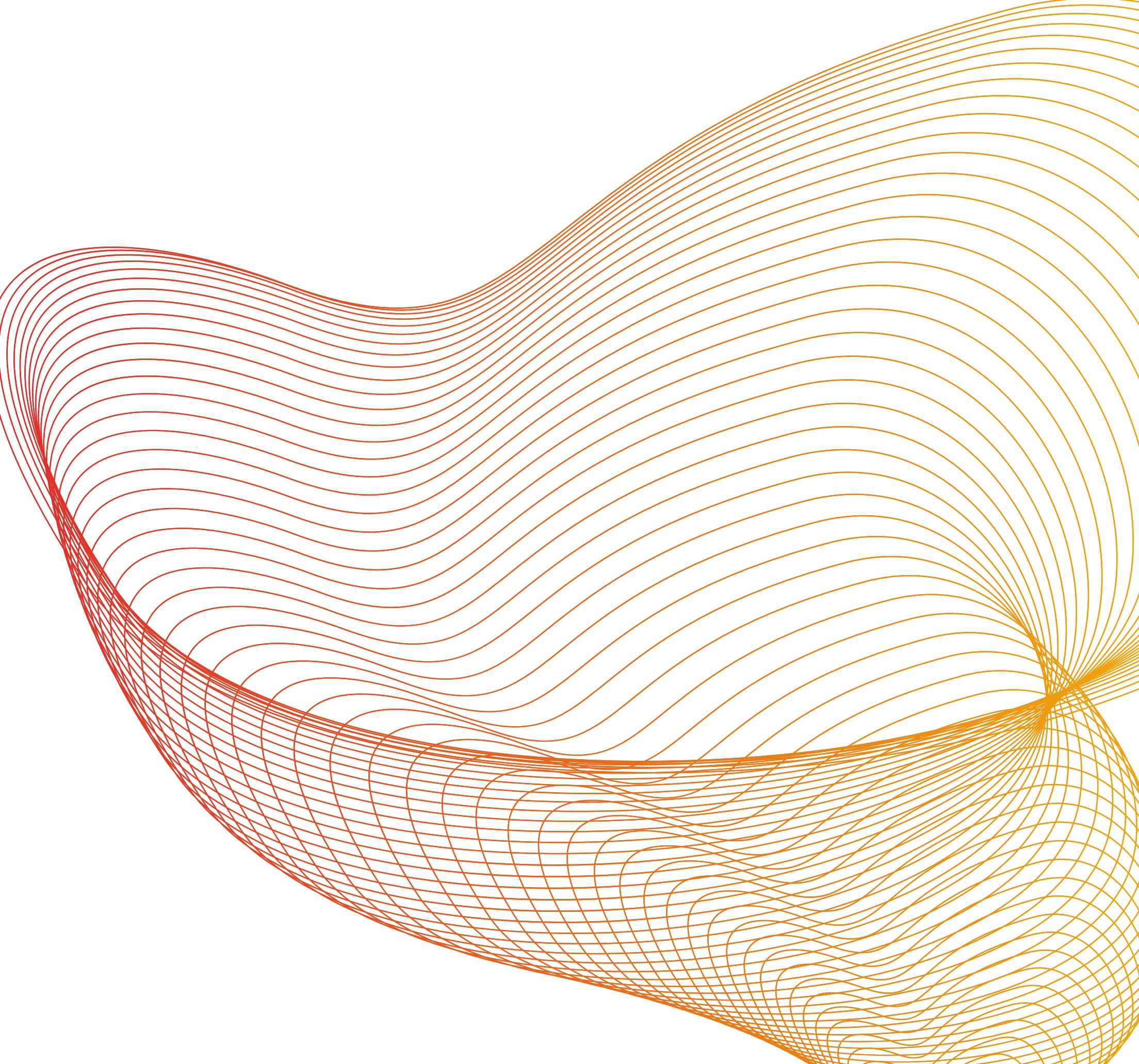
>> BEYOND TOMORROW

What's Next





live Q & A



success



back to
our regularly-
scheduled
presentation



What is a linked list?



Singly-Linked Lists vs Arrays

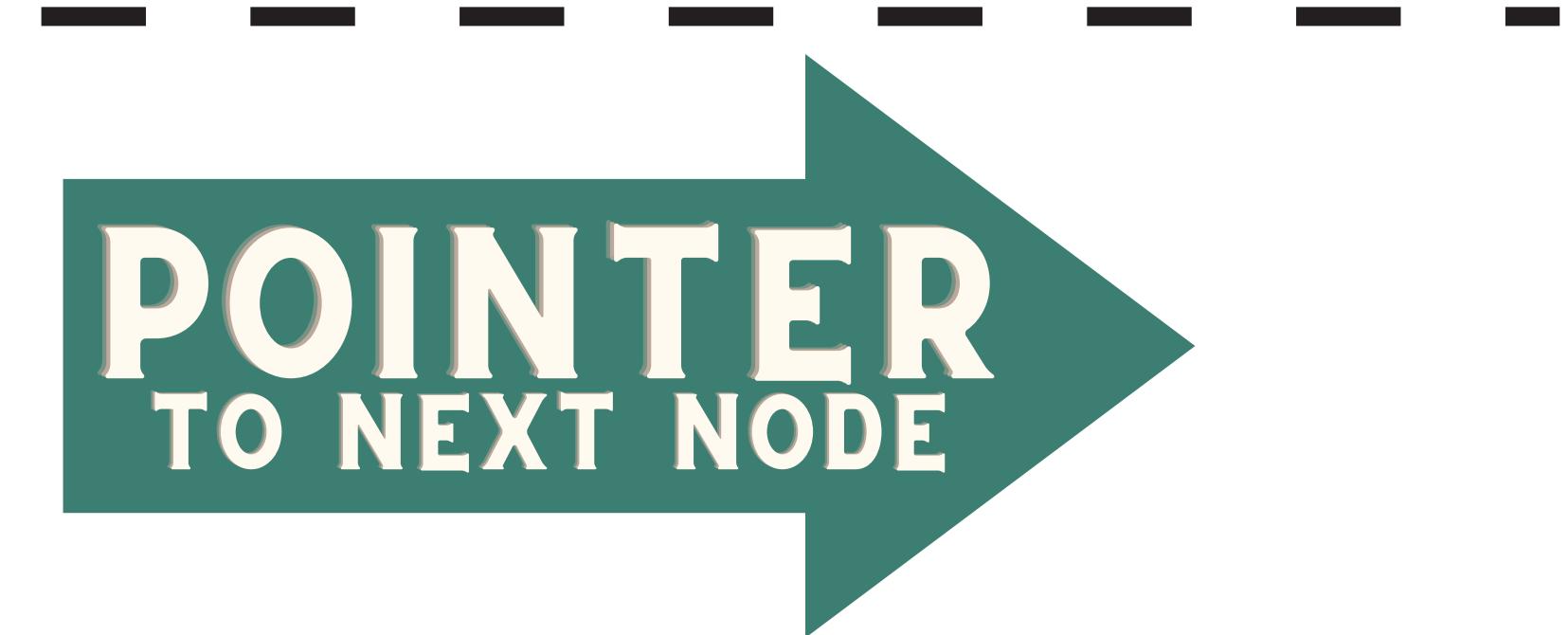
Arrays	Singly-Linked Lists
Fixed size	Flexible size!
Adjacent memory locations	Members live anywhere, like remote-working software engineering teams
Access by index	Access in order, by traversing



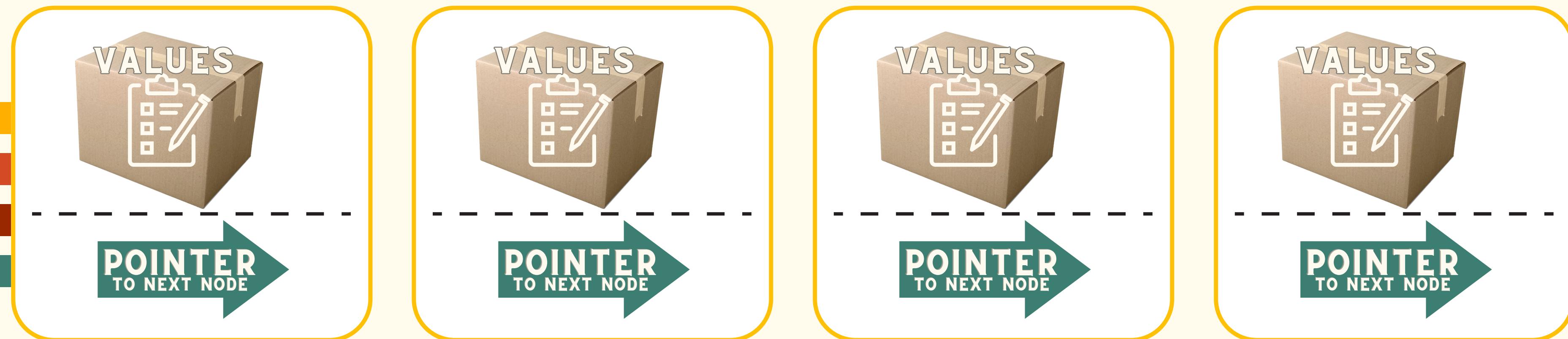
Umm, but
What do they
look like?



a
Node



a bunch of nodes



pointing to each other

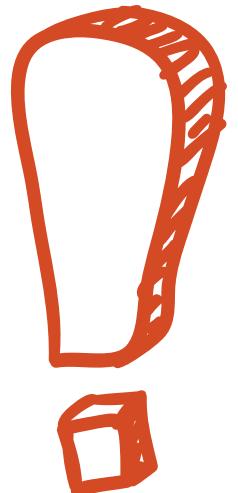
almost there...

let's try it

in person



special pointers



HEAD

points to start

NEXT

NOTE

on the last node





finally, a singly-linked list!

let's code!



simple project:

friends list

make the node structure

- name
- phone
- pointer to next node

let's talk traversal

going through the list

- to look at it
- to change it
- to perform operations

based on it

example: list printer



let's start
with one
best friend



let's make a
new BEST
friend, but
stay friends
with our
old bff



now let's add
some lesser
friendships



finally, let's
cut a friend
out of our
lives

hey, sometimes people move!





Douglas Davison
Student Tutor

The End

(slide->next == NULL)



