

## CSC151.02 2013F, Class 36: Vectors

---

### *Overview*

- Admin.
- Rethinking lists: Good aspects, bad aspects.
- An Alternative: Vectors.
- Yay! A new data type!
- Behind the scenes: How Scheme implements vectors.
- Important vector procedures.

### *Admin*

- Today's writeup: Exercise 6
- New assignment: Mostly building color trees.
- Review session needed. Yes!
- Upcoming extra credit opportunities:
  - Any one Grinnell prize event this week.
  - Thursday extras this week: Reports from internships (I think)
  - Grinnell Town Hall Meeting, Nov 13 noon or 7:30, I believe
  - CS Table, Friday: TBD (it's been a bad week)
  - Orchestra, Saturday, 7:30 in SL
  - 24-hour theatre and dance production, 7:30 Saturday
  - Drag Show
  - Some other love your body week event
  - Alumni Basketball game 3pm Saturday at Bear
- Cool talk on supernovas 7:30 in ARH 302

## **Rethinking lists**

- They collect values and let you do things to each value!
- Extensible!
- And shrinkable
- ANd reversable
- But ...
  - Slow to access the nth element
  - Hard to change the middle

## A solution: Vectors

- Collects values!
- Fast to access the nth element!
- Can change the nth element!
- But not easily growable or shrinkable.
  - "Not easily" == "Computational expensive."
- About has hard to reverse as a list
  - Although semantics issues - Does vector-reverse create a new vector or just mutate a vector?
    - Mutates! Mutation is good. It's the point.
  - Depends on what you name it
    - vector-reverse
    - vector-reverse!

## Thinking about a new type!

- What's its name? Vectors
- What do we use it for? Store collections of values, using indices
- What are situations in which we use them?
  - To keep track of all the images we've created
  - To keep track of students by id number
- How do we express them? / How are they created?
  - (vector val val val)
  - #(val val val)
  - (make-vector n val)
- What procedures can we use?
  - (vector-ref vec pos) - gets a value at a position
  - (vector-set! vec pos newval) - sets a value
  - Is there a non-mutation vector-set? No. You can write it.
- How does Racket present them to us?
  - '#(...)
- How are they different from other types?
- How do we do recursion with them?
  - Recurse over position
- How do we break them?

Specific to vectors \* What kinds of values can we put in them? Any!

## Behind the scenes: How Scheme implements vectors

---

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2013 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.