Conses in Ruby: So Much More Than Lists

Christopher Mark Gore

cgore.com

Monday, September 12, AD 2016

Ruby is my ~~most second~~ third favorite programming language of all time.

1. My language, Teepee, eventually *(but not yet)*
2. Common Lisp
3. Ruby

.

.

.

999. Java

Nearly every programming language has some way to do things to a list/array/vector/whatever of things.

Ruby has arrays.

```ruby
a = [1,2,3,4,5]
a.class # Array
a.length # 5
a.first # 1
a.map {|ai| ai*2} # [2,4,6,8,10]
```

Common Lisp prefers linked lists.

```
(setf a '(1 2 3 4 5))
(class-of a) ; #<BUILT-IN-CLASS COMMON-LISP:CONS>
(length a) ; 5
(first a) ; 1
(mapcar (lambda (i) (* 2 i)) a) ; '(2 4 6 8 10)
```

But Common Lisp also has vectors, which are basically the same as Ruby arrays.

```
(let v (vector 1 2 3 4 5))
(class-of v)
;; #<BUILT-IN-CLASS COMMON-LISP:SIMPLE-VECTOR>
(length v) ; 5
(elt v 0) ; 1
(map 'vector (lambda (i) (* 2 i)) v)
;; #(2 4 6 8 10)
```

Questions?