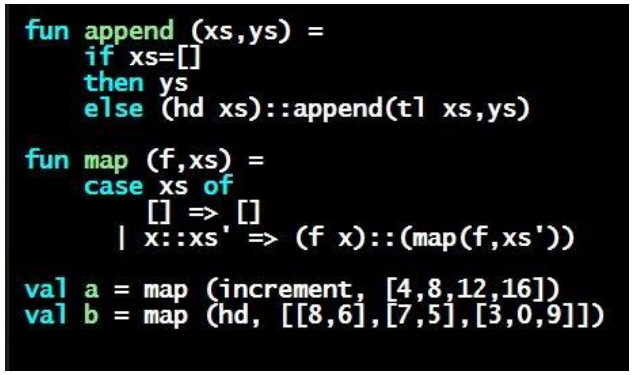


# Programming Languages

## Dan Grossman

Racket Lists

# *Another old-friend: List processing*

Empty list: **`null`**

Cons constructor: **`cons`**

Access head of list: **`car`**

Access tail of list: **`cdr`**

Check for empty: **`null?`**

Notes:

- Unlike Scheme, **`()`** doesn't work for **`null`**, but **`'()`** does
- **`(list e1 … en)`** for building lists
- Names **`car`** and **`cdr`** are a historical accident

# *Examples*

```
(define (sum xs)
  (if (null? xs)
      0
      (+ (car xs) (sum (cdr xs)))))

(define (my-append xs ys)
  (if (null? xs)
      ys
      (cons (car xs) (my-append (cdr xs) ys))))

(define (my-map f xs)
  (if (null? xs)
      null
      (cons (f (car xs)) (my-map f (cdr xs)))))
```