https://v.gd/ewojan

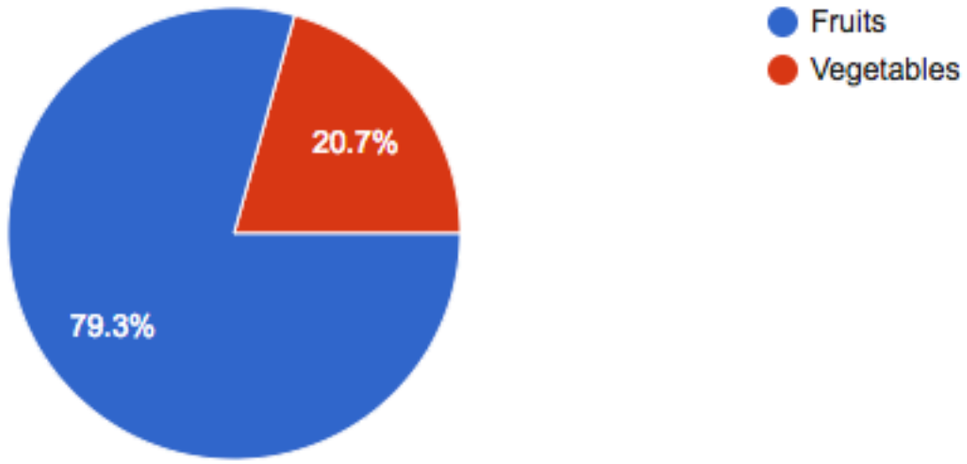
#9 Macros and Streams

TA: Jerry Chen (jerry.c@berkeley.edu)

Have you seen the new Yelp, but for sequences? It's a great "iter rater."

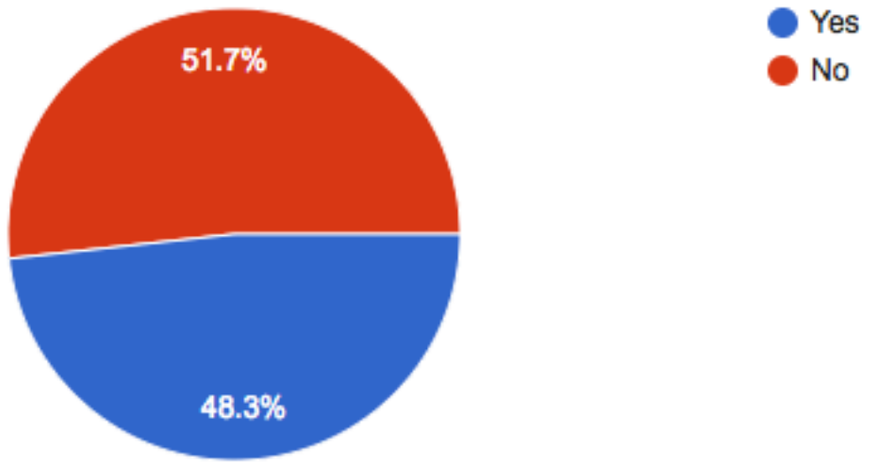
Are tomatoes fruits or vegetables?

29 responses



Do tomatoes belong in fruit salads?

29 responses



Macros

A new special form (define-macro)

- Special evaluation rules for ALL special forms
  - Ex: if, cond, and, etc.
- Special in two aspects:
  - How is a macro form defined?
  - What gets evaluated in a macro form?

```
1 (define (get-oper-def a)
2         (car a))
3
4 (define-macro (get-oper-mac a)
5                 (car a))
```

[Function/Macro] name

```
1 (define (get-oper-def a)
2         (car a))
3
4 (define-macro (get-oper-mac a)
5                 (car a))
```

(List of) arguments

```
1 (define (get-oper-def a)
2         (car a))
3
4 (define-macro (get-oper-mac a)
5                 (car a))
```

[Function/Macro] body

```
1 (define (get-oper-def a)
2         (car a))
3
4 (define-macro (get-oper-mac a)
5                 (car a))
```

Macros

What's the difference?

- Macro and function definition look very similar
- Call expression evaluation is ALSO different
- Operands are unevaluated

```scheme
(define (get-oper-def a)
        (car a))
```

```
scm> (get-oper-def '(+ 1 2))
+
```

```
(define-macro (get-oper-mac a)
              (car a))

scm> (get-oper-mac '(+ 1 2))
Traceback (most recent call last):
  0       (get-oper-mac (quote (+ 1 2)))
  1     quote
Error: unknown identifier: quote
```

```
(define-macro (get-oper-mac a)
              (car a))

scm> (get-oper-mac '(+ 1 2))
Traceback (most recent call last):
  0       (get-oper-mac (quote (+ 1 2)))
  1     quote
Error: unknown identifier: quote

scm> quote
Traceback (most recent call last):
  0     quote
Error: unknown identifier: quote
```

```
(define (get-oper-def a)
        (car a))

(define-macro (get-oper-mac a)
              (car a))

scm> (get-oper-def '(+ 1 2))
+
scm> (get-oper-def (list '+ 1 2))
+
```

```
(define (get-oper-def a)
        (car a))

(define-macro (get-oper-mac a)
              (car a))

scm> (get-oper-def '(+ 1 2))
+
scm> (get-oper-def (list '+ 1 2))
+
scm> (get-oper-mac '(+ 1 2))
Error: unknown identifier: quote
scm> (get-oper-mac (list '+ 1 2))
#[list]
```

```
(define (get-oper-def a)
        (car a))

(define-macro (get-oper-mac a)
              (car a))

scm> (get-oper-def '(+ 1 2))
+
scm> (get-oper-def (list '+ 1 2))
+
scm> (get-oper-mac '(+ 1 2))
Error: unknown identifier: quote
scm> (get-oper-mac (list '+ 1 2))
#[list]
```

Depends on argument structure, not just what they eval to

(1) Eval body, do not eval args!
(2) Eval result expression

```
(define-macro (get-oper-mac a)
  (car a))
```

Streams

Being lazy pays off

Lists, but better

Streams

New primitives

`cons-stream` allows us to delay evaluation of **`second`**

`cdr-stream` forces that evaluation