Lists

- Express a collection of items, not just one type

Selectors & Constructors

- List
  - Takes multiple arguments and returns a list with all of them evaluated
  - > (list '(i am) '(the walrus))
    ((I AM) (THE WALRUS))
- Cons
  - Takes two elements (element & list) and adds the element to the beginning of the list
  - > (cons '(i am) '(the walrus))
    ((I AM) THE WALRUS)
- Append
  - Combines two or more lists into one list
  - > (append '(i am) '(the walrus))
    (I AM THE WALRUS)
- Assoc
  - Searches for an entry in a list
  - Compares entry to first of the list and if true, returns the whole portion of the list
- Map
  - Takes a function and a list, and returns a list containing the result of applying the function to each element of the list

Other

- Car → first
- Cdr → butfirst
- Cadr → first of butfirst (takes out first element and finds the first of the resultant list)
- Caar → first of first
- Caadr (car (car (cdr x)))
- Null? → equivalent to empty?