

# List Exercises

---

```
(lon-equal? list1 list2)
```

is true only if `list1` and `list2` (both lists of number) have the same numbers in the same order.

```
(lon-equal? empty empty) => true
```

```
(lon-equal? (cons 1 empty) empty) => false
```

```
(lon-equal? (cons 1 (cons 2 empty))
```

```
  (cons 2 (cons 1 empty))) => false
```

```
(lon-equal?
```

```
  (cons 2 (cons 3 (cons 5 empty)))
```

```
  (cons 2 (cons 3 (cons 5 empty)))) => true
```

# add item to end of a list

---

- cons only adds to the beginning – write a function that creates a new list identical to the original but with a 'blah on the right:

```
(addright 'blah (cons 1 (cons 'hi empty))) =>  
  (cons 1 (cons 'hi (cons 'blah empty)))
```

# Remove items

---

```
(remove-first (cons 1 (cons 2 empty))) =>  
  (cons 2 empty)
```

```
(remove-last (cons 1 (cons 2 empty))) =>  
  (cons 1 empty)
```

# Add two lists of numbers (as vectors)

---

Given two lists of numbers (the lists are the same size), this function produces a new list of numbers (of the same size) in which each element is the sum of the corresponding elements from the original lists.

```
(sum-1 on
  (cons 1 (cons 3.3 empty))
  (cons 12 (cons 4 empty))) =>
  (cons 13 (cons 7.3 empty))
```

# reverse a list of numbers(tricky?!?)

---

```
(reverse-lon
```

```
  (cons 18 (cons 2 (cons 3.14 empty)))) =>
```

```
  (cons 3.14 (cons 2 (cons 18 empty))))
```

```
(define (check-reverse l)
```

```
  (lon-equal? l
```

```
    (reverse-lon (reverse-lon l))))
```