



Equality

Or, why you can say that, in Scheme,
“All animals are equal, but some animals
are more equal than others”



Our Running Example

- `(define a (make-posn 1 2))`
- `(define b (make-posn 1 2))`
- `(define c b)`

- We know `a` and `b` are equal



Our Running Example

- `(define a (make-posn 1 2))`
 - `(define b (make-posn 1 2))`
 - `(define c b)`
-
- We know `a` and `b` are equal
 - But isn't `c` "more equal" to `b`?



More on What to do with set!

- We'll always start by thinking purely
- Two examples:
 - insert-sort
 - quick-sort
- How were these built?

- How can we use imperative features?