Set

Sample Set: [ Red, Blue, Green, Orange ]

## Size

We will need a size function to return the number of items that are within the set. To do this we will use the Length method to return the number of items.

Q.Size() : { Length[Q] }

i.e. Q.Size = Length = 4

## Empty

The empty function will return the set if the set is equal to null or empty, if it has elements then it will return nil.

Q.Empty() : { null } => NIL

Since there are items in the set (Red, Blue, etc.) it will return nil.

Found answer on: <http://clhs.lisp.se/Body/f_null.htm>

## Push

If someone wants to push an item onto the end of the set, they will use the push function and pass in the item to be added. We will use the cons method to add this item onto the end.

Q.push(e) : { S’ = cons[S;e] }

i.e Q.push(pine) = cons[ (fir oak maple); (pine)] = (fir oak maple pine)

Q.push(e) : { S” = cons[ (S.top, S.pop)\*, (e) }

## Pop

## Top