

---

---

## nth

*Accessor*

---

### Syntax:

`nth n list` → *object*  
`(setf (nth n list) new-object)`

### Arguments and Values:

*n*—a non-negative *integer*.  
*list*—a *list*, which might be a *dotted list* or a *circular list*.  
*object*—an *object*.  
*new-object*—an *object*.

### Description:

`nth` locates the *n*th element of *list*, where the *car* of the *list* is the “zeroth” element. Specifically,  
`(nth n list) ≡ (car (nthcdr n list))`  
`nth` may be used to specify a *place* to `setf`. Specifically,  
`(setf (nth n list) new-object) ≡ (setf (car (nthcdr n list)) new-object)`

### Examples:

```
(nth 0 '(foo bar baz)) → FOO
(nth 1 '(foo bar baz)) → BAR
(nth 3 '(foo bar baz)) → NIL
(setq 0-to-3 (list 0 1 2 3)) → (0 1 2 3)
(setf (nth 2 0-to-3) "two") → "two"
0-to-3 → (0 1 "two" 3)
```

### See Also:

`elt`, `first`, `nthcdr`

---