Why does the following code evaluate to #f

(eq? - (car '(-)))

One evaluates to - and the other evaluates to <builtin -(@)}. Why?

Because - is defined as a function, while the (car '(-)) is a symbol.

If you want to check if they're equal I think the correct call would be (eq? '- (car '(-))) which should evaluate to true.

Ah. However, I don't that that's technically the semantically correct way to do it. I guess I didn't fully understand what the apostrophe did. It looks like (eq? - (car (list -))) seems to work. Thanks!

tsccrompton wrote on Sat, 20 February 2016 20:31

Ah. However, I don't that that's technically the semantically correct way to do it. I guess I didn't fully understand what the apostrophe did. It looks like (eq? - (car (list -))) seems to work. Thanks!

tscrompton on Mon, 22 Feb 2016 21:20:53 GMT

If I do (define minus -), the following would evaluate to #f:
Whereas, the following would evaluate to #t.

(eq? - (car (list minus)))

---

Subject: Re: Functional Equality
Posted by jarobinson3 on Mon, 22 Feb 2016 22:03:34 GMT
View Forum Message <> Reply to Message

Lets take a look at each separately,

Here is your first piece of code:

(eq? ' - (car '(minus)))

This is a symbol (use symbol? to check)
' -

This is a list (use list? to check) which contains a function called minus.
'(minus)

Does the symbol minus equal the function minus? I would say not.

Let's look at the second piece of code:

(eq? - (car (list minus)))

This is a function (builtin)
-

This is a list containing the function minus:
(list minus)

Since you defined minus to be - they are the same. So you should get true.

---

Subject: Re: Functional Equality
Posted by lusth on Mon, 22 Feb 2016 23:21:48 GMT
View Forum Message <> Reply to Message

(define minus -)
(eq? - (car (list minus)))

is the same as:
(define a 3)
(= 3 (car (list a)))

On the other hand:

(eq? '- (car '(minus)))

is the same as:

(eq? '- (car (list 'minus)))

The symbol - is not the same symbol as the symbol minus.