avoiding the infamous “eval(parse())” construct

Ok, so I'm running some loops to process data stored in list objects. Ever mindful of the infamous `fortune` admonishment not to use `eval(parse(mystring))` , I came up with this:

```
Rgames> bar
$foo
$foo$fast
[1] 1 2 3 4 5

$foo$slow
[1]  6  7  8  9 10


$oof
$oof[[1]]
[1]  6  7  8  9 10

$oof[[2]]
[1] 1 2 3 4 5


Rgames> rab<-'bar'
Rgames> do.call('$',list(as.name(rab),'oof'))
[[1]]
[1]  6  7  8  9 10

[[2]]
[1] 1 2 3 4 5
```

Typically I'd be selecting a list (of which `bar` is one such) and then one element of the list (e.g. `oof`) which contains my data. The code above does the same thing as `eval(parse(text=paste(rab,'$','oof',sep='')))` .
I'm doing all this specifically because I want to use the lists' names rather than `[[x]]` notation as a safety mechanism (because not all list objects have their contents in the same order). Should I stick with the advice from DWin in R: eval(parse(...)) is often suboptimal ?

r

asked Nov 30 '12 at 14:24

Carl Witthoft
12.2k 3 16 43

2 Answers

Using `get` and `[[` :

```
bar <- list(foo = list(fast = 1:5, slow = 6:10),
            oof = list(6:10, 1:5))

rab <- 'bar'

get(rab)[['oof']]
# [[1]]
# [1]  6  7  8  9 10
#
# [[2]]
# [1] 1 2 3 4 5
```

answered Nov 30 '12 at 14:33

flodel
50k 6 64 114

If the name of your top list is going to change and be accessed by a variable with the name then it is best to put those lists into another list, then you can access the list you want using `[[` . Also read `fortune(312)` and the help on `?'[['` .

You can then access the pieces in a different ways (detailed on the help page `?'[['`).

```
mylist <- list()
mylist$bar <- bar

mylist[[rab]][['oof']]
#or
mylist[[ c(rab,'oof') ]]
```

answered Nov 30 '12 at 17:20

Greg Snow
26.3k 1 21 42

This is a very good point, but it does assume I'm organized enough :-) to know all the lists I'll be using in advance. – Carl Witthoft Nov 30 '12 at 17:38

@CarlWitthoft, any time that you go to use a list that is not already in your master list you can add it using `get` . – Greg Snow Nov 30 '12 at 20:37

Yep. Thanks again. – Carl Witthoft Nov 30 '12 at 21:33

Or do `globalenv()[["mylist"]]` - lists and environments work very similarly. – hadley Dec 1 '12 at 0:44

@hadley, I considered making the same suggestion, but it should be noted (this is to others reading this, not Hadley) that that can get very dangerous if new users start overusing 'globalenv()` and similar rather than thinking through better approaches. – Greg Snow Dec 2 '12 at 5:47