

Ruby - Bug #8359

If p is a proc, p.call(x) can take a block, but p[x] cannot

05/02/2013 07:46 PM - alexeymuranov (Alexey Muranov)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	2.0.0	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN
Description <pre>=begin format = lambda { l, s = "", &w l.map { e w ? w[e] : e}.join(s) } format.call([1, 2], "\n") { c "[#{ c }]"} => "[1]\n[2]" format[[1, 2], "\n"] { c "[#{ c }]"} => SyntaxError: unexpected { arg, expecting end-of-input =end</pre>		

History

#1 - 05/02/2013 08:45 PM - Anonymous

- Status changed from Open to Rejected

```
=begin
```

You can pass a block to `(([ #[] ]))` in two ways:

```
format[[1, 2], "\n", &proc {|c| "[ #{ c } ]"}]
```

or:

```
format.[]([1, 2], "\n") {|c| "[ #{ c } ]"}
```

```
=end
```

#2 - 05/02/2013 09:04 PM - alexeymuranov (Alexey Muranov)

```
=begin
```

Ok, maybe this is not a bug report but a feature request. Is there a reason to not allow `(([ p[x] do ...]))`?

```
=end
```