

CommonRuby - Feature #11605

Ruby should raise on def(a='a', b) [optional param that can't be used]

10/19/2015 04:49 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:		
Description		
Why doesn't Ruby complain when defining a method like this?		
<pre>def a(opt='opt', required) end</pre>		
I believe it's impossible to call method 'a' passing only required, while sticking to the default value for 'opt', right? So, why should Ruby allow such definition?		

History

#1 - 10/19/2015 05:12 PM - usa (Usaku NAKAMURA)

- Status changed from Open to Rejected

Why did not you try actually?

```
irb(main):001:0> def a(opt='opt', required)
irb(main):002:1>   p [opt, required]
irb(main):003:1> end
=> :a
irb(main):004:0> a('required')
["opt", "required"]
=> ["opt", "required"]
```

#2 - 10/19/2015 05:17 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Oh, thanks. I had no idea on how to try this. I've tried a('required'). Actually, I was reading a method like a(req1, opt='opt', req2) and thought it would be impossible to use the default value, but now I understand how optional arguments work in Ruby. Sorry for the confusion.