Description

Using ruby 2.2.2 I see this work:

```
[18] pry(main)> def test(cmd, opts={})
[18] pry(main)>   puts cmd,opts
[18] pry(main)> end
[19] pry(main)> test(:ads, :d => 6, :f => :gah, {a:3,b:4} => 3)
ads
{:d=>6, :f=>:gah, {a=>3, :b=>4}=>3}
=> nil
```

But this fails:

```
[2] pry(main)> def test(cmd, **opts)
[2] pry(main)>   puts cmd, opts
[2] pry(main)> end
[9] pry(main)> test(:ads, :d => 6, :f => :gah, {a:3,b:4} => 3)
ArgumentError: wrong number of arguments (2 for 1)
from (pry):2:in `test'
```

Related issues:

Related to Ruby master - Feature #14183: "Real" keyword argument

History

#1 - 06/08/2015 09:15 PM - jeremyevans0 (Jeremy Evans)

This is expected behavior, keyword argument hashes only support symbol keys, not hash keys. I suppose the error message could be better, though.

#2 - 06/09/2015 07:39 AM - nobu (Nobuyoshi Nakada)

- Description updated

What message would be better?

#3 - 06/09/2015 08:25 AM - akostadinov (Aleksandar Kostadinov)

Nobuyoshi Nakada wrote:

What message would be better?

Unsupported key type?

btw what is the reason to have that limitation? Not that I need this functionality, I just happened to notice the difference...

#4 - 12/14/2017 07:24 AM - hsbt (Hiroshi SHIBATA)

- Related to Feature #14183: "Real" keyword argument added

#5 - 09/05/2018 04:11 PM - marcandre (Marc-Andre Lafortune)

When last argument is hash-like but has keys that are not symbols, we could output instead "wrong number of arguments (2 for 1); note that the last argument has keys that are not symbols and thus was not considered as keyword parameters"
With the changes in #14183, keyword splats can support arbitrary keys, and the second example now passes:

test(:ads, :d => 6, :f => :gah, {a:3,b:4} => 3)
# ads
# {:d=>6, :f=>:gah, {a=>3, :b=>4}=3}