

Ruby master - Feature #5735

Extending the use of splat operator to when it is inside a hash

12/09/2011 01:31 PM - sawa (Tsuyoshi Sawada)

Status:	Closed
Priority:	Normal
Assignee:	matz (Yukihiro Matsumoto)
Target version:	
Description	
Ruby convention of allowing omission of the curly brackets for the last argument is convenient:	
<pre>foo(arg1, arg2, 1 => :a, 2 => :b)</pre>	
Sometimes, I want to pass a hash with some modifications. For example, suppose <code>h = {3 => :c, 4 => :d, 5 => :e}</code> is a hash already defined and that I want to add some key-value pairs as well as overwrite some values of <code>h</code> , such as <code>{3 => :c, 4 => :f, 5 => :e, 6 => :g}</code> , and pass that. The current convention only allows:	
<pre>foo(arg1, arg2, h.merge(4 => :f, 6 => :g))</pre>	
but it would be more convenient if a hash preceded by the splat operator is placed in a hash, it is interpreted as part of the hash, allowing notations like:	
<pre>foo(arg1, arg2, *h, 4 => :f, 6 => :g)</pre>	
or, if I want to overwrite the hash <code>{4 => :f, 6 => :g}</code> with <code>h</code> , then:	
<pre>foo(arg1, arg2, 4 => :f, 6 => :g, *h)</pre>	
Or besides the argument position, in general, usages like the following:	
<pre>{3 => :h, *h, 4 => :f, 6 => :g}</pre>	
This is an analogy from the splat operator used within an array:	
<pre>[1, 2, *[4, 5, 6], 7, 8]</pre>	

History

#1 - 01/05/2012 01:06 PM - kernigh (George Koehler)

=begin

Ruby trunk has a hash splat operator because of feature [#5474](#), keyword argument. See ([URL:http://bugs.ruby-lang.org/issues/5474](http://bugs.ruby-lang.org/issues/5474)).

With this version of Ruby,

`ruby 2.0.0dev (2012-01-04 trunk 34211) [x86_64-openbsd5.0]`

I can run this code,

```
def eat(i: 1, **hash)
  printf "i is %p, hash is %p\n", i, hash
end
eat(h: 7, i: 8, j: 9)
# => "i is 8, hash is {:h=>7, :j=>9}"
```

So far, `**hash` only works when defining a method. We cannot (yet) use `**hash` in a method call or a hash literal.

```
args = {h: 2, i: 3, j: 4}
eat(**args) # syntax error!
args = {k: 5, **args} # syntax error!
```

There is much more discussion of hash splat at feature [#5474](#).

=end

#2 - 03/29/2012 12:57 AM - mame (Yusuke Endoh)

- Status changed from Open to Feedback
- Assignee set to matz (Yukihiko Matsumoto)

Marc-Andre proposed a similar idea at [ruby-core:41772], and Matz commented a little at [ruby-core:41822]. It would be good to investigate how many use cases occur in real world.

matz wrote:

I am still not sure if we need hash splat nor hash decomposition, it might be useful in some cases, but also makes syntax more complex. So we need more discussion before picking it.

But if we could made consensus I'd make small changes to proposed syntax.

For hash splat, I'd rather use ** instead of *, because splat in array and splat in hash are different. Using same operator could cause confusion. Besides that, I would also restrict hash splat position to the end, since hashes do not have order.

--

Yusuke Endoh mame@tsg.ne.jp

#3 - 11/20/2012 09:57 PM - mame (Yusuke Endoh)

- Target version set to 2.6

#4 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)

- Target version deleted (2.6)

#5 - 07/08/2019 09:53 AM - sawa (Tsuyoshi Sawada)

A similar feature has already been realized as the double splat operator. Please close this issue.

#6 - 07/08/2019 10:02 AM - matz (Yukihiko Matsumoto)

- Status changed from Feedback to Closed