

## Ruby master - Feature #6414

### Destructuring Assignment

05/09/2012 03:19 AM - edtsech (Edward Tsech)

<b>Status:</b>	Feedback	
<b>Priority:</b>	Normal	
<b>Assignee:</b>		
<b>Target version:</b>		
<b>Description</b>		
I mean:		
<pre>john = {name: "John", last: "Smith"} {name, last} = john # extract value by key from the hash and set to local variable with same name name # =&gt; "John" last # -&gt; "Smith"</pre>		
Useful for ex. for "options" hashes:		
<pre>def select(options={})   {only, except} = options   ... end</pre>		
As extra part of this feature can be hash constructing in this way:		
<pre>name = "John" last = "Smith" find({name, last}) # equals to =&gt; find({name: "John", last: "Smith"})</pre>		
I think both really nice to have in Ruby. Thanks.		
<b>Related issues:</b>		
Related to Ruby master - Feature #5474: keyword argument	<b>Closed</b>	<b>10/23/2011</b>

### History

#### #1 - 05/17/2012 12:30 AM - mame (Yusuke Endoh)

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

FYI: There is similar (but not same) proposal: [ruby-core:41772]

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

#### #2 - 07/01/2012 06:27 AM - marcandre (Marc-Andre Lafortune)

- File *destructuring.pdf* added
- Category set to core
- Target version set to 2.0.0

Attaching a one-minute slide. It doesn't cover the "construction" part of this request though.

#### #3 - 07/01/2012 10:21 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

-1. I'd still like to see Ruby supporting CoffeeScript hash style for this common pattern:

Instead of  
a = 1; b = 2; c = {a: a, b: b}

I'd prefer to write just c = {a, b}. The current proposed syntax wouldn't allow such a feature in the future.... :(

By the way, I can't remember what is the current situation of an old proposition to allow things like {"#{interpolation}": value} as well as {'some string':

value}.

Was it already accepted? Is there already a slide for it?

#### #4 - 07/02/2012 03:22 AM - mame (Yusuke Endoh)

Marc-Andre, your slide is received. Thank you!

Rodrigo, I don't know CoffeeScript, but what is the difference between OP's (second) proposal and yours?

```
name = "John"
last  = "Smith"
find({name, last}) # equals to => find({name: "John", last: "Smith"})
```

Instead of  
a = 1; b = 2; c = {a: a, b: b}  
I'd prefer to write just c = {a, b}.

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

#### #5 - 07/03/2012 11:01 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

sorry but I couldn't find the proposal you're talking about, but looking at the examples it seems there is no difference.

But one thing that is supported in CoffeeScript and is not in the examples is to mix both forms. For example, in Ruby, that would mean:

```
b = 2; d = 4
{a: 1, b, 'c' => 3, d } == { :a => 1, :b => b, 'c' => 3, :d => d }
```

#### #6 - 07/24/2012 12:10 AM - mame (Yusuke Endoh)

Edward Tsech and Marc-Andre Lafortune,

We discussed your slide at the developer meeting (7/21).

We were not sure if we can actually implement this feature in terms of yacc. Could you create an experimental patch?

Matz was basically positive to the feature itself, but wanted to focus on surely implementable syntax (with no parser conflict and with reasonable code).

--

Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

#### #7 - 07/24/2012 01:20 AM - trans (Thomas Sawyer)

So this is a shortcut for?

```
name, last = john.values_at(:name, :last)
```

If so, long ago it was suggested Hash#[] support multiple keys:

```
name, last = john.name..last
```

Not as short as {name, last} = john, but not quite as esoteric either.

If assigning from an array can work, eg.

```
john = ['John', 'Smith']
name, last = john
```

Why not hash? So, just

```
john = {:name=>'John', :last=>'Smith'}
name, last = john
```

#### #8 - 07/24/2012 07:23 AM - cjheath (Clifford Heath)

On 24/07/2012, at 2:20 AM, trans (Thomas Sawyer) wrote:

If so, long ago it was suggested Hash#[] support multiple keys:

name, last = john:name. :last

I'm very glad this didn't happen. I often have reason to use arrays as hash keys, and this would make that impossible.

Clifford Heath.

**#9 - 07/24/2012 09:28 AM - trans (Thomas Sawyer)**

I made a mistake with that example.

name, last = john[:name, :last]

**#10 - 07/24/2012 11:53 AM - rosenfeld (Rodrigo Rosenfeld Rosas)**

I'd much prefer something like this and reserve the {a, b} syntax to be used as a shortcut for {a: a, b: b} like in CoffeeScript in the future...

Also, I find it clearer to read as well

**#11 - 07/24/2012 12:01 PM - shyouhei (Shyouhei Urabe)**

- Status changed from Assigned to Feedback

- Assignee deleted (matz (Yukihiro Matsumoto))

This request won't progress without a working proof-of-concept. Feel free to send us your pull request. <https://github.com/ruby/ruby>

**#12 - 10/25/2012 06:43 PM - yhara (Yutaka HARA)**

- Target version changed from 2.0.0 to 3.0

**#13 - 10/25/2012 06:56 PM - yhara (Yutaka HARA)**

- Target version changed from 3.0 to 2.6

**#14 - 12/25/2017 06:15 PM - naruse (Yui NARUSE)**

- Target version deleted (2.6)

**Files**

---

destructuring.pdf	128 KB	07/01/2012	marcandre (Marc-Andre Lafortune)
-------------------	--------	------------	----------------------------------