

Ruby trunk - Feature #12578

Instance Variables Assigned In parameters (ala Crystal?)

07/09/2016 08:42 PM - lankford (Matt Lankford)

Status:	Rejected
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>When I first saw the instance assignments in Crystal, I was floored... I have looked at all 18 pages in this tracker and thought it "must" have been mentioned in here... but I don't see it... pardon me if I missed it...</p> <p>BUT</p> <p>so in Crystal, this :</p> <pre>class MyStuff def initialize @thing_one , @thing_two end end</pre> <p>would replace :</p> <pre>class MyStuff def initialize thing_one , thing_two @thing_one = thing_one @thing_two = thing_two end end</pre> <p>One of the things I loved about ruby was attr_accessor / reader / writer...</p> <p>I know in Crystal, it is a macro, but still at the millions of lines of boilerplate that would go away...</p> <p>Has no one thought of stealing this from Crystal or is there a good reason this does not exist? (and I will go sit down and color now)</p>	
Related issues:	
Is duplicate of Ruby trunk - Feature #5825: Sweet instance var assignment in ...	Assigned
Is duplicate of Ruby trunk - Feature #8563: Instance variable arguments	Rejected
Is duplicate of Ruby trunk - Feature #12023: Allow ivars to be used as method...	Open
Has duplicate Ruby trunk - Feature #12820: Shorter syntax for assigning a met...	Rejected

History

#1 - 07/09/2016 11:10 PM - hsbt (Hiroshi SHIBATA)

- Assignee deleted (ruby-core)

#2 - 07/10/2016 07:33 AM - duerst (Martin Dürst)

I personally think this syntax looks quite good. I wonder whether it should be limited to initialize or not.

Matt Lankford wrote:

I have looked at all 18 pages in this tracker and thought it "must" have been mentioned in here... but I don't see it... pardon me if I missed it...

Can you link some of these pages with "related issues"?

#3 - 07/10/2016 09:26 AM - phluid61 (Matthew Kerwin)

Here is some history of this proposal:

- [#5825 Sweet instance var assignment in the object initializer](#)
- [#8563 Instance variable arguments](#)
- [#12023 Allow ivars to be used as method arguments](#)

#4 - 07/10/2016 12:58 PM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #5825: Sweet instance var assignment in the object initializer added

#5 - 07/10/2016 12:59 PM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #8563: Instance variable arguments added

#6 - 07/10/2016 12:59 PM - nobu (Nobuyoshi Nakada)

- Is duplicate of Feature #12023: Allow ivars to be used as method arguments added

#7 - 07/10/2016 12:59 PM - nobu (Nobuyoshi Nakada)

- Description updated

#8 - 07/10/2016 10:27 PM - shevegen (Robert A. Heiler)

I like the idea behind it.

I am not sure about the syntax itself. Would this break existing code? What are the constraints how it is used?

#9 - 07/11/2016 03:51 PM - lankford (Matt Lankford)

Matthew Kerwin wrote:

Here is some history of this proposal:

- [#5825 Sweet instance var assignment in the object initializer](#)
- [#8563 Instance variable arguments](#)
- [#12023 Allow ivars to be used as method arguments](#)

You sir, are a much better "looker" than I ;)

well done!

#10 - 07/19/2016 08:33 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Feedback

Arguments are giving names to passed values, which is different from attribute (instance variables) initialization. I think they should be separated clearly. And for most of the cases, only initialize needs this kind of initialization.

My idea is <https://bugs.ruby-lang.org/issues/8563#note-3>

Matz.

#11 - 09/04/2016 04:40 PM - jwmittag (Jörg W Mittag)

By the way, just as a historical note: up to and including Ruby 1.8.7, you could actually do this:

```
define_method(:initialize) {|@thing_one, @thing_two|}
```

I have never actually seen this, though, which you can take as an indication of the usefulness of your suggestion.

#12 - 10/08/2016 02:38 AM - nobu (Nobuyoshi Nakada)

- Has duplicate Feature #12820: Shorter syntax for assigning a method argument to an instance variable added

#13 - 11/29/2017 06:26 AM - matz (Yukihiro Matsumoto)

- Status changed from Feedback to Rejected

My opinion has not been changed from <https://bugs.ruby-lang.org/issues/8563#note-3>

I am strongly against code like

```
def initialize(@foo, @bar)
```

end

I don't care about `define_attr_initialize` not being "the solution".

Matz.