

Ruby master - Feature #10883

Passing a block to itself

02/21/2015 09:41 PM - Gondolin (Damien Robert)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
In the discussion of itself some people proposed that passing a block to itself could return the value of the block:	
<pre>def itself if block_given? yield self else self end end</pre>	
It would be very usefull in method chains	
<pre>#this would allow an_array.foo.bar.itself { x x[1..x.length-1]}.baz #which flows better than a=an_array.foo.bar a[1..a.length-1].baz</pre>	

History

#1 - 02/22/2015 11:03 PM - phluid61 (Matthew Kerwin)

Not exactly a duplicate, but also see the discussion at [#6373](#)

#2 - 02/23/2015 04:08 PM - recursive-madman (Recursive Madman)

Note that for your example that wouldn't be needed:

```
a = an_array.foo.bar
a[1..a.length-1].baz
```

is equivalent to

```
an_array.foo.bar[1..-1].baz
```

#3 - 03/06/2015 08:06 PM - shan (Shannon Skipper)

I think having a block form of #itself would be great. Here's a highly contrived example:

```
(("123".to_i - 3).to_s.reverse.to_i * 2).to_s
#=> "42"
```

Instead of nested parens, the following reads left to right:

```
"123".to_i.itself { |n| n - 3 }.to_s.reverse.to_i.itself { |n| n * 2 }.to_s
#=> "42"
```

It also allows a variety of other niceties:

```
["Hello", "Word"].itself { |greeting, subject| "#{greeting}... #{subject}!!" }
#=> "Hello... Word!!"
```

So like #tap except it returns itself instead of just tapping in.

#4 - 03/07/2015 02:14 AM - nobu (Nobuyoshi Nakada)

Why itself doesn't return itself?

It seems very confusing.

#5 - 03/19/2015 08:32 PM - baweaver (Brandon Weaver)

Nobuyoshi Nakada wrote:

Why itself doesn't return itself?
It seems very confusing.

Why can't we just tie into the current verbiage?

```
# Yes, contrived
1.yield { |v| v + 5 } # => 6

def defaults_to(default)
  -> v { v || default }
end

nil.yield(&defaults_to(5)).times do # ....
```

#6 - 10/11/2015 04:37 AM - avit (Andrew Vit)

Duplicate of: [#6373](#), [#10095](#)