Ruby master - Feature #14821
aditional arguments to yield_self
06/04/2018 07:37 AM - DarkWiiPlayer (Dennis Fischer)

Status: Rejected
Priority: Normal
Assignee:
Target version:

Description
Currently, yield_self only accepts a block as its single argument. I propose that it should also accept a series of additional arguments that get passed to the block after self

This is what I mean:

```ruby
def yield_self(*args, &block)
  block.call(self, *args)
end
```

which would allow writing code like this

```ruby
some_object.yield_self(:name, :surname, &sort_stuff).do_more_stuff()
```

instead of code like this

```ruby
some_object.yield_self{|x| sort_stuff.call(x, :name, :surname).do_more_stuff()}
```

This 1) is shorter 2) expresses intention rather than implementation.

History
#1 - 06/04/2018 07:43 AM - duerst (Martin Dürst)
- Backport deleted (2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN)
- Tracker changed from Bug to Feature

This is clearly a feature, so I have changed it from bug to feature.

The usage examples are still a bit abstract, some more concrete examples would be desirable.

#2 - 06/04/2018 09:28 AM - matz (Yukihiro Matsumoto)
- Status changed from Open to Rejected

#3 - 06/04/2018 09:37 AM - matz (Yukihiro Matsumoto)
Your proposal makes the specific case simpler like your example, but in general, it makes the intention of yield_self (or then) less clear.

Matz.

p.s.
I removed the previous comment because I made mistake.