

Ruby trunk - Feature #13601

Remove yield_self from Ruby 2.5 (already implemented)

05/26/2017 07:35 PM - danielpclark (Daniel P. Clark)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>Issue https://bugs.ruby-lang.org/issues/6721 for adding yield_self is unnecessary as BasicObject#instance_eval will do the exact same thing.</p> <pre>class BasicObject alias :yield_self :instance_eval end 2.yield_self { x x*x } # => 4</pre> <p>I've tried all the examples of its given use cases and BasicObject#instance_eval will do them all.</p> <p>In issue #6721 nobu said he didn't like the name yield_self. I'm in agreement about the name. Since BasicObject#instance_eval will already pipe in self and return the value of the block yield_self is not necessary.</p>	

History

#1 - 05/26/2017 08:33 PM - shan (Shannon Skipper)

Note there are differences, for example

```
42.instance_eval { abs2 } #=> 1764
```

won't work with Kernel#yield_self and

```
42.yield_self #=> #<Enumerator: 42:yield_self>
```

won't work with BasicObject#instance_eval, etc.

#2 - 05/26/2017 09:37 PM - zverok (Victor Shepelev)

instance_eval changes scope, while yield_self does not.

```
class A
  def foo
    1
  end

  def yield_self
    yield self
  end
end
```

```
def self.foo
  2
end
```

```
p A.new.instance_eval { self.foo * 3 } # => 3
p A.new.yield_self { self.foo * 3 } # => 6
```

An example is synthetic, yet the difference is important.
...and was discussed multiple times on the road to yield_self, to be honest.

#3 - 05/27/2017 12:23 AM - danielpclark (Daniel P. Clark)

You both have provided better examples as to the differences. Also finding out yield_self returns an Enumerator was a new one to me. I guess this feature request can be closed.

#4 - 05/27/2017 01:40 AM - shyuhei (Shyouhei Urabe)

- Status changed from Open to Closed

Closing as per request.