Ruby trunk - Feature #6126

Introduce yes/no constants aliases for true/false

03/10/2012 05:29 AM - homakov (Egor Homakov)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

I propose to have predefined constants of TrueClass and FalseClass - yes/no accordingly.
Benefits:

1. 'truefalse'.size - 'yesno'.size = 4 (!)
2. much more understandable and reasonable words. Well known by everybody on this planet.
3. would be nice feature to introduce. Object#no? so we could use. puts 'horay!' unless will_you_marry_me.no? but it is very small thing, !will_you_marry_me behaves the same.. nevermind if it doesn't look useful
4. in further releases of ruby we could use them by default and keep true/false only for compatibility e.g.: [2] pry(main)> true => yes

to discuss:

1. how to manage with true-false constants
2. do you like it?

P.S. I hope it is not crazy proposal for Ruby 4.9.3. I believe that ruby is agile enough. thoughts?

Related issues:

Has duplicate Ruby trunk - Feature #9107: Introduce YES and NO as aliases of ...	**Rejected**	**11/13/2013**

History

#1 - 03/10/2012 05:45 AM - burnhype (Sebastian Sito)

Considering your poor english I'm able to belive that you are not joking.

#2 - 03/10/2012 05:57 AM - homakov (Egor Homakov)

@sebastian will you forgive me if I ask you to align topic?

#3 - 03/10/2012 06:21 AM - homakov (Egor Homakov)

References:
http://www.otierney.net/objective-c.html

it's common practice to define constants YES Yes in all languages and it seems widely used.

Plain and straight syntax. It's all about ruby, isn't it? :
puts -> p
null -> nil
.to_string -> to_s

#4 - 03/10/2012 06:42 AM - burnhype (Sebastian Sito)

That wasn't offense at all. There was a hype on Twitter about your Borat-like sentences (which were treated positive and funny) so this was a little pinch to them.

Anyway, to say something on topic. I don't like the idea.

"much more understandable and reasonable words. Well known by everybody on this planet"
Programming language is for programmers and true/false are very well known among them. No need for new aliases IMO.

#5 - 03/10/2012 07:00 AM - homakov (Egor Homakov)

@sebastian Yes they are for programmers. And there were no need for C/C++/... because assembler commands were very well known among programmers. Is it fair enough?
PLs need to evolve. And move on with standards too.

I don't mind jokes on me but I didn't get that one, nevermind then.

#6 - 03/10/2012 08:32 AM - greyblake (Sergey Potapov)

New aliases lead to more code mess and mixing yes/no and true/false styles. Newbies would get only more confused seeing things like this.

#7 - 03/10/2012 08:39 AM - homakov (Egor Homakov)

It is good point but would you consider it like if/unless? they do same job but used in proper places.
p 1 if yes
p 2 unless no

Newbies learn rapidly. And 'yes/no' thing will be the first thing they will like in ruby(in the same time obscure 'true/false' is what other languages can suggest)

the issue seems even newbie-friendly and it is benefit too.

ps just, isn't this awesome?
user_authenticated = yes

#8 - 03/10/2012 08:53 AM - ujihisa (Tatsuhiro Ujihisa)

I don't agree with the suggestion.

yes = either true or false

"Did you eat sushi?": "Yes I did" = true
"Didn't you eat sushi?": "Yes I did" = false

Giving Hai/Iie as true/false makes more sense in this case. Yes/no are ambiguous.

#9 - 03/10/2012 09:20 AM - homakov (Egor Homakov)

ujihisa (Tatsuhiro Ujihisa) hrm.. Are you trying to inject english grammar in the logic? And, What is wrong with
"Didn't you eat sushi?": "Yes I did" = false
There is an ambiguity if you only use x.didnt_you_do_y? instead of x.did_you_do_y? and that is a very rare case.
But, honestly, I see some ambiguous things. In those cases you should better use true/false if would like to. Anyway it is just a habit IMO

#10 - 03/10/2012 01:20 PM - nobu (Nobuyoshi Nakada)

Egor Homakov wrote:

> 1. do you like it?

false

#11 - 03/10/2012 02:00 PM - daz (Dave B)

```
#- You can please yourself in your own programs by
#-  adding a couple of lines at the top.

#====================#
class Object
def yes?; self end
def no?; !self end
end
#====================#

def test(proposal_accepted)
puts
puts  proposal_accepted       ? 'Hooray!' : 'Booooo!'
puts  proposal_accepted.yes?  ? 'Hooray!' : 'Booooo!'
puts !proposal_accepted       ? 'Hooray!' : 'Booooo!'
puts  proposal_accepted.no?   ? 'Hooray!' : 'Booooo!'
end

test true

#=> Hooray!
#=> Hooray!
#=> Booooo!
#=> Booooo!

test false

#=> Booooo!
#=> Booooo!
```

#=> Hooray!
#=> Hooray!

#- I wouldn't recommend it; your suggestion
#- seems to require extra typing:

p '.yes?'.size - ''.size   #=> 5
p '.no?'.size - '!'.size   #=> 3

#- daz

#12 - 03/10/2012 03:37 PM - homakov (Egor Homakov)

@Dave wow, you win.
I just was jealous to this obj-c thing.

Anybody pls close the issue

#13 - 03/10/2012 09:40 PM - ujihisa (Tatsuhiro Ujihisa)

- Status changed from Open to Rejected

#14 - 03/11/2012 06:23 AM - mame (Yusuke Endoh)

Off topic.

I really hate the names "true" and "false" for another reason:
their length should be equal. For the same reason, I also hate
yes/no, in/out, width/height, left/right, etc.

I want to see these BUGS were fixed in English 2.0.

--
Yusuke Endoh mame@tsg.ne.jp

#15 - 03/11/2012 10:23 AM - nobu (Nobuyoshi Nakada)

Hi,

(12/03/11 6:03), Yusuke Endoh wrote:

> I really hate the names "true" and "false" for another reason:
> their length should be equal. For the same reason, I also hate
> yes/no, in/out, width/height, left/right, etc.

Third party's issue. Report to the upstream.

You can use :true and false instead, in Ruby at least.

--
Nobu Nakada