

Ruby master - Feature #13860

A proposal for a new, simpler class-method addition to Time - for the time being calling it Time.date() but another name is fine; and the ability to omit '%' tokens as arguments to it

09/02/2017 10:10 AM - shevegen (Robert A. Heiler)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
Description	
Hello ruby-folks,	
On reddit someone linked in this:	
http://www.w3school.in/w3schools/php-tutorial/php-date-time	
Just in the event that this may not be available lateron, or nobody wants to look at external links, I'll very briefly note that this is about PHP; and how PHP is handling time - at the least to a degree.	
I am not interested in most of the examples from there, except for the first one, which is this:	
<pre><?php echo date("Y-m-d"); ?></pre>	
Output: 2012-11-25	
Specifically, the date("Y-m-d") notation.	
I believe that this is quite easy to read for many people - Year, month, day. Seems simple to me.	
In Ruby, there are or may be many other ways - the two that I tend to use the most, all come via Time.now.strftime() :	
Time.now.strftime "%d.%m.%Y" # My favourite dd.mm.yyyy notation Time.now.strftime "%H:%M:%S" # And to complement, hh.mm.ss notation	
To me it seemed to be the most flexible; I could also of course use both together such as "%d.%m.%Y-%H:%M:%S". The primary reason why I tend to keep them different is that I often have to return the current date, but not always the hour-minutes-seconds notation. So I need the first variant more frequently than the second variant.	
However had, although the above does not bother me at all really, as I usually use custom helper-methods, such as return_date or something to point to these two invocations, I do have to say that the PHP date() variant may actually be simpler and shorter to read - both from the arguments that are passed to it, and also from the API call, than the ruby variant.	
Which brings me to the issue request here:	
<ul style="list-style-type: none">• Would it be possible for ruby to also use a short or shorter variant, similar to PHP? It can be the same API, or a different name; I think that this is mostly secondary what the method name ultimately is, compared to the question over whether ruby would provide a simpler API here or not. And also the possibility to omit the '%' of course. I think that in the above examples, the '%' should not really be necessary if we already assigned a special meaning to the individual characters - at the least for a new method. It's ok to retain Time.now.strftime() behaviour how it is.	
I do not mean to just copy/paste the PHP solution per se, necessarily,	

but more generally, these two requests for the issue report here:

(a) Allow for arguments to have the '%' be omitted.

This may be for another method, so I do NOT suggest this for `strftime()` to be the case, `strftime()` can remain as it is just fine; I just would like to also see a shorter variant that is possible to use in ruby, just like the PHP variant. This method could automatically prepend a '%' when it "sees" something such as "Y-m-d". So the latter would then be similar to "%Y-%m-%d". I hope I could explain what I mean there. :)

This may be fully equivalent to exactly this API call:

```
Time.now.strftime "%Y-%m-%d"
```

I am currently unaware of time/date variants in Ruby that allow for this or a similar use of arguments.

(b) Allow for simpler invocation examples, such as perhaps `Time.date()` ...

I am not sure if `date()` is a good method name there, but if people have a better idea, I am all for it. Giving things a proper name is hard.

I would suggest a short name though, as otherwise there is no real advantage compared to `Time.now.strftime()`. So perhaps `Time.date()` would be ok nonetheless?

Ultimately, this would allow us to have a very short variant available as well, similar as to what PHP would allow for. So, for example, the PHP variant here:

```
echo date("Y-m-d");
```

would "translate" to this variant in Ruby, then:

```
puts Time.date "Y-m-d"
```

Almost the same amount of characters used. :D

For anyone who wants to have a look at **Time** and **Date** in Ruby before chiming in, here are the links to the documentation of these two:

```
https://ruby-doc.org/core/Time.html  
https://ruby-doc.org/stdlib/libdoc/date/rdoc/Date.html
```

On a side note, I also wondered whether one could use `Time.now("Y-m-d")` but I think that everything else aside, this might confuse people more than do good - so a new method may be better.

I have also noticed `Date.today()` ... so I thought about using it, such as via `Date.today 'H:M:S'` - but I think that it may be better to have it reside within `Time`, rather than `Date`. `Date.today()` presently only accepts Floats rather than Strings as input, so it is probably also easier to add a separate, new and different method rather than change an existing method.

I only had a brief look whether existing methods could be adapted but I think - a new method may be better and easier altogether.

Internally, the proposal for `Time.date()` would then just use `Time.now.strftime()` and also insert '%' characters accordingly, unless these characters already exist there. So people could use both the '%' variant but also omit it and use the non '%' variant, simply for more flexibility. Even though the variant with fewer characters would be the main point why this issue request has been written here. But if people want to use '%' even there, that is fine - I do not intend to reduce flexibility, only to extend flexibility. :)

Sorry for this lengthy request here - please do feel free to close it at

any time if you feel it is not a good addition/idea/fit.

History

#1 - 09/04/2017 01:45 AM - shyouhei (Shyouhei Urabe)

Is there any reasons this has to be done in-core, instead of a gem? It seems everything needed to implement such thing as a gem is available... or do we miss something?