

Ruby master - Feature #12173

``Time#till_now``

03/14/2016 04:57 PM - sawa (Tsuyoshi Sawada)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
Description It is very frequent to have a time instance: <pre>t = Time.now</pre> and then after some operations, do: <pre>Time.now - t</pre> I propose <code>Time#till_now</code> , which is equivalent to: <pre>class Time def till_now; self.class.now - self end end</pre> and similar methods can perhaps be defined on <code>Date</code> and <code>DateTime</code> classes as well. Another candidate for the method name is <code>until_now</code> . Then we can do: <pre>t = Time.now # some heavy operation puts "It took #{t.till_now} secs."</pre>	
Related issues:	
Related to CommonRuby - Feature #8640: Add <code>Time#elapsed</code> to return nanoseconds...	Open 07/16/2013
Related to CommonRuby - Feature #8096: introduce <code>Time.current_timestamp</code>	Feedback 03/15/2013

History

#1 - 03/14/2016 07:48 PM - sawa (Tsuyoshi Sawada)

This can also be used to wait until a preset time:

```
target_time = Time.new(2017, 1, 1, 0, 0, 0)
sleep(-target_time.till_now)
```

Or perhaps, the opposite (multiplied by -1) might be more intuitive:

```
class Time
  def from_now; self - self.class.now end
end
```

so that we can do:

```
target_time = Time.new(2017, 1, 1, 0, 0, 0)
sleep(target_time.from_now)
```

or

```
t = Time.now
# some heavy operation
puts "It took #{-t.from_now} secs."
```

#2 - 03/14/2016 08:41 PM - phluid61 (Matthew Kerwin)

I would use both of these if they existed, and have defined my own equivalent versions in past projects.

Please note that "till_now" is not correct spelling; it should be "until_now" or "til_now"

#3 - 03/14/2016 08:55 PM - phluid61 (Matthew Kerwin)

This is related to (but different from) [#8640](#), [#8096](#)

#4 - 03/14/2016 08:57 PM - kosaki (Motohiro KOSAKI)

- Related to Feature #8640: Add Time#elapsed to return nanoseconds since creation added

#5 - 03/14/2016 08:57 PM - kosaki (Motohiro KOSAKI)

- Related to Feature #8096: introduce Time.current_timestamp added

#6 - 03/14/2016 10:24 PM - sawa (Tsuyoshi Sawada)

Matthew Kerwin wrote:

Please note that "till_now" is not correct spelling; it should be "until_now" or "til_now"

till and *until* are the correct spelling. Perhaps you are confusing it with *'til* (which is pretty much informal).

#7 - 03/15/2016 12:10 AM - phluid61 (Matthew Kerwin)

Tsuyoshi Sawada wrote:

Matthew Kerwin wrote:

Please note that "till_now" is not correct spelling; it should be "until_now" or "til_now"

till and *until* are the correct spelling. Perhaps you are confusing it with *'til* (which is pretty much informal).

till means cash register, or to plough soil. *'til* is an abbreviation of *until*, which isn't allowed in Ruby syntax so I dropped the apostrophe.

#8 - 03/15/2016 12:32 AM - sawa (Tsuyoshi Sawada)

Matthew Kerwin wrote:

till means cash register, or to plough soil.

That is a homonym (as well as a homograph) of the word in question.

#9 - 03/15/2016 01:06 AM - phluid61 (Matthew Kerwin)

Tsuyoshi Sawada wrote:

Matthew Kerwin wrote:

till means cash register, or to plough soil.

That is a homonym (as well as a homograph) of the word in question.

I've just done some research and you're right, I retract my opposition.

#10 - 03/15/2016 01:41 AM - kosaki (Motohiro KOSAKI)

2016-03-14 21:06 GMT-04:00 matthew@kerwin.net.au:

Issue [#12173](#) has been updated by Matthew Kerwin.

Tsuyoshi Sawada wrote:

Matthew Kerwin wrote:

till means cash register, or to plough soil.

That is a homonym (as well as a homograph) of the word in question.

I've just done some research and you're right, I retract my opposition.

dictionary.com includes this usage:

<http://www.dictionary.com/browse/till?s=t>

but freedictionary.org doesn't.

<http://freedictionary.org/?Query=till&button=Search>

So, I'm afraid Matthew is not alone and I'd suggest avoiding 'till'.

#11 - 03/15/2016 04:54 AM - avit (Andrew Vit)

Tsuyoshi Sawada wrote:

That is a homonym (as well as a homograph) of the word in question.

I also think "till" is the traditional spelling ("till the cows come home") and I've seen the apostrophe 'til used more frequently (USA/Canada) in modern writing but I still prefer "until" anyway.

For reference, ActiveSupport also defines similar methods for numeric classes:

<http://api.rubyonrails.org/classes/ActiveSupport/Duration.html#method-i-until>

(These are the inverse receiver/types of the proposed method.)

#12 - 03/15/2016 08:21 AM - Eregon (Benoit Daloze)

Tsuyoshi Sawada wrote:

It is very frequent to have a time instance:

```
t = Time.now
```

and then after some operations, do:

```
Time.now - t
```

This seems fairly related to [#8640](#).

Just a note: the `Time.now - t` is typically not what you want if you want to benchmark some piece of code.

A monotonic clock like `Process.clock_gettime(Process::CLOCK_MONOTONIC)` should be used instead (I know, it's a bit lengthy).

Or well if you can require 'benchmark' then it's just `Benchmark.realtime { ... }`.

#13 - 03/15/2016 11:23 PM - elifoster (Eli Clemente Gordillo Foster)

I think it would be best for these `from_now` and `until_now` methods to return a Time object rather than a Float.

#14 - 03/16/2016 10:19 AM - shevegen (Robert A. Heiler)

I think it would be best for these `from_now` and `until_now` methods to return a Time object rather than a Float.

I don't know the current way - is it custom for Ruby to return a time object for time-related methods? I mean it would not be a huge issue, one could call `.seconds` to get the seconds.

Anyway +1 on the idea of the initial suggestion above. I have no particular preferences on the name itself, `till_now`, `until_now` but we should all settle for something in the end.