Ruby master - Bug #16438
Check warning messages for Ruby 2.7
12/20/2019 01:50 AM - ko1 (Koichi Sasada)

Status: Closed
Priority: Normal
Assignee: matz (Yukihiro Matsumoto)
Target version: 2.7
ruby -v:
Backport: 2.5: UNKNOWN, 2.6: UNKNOWN

Description
Please check Ruby 2.7 new introduced warning messages.

- deprecation warning

```ruby
def foo(**kw)
end

foo({}) #=> test.rb:4: warning: The last argument is used as the keyword parameter
    #    test.rb:1: warning: for `foo' defined here; maybe ** should be added to the call?
    # default: on

_1 = 2
#=>
# test.rb:1: warning: `_1' is used as numbered parameter
# default: on

def _1; end
#=> test.rb:1: warning: `_1' is used as numbered parameter
# default: on
# added at Today!!

def foo
  proc.call #=> 1
end

foo(p 1)
#=>
# test.rb:2: warning: Capturing the given block using Kernel#proc is deprecated; use `&block' instead
# 1
# default: on

$; = //
#=> test.rb:1: warning: non-nil $; will be deprecated

$, = '
#=> test.rb:2: warning: non-nil $, will be deprecated
# default: on

  def foo
    class << Object.new
      yield
    end
    end
    foo { p :ok } #=> warning: `yield' in class syntax will not be supported from Ruby 3.0.
    # default: on

require 'open-uri'
```
open('http://atdot.net')
#=> test.rb:2: warning: calling URI.open via Kernel#open is deprecated, call URI.open directly or use URI.open

# default: on
  • experimental feature

$ ./miniruby -e 'case 1; in 1; end'
-e:1: warning: Pattern matching is experimental, and the behavior may change in future versions of Ruby!
# default: on

Associated revisions
Revision a8bddd3a - 12/20/2019 05:34 AM - nobu (Nobuyoshi Nakada)
Refined the warning message for numbered-parameter like variables
[Bug #16438]

Revision 7a9425e - 12/20/2019 05:58 AM - nobu (Nobuyoshi Nakada)
Refined the warning message for numbered-parameter like variables
[Bug #16438]

Revision 2b2030f2 - 12/20/2019 06:09 AM - nobu (Nobuyoshi Nakada)
Refined the warning message for $, and $;
[Bug #16438]

History
#1 - 12/20/2019 02:00 AM - osyo (manga osyo)

    def _1; end
    #=> test.rb:1: warning: `_1' is used as numbered parameter

Warn by parameter name instead of method name.
Became warning.

    # warning: `_1' is used as numbered parameter
    def _1; end
    # warning: `_1' is used as numbered parameter
    def hoge(_1); end

#2 - 12/20/2019 02:50 AM - ko1 (Koichi Sasada)

    $SAFE = 1
    #=> t.rb:1: warning: $SAFE will become a normal global variable in Ruby 3.0

    no warning

    pls check current master.

#3 - 12/20/2019 02:56 AM - osyo (manga osyo)

    pls check current master.

Sorry, became warning.
Refined the warning message for numbered-parameter like variables

[Bug #16438]

#5 - 12/20/2019 06:00 AM - ko1 (Koichi Sasada)

At developer's meeting, the followings are proposed. They will be merged into Ruby 2.7 RC2.

def foo(**kw)
end

foo({})

# current:
#=> test.rb:4: warning: The last argument is used as the keyword parameter
# test.rb:1: warning: for `foo' defined here; maybe ** should be added to the call?

# modified (conclusion):
#=> test.rb:4: warning: The last argument is used as keyword parameters; maybe ** should be added to the call
# test.rb:1: warning: The called method `foo' is defined here

#=> test.rb:1: warning: `_1' is used as numbered parameter
->
#=> t.rb:1: warning: `_1' is reserved for numbered parameter; consider another name

#=> t.rb:1: warning: non-nil $; will be deprecated
=>
#=> t.rb:1: warning: `$;' is deprecated

#6 - 01/02/2020 03:30 PM - puchuu (Andrew Aladjev)

Hello. I can't understand why setting OFS to not nil value was deprecated. I want to simulate print method behaviour and I am using $OUTPUT_FIELD_SEPARATOR between objects. Than I want to test this method, so I am settings OFS to not nil value in tests. Ruby 2.7.0 throws deprecated warning.

#7 - 01/02/2020 05:23 PM - puchuu (Andrew Aladjev)

I see the only one way to fix this deprecation. Change write(*objects) signature to print(*objects, field_separator: $OUTPUT_FIELD_SEPARATOR, record_separator: $OUTPUT_RECORD_SEPARATOR) and test print by passing separators directly. Will ruby change print signature in the same way?