Ruby master - Feature #16128

Would it be possible for ruby to warn about case/when menu options separated by a trailing, but accidental ','?

08/25/2019 03:52 PM - shevegen (Robert A. Heiler)

Status:	Open	
Priority:	Normal	
Assignee:		
Target version:		

Description

I was not sure whether the following behaviour is a bug or not, so I filed this under "feature", mostly because this may change existing behaviour; and even if I think the current behaviour in this context may not make a lot of sense, perhaps there are caveats; or it may be too insignificant to want to change.

Anyway, without further ado, I will next show the ruby code that can be used to reproduce the issue/behaviour that I refer to:

```
class Foo

  def initialize
    menu(:random_colour)
  end

  def menu(i)
    case i.to_s
    when /^-?-?random(-|_)?colou?r$/i,
         /rcolour$/i,
         'RCOL',
      print_foobar
    end
  end

  def print_foobar
    print 'foobar'
  end

end

Foo.new
```

To those who may not immediately see the problem - it is the last ',' character in the case/when menu, right after the string 'RCOL'.

The above snippet was part of a much larger class/codebase, so I narrowed it down to this smaller example.

If you run this code, you will see no output. The reason is due to the ','.

If you remove the ',', then you get the desired output - the invocation of the method called print_foobar() which will print 'foobar'.

I discovered this strange behaviour by accident in a much larger case/when menu (my case/when menu interfaces can be excessively long, I admit this).

Sometimes I re-arrange the case menu, and then I may forget the ',' there, so I paste the ',' with the line, which sometimes leads to above situation, which tends to confuse me. I am quite used to this at this point, so discovering the problem does not take me long - but I was wondering whether this current behaviour is useful for anything?

Because if not then perhaps there could be a warning by ruby, possibly by the did-you-mean gem, or by ruby directly even without the did-you-mean

gem (although I think this may fit the did-you-mean gem). I have, however had, decided to first report this to ruby core anyway, because perhaps I am missing something obvious that may explain the behaviour. Or perhaps it is difficult to distuingish what the next lines should be. But I assume that the ruby parser assumes another case/when statement here, after the ',', so does not treat the above as an error or problematic behaviour.

Perhaps the above can still be reported based on additional information, such as correct indent level? I indent uniformly and consistently, so the above could provide additional cues what the ruby user at hand may have wanted to see - as in the example above, I don't think the trailing ',' is useful; it was just a typo. I am not sure how easy it is to distinguish this case between cases where the user wanted to have a ','.

I am not sure how easy it would be to change the behaviour of ruby in this regard, or if it is wanted, or if it takes too much time, but I thought it is better to report it anyway - others can give their opinion in this case. Thanks!

History

#1 - 08/26/2019 03:37 AM - nobu (Nobuyoshi Nakada)

- Description updated

shevegen (Robert A. Heiler) wrote:

> Perhaps the above can still be reported based on additional information, such as correct indent level? I indent uniformly and consistently, so the above could provide additional cues what the ruby user at hand may have wanted to see - as in the example above, I don't think the trailing ',' is useful; it was just a typo. I am not sure how easy it is to distinguish this case between cases where the user wanted to have a ','.

There is no way to distinguish these cases syntactically.
Only what I think of is the indent level, too.

#2 - 08/26/2019 06:23 AM - sawa (Tsuyoshi Sawada)

In fact, even indentation level in this particular code does not seem to tell that it was a mistake. The print_foobar line is aligned with the other conditions. I would take it that they all belong to the condition.

However, you can tell that this code is wrong from the fact that the body for the conditioned case is empty, and there is no other case that does anything. Perhaps you can warn when all conditional branches in a case construction have an empty body.

#3 - 08/26/2019 07:36 AM - Hanmac (Hans Mackowiak)

the problem is that the case body can be empty, that doesn't mean the code is wrong.

and like sawa said, it probably needs to check all other braches too, which i don't know is possible there

#4 - 08/26/2019 09:56 AM - nobu (Nobuyoshi Nakada)

- Description updated

Comparing indents is easy in very simple cases, but it will harder in complex cases, for instance, after a multiple line expression.
And I suspect this can be found by editors auto-indent.