When I use pattern match with instance variables, I get an error message like "expecting local variable or method"

```ruby
$n = 1
case 1
  in $n
    # ...
end
#=> syntax error, unexpected instance variable, expecting local variable or method (SyntaxError)
```

However using method is not allowed, either.

```ruby
def n = 1
case 1
  in ^n
    # ...
end
#=> n: no such local variable (SyntaxError)
```

I think the message "expecting local variable or method" is confusing and should be fixed like "expecting variable."

Associated revisions

Revision fa87172e - 07/15/2021 04:56 PM - jeremyevans (Jeremy Evans)
Add pattern matching pin support for instance/class/global variables

Pin matching for local variables and constants is already supported, and it is fairly simple to add support for these variable types.

Note that pin matching for method calls is still not supported without wrapping in parentheses (pin expressions). I think that's for the best as method calls are far more complex (arguments/blocks).

Implements [Feature #17724]

Revision 25689024 - 07/16/2021 02:26 AM - nobu (Nobuyoshi Nakada)
Added code fence to the example in [Feature #17724] [ci skip]

History

#1 - 03/15/2021 02:57 AM - marcandre (Marc-Andre Lafortune)
Agreed. Ideally instance variable would be allowed, no?

#2 - 04/11/2021 07:08 AM - ktsj (Kazuki Tsujimoto)
- Assignee set to ktsj (Kazuki Tsujimoto)
- Status changed from Open to Assigned

#3 - 05/14/2021 12:15 AM - jeremyevans0 (Jeremy Evans)
I've added a pull request to directly support instance/class/global variables with the pin operator: [https://github.com/ruby/ruby/pull/4502](https://github.com/ruby/ruby/pull/4502)

You could already use all of those variables with the pin expression support by surrounding them in parentheses. However, since local variables and constants are supported, I think it makes sense to also support instance/class/global variables directly. I don't think it makes sense to support
methods, as doing so is much more complex (arguments/blocks). Users can use the pin expression support for methods.

My approach to fixing this bug is really introducing a new feature so the bug reported (confusing error message) isn't a problem anymore. Since this changes the pattern matching syntax, it probably should be approved by both ktsj (Kazuki Tsujimoto) and matz (Yukihiro Matsumoto).

#4 - 05/16/2021 08:12 AM - ktsj (Kazuki Tsujimoto)
- Backport deleted (2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN)
- ruby -v deleted (ruby 3.0.0p0 (2020-12-25 revision 95aff21468) [x86_64-darwin20])
- Subject changed from Pattern matching error message is confusing to Make the pin operator support instance/class/global variables
- Tracker changed from Bug to Feature

I think it makes sense to also support instance/class/global variables directly.

I agree.

What do you think, matz (Yukihiro Matsumoto)?

#5 - 07/15/2021 05:42 AM - matz (Yukihiro Matsumoto)
Sounds reasonable. Accepted.

Matz.

#6 - 07/15/2021 04:56 PM - jeremyevans (Jeremy Evans)
- Status changed from Assigned to Closed

Applied in changeset git|fa87f72e1e84e2b55516be188f00434a683b924c.

Add pattern matching pin support for instance/class/global variables

Pin matching for local variables and constants is already supported, and it is fairly simple to add support for these variable types.

Note that pin matching for method calls is still not supported without wrapping in parentheses (pin expressions). I think that's for the best as method calls are far more complex (arguments/blocks).

Implements [Feature #17724]