Ruby master - Bug #16293
Numbered parameter confirmation in Ruby 2.7
11/05/2019 12:14 PM - osyo (manga osyo)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v:</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: UNKNOWN, 2.6: UNKNOWN</td>
</tr>
</tbody>
</table>

Description

Overview

I want to make a final check on the behavior of Numbered parameter (No warning or Warning or Error).
There is a difference in DevelopersMeeting20190829Japan logs and behavior.

Ruby version

```ruby
p RUBY_VERSION
# => "2.7.0"
p RUBY_DESCRIPTION
# => "ruby 2.7.0dev (2019-11-02T06:32:49Z trunk 772b0613c5) [x86_64-linux]"
p RUBY_RELEASE_DATE
# => "2019-11-02"
p RUBY_REVISION
# => "772b0613c583773cd2eda23bce8275926a351e79"
```

DevelopersMeeting20190829Japan logs

- local variables (parameters and assigned variable) → force warning (Don’t use) on Ruby 2.7 and syntax error on Ruby 3.
- method invocation
  - `vcall: x = _0` outside block. → force warning on Ruby 2.7 and syntax error on Ruby 3.
  - `vcall: 1.times{ _0 }` → block parameter (incompatibility)
  - `vcall: 1.times{|i| _0 }` → force warning on Ruby 2.7 and syntax error on Ruby 3.
- method invocation (`x = _0(); x = foo._0`) → no warning
- method name (def _0(); end) → no warning

Proposal

```ruby
def _1
# expected: warning: `_1` is used as numbered parameter
# actual: No warning
# reason: Because "`x = _0 # expect _0()` outside block. → force warning on Ruby 2.7 and syntax error on Ruby 3" is written in the log
x = _1
# expected: warning: `_1` is used as numbered parameter
# and call to `_1()`
# actual: Error: numbered parameter outside block (SyntaxError)
# reason: Because hoge() method is called with `eval("hoge")`
eval("_1")
def _1
# expected: warning: `_1` is used as numbered parameter
# and call to `_1()`
# actual: Error: ordinary parameter is defined
# reason: Because "`1.times{|i| _0 }` → force warning on Ruby 2.7 and syntax error on Ruby 3." is written in the log
proc { |i| _1 }.call 42
```
# expected: warning: `_1' is used as numbered parameter
# actual: No warning
# reason: Because define local variable `_1' is warning

def hgoe(_1)
end

proc {
    # Warning
    _1 = 42
}

proc {
    # expected: warning: `_1' is used as numbered parameter
    # actual: No warning
    # reason: Because define local variable `_1' is warning
    _1 = 42
}

:MEMO: Other behavior

- Other current behavior: [https://gist.github.com/osyo-manga/f332ba1f31dbc3a437acd4d86d7986dc](https://gist.github.com/osyo-manga/f332ba1f31dbc3a437acd4d86d7986dc)
- Is there any other syntax to change No warning Warning Error?
- Considering compatibility, make it Warning instead of Error?

Are there other edge cases for Numbered parameter?

Thank you :)

Associated revisions

Revision 82e840ad - 11/06/2019 03:44 AM - nobu (Nobuyoshi Nakada)
Numbered parameter cannot appear outside block now [Bug #16293]

Revision 2fd46554 - 11/06/2019 03:45 AM - nobu (Nobuyoshi Nakada)
Numbered parameter is an ID_LOCAL now [Bug #16293]

History

#1 - 11/05/2019 12:56 PM - habt (Hiroshi SHIBATA)
I couldn't understand your request. Please summarize actual and expect behavior with your request.

#2 - 11/05/2019 01:05 PM - mame (Yusuke Endoh)
It is difficult for me to analyze this ticket, too. Could you write your expectations and the actual results?

# expected: warning: `_1' is used as numbered parameter
# actual: no warning

def hgoe(_1)
end

    def _1; end

# expected: warning: `_1' is used as numbered parameter
# actual: no warning

    x = _1

It would be good to write the reason why you expect so. But firstly I need to understand what you expect.

#3 - 11/05/2019 04:18 PM - shevegen (Robert A. Heiler)
The present behaviour:

    def _1; end

seems correct and osyo's example of expecting a warning instead seems incorrect to me, unless he meant it in in some other context. (It is indeed difficult to break the suggestion down since it is so long. On a semi-related note, I know understand the ruby core team when it encourages ruby users to explain what they want to see discussed at a developer meeting, since they have to understand it first. I picture a round of question marks on the
def _1; end

# expected: warning: `_1` is used as numbered parameter
# actual: No warning
# reason: Because "x = _0 # expect _0()" outside block. → force warning on Ruby 2.7 and syntax error on Ruby 3" is written in the log
x = _1

The "outside block" warning is shown at assignment to a local variable looks like a numbered parameter. There is no assignment.

# expected: warning: `_1` is used as numbered parameter
# and call to _1()
# actual: Error: numbered parameter outside block (SyntaxError)
# reason: Because hoge() method is called with `eval("hoge")`

This is a bug.
As numbered parameter looks like an ordinary variable now, the warning doesn't make sense.

def _1; end

# expected: warning: `_1` is used as numbered parameter
# and call to _1()
# actual: Error: ordinary parameter is defined
# reason: Because `1.times{|i| _0 }` → force warning on Ruby 2.7 and syntax error on Ruby 3." is written in the log
proc { |i| _1 }.call 42

The method is irrelevant, Ruby doesn't know if it is defined or not until calling it. And _1 in a block is considered as a numbered parameter, it conflicts with [i].

# expected: warning: `_1` is used as numbered parameter
# actual: No warning
# reason: Because define local variable _1 is warning
def hgoe(_1)
end

It is outside block.

```
08/30/2022
```

08/30/2022 3/4
This is a bug. It must be a syntax error. It was a syntax error in old versions actually.

#7 - 11/06/2019 03:46 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset 82e840ad15bba42b999a21de746967a731240ec2.

Numbered parameter cannot appear outside block now [Bug #16293]