Ruby master - Bug #14374

`for` does not splat elements

01/19/2018 08:24 AM - nobu (Nobuyoshi Nakada)

<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>ko1 (Koichi Sasada)</td>
</tr>
<tr>
<td>Target version:</td>
<td>ruby -v: 2.6</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.3: REQUIRED, 2.4: REQUIRED, 2.5: REQUIRED</td>
</tr>
</tbody>
</table>

Description

In general, it is said that for statement and each method call are similar, except for the scope of loop variables.

I found another difference, since 1.9.

```ruby
a = [Struct.new(:to_ary).new([1, 2])]
a.each {|i, j| p [i, j]} #=> [1, 2]
for i, j in a; p [i, j]; end #=> [#<struct to_ary=[1, 2]>, nil]
```

In 1.8, the latter result equals the former.
Is this intentional?

Associated revisions

Revision 7ea65f9b - 01/27/2018 10:41 AM - nobu (Nobuyoshi Nakada)
compile.c: try to convert in massign for
  - compile.c (compile_for_masgn): try to convert to an Array if an element of massign for is not an Array. [ruby-core:84931] [Bug #14374]
git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62073 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62073 - 01/27/2018 10:41 AM - nobu (Nobuyoshi Nakada)
compile.c: try to convert in massign for
  - compile.c (compile_for_masgn): try to convert to an Array if an element of massign for is not an Array. [ruby-core:84931] [Bug #14374]

Revision 62073 - 01/27/2018 10:41 AM - nobu (Nobuyoshi Nakada)
compile.c: try to convert in massign for
  - compile.c (compile_for_masgn): try to convert to an Array if an element of massign for is not an Array. [ruby-core:84931] [Bug #14374]

History

#1 - 01/27/2018 10:41 AM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset trunk:62073.

#2 - 01/28/2018 12:21 PM - nagachika (Tomoyuki Chikanaga)
- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: REQUIRED, 2.4: REQUIRED, 2.5: REQUIRED

#3 - 12/05/2018 04:55 PM - nobu (Nobuyoshi Nakada)
I found each does not call Array.try_convert so only for is affected by redefinition of that method.

```
$ ruby -e 'a = [Struct.new(:to_ary).new([1, 2])]; def Array.try_convert(*); raise; end; a.each {|i, j| p [i, j]}; for i, j in a; p [i, j]; end'
[1, 2]
Traceback (most recent call last):
```

5/10/2020
3: from -e:1:in `<main>'
2: from -e:1:in `each'
1: from -e:1:in `block in <main>'
-e:1:in `try_convert': unhandled exception

https://github.com/nobu/ruby/pull/new/feature/expandarray-for-masgn