

Ruby - Bug #21132

Changed postposition `rescue` and `if` behavior since Ruby 3.4

02/13/2025 02:14 AM - watson1978 (Shizuo Fujita)

Status:	Closed	Backport: 3.1: UNKNOWN, 3.2: UNKNOWN, 3.3: UNKNOWN, 3.4: UNKNOWN
Priority:	Normal	
Assignee:	prism	
Target version:		
ruby -v:		

Description

When I use postposition rescue and if at same line, the behavior is changed since Ruby 3.4.

Reproduction

```
def foo
  return "hoge hoge" rescue nil if false
  return "fuga fuga"
end

foo
```

Result with Ruby 3.3

```
irb(main):001> RUBY_VERSION
=> "3.3.6"
irb(main):002* def foo
irb(main):003*   return "hoge hoge" rescue nil if false
irb(main):004*   return "fuga fuga"
irb(main):005> end
=> :foo
irb(main):006> foo
=> "fuga fuga"
```

YARV bytecode

```
ruby 3.3.6 (2024-11-05 revision 75015d4c1f) [x86_64-linux]
== disasm: #<ISeq:<compiled>@<compiled>:1 (1,0)-(6,3)>
0000 definemethod          :foo, foo          ( 1) [Li]
0003 putself               ( 6) [Li]
0004 opt_send_without_block <calldata!mid:foo, argc:0, FCALL|VCALL|ARGS_SIMPLE>
0006 leave

== disasm: #<ISeq:foo@<compiled>:1 (1,0)-(4,3)>
0000 putstring              "fuga fuga"        ( 3) [LiCa]
0002 leave                  ( 4) [Re]
```

Result with Ruby 3.4

```
irb(main):001> RUBY_VERSION
=> "3.4.1"
irb(main):002* def foo
irb(main):003*   return "hoge hoge" rescue nil if false
irb(main):004*   return "fuga fuga"
irb(main):005> end
=> :foo
irb(main):006> foo
=> "hoge hoge"
```

YARV bytecode

```
ruby 3.4.1 (2024-12-25 revision 48d4efcb85) +PRISM [x86_64-linux]
== disasm: #<ISeq:<compiled>@<compiled>:1 (1,0)-(1,54)>
0000 putself ( 1)[Li]
0001 opt_getconstant_path <ic:0 RubyVM::InstructionSequence>
0003 opt_getconstant_path <ic:1 DATA>
0005 opt_send_without_block <calldata!mid:new, argc:1, ARGS_SIMPLE>
0007 opt_send_without_block <calldata!mid:disassemble, argc:0, ARGS_SIMPLE>
0009 opt_send_without_block <calldata!mid:puts, argc:1, FCALL|ARGS_SIMPLE>
```

Related issues:

- | | |
|--|-----------------|
| Related to Ruby - Bug #21097: `x = a rescue b in c` and `def f = a rescue b i... | Assigned |
| Is duplicate of Ruby - Bug #21048: [Prism] rescue in modifier form with condi... | Closed |

History

#1 - 02/13/2025 02:30 AM - watson1978 (Shizuo Fujita)

- Description updated

#2 - 02/13/2025 02:32 AM - watson1978 (Shizuo Fujita)

- Subject changed from Changed postposition `resque` and `if` behavior since Ruby 3.4 to Changed postposition `rescue` and `if` behavior since Ruby 3.4

#3 - 02/13/2025 02:44 AM - watson1978 (Shizuo Fujita)

When I switched parser to parse.y, it has same result with Ruby 3.3.

```
[~/tmp]$ ruby -v --parser=parse.y t.rb
ruby 3.4.1 (2024-12-25 revision 48d4efcb85) [x86_64-linux]
"fugafuga"
[~/tmp]$ ruby -v --parser=prism t.rb
ruby 3.4.1 (2024-12-25 revision 48d4efcb85) +PRISM [x86_64-linux]
"hogehoge"
```

#4 - 02/13/2025 02:49 AM - nobu (Nobuyoshi Nakada)

- Related to Bug #21097: `x = a rescue b in c` and `def f = a rescue b in c` parsed differently between parse.y and prism added

#5 - 02/13/2025 02:49 AM - nobu (Nobuyoshi Nakada)

- Assignee set to prism

#6 - 02/13/2025 08:52 AM - hsbt (Hiroshi SHIBATA)

- Is duplicate of Bug #21048: [Prism] rescue in modifier form with condition behaves differently added

#7 - 02/13/2025 10:52 AM - watson1978 (Shizuo Fujita)

My problem will be solved by <https://github.com/ruby/ruby/commit/c2908613368b2ae404d094a15df61d830fc46dc9>
Thanks

#8 - 02/13/2025 10:53 AM - watson1978 (Shizuo Fujita)

- Status changed from Open to Closed