

Ruby master - Feature #5400

Remove flip-flops in 2.0

10/04/2011 04:43 PM - judofyr (Magnus Holm)

Status:	Rejected	
Priority:	Normal	
Assignee:	matz (Yukihiro Matsumoto)	
Target version:	Next Major	
Description		
Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?		

Associated revisions

Revision bae638ad - 06/15/2018 08:53 AM - mame (Yusuke Endoh)

parse.y: Deprecate flip-flops

Ref #5400

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63667 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 63667 - 06/15/2018 08:53 AM - mame (Yusuke Endoh)

parse.y: Deprecate flip-flops

Ref #5400

Revision 63667 - 06/15/2018 08:53 AM - mame (Yusuke Endoh)

parse.y: Deprecate flip-flops

Ref #5400

Revision 4e038a7e - 07/11/2019 05:52 AM - nobu (Nobuyoshi Nakada)

Revert "parse.y: Deprecate flip-flops"

This reverts commit bae638ad5b782c44c80efe33834cb9039279af46.

[Feature #5400]

Revision aae25ddb - 08/28/2019 05:41 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 4e038a7e64a9d52eed59b8f05647d4e58d265ec3: [Backport #5400]

```
Revert "parse.y: Deprecate flip-flops"
```

```
This reverts commit bae638ad5b782c44c80efe33834cb9039279af46.
```

```
[Feature #5400]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_6@67792 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 67792 - 08/28/2019 05:41 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 4e038a7e64a9d52eed59b8f05647d4e58d265ec3: [Backport #5400]

```
Revert "parse.y: Deprecate flip-flops"
```

```
This reverts commit bae638ad5b782c44c80efe33834cb9039279af46.
```

```
[Feature #5400]
```

Revision d92289cd - 08/29/2019 02:10 AM - nobu (Nobuyoshi Nakada)

Revert "Remove warnings of flip-flop deprecation from tests and specs"

This reverts commit bf7a32d22079cc44eb19794e41d82b886d5d17b3.

flip-flop is no longer deprecated.

[Feature #5400]

History

#1 - 10/04/2011 05:36 PM - yimutang (Joey Zhou)

Magnus Holm wrote:

Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?

I disagree. The flip-flop syntax is obscure, but very useful, especially in text manipulation.

For example, I want to fetch some chunks of lines :

```
DATA.each_line do |line|
  print line if (line =~ /begin/)..(line =~ /end/)
end
```

```
__END__
0a
1begin
2c
3end
4e
5f
6begin
7end
8i
9j
```

this will print:

```
1begin
2c
3end
6begin
7end
```

flip-flop syntax comes from Perl, the Perl idiom looks like "print if /begin/..end/;" or "print if 5..8;" (which means print line5 to line8).

Perl idiom is implicit, not so easy to read, Ruby idiom is a little hard to write.

I'm afraid Rudy didn't take advantage of flip-flops, I try to write in this way:

```
DATA.readlines.select {|line| (line =~ /begin/)..(line =~ /end/) } # error
```

it seems that ruby treat (line =~ /begin/)..(line =~ /end/) as a range object, that is not what I mean.

Maybe it's difficult for the parser to distinguish the range token ".." and flip-flop token "..", Perl use the same token because it has "context", but Ruby hasn't.

#2 - 10/04/2011 07:58 PM - yimutang (Joey Zhou)

Magnus Holm wrote:

Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?

Well, the flip-flop behavior is useful, so it should not be removed.

However, I agree that the syntax is a bit confusing.

Flip-flop in Ruby is not so powerful as in Perl (<http://perldoc.perl.org/perlop.html#Range-Operators>).

It seems like expression leading to true or false, but it cannot be assigned to a variable, and often be treated as a range literal.

So maybe we can get rid of the ".." syntax, instead, introduce a class to do the same thing.

I've implemented a simple class FlipFlop, which simulates the behavior of flip-flop in Perl.

```
class FlipFlop
  def initialize(test_right_same_time=false)
```

```

    @bool = false
    @sequence_num = 0
    @same_time = test_right_same_time
end

def rewind
  initialize(@same_time)
end

def test(condition_left, condition_right)
  if @bool == false and condition_left
    @sequence_num = 1
    @bool = true
    if @same_time == true and condition_right
      @sequence_num = 1.0
      @bool = false
    end
    return true
  elsif @bool == true and not condition_right
    @sequence_num += 1
    return true
  elsif @bool == true and condition_right
    @sequence_num += 1.0
    @bool = false
    return true
  else # @bool == false and condition_left == false
    @sequence_num = 0
    return false
  end
end

def true?
  @bool
end

def value
  @sequence_num
end

def end?
  @sequence_num.is_a? Float
end

end

flipflop = FlipFlop.new

# take only line3 ~ line5 from a chunk of lines (from /begin/ to /end/)
lines = DATA.readlines.select do |line|
  t = flipflop.test(line =~ /begin/, line =~ /end/)
  t and flipflop.value.between?(3,5)
end

p lines # ["04end(x)\n", "09(x)\n", "10(x)\n", "11(x)\n", "17(x)\n", "18(x)\n", "19end(x)\n"]

__END__
01
02begin
03
04end(x)
05
06
07begin
08
09(x)
10(x)
11(x)
12end
13
14
15begin
16
17(x)
18(x)
19end(x)

```


have to wait until 3.0.

matz.

#7 - 10/11/2011 12:53 AM - judofyr (Magnus Holm)

I have no objection to deletion, but I'm just curious.
Why do you want to delete it aggressively?

--
Yusuke Endoh mame@tsg.ne.jp

I just want to get rid of complexity in the language.

Currently, flip-flops aren't well known for Rubyists, so I don't feel comfortable of using them in code. I fear that it won't be readable. And if you *don't* know them, it's easy to confuse them for literal ranges, which makes it even more confusing. If it had a distinct syntax, you would at least realize that you don't know about them, now you would go "what? a Range literal is always true, no?".

#8 - 10/18/2011 09:16 AM - naruse (Yui NARUSE)

- Project changed from *Ruby master* to *CommonRuby*
- Target version deleted (*Next Major*)

#9 - 10/23/2011 05:21 PM - naruse (Yui NARUSE)

- Project changed from *CommonRuby* to *Ruby master*

#10 - 03/27/2012 03:33 AM - mame (Yusuke Endoh)

- Target version set to *Next Major*

#11 - 11/18/2012 10:40 AM - Anonymous

For Endo's sake, please don't remove this jewel, bring it to perfection, somehow :)))

#12 - 02/26/2013 09:34 AM - ko1 (Koichi Sasada)

- Category set to *core*
- Assignee set to *matz* (Yukihiro Matsumoto)

#13 - 05/17/2018 08:47 AM - nobu (Nobuyoshi Nakada)

judofyr (Magnus Holm) wrote:

Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?

I know, and use them.

#14 - 05/17/2018 08:51 AM - matz (Yukihiro Matsumoto)

Finally, we are going to remove it. 2.6 will warn flip-flop usage of ranges.

Matz.

#15 - 06/15/2018 08:59 AM - mame (Yusuke Endoh)

- Status changed from *Open* to *Assigned*

I've committed r63667..r63669 to deprecate flip-flops. I leave this ticket open to remove the feature in 3.0 (?).

There were some programs that used flip-flops in build scripts. I rewrote them to a code that does not use flip-flops. The work was harder than I expected.

Honestly I'm unsure if deprecation of flip-flops is really a right way...

#16 - 09/18/2018 09:22 AM - normalperson (Eric Wong)

mame@ruby-lang.org wrote:

There were some programs that used flip-flops in build scripts. I rewrote them to a code that does not use flip-flops. The work was harder than I expected.

Honestly I'm unsure if deprecation of flip-flops is really a right way...

<https://bugs.ruby-lang.org/issues/5400#change-72504>

I don't think it is a good idea to deprecate or remove features we've supported for so long.

I've been pondering on this issue more while I away...

These warnings in scripting languages annoy USERS. In contrast with compiled languages: only programmers see the warning when they compile, most users never see warnings from gcc/clang. This gives languages like C more freedom than us to deprecate things (e.g. gets(3)).

Furthermore, with compiled languages, the old build will continue working forever without annoying the user. I have small C programs which haven't been rebuilt in a decade or more, yet still run fine.

Scripting language users don't have that luxury and will be affected by breakage when their distro upgrades Ruby for them.

Not every user is a programmer and can fix every warning they encounter. And often times, the programmer who originally wrote the script has long moved on and a new user will choose something written in a different language.

Looking back to a decade ago, I saw many people leave Ruby because migrating to 1.8 to 1.9 was too painful and the language was viewed as too volatile. Yet we still keep making the same mistakes and lose users as a result :<

#17 - 09/18/2018 09:58 AM - Eregon (Benoit Daloz)

Just a quick opinion, deprecation is part of the migration part, better warn in a release in between than break functionality immediately. We also warn for continuation FWIW.

I support removing flip-flops, it looks like a legacy strange behavior from Perl that doesn't belong to Ruby for me. I think there is no need for such magic and confusing syntax to do something like this.

<https://chrisseaton.com/truffleruby/flip-flops/> makes it clear there is little to no code using it, so I don't think it's an issue to deprecate and remove it.

[#14183](#) is likely much more relevant for a discussion about compatibility ;)

#18 - 09/21/2018 08:14 AM - nobu (Nobuyoshi Nakada)

I agree with Eric.

It is too annoying to rewrite flip-flop.

A patch for flip-flop I found today is:

```
diff --git a/tool/enc-unicode.rb b/tool/enc-unicode.rb
index d953014952..c05d02358c 100755
--- a/tool/enc-unicode.rb
+++ b/tool/enc-unicode.rb
@@ -538,6 +538,7 @@
   IO.popen(%W[diff -DUSE_UNICODE_AGE_PROPERTIES #{fds[1].path} #{fds[0].path}], "r") {|age|
     IO.popen(%W[diff -DUSE_UNICODE_PROPERTIES #{fds[2].path} -], "r", in: age) {|f|
       ansi = false
+     in_hash = false
+     f.each {|line|
+       if /ANSI-C code produced by gperf/ =~ line
         ansi = true
@@ -545,7 +546,7 @@
       line.sub!(/\/*ANSI*\//, '1') if ansi
```

```

line.gsub!(/\(int\)\((?:long|size_t)\)&\(\(struct uniname2ctype_pool_t \*\)0\)->uniname2ctype_pool_(s
tr\d+),\s+/,
      'uniname2ctype_offset(\1), ')
-   if (/^(uniname2ctype_hash) / =~ line) .. (/^\} / =~ line)
+   if !in_hash ? (in_hash = /^(uniname2ctype_hash) / =~ line) : (in_hash = /^\} / ! =~ line; true)
      line.sub!(/^( *(?:register\s+)?(.*\S)\s+hval\s*=\s*)(?=\len;)/, '\1(\2)')
    end
    puts line

```

Does this gain readability?

#19 - 12/10/2018 10:11 PM - jnchito (Junichi Ito)

Could you list this change on NEWS page?

This is because flip-flop is not used so often, but many people know it as one of Ruby's strange features. So this must be an important change.

Please refer my pull request: <https://github.com/ruby/ruby/pull/2047>

#20 - 12/11/2018 12:05 AM - shevegen (Robert A. Heiler)

I only got to know it after having read the news entry just now. :-)

I think the impact of its removal will be very, very little - not many need it; and I am sure even less depend on it. Personally after having seen it, I would not use it because it (to me) seems not worth getting my brain to try to understand what it is doing. I like to write extremely simple code that I can very quickly understand.

I think the only question is whether matz is ok with the removal (possibly if it was changed already), considering his statement about avoiding backwards-incompatible change until 3.0 - but I think in this case, it will really just about not affect anyone. I do not recall even having seen flip flop used in real code in the last ~10 years or so even including having checked quite a few gems by other people on rubygems.org. Personally I think it is good to remove it - makes ruby a tiny bit simpler to understand. :D
(Ternary + flip-flop is a bit to the mind like this language that starts with Brain ...)

By the way, since 7 years ago at the least two wrote that they find it useful - features are useful, but there are trade offs. I do not find the flip-flop operator or way extremely useful for example.

Ruby took concepts and ideas from many other languages, including perl, but ruby always was a LOT more readable than perl and perl really did not care much at all about readability. So when you compare ruby and perl, you also should compare the readability - if you have a language that is "uglier", then a feature may be less at odds with other parts of a language if the rest of the language is already quite ugly to begin with (perl), whereas I think syntax that is not "ideal", is more distracting in a language that has a better syntax (ruby). Not sure if I managed to explain that...

#21 - 04/08/2019 04:52 AM - nobu (Nobuyoshi Nakada)

I **STRONGLY** object removal of flip-flop, in "-e" option at least. It is too tiresome to rewrite with a flag variable in one-liners.

#22 - 06/10/2019 12:38 PM - scub8040 (Saverio M.)

judofyr (Magnus Holm) wrote:

Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?

I'll chip in as a dev who spends a significant amount of time scripting.

First, the description of the issue is false. I do use the flip-flop operator, so statements like "Nobody knows/uses it" are false.

I'm not being pedantic here - I'm pointing out that there is a very strong bias in presenting the issue in these terms, which is not a good starting point for a discussion.

Second - I find underwhelming to conflate the functionality with its syntax, throwing away the former because the latter is arguably poor, without evaluating any alternatives.

Flip-flop logic exists in the major scripting languages - Perl and AWK, and I think sed as well; it's arguably seldom used, but it has a very specific use case, where it fits very well. Devs who frequently process text do use it.

My last issue is somewhat subtle. I see an underlying philosophy of identifying Ruby as the "Rails language" and nothing else; removing the flip-flop feature is a symbolic detachment of Ruby from text processing.

Making Ruby the "Rails language" is a respectable direction, but is it really desirable?

#23 - 06/18/2019 12:47 PM - rovf (Ronald Fischer)

judofyr (Magnus Holm) wrote:

Nobody knows them. Nobody uses them. Let's just get rid of flip-flops, shall we?

Do NOT remove this useful feature!!! I used it a lot already in Perl, and now using it in Ruby too. The deprecation warning also causes a lot of headache, see for instance my posting at <https://stackoverflow.com/questions/56648847/workaround-flip-flop-deprecated-warning-in-ruby?>.

#24 - 06/18/2019 03:48 PM - CodeGnome (Todd Jacobs)

rovf (Ronald Fischer) wrote:

Do NOT remove this useful feature!!! I used it a lot already in Perl, and now using it in Ruby too. The deprecation warning also causes a lot of headache

Very much this. I find the flip-flop operator extremely useful for text processing, and don't see *any* value-add to removing such a useful feature. If the problem is that the new explicit-line syntax or other changes have made it harder to use, or that the documentation on this useful feature is lacking, then that's a separate (but potentially useful) issue. The OP's view that they don't use it, so the feature should be removed, is definitely something that would punish *me* personally. You're not alone in relying on this useful feature.

The deprecation warning is also an anti-feature. It makes a legitimate language feature look like a bug. That helps nobody. I hope the flip-flop operator stays, and that the deprecation warning gets removed.

#25 - 07/11/2019 05:27 AM - matz (Yukihiro Matsumoto)

- Status changed from Assigned to Rejected

I hear the negative feedback from the community. OK, I give up. The warning should be removed.

Matz.

#26 - 07/14/2019 03:23 AM - normalperson (Eric Wong)

matz@ruby.or.jp wrote:

I hear the negative feedback from the community. OK, I give up. The warning should be removed.

Thank you! This saves me the trouble of rewriting a bunch of random scripts I have.

It seems your position regarding keeping backwards compatibility (backtick, frozen literals) seems stronger nowadays; so maybe I can consider returning from my sabbatical next year...

(Or maybe Ruby is better off without a Free Software extremist :P)

#27 - 08/27/2019 07:26 PM - graywolf (Gray Wolf)

Fyi this currently causes ri tab completion to display a warning (<https://github.com/scop/bash-completion/issues/343>)

```
$ ri mktmpd-e:1: warning: flip-flop is deprecated
Nothing known about RSS::Maker::RSS09::Channel
Did you mean?  RSS::Maker::RSS09::Channel
               RSS::Maker::RSS092::Channel
               RSS::Maker::RSS091::Channel
               RSS::Maker::RSS20::Channel
               RSS::Maker::RSS10::Channel
               RSS::Maker::RSS09::Channel::Title
```

```
RSS::Maker::RSS09::Channel::Links
RSS::Maker::RSS09::Channel::Cloud
RSS::Maker::RSS09::Image
RSS::Maker::RSS092::Image
RSS::Maker::RSS091::Image
RSS::Maker::RSS20::Image
RSS::Maker::RSS10::Image
RSS::Maker::RSS09::Items
```

ir ^C

So much for Nobody uses them....

I appreciate very much that this warning was reverted. Hopefully it will be in 2.6.4.

#28 - 08/28/2019 05:42 AM - nagachika (Tomoyuki Chikanaga)

I backported 4e038a7e into ruby_2_6 at r67792.