Hello.

I might have missed something, but examples like:

```ruby
if
  puts "!!!!1"
elsif
  puts "!!!!2"
elsif
  puts "!!!!3"
else
  puts "!!!!4"
end

# Output:
# !!!!1
# !!!!2
# !!!!3
# !!!!4
```

```ruby
if false
  puts "!!!!1"
elsif 1==2
  puts "!!!!2"
elsif
  puts "!!!!3"
else
  puts "!!!!4"
end

# Output:
# !!!!3
# !!!!4
```

are confusing. We probably should raise SyntaxError in the case when if/elsif is being used without any condition. What do you think about it?

Original source: [https://twitter.com/bogdanvlviv/status/1154356514628493313](https://twitter.com/bogdanvlviv/status/1154356514628493313)

Edited:

[https://twitter.com/mamantoha/status/1154369189647454213](https://twitter.com/mamantoha/status/1154369189647454213) helped me to figure out that

The code like

```ruby
if false
  puts "!!!!1"
elsif 1==2
  puts "!!!!2"
elsif
  puts "!!!!3"
```
else
  puts "!!!!4"
end

is similar to

if false
  puts "!!!!1"
elsif 1==2
  puts "!!!!2"
elsif (puts "!!!!3")
else (puts "!!!!4")
end

Probably Ruby should be more strict in those cases because such code is more prone to bugs.

History
#1 - 07/25/2019 12:28 PM - bogdanvlviv (Bogdan Denkovych)
  - ruby -v changed from ruby 2.6.3p62 (2019-04-16 revision 67580) [x86_64-linux] to ruby 2.6.3p62 (2019-04-16 revision 67580) [x86_64-linux]

#2 - 07/25/2019 01:06 PM - bogdanvlviv (Bogdan Denkovych)
  - Description updated

#3 - 07/25/2019 01:13 PM - bogdanvlviv (Bogdan Denkovych)
  - Description updated

#4 - 07/25/2019 01:56 PM - znz (Kazuhiro NISHIYAMA)
Rubocop can detect such bug.

#5 - 07/25/2019 05:49 PM - marcandre (Marc-Andre Lafortune)
  - Status changed from Open to Rejected