Ruby - Feature #12608

Proposal to replace unless in Ruby

07/21/2016 01:00 PM - LucianCancescu (Lucian Cancescu)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

Hi,

I would like to propose the replacement of the unless statement in Ruby.

Problem description:

Unless is complex and my arguments are:

- I often see in projects unless ... else:

```
unless user
  redirect_to_login
else
  render_profile
end
```

- Using unless with conditionals makes it very hard to understand:

```
unless user || tenant
  user_and_tenant_are_present
end
```

Put an else to the conditional above and you can easily spend a few minutes to correctly understand what the code does.

Ruby is focuses on simplicity. I find both the examples above speaking against it. I have seen many Ruby projects, written by different people, and in most of them you see unless being abused in some way.

There is another aspect, when you see unless you have to keep a not in mind. The problem with that is that you have to negate all you read afterwards. You have to remove the negation when an else appears and you have to carefully read the negation with parenthesis in mind when unless contains boolean conditions. It's a huge complexity which can be avoided by simply using something else.

Proposed solution:

How can we reduce the complexity? Other languages, for instance Swift, have introduced the guard [1] statement.

guard can replace unless in Ruby and the complex scenarios from above would become impossible:

Example 1 from above translates to:

```
guard user else
  return redirect_to_login
end
# do something with the user
render_profile
```

Example 2 from above translates to:

```
guard user && tenant else
  return user_or_tenant_missing
end
# do something with the user and the tenant
```

```
user_and_tenant_are_present
```

Not only the conditions become easier to read, but it reduces a lot the complexity of the code.

My suggestion is to add guard to the language and make it replace unless. What do you think?

Thanks,
Lucian

[1] Guard in Swift:
https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Statements.html#//apple_ref/swift/grammar/guard-statement

History

#1 - 07/21/2016 01:04 PM - LucianCancescu (Lucian Cancescu)

- Description updated

#2 - 07/21/2016 01:05 PM - LucianCancescu (Lucian Cancescu)

- Description updated

#3 - 07/21/2016 02:18 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

I often use unless and never use it with else. While you might get some discussion around forbidden "else" when "unless" is used, I don't really think you'd be able to convince anyone to simply remove "unless" from Ruby. It may be hard for you to read unless but I guess it's not hard for most people. Particularly I don't find it easier to understand an "if" than to understand an "unless" condition...

#4 - 07/21/2016 02:54 PM - sawa (Tsuyoshi Sawada)

Rodrigo Rosenfeld Rosas wrote:

> I often use unless and never use it with else. While you might get some discussion around forbidden "else" when "unless" is used, I don't really think you'd be able to convince anyone to simply remove "unless" from Ruby. It may be hard for you to read unless but I guess it's not hard for most people. Particularly I don't find it easier to understand an "if" than to understand an "unless" condition.

Exactly. At least, unless is much easier to understand, more concise, and much better than whatever is proposed in this thread.

#5 - 07/21/2016 03:16 PM - rosenfeld (Rodrigo Rosenfeld Rosas)

Yes, I haven't read the proposals yet and they are indeed pretty confusing :)

Also, I just read the second example and it's wrong:

```
unless user || tenant
  user_and_tenant_are_present
end
```

This condition means either user or tenant (or both) are present rather than what the method states...

#6 - 07/21/2016 04:45 PM - mmaglana (Mark Maglana)

The following example:

```
unless user
  redirect_to_login
else
  render_profile
end
```

is a result of poor choices made by the programmer and not the fault of unless. The above is better written as:

```
if user
  render_profile
else
  redirect_to_login
end
```

Also, unless is best used in a single line. So the following:

```
unless user && tenant
  user_and_tenant_are_absent
```

```
end
```

is better written as:

```
user_and_tenant_are_absent unless user && tenant
```

#7 - 07/26/2016 12:17 PM - LucianCancescu (Lucian Cancescu)

Thanks all for the replies. A few examples will obviously not cover everything to convince everyone. My examples were probably not the best but you get the idea, it happens a lot and it can so easily be changed.

However, I continue see so many wrong usages of unless that it makes me question whether its presence in the language is a good idea at all. I am sure I'm not the only one who has ever struggled to read a complicated unless/else structure. You can call it a poor design, but it's something that allows people to so easily shoot themselves in the foot. And they seem to enjoy doing it.

We have two things to achieve the same result, one (if) is positive and the other (unless) is negative. I think this brings the confusion.

I realize that my initial proposal has a very big impact and will probably be rejected.

However, I still believe we can improve the situation by not allowing unless to be used together with else.

#8 - 08/09/2016 08:51 AM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I don't think proposed guard syntax is more readable than unless.
Even if some may agree with you, I doubt it's worth adding a new keyword and introducing possible incompatibility.

Matz.