Ruby master - Feature #9442

Multiple comparison construction with `==` and `===`

01/23/2014 08:23 AM - sawa (Tsuyoshi Sawada)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		

Description

I often want to write a condition that depends on multiple variables like this:

```
if a == 1 and b == 2 then ...
elsif a == 2 and b == 1 then ...
elsif a == 2 and b == 2 then ...
...
```

In order to make this compact, I sometimes write these cases like this:

```
case [a, b]
when [1, 2] then ...
when [2, 1] then ...
when [2, 2] then ...
...
```

or even

```
if [a, b] == [1, 2] then ...
elsif [a, b] == [2, 1] then ...
elsif [a, b] == [2, 2] then ...
...
```

but I feel that constructing these arrays is a waste of resource, as well as is not elegant to write.

I request a syntax feature that allows multiple comparison using == and === in the spirit of multiple assignment with = :

```
x, y, z = :foo, :bar, :baz
```

so that the two examples above would be respectively written as:

```
case a, b
when 1, 2 then ...
when 2, 1 then ...
when 2, 2 then ...
...
```

and

```
if a, b == 1, 2 then ...
elsif a, b == 2, 1 then ...
elsif a, b == 2, 2 then ...
...
```

History

#1 - 01/23/2014 08:47 AM - sawa (Tsuyoshi Sawada)

The conventional case construction allows alternatives to be separated with a comma:

```
case a
when 1, 2, 3, 4 then ...
when 5 then ...
...
```

In order to resolve ambiguity/confusion, parentheses may be used when multiple comparison and alternatives are used in combination:

```
case a, b
when (1, 1), (1, 2) then ...
when 2, 2 then ...
...
```

#2 - 01/23/2014 12:16 PM - matz (Yukihiro Matsumoto)

- Status changed from Open to Rejected

I am sorry but the proposed syntax conflicts with the existing syntax, especially the equal operator one.
You have proposed improvement on case statement, but it still conflicts.

So (until someone come up with the new non-conflicting syntax), I have to reject this proposal.

Matz.

#3 - 01/23/2014 12:30 PM - mame (Yusuke Endoh)

I think that it is better to optimize the specific form of case statements than to introduce a new syntax.

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 01/23/2014 04:03 PM - sawa (Tsuyoshi Sawada)

Yukihiro Matsumoto wrote:

> I am sorry but the proposed syntax conflicts with the existing syntax, especially the equal operator one.
> You have proposed improvement on case statement, but it still conflicts.
>
> So (until someone come up with the new non-conflicting syntax), I have to reject this proposal.
>
> Matz.

My understanding is that

```
a, b == 1, 2
```

or

```
1, 2 === a, b
```

among which I am not sure Matz is mentioning which as "the equal operator one", but they are both currently ungrammatical. Can you point me to how these would conflict with the current syntax? Is it about operator precedence?