

Ruby master - Feature #16960

Feedback regarding => in 'As' Pattern Matching

06/13/2020 12:30 AM - ParadoxV5 (XXXXXXXXXXXXXXXXXXv0)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>The only obstacle preventing the use of non-symbol keys in hash patterns is simply the fact that => is used for both Hashes and 'as' patterns.</p>	
<pre>case x in Integer => a, Float => b # Ambiguous in key: Symbol => value, **rest # Poor readability in Symbol => key => Symbol => value, **rest # ??? end</pre>	
<p>I recommend changing to something else. We can introduce the and operator &:</p>	
<pre>case array in Float & 5 & five five.eql? 5.0 #=> true end</pre>	
<p>And/Or, implement another operator explicitly for 'as' patterns, say : (otherwise only used for <i>the</i> ternary operation) –</p>	
<pre>case x in Integer : a, Float : b in key: Symbol : value, **rest in Symbol : key => Symbol : value, **rest end</pre>	
<p>– which renders the pin operator ^ (and for those who <i>oppose</i>, their <i>unpin</i> operator(s)) unnecessary: expressions before it are always pinned values, and expressions after it are always case-in variable declarations.</p>	
<pre>good = 8..10 okay = 5..7 skip = 0 case x in skip : # Syntax sugar for skip : _ # skip in good : good_value puts good_value in okay : okay_value warn okay_value else : bad_value # Syntax sugar for BasicObject : bad_value raise bad_value.to_s end</pre>	
<p>The second option still looks like an unpin operator for sugary code, however, if pinning values –</p>	
<pre>case array in :, : *all_but_first # Syntax sugar for BasicObject : _, BasicObject : *all_but_first p all_but_first end</pre>	
<p>– are somehow preferred over declaring variables:</p>	
<pre>case array in _, *all_but_first # Current implementation p all_but_first end</pre>	

Regarding rescue

We can't change the => part in rescue, of course; but we don't need to. Hashes are not Exceptions anyway.

History

#1 - 06/13/2020 12:32 AM - ParadoxV5 ()

- Subject changed from *Feedback regarding `=>` in 'As' Pattern Matching* to *Feedback regarding => in 'As' Pattern Matching*