

Ruby master - Feature #17006

Let `Kernel#Hash` take a block to provide the default value

07/03/2020 08:09 AM - sawa (Tsuyoshi Sawada)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
<p>Sometimes, we need to create a hash that has explicit key-value pairs as well as a default value, but there has not been a way to do that at once. The most naive way is to do like this:</p> <pre>h = {a: 1, b: 2} h.default_proc = ->(h, k){raise "Unknown key #{k}"} h # => ...</pre>	
<p>A more sophisticated way is this:</p> <pre>Hash.new{ h, k raise "Unknown key #{k}"}.merge({a: 1, b: 2})</pre>	
<p>But that is still not short enough, and it also looks backwards, i.e., we usually want to specify the explicit key-value pairs before the default value.</p>	
<p>My proposal is to allow Kernel#Hash to take a block that provides the default value in the same way as the block of Hash.new:</p> <pre>Hash({a: 1, b: 2}){ h, k raise "Unknown key #{k}"}</pre>	

History

#1 - 07/03/2020 08:11 AM - sawa (Tsuyoshi Sawada)

- Description updated

#2 - 07/04/2020 07:35 AM - nobu (Nobuyoshi Nakada)

The description mentions the default value first, but the proposal will solve the default proc only.

#3 - 07/04/2020 07:51 AM - sawa (Tsuyoshi Sawada)

nobu (Nobuyoshi Nakada) wrote in [#note-2](#):

The description mentions the default value first, but the proposal will solve the default proc only.

Default value is a special (static) case of default proc, so I think having the latter would be enough:

```
Hash({a: 1, b: 2}){0}
```

If there is concern for efficiency, at least for strings, we can turn on the frozen string mode.

```
# frozen_string_literal: true
```

```
Hash({1 => "a", 2 => "b"}){"c"}
```