

Ruby - Feature #21182

Add Hash#rename

03/11/2025 07:04 PM - calebm (Caleb Meyer)

<div>Status:Open</div> <div>Priority:Normal</div> <div>Assignee:</div> <div>Target version:</div>	
<div>Description</div> <div>Abstract: Implement Hash#rename which takes as arguments a from key and a to key and renames the from key to the new name.</div> <div>Background: One of the most common data transformations for a hash (in my experience) is renaming a single key. You can do this with the existing hash methods (e.g. hash['new'] = hash.delete('old') or hash.transform_keys('old' =&gt; 'new')) but they feel a bit clunky.</div> <div>Proposal:</div> <div>Add a new method to Hash, something like this:</div> <div><pre>class Hash   # usage { old: 'value' }.rename(:old, to: :new)   # =&gt; { new: 'value' }   def rename(from, to:)     transform_keys(from =&gt; to)   end    def rename!(from, to:)     transform_keys!(from =&gt; to)   end end</pre></div>	

History

#1 - 03/12/2025 05:41 AM - nobu (Nobuyoshi Nakada)

transform\_keys('old' => 'new') feels good enough.  
At least, ('old' => 'new') (or even ('old', 'new'), (from: 'old', to: 'new')) looks clearer than (:old, to: :new) to me.

And another concern is if the method name rename express the semantics well.  
Is there a consensus that name means key?