

Ruby master - Feature #17036

Regexp deconstruction keys to allow pattern matching

07/19/2020 02:21 PM - zechris (Chris Ottrey)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

This is to allow Regexp matches in Ruby 2.7's new experimental Pattern Matching.

eg.

```
case /(?<a>.) (?<b>.)/.match("xy")
in a: "x", b:
  "a was 'x' and b was matched to #{b.inspect}"
end
```

```
#=> "a was 'x' and b was matched to 'y'"
```

[MatchData#named_captures](#) is *close* but not close enough as the Hash required from a [deconstruct_keys\(\)](#) method that gets used during Pattern Matching requires symbolized keys.

NB. This PR (<https://github.com/ruby/ruby/pull/3333>) is WIP as the C code hasn't actually been written yet..

Although, it could be used in the wild now with a refinement of MatchData like so:

```
module RegexpDeconstructKeys
  refine MatchData do
    # Temporary patch in ruby to make tests pass until the code gets re-written in C
    def deconstruct_keys(*keys)
      h0 = named_captures.transform_keys(&:to_sym)
      if keys
        keys = keys.flatten.compact
        raise TypeError unless keys.all? { |k| k.is_a?(Symbol) }
        h0 = h0.slice(*keys) unless keys.empty?
      end
      h0.inject({}) { |h, (k, v)| h[k] = v if v; h }
    end
    # Temporary patch in ruby to make tests pass until the code gets re-written in C
  end
end
```

```
using RegexpDeconstructKeys
```

```
case /(?<a>.) (?<b>.)/.match("xy")
in a: "x", b:
  puts "a was 'x' and b was #{b.inspect}"
end
```

```
#=> "a was 'x' and b was matched to 'y'"
```

(tested in ruby v2.7.1)

History

#1 - 07/19/2020 02:39 PM - zechris (Chris Ottrey)

- Description updated

#2 - 07/19/2020 02:45 PM - zechris (Chris Ottrey)

- Description updated