

Ruby master - Feature #11529

extensible % literal declarations

09/15/2015 09:07 PM - eike.rb (Eike Dierks)

Status:	Feedback
Priority:	Normal
Assignee:	
Target version:	

Description

The ruby syntax provides for declaring literals by using the % escapes:

- %q(foo) => 'foo'
- %Q(foo) => "foo"
- %w(foo bar) => ['foo', 'bar']
- %i(foo bar) => [:foo, :bar]

It should be possible to define new % escapes.

Use cases:

I currently use:

```
%q{select foo from bar}
to quote my sql statements.
```

It would improve readability to spell that:

```
%sql{select foo from bar}
(with %sql being an alias of %q [see parsing problems below])
```

But there could be more interesting uses:

```
%hash[
  a 1
  b 2]
```

```
%triples[
  a b c
  1 2 3]
```

```
%json/{"foo":"bar", "zip":"zap"}/
```

```
%octal_data[012 345 677]
```

and the extension code would be something like

```
def %hash(s)
  return ...
end
```

Notes:

- I did expect this to be a problem. But it's not.
- in my faint memory, old ruby versions were rumoured to parse %sqfooq => "foo" but this is no longer true: %sqfooq => SyntaxError: (eval):2: unknown type of %string
- the characters used for enclosing should be clearly defined
- we have two classes:
 - paired chars like () [] {} <>
 - quoting chars like ' " / @ _ # \$ %

My Suggestion:

- the quoting chars should be deprecated (try this: %Q% %%)
- the set of paired chars should be extended to include selected pairs from unicode math symbols
 - unicode does not yet define a plane for open-close-brackets
 - see <http://stackoverflow.com/questions/13535172/list-of-all-unicodes-open-close-brackets>

History

#1 - 09/16/2015 01:01 AM - nobu (Nobuyoshi Nakada)

- Description updated

- Status changed from Open to Feedback

Eike Dierks wrote:

- in my faint memory, old ruby versions were rumoured to parse %sqfooq => "foo" but this is no longer true: %sqfooq => SyntaxError: (eval):2: unknown type of %string

It has never been true, at least since 1.0.

- the quoting chars should be deprecated (try this: %Q% %%)

What's the rationale?

I don't think it acceptable because of backward compatibility.

And what does the former half mean?

#2 - 09/17/2015 05:57 PM - mame (Yusuke Endoh)

I like the idea. Though, I guess it will be hard to agree with the definition syntax and precise semantics.

```
and the extension code would be something like
def %hash(s)
  return ...
end
```

It conflicts with an existing definition syntax of method named %.

- the quoting chars should be deprecated (try this: %Q% %%)

Try this: %Q% %%

This is one of the cutest parts of Ruby syntax that I love.

```
p %%%
p %%\% \%%
p %%%%%
puts %: %+%- \s%%%) %
```

See also: <https://github.com/tric/trick2013/blob/master/yoshi-taka/entry.rb>

Anyway. This suggestion is a completely different topic than the first idea (extensible %). We should discuss it in another thread, if really needed.

--

Yusuke Endoh mame@ruby-lang.org