MatchData#sublen to return the length of a substring

09/16/2021 07:34 AM - nobu (Nobuyoshi Nakada)

There are many code taking the length of a substring matched by Regexp. For instance, in rdoc/markup/attribute_manager.rb:

```ruby
attr_updated = attrs.set_attr($\'.length + $1.length + $2.length, $3.length, attr)
if attr_updated
  $1 + NULL * $2.length + $3 + NULL * $2.length + $4
```

People often tends to use such code (although the first addition can be simpler as $~.begin(3)), that creates and soon drops substrings, just to take the length.

Therefore, how about the new method to calculate the length, MatchData#sublen?

```ruby
/(\d+)\W(\w+)/ =~ "1:foo"
$~.sublen(1) #=> 1
$~.sublen(2) #=> 3
```

Characteristics

- **Status**: Closed
- **Priority**: Normal
- **Assignee**: nobu
- **Target version**: 

Description

There are many code taking the length of a substring matched by Regexp. For instance, in rdoc/markup/attribute_manager.rb:

```ruby
attr_updated = attrs.set_attr($\'.length + $1.length + $2.length, $3.length, attr)
if attr_updated
  $1 + NULL * $2.length + $3 + NULL * $2.length + $4
```

People often tends to use such code (although the first addition can be simpler as $~.begin(3)), that creates and soon drops substrings, just to take the length.

Therefore, how about the new method to calculate the length, MatchData#sublen?

```ruby
/(\d+)\W(\w+)/ =~ "1:foo"
$~.sublen(1) #=> 1
$~.sublen(2) #=> 3
```

Associated revisions

Revision 09d724e6 - 09/16/2021 10:55 AM - nobu (Nobuyoshi Nakada)
[Feature #18172] Add MatchData#match

The method to return the single matched substring corresponding to the given argument.

Revision f2cb6288 - 09/16/2021 10:55 AM - nobu (Nobuyoshi Nakada)
[Feature #18172] Add MatchData#match_length

The method to return the length of the matched substring corresponding to the given argument.

Revision aa3d8388 - 09/16/2021 10:55 AM - nobu (Nobuyoshi Nakada)
NEWS for [Feature #18172] [ci skip]

Revision 475a4651 - 09/16/2021 11:38 PM - nobu (Nobuyoshi Nakada)
[Feature #18172] Fix duplicate test name

Revision 8f41c791 - 09/17/2021 01:27 AM - nobu (Nobuyoshi Nakada)
Add spec for MatchData#match and MatchData#match_length [Feature #18172]

History

#1 - 09/16/2021 09:04 AM - matz (Yukihiro Matsumoto)
In Ruby C source code, we use names like sublen but Ruby methods tend to be fully spelled. So I propose MatchData#match(n) and MatchData#match_length(n).

Matz.

#2 - 09/16/2021 12:16 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

Applied in changeset g8l09d724e6f846b4e53e8571d41ca7d3055d732d9f.
[Feature #18172] Add MatchData#match

The method to return the single matched substring corresponding to the given argument.

#3 - 09/16/2021 01:36 PM - Hanmac (Hans Mackowiak)
nobu (Nobuyoshi Nakada), isn't your MatchData#match the same as MatchData#[]?

#4 - 09/16/2021 02:28 PM - nobu (Nobuyoshi Nakada)
Hanmac (Hans Mackowiak) wrote in #note-3:

    nobu (Nobuyoshi Nakada), isn't your MatchData#match the same as MatchData#[]?

Similar, but #match accepts only single index/name, but not a range or an optional length.

#5 - 09/16/2021 04:02 PM - Hanmac (Hans Mackowiak)
nobu (Nobuyoshi Nakada) wrote in #note-4:

    Hanmac (Hans Mackowiak) wrote in #note-3:
        nobu (Nobuyoshi Nakada), isn't your MatchData#match the same as MatchData#[]?

    Similar, but #match accepts only single index/name, but not a range or an optional length.

i just wonder why not use the functions there like done in the other method?

    if (FIXNUM_P(idx)) {
        return rb_reg_nth_match(FIX2INT(idx), match);
    } else {
        int num = namev_to_backref_number(RMATCH_REGS(match), RMATCH(match)->regexp, idx);
        if (num >= 0) {
            return rb_reg_nth_match(num, match);
        } else {
            return match_ary_aref(match, idx, Qnil);
        }
    }

#6 - 09/17/2021 12:29 AM - nobu (Nobuyoshi Nakada)
match_ary_aref may return an Array of substrings when idx is a Range.