Ruby - Feature #16287

Proposal to add .second and .third in particular to class Array

10/31/2019 12:23 PM - shevegen (Robert A. Heiler)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	

Description

I meant to suggest this earlier, but then I think I did not; if I actually did, then please excuse my forgetfulness and close the issue request here. I am very unorganized (also in reallife).

Anyway - I will be as short as possible here.

I would like to propose that we add .second and .third, in particular for Arrays. I don't care that much for other data structures, so I will limit my initial suggestion here to the "minimum" case.

I will next explain as to why I suggest it (as many ruby core members said before, to demonstrate the use case, needs, trade-offs etc...).

Consider the following Array, aptly called array:

```
array = ['cat','dog','horse','fox','snake']
```

So five elements, five animals.

The archetypical "default" (main) way to query the "slot" at a specific position, in ruby, is via the index number, and []. So for example:

```
array[2] # => "horse"
```

So in this case we asked the array object for its third element (array index starts at 0).

This is all fine; we ruby users use this a lot.

We also have this available in ruby:

```
array.first # => "cat"
array.last  # => "snake"
```

This is often useful too, and I personally like this because it reads "naturally". But I think most of us may agree that the index specifier via [] is the main way for ruby to query elements at a certain position. We can also custom-define [] on our classes to allow for the [] to work (as syntactic sugar) on our own custom classes.

Now consider the following situation - we wish to return the first element, the last, and the second element.

This way works:

```
array[0]
array[1]
array[-1]
```

This way also works:

```
array.first
array[1]
array.last
```

The last solution, however had, puts me a bit off, in the sense that I dislike the mixing of [] with longer .method_name, such as .first and .last.

In the last case, I think this would read better:

```
array.first
array.second
array.last
```

I do not know if this has been suggested before; quite possibly it may have been suggested before. If anyone knows then perhaps there was a reason why the latter was not added.

To me, personally, from my point of view, I like a "symmetry" in the code here.

I have found in my own code when I end up with a situation like this:

```
array.first
array[1]
array.last
```

Then I tend to much prefer that one instead:

```
array[0]
array[1]
array[-1] # Or another way to get the last element such as array.size instead.
```

The reason for the latter is mostly that I feel it "reads" better. But if I use .first and .last, then I would prefer a positional name, such as .second.

You may wonder why I do not suggest .fourth, .fifth and so forth. Well, yes, this could be done too, but I think people will use these variants less and less. How often may it happen that you wanted to use .fifth, aka [4]? Probably not that often compared to .first, or .last, or even [1]. So my reasoning here is mostly about common use cases; I think .second and perhaps .third will be more common than e. g. .fifth. .second is probably significantly more frequent than .third too - but I have no problem if more names would be available either. To me this is mostly a detail, not the big picture. I am trying to keep my suggestion here smallish, though, to make it simpler if it is decided that this may be a useful or wanted addition.

I should also add that I do not have any problem keeping using the [] variants. My use case is really motivated almost exclusively from .first and .last alone. These two names are probably by far the most commonly used method names, but I think .second may also be useful. And my reasoning is really mostly with the above explanation alone e. g. "symmetry" in code use between array.first and array[1].

Please feel free to comment in any way as you see fit. I think this was perhaps suggested before but I can not find any older discussion (or perhaps it was many years ago or so, in which case it may be useful to have this more recent discussion too, if only to be able to point others to it if it may come up again in the future).

I also agree that the real benefit will be very small; at the same time, there should not be any major problem with the feature either. (Although, perhaps if matz may approve, it should come after ruby 3.0, just to not confuse people about this working; it may be easier to add it past ruby 3.0, so that ruby users do not have to wonder for now whether more method names to numbers may be available. I assume that Active* in rails may have this, but I don't know, I am still not using rails really, so I come from a non-rails point of view as well; I do use sinatra though, it's great - minimal code is great for being able to "build" flexible software.)

PS: I limited this to class Array. Perhaps the above would fit to other enumerable or enumerators (I mix these up all the time), but my common use case is really motivated almost exclusively by and for class Array. The reason is because I tend to end up with array data structures often; they are so easy to work with in ruby.

History

#1 - 10/31/2019 12:32 PM - naruse (Yui NARUSE)

- Subject changed from [Feature suggestion] Proposal to add .second and .third in particular to class Array to Proposal to add .second and .third in particular to class Array

#2 - 10/31/2019 01:47 PM - nobu (Nobuyoshi Nakada)

- Description updated

If I need such repeating accesses, I'll use values_at instead.

Probably, the word first is misleading.
If it were head or lead, the meaning would be clear.