Ruby master - Feature #16806

Struct#initialize accepts keyword arguments too by default

04/22/2020 06:14 AM - k0kubun (Takashi Kokubun)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>k0kubun (Takashi Kokubun)</td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
</tbody>
</table>

### Description

#### Proposal

Post = Struct.new(:id, :name)

# In addition to this,
Post.new(1, "hello") => #<struct Post id=1, name="hello">

# Let the following initialization also work
Post.new(id: 1, name: "hello") => #<struct Post id=1, name="hello">

#### Known incompatibility

- Post.new(id: 1, name: "hello") will be #<struct Post id=1, name="hello"> instead of #<struct Post id=1, name="hello">, name=nil>
  - Struct initialization only using keyword arguments should be warned in Ruby 3.0. **This feature should be introduced in Ruby 3.1 or later.**

#### Edge cases

- When keyword arguments and positional arguments are mixed: Post.new(1, name: "hello")
  - This should continue to work like Ruby 2: #<struct Post id=1, name="hello">
- Only keywords are given but they include an invalid member: Post.new(foo: "bar")
  - ArgumentError (unknown keywords: foo)
- When keyword_init is used
  - true: Require keyword init, disallow positional init.
  - false: Treat keywords as positional hash.

#### Use cases

- Simplify a struct definition where [Feature #11925](https://bugs.ruby-lang.org/issues/11925) is used.
  - When we introduced [Feature #11925](https://bugs.ruby-lang.org/issues/11925), name (Yusuke Endoh) thought we don't need keyword_init: true once keyword args are separated ([https://docs.google.com/document/d/1XbUbc8_eTqgh21FOw9a_X-ZyLyCBjvq8rWwpf5BM/edit#](https://docs.google.com/document/d/1XbUbc8_eTqgh21FOw9a_X-ZyLyCBjvq8rWwpf5BM/edit#)). That's what this ticket is all about.
  - However, the keyword arguments separation was done differently from what we expected at the moment. So we need to deal with the "Known incompatibility".
  - Matz objected to having a new keyword argument (immutable: true) in Struct.new at [https://bugs.ruby-lang.org/issues/16769#note-8](https://bugs.ruby-lang.org/issues/16769#note-8). So keyword_init: true seems also against Ruby's design. Now we should be able to skip specifying the option for consistency in the language design.

### Related issues:

- Related to Ruby master - Feature #11925: Struct construction with kwargs - Closed

### Associated revisions

Revision 8d099aa0 - 01/17/2021 09:35 AM - k0kubun (Takashi Kokubun)
Warn Struct#initialize with only keyword args (#4070)

- Warn Struct#initialize with only keyword args
- Do not warn if `keyword_init: false` is explicitly specified
- Add a NEWS entry
- `s/in/from/
- Make sure all fields are initialized

Revision fbb3cab9 - 01/29/2022 08:32 AM - k0kubun (Takashi Kokubun)
Add a NEWS entry about [Feature #16806]

History

#1 - 04/22/2020 06:23 AM - k0kubun (Takashi Kokubun)
- Related to Feature #11925: Struct construction with kwargs added

#2 - 04/22/2020 06:29 AM - k0kubun (Takashi Kokubun)
- Description updated

#3 - 04/22/2020 02:57 PM - jeremyevans0 (Jeremy Evans)
I'm OK with the basic idea of allowing keyword init for Structs by default. However, because this changes behavior, I think we should keep existing behavior and warn in Ruby 3.0 and not make this change until Ruby 3.1. I think this change should only affect cases where keywords are passed without positional arguments during Struct initialization.

For `Post.new(foo: "bar")`, this should still be treated as keyword init, even though `foo` is not a member of `Post`. In Ruby 3.0, that would warn and use `{foo=>"bar"}` as the first member of `Post`. In Ruby 3.1, that would raise ArgumentError, since `foo` is not a member of `Post` and the keywords will be treated as named members.

For `Post.new(1, foo: "bar")`, `Post.new(1, id: 1);` I think these should be treated the same as Ruby 2. Any non-keyword argument should treat keywords as a positional hash argument. I think allowing mixing of the arguments would result in confusion, brings up additional edge cases, and there is no practical use case for it.

Do we want to support `Post = Struct.new(:id, :name, keyword_init: false)` to disallow keyword initialization and always treat keywords as a positional hash? I think we should as that is the intent of the code. Basically, `keyword_init` would allow 3 values:

- true: Require keyword init, disallow positional init.
- false: Treat keywords as positional hash.

#4 - 04/23/2020 06:10 AM - k0kubun (Takashi Kokubun)
- Description updated

I agreed with your ideas and reflected them to the description.

#5 - 01/13/2021 06:30 AM - matz (Yukihiro Matsumoto)
I am OK with 3.1 to warn, 3.2 to change.

Matz.

#6 - 01/17/2021 09:36 AM - k0kubun (Takashi Kokubun)
- Status changed from Open to Closed

Applied in changeset git|8d099aa040427aede04e42c3ec9380afd431ffe3.

Warn Struct#initialize with only keyword args (#4070)

- Warn Struct#initialize with only keyword args

A part of [Feature #16806]

- Do not warn if `keyword_init: false`
is explicitly specified

- Add a NEWS entry
- s/in/from/
- Make sure all fields are initialized

#7 - 01/19/2021 12:05 AM - k0kubun (Takashi Kokubun)
- Assignee set to k0kubun (Takashi Kokubun)
- Status changed from Closed to Assigned

#8 - 01/29/2022 08:29 AM - k0kubun (Takashi Kokubun)
- Status changed from Assigned to Closed

nobu (Nobuyoshi Nakada) already committed the last step at c956f979e5d05900315d2753d5c3b1389a6dae4. Closing.