Disallow "in" as a keyword argument name

"in" is a reserved keyword and thus can't be accessed directly anyway.

e.g. the following is a syntax error:

```ruby
def m(in:);
p in; end; m(in: 1)
```

And as usual, doing `in = 1` doesn't work.

"in" could still be passed as a keyword argument if using some kind of catch-all, e.g. `m(args)`, `m(*args)`, etc.

Source: [https://twitter.com/drbrain/status/1104152696339087361](https://twitter.com/drbrain/status/1104152696339087361)

History

**#1 - 03/08/2019 11:14 PM - drbrain (Eric Hodel)**

This would break the Kernel#spawn family's backward compatibility make wrapping methods in this family harder than it already is, as you can work around this with Binding#local_variable_get.

I think this should be rejected.

**#2 - 03/08/2019 11:36 PM - nobu (Nobuyoshi Nakada)**

- Status changed from Open to Rejected
- Description updated

It is known that reserved words can be keyword arguments.
And Binding#local_variable_get has been added for that purpose, if argument for instance.