## Description

```ruby
# given
module Bam
  def self.[](bam)
    yield bam
  end
end

# SyntaxError: unexpected keyword_do_block
Bam['bam'] do |b|
  puts b
end

# SyntaxError: unexpected { arg, expecting end-of-input
Bam['bam'] { |b|
  puts b
}

# Valid but more verbose
Bam.[]('bam') do |b|
  puts b
end
Bam.[]('bam') { |b|
  puts b
}
```

### History

**#1 - 06/03/2017 04:39 PM - shevegen (Robert A. Heiler)**

I agree. I always wondered about this too.

I did not make a suggestion like this because I suspect that one reason may be a parser issue, as otherwise it would most likely have been done already.

In particular for elements without any arguments like:

```ruby
Bam.[] { |b| puts b }
```

Or more verbose to use your example:

```ruby
module Bam
  def self.[](bam=42)
    yield bam
  end
end

Bam.[] { |b| puts b } # => 42
Bam[] { |b| puts b } # => SyntaxError: (irb):14: syntax error, unexpected { arg, expecting end-of-input
```

Only difference here between failure and success is one lonely `. `. I am not sure if it is technically possible to get rid of the `. `. but if it would be possible, that would be great. The `.` visually disturbs me when I see `[]` ... but not at other places, it's weird. :D

Edit: Ok, my example was not good but I guess the difference between `[]` and `[]` can be seen which is the main point.
#2 - 08/31/2017 05:25 AM - nobu (Nobuyoshi Nakada)
The trunk accepts a block after [] now.

#3 - 09/25/2017 12:18 PM - shyouhei (Shyouhei Urabe)
- Status changed from Open to Closed

We looked at this at a developer meeting today and confirmed that this is already done.

#4 - 09/25/2017 01:31 PM - matz (Yukihiro Matsumoto)
And now mruby does too.

Matz.