Ruby master - Feature #17292
Hash Shorthand / Punning
10/29/2020 05:07 AM - baweaver (Brandon Weaver)

Status: Closed  
Priority: Normal  
Assignee:  
Target version:  

Description

Set Literal vs Javascript Object Punning

There was a proposal for a Set literal here: https://bugs.ruby-lang.org/issues/16989

```ruby
set = { 1, 2, 3 }
```

...but it was brought up that this is similar to the Javascript Object punning, or Object shorthand syntax:

```javascript
const a = 1, b = 2, c = 3;
const punnedObject = { a, b, c }
// => { a: 1, b: 2, c: 3 }
```

**Proposition:** I believe we should use brackets ( `{` ) for a shorthand Hash syntax similar to Javascript.

Hash Punning

My first proposal in this feature request is Hash punning, or Hash shorthand:

```ruby
a = 1
b = 2
c = 3
{ a: , b: , c: }
# => { a: 1, b: 2, c: 3 }
```

This syntax avoids the ambiguous syntax of empty block ( `{` ) versus empty set ( `{` ), and with the presence of Symbols it introduces a distinct syntax that would be easier to parse against.

One potential issue would be mixed syntax:

```ruby
{ a: , b: 2 }
# => { a: 1, b: 2 }
```

Method Punning

This syntax can also be used for keyword argument and method call punning:

```ruby
def method_name(a, b, c):
    a + b + c
end
```

```ruby
a = 1
b = 2
c = 3
method_name(a, b, c)
# => 6
```

I believe this existing syntax for required keywords gives credence to the idea of introducing punning to Ruby, as it's very similar to existing syntax, and therefor feels "Ruby-like".

Pattern Matching
This syntax is also already present and used in pattern matching, making it already part of the language:

```ruby
case { x: 1, y: 2 }
in { x:, y: }
  { x:, y: y + 1 } # new
else
  # ...
end
```

I believe this further justifies the case for punning syntax.

Related issues:
- Related to Ruby master - Feature #18124: Hash shorthands (matching constructors functionality in JS) added
- Is duplicate of Ruby master - Feature #14579: Hash value omission added

History

**#1 - 10/29/2020 05:48 AM - mame (Yusuke Endoh)**

- Is duplicate of Feature #14579: Hash value omission added

**#2 - 10/29/2020 06:26 AM - baweaver (Brandon Weaver)**

I would agree this is a duplicate, but in the last year we have had the introduction of the pattern matching syntax which makes this less of a syntactical oddity than it may have been in the past.

**#3 - 08/20/2021 06:18 PM - mame (Yusuke Endoh)**

- Related to Feature #18124: Hash shorthands (matching constructors functionality in JS) added

**#4 - 09/11/2021 09:59 AM - shugo (Shugo Maeda)**

- Status changed from Open to Closed

Thank you.
Committed in c60dbcd1c55cd77a24c41d5e1a9555622be8b2b8.