Ruby master - Feature #10406
Method starting with a capital is not detected when parentheses and receiver are omitted
10/20/2014 11:32 AM - sawa (Tsuyoshi Sawada)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>matz (Yukihiro Matsumoto)</td>
</tr>
<tr>
<td>Target version:</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Description
A method whose name starts with a capital is not detected when parentheses and receiver are omitted.

```ruby
def Foo; puts "foo" end
Foo # => uninitialized constant Foo
```

If I disambiguate it as a method, then the method is detected:

```ruby
Foo() # => foo
```

This kind of consideration if unnecessary if the method name does not start with a capital.

```ruby
def foo; puts "foo" end
foo # => foo
```

So I expect method Foo to be detected without explicit receiver and arguments (unless there is a constant with the same name).

```ruby
def Foo; puts "foo" end
```

I first thought this as a bug. If it is not a bug, then I would like to ask this as a feature request.

History
#1 - 10/20/2014 11:42 AM - nobu (Nobuyoshi Nakada)
- Tracker changed from Bug to Feature
- Category set to syntax
- Status changed from Open to Assigned
- Assignee set to matz (Yukihiro Matsumoto)
- Target version set to 3.0

It's a ruby's naming rule.

#2 - 10/20/2014 11:46 AM - mame (Yusuke Endoh)
I believe this is intentional. Otherwise, we would become unable to refer Array, Integer, Float, String, and Hash classes because of Kernel#Array and so on :-(

--
Yusuke Endoh mame@ruby-lang.org

#3 - 10/20/2014 11:51 AM - sawa (Tsuyoshi Sawada)

Yusuke Endoh wrote:

```
I believe this is intentional. Otherwise, we would become unable to refer Array, Integer, Float, String, and Hash classes because of Kernel#Array and so on :-(
```

Just like local variables have precedence over method look up, I expect that constants have precedence over method look up, so I don't think that would be a problem.

#4 - 10/20/2014 12:07 PM - mame (Yusuke Endoh)
Oops, I see. Sorry for the noise!
Unlike local variables, constants are not resolved statically. That's the reason behind this behavior. If you have further demand, reopen the issue.

Matz.