Ruby - Feature #5893

named return,next...

01/13/2012 10:51 PM - neleai (Ondrej Bilka)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:	2.0.0	

Description

Hello. As returning from various blocks is recurring theme at list I propose following.

You can assign label to each block. You can exit given block by referring to its label.
Syntax can be ::label to avoid ambiguity(But prettier syntax would be desirable).

return,next,break,redo would work as they do now.

Here is example

```
while::loop true
10.times{|i|::outer
10.times{|j|::inter
next::outer    if i>6
return::inner 42 if j>4
puts i,j
}
}::outer #For readability we could optionaly add label at end of block.
break::loop
end
```

History

#1 - 01/13/2012 11:23 PM - yeban (Anurag Priyam)

On Fri, Jan 13, 2012 at 7:21 PM, Ondrej Bilka neleai@seznam.cz wrote:

> Hello. As returning from various blocks is recurring theme at list I propose following.
>
> You can assign label to each block. You can exit given block by referring to its label.
> Syntax can be ::label to avoid ambiguity(But prettier syntax would be desirable).

Doesn't throw/catch to that already?

--
Anurag Priyam

#2 - 01/13/2012 11:29 PM - neleai (Ondrej Bilka)

On Fri, Jan 13, 2012 at 11:11:28PM +0900, Anurag Priyam wrote:

> On Fri, Jan 13, 2012 at 7:21 PM, Ondrej Bilka neleai@seznam.cz wrote:
>
>> Hello. As returning from various blocks is recurring theme at list I propose following.
>>
>> You can assign label to each block. You can exit given block by referring to its label.
>> Syntax can be ::label to avoid ambiguity(But prettier syntax would be desirable).
>
> Doesn't throw/catch to that already?
> No unless you are happy with passing result by global variable.

#3 - 01/14/2012 12:23 AM - neleai (Ondrej Bilka)

On Fri, Jan 13, 2012 at 11:34:17PM +0900, Yukihiro Matsumoto wrote:

> Hi,
>
> In message "Re: [ruby-core:42115] Re: [ruby-trunk - Feature #5893][Open] named return,next..."
> on Fri, 13 Jan 2012 23:26:36 +0900, Ondřej Bílka neleai@seznam.cz writes:
>
> |> Doesn't throw/catch to that already?
> |No unless you are happy with passing result by global variable.
>
> Have you ever tried? throw can pass the result to catch.
>
> ```
>         matz.
> ```

I did not know this one. In code I have seen it is typically
something like this.

def return_it(foo)
$foo=foo
throw(:my_constant)
end

$foo=nil
t=try(:my_constant){
yield
}
t=$foo if $foo

#4 - 01/14/2012 12:23 AM - yeban (Anurag Priyam)

2012/1/13 Ondřej Bílka neleai@seznam.cz:

> Yukihiro Matsumoto wrote:
>
> > throw can pass the result to catch.
>
> I did not know this one.

In my opinion, Sinatra is a very good example here. Look for
throw/catch in lib/sinatra/base.rb.

--
Anurag Priyam

#5 - 03/05/2012 11:46 AM - matz (Yukihiro Matsumoto)

Hi,

In message "Re: [ruby-core:42115] Re: [ruby-trunk - Feature #5893][Open] named return,next..."
on Fri, 13 Jan 2012 23:26:36 +0900, Ondřej Bílka neleai@seznam.cz writes:

|> Doesn't throw/catch to that already?
|No unless you are happy with passing result by global variable.

Have you ever tried? throw can pass the result to catch.

```
        matz.
```

#6 - 03/29/2012 01:45 AM - mame (Yusuke Endoh)

- Status changed from Open to Rejected

I believe this is already solved. Closing.

--
Yusuke Endoh mame@tsg.ne.jp