

Ruby master - Feature #16621

Second block parameter of Pathname#glob to be relative path from self

02/10/2020 12:44 PM - 5.5 (5 5)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
Description	
I often write such a code:	
<pre>some_dir = Pathname("some/dir") some_dir.glob("**/*.html") do path rel_path = path.relative_path_from(some_dir) # ... end</pre>	
If Pathname#glob yields the relative path from self as a second block parameter, I can write the above as follows:	
<pre>Pathname("some/dir").glob("**/*.html") do path, rel_path # ... end</pre>	

History

#1 - 02/10/2020 02:56 PM - shevegen (Robert A. Heiler)

(Just trying to get some more details - I have no particular opinion per se as I tend to almost exclusively use Dir[] myself, so this is more for ruby users who use Pathname in general.)

Would this change have any potential backwards incompatibility/problem? This is a genuine question, by the way, I really don't know, since I don't use Pathname since many years, and I have no pro/con opinion about the suggestion itself.

#2 - 02/12/2020 12:30 PM - 5.5 (5 5)

I think no problem in the backward compatibility.

If we use the *proposed* glob method with *one* block parameter, it behaves same as the current glob method.

#3 - 02/12/2020 02:59 PM - Dan0042 (Daniel DeLorme)

What you want is this:

```
Dir.glob("**/*.html", base: "some/dir")
```

It would be nice if this was exposed via Pathname. There's something weird with the documentation for Pathname#glob though; it says "This method uses the base keyword argument of Dir.glob." but this doesn't appear to be true. Dir.glob(g, base: p) is not the same as p.glob(g)

#4 - 02/12/2020 03:24 PM - jeremyevans0 (Jeremy Evans)

5.5 (5 5) wrote in [#note-2](#):

I think no problem in the backward compatibility.

If we use the *proposed* glob method with *one* block parameter, it behaves same as the current glob method.

This is incorrect if the block passed is a lambda.

#5 - 02/12/2020 11:22 PM - 5.5 (5 5)

jeremyevans0 (Jeremy Evans) wrote in [#note-4](#):

This is incorrect if the block passed is a lambda.

Oops. You are right.

My proposal contains the backward incompatibility.

So some option is needed ...?

#6 - 02/12/2020 11:27 PM - 5.5 (5 5)

Dan0042 (Daniel DeLorme) wrote in [#note-3](#):

What you want is this:

```
Dir.glob("**/*.html", base: "some/dir")
```

Thank you for your suggestion.

But I want *both* its absolute path and its relative path from self.