

Ruby trunk - Bug #7975

Why `__dir__`, not `__DIR__`

02/27/2013 10:22 AM - yimutang (Joey Zhou)

Status: Rejected	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: 2.0	Backport:
Description There's a FILE in Ruby 1.x, no file Why dir in Ruby 2.0 is downcase? It's not consistent.	
Related issues: Is duplicate of Ruby trunk - Feature #3346: <code>__DIR__</code> revisited Closed 05/26/2010	

History

#1 - 02/27/2013 10:31 AM - zzak (Zachary Scott)

- Status changed from Open to Rejected

see [#3346](#)

#2 - 02/27/2013 10:48 AM - yhara (Yutaka HARA)

That is because **dir** is a method, whereas **FILE** is not.

```
$ ruby -ve 'p Kernel.methods'
ruby 2.0.0p0 (2013-02-24 revision 39474) [x86_64-darwin12.2.1]
[:sprintf, :format, :Integer, :Float, :String, :Array, :Hash, :warn, :raise, :fail,
:global_variables, :method, :callee, :dir, :eval, .....
```

dir is preferred to avoid adding a new keyword (see [#1961](#).)

#3 - 02/27/2013 08:19 PM - wardrop (Tom Wardrop)

=begin

We did not come to any sort of resolution to the naming inconsistency between **dir/method** and **FILE/LINE** in any of those mentioned issues.

Should we not add **file** and **line** as methods, and perhaps deprecate the keywords **FILE** and **LINE**. This would keep it consistent with all the other double-underscore methods. To most developers who perhaps do not know Ruby as intricately as most of the people on this issue tracker, the inconsistency between **dir** and **FILE** is not just confusing by name, but the fact that one is a method and one isn't, is doubly confusing. Definitely not principle of least surprise.

This needs to be addressed in my opinion, either through deprecation of **FILE** and **LINE**, or by keeping those keywords and simply creating Kernel method equivalents for the sake of a consistent API.

While on the topic, someone also suggested in one of those previous issues, to give **dir** an optional join argument, so you could do something like this:

```
(({dir('somefile.txt') # => /Users/admin/somefile.txt}))
```

I'd predict that at least 90% of use cases for **dir** will involve joining it to another path or filename. I can't see any harm in adding this. The naming inconsistencies are my main concern however.

=end