

It's back! [Take the 2018 Developer Survey today »](#)



Why do people write `#!/usr/bin/env python` on the first line of a Python script?

It seems to me like the files run the same without that line.

[python](#) [shell](#) [shebang](#)

edited Mar 11 '10 at 23:54



[Sinan Ünür](#)

103k

14

162

305

asked Mar 11 '10 at 23:50



[John Garcia](#)

3,737

3

10

3

The answer below that states that it is just a comment line. That's not always the case. I have a "Hello, World!" CGI script(.py) that will only run and display the webpage with `#!/usr/bin/env python` at the top. – [Chakotay](#) Nov 9 '13 at 22:56

possible duplicate of [What's the difference between these two python shebangs](#) – [fuad](#) Feb 18 '14 at 12:15

They may run, but not in the intended environment – [Nicholas Hamilton](#) Jul 28 '14 at 9:09

I've visited this post so many times in 7 years because I sometimes forget the env hashbang. Copy pasta :) – [BugHunterUK](#) Jul 15 '17 at 17:10

20 Answers

If you have several versions of Python installed, `/usr/bin/env` will ensure the interpreter used is the first one on your environment's `$PATH`. The alternative would be to hardcode something like `#!/usr/bin/python`; that's ok, but less flexible.

In Unix, an *executable* file that's meant to be interpreted can indicate what interpreter to use by having a `#!` at the start of the first line, followed by the interpreter (and any flags it may need).

If you're talking about other platforms, of course, this rule does not apply (but that "shebang line" does no harm, and will help if you ever copy that script to a platform *with* a Unix base, such as Linux, Mac, etc).

edited Jul 5 '15 at 2:20



[nbro](#)

4,807

6

36

76

answered Mar 11 '10 at 23:52



[Alex Martelli](#)

557k

106

965

1229

188 Just to add: this applies when you run it in Unix by making it executable (`chmod +x myscript.py`) and then running it directly: `./myscript.py` , rather than just `python myscript.py` . – [Craig McQueen](#) Mar 12 '10 at 2:48

17 using `env` gives maximum flexibility in that the user can select the interpreter to use by changing the `PATH`. Often this flexibility is not required though and the downside is that linux for example can't use the script name for the name of the process in `ps` and reverts to "python". When packaging python apps for distros for example I would advise not to use `env .` – [pixelbeat](#) Mar 12 '10 at 13:26

7 [py launcher](#) can use the shebang line on Windows. It is included in Python 3.3 or [it can be installed independently](#). – [jfs](#) Jan 17 '13 at 0:35

Note that you have to have a full path to an interpreter in the shebang. If not `#!/python` would have sufficed instead of `#!/usr/bin/env python` – [Aneesh](#) Sep 8 '13 at 6:29

2 @Aneesh: This is true for UNIX, but not for Windows. Read the [python.org/dev/peps/pep-0397](#) – [pepr](#) Sep 15 '13 at 21:10